# Table of Contents

How to use the Moore_RateTest script.................................................................1

List of example instructions..................................................................................2
  Example: Using trigger-dev to test the HLT2 line rate with HLT1 filtered input data...........................................2
  Example: Testing the rate of your development HLT2 line with the production version of Moore, and HLT1 filtered input data...........................................................................................................2
  Example: Measure the rate w.r.t. L0 filtered data while tuning some HLT2 lines with the Moore v26r0 release.................................................................................................................................2
  Example: Run straight from a nightly or release build without lb-dev..............................................................3
  Example: Measure signal efficiencies....................................................................................................................3
How to use the Moore_RateTest script

The script lives in the DBASE package, PRConfig. You can see it here in websvn. You can refer to these slides for some examples on how to use the tool. It is visible to a Moore environment, e.g. lb-run Moore/prod cat '$PRCONFIG/python/MooreTests/Moore_RateTest.py'. Many of the common uses of the Moore_RateTest are tested with nightly qmtests, which are configured here.
List of example instructions

Please let us know if you have a use case that isn’t covered by the following list of examples. The recommended example for line development is this one.

Example: Using `trigger-dev` to test the HLT2 line rate with HLT1 filtered input data.

This is the recommended example for testing the rate of your development HLT2 line for 2017 datataking. Check the comprehensive up-to-date instructions at https://gitlab.cern.ch/lhcb-HLT/trigger-dev

```
source /cvmfs/lhcb.cern.ch/group_login.sh -c x86_64-centos7-gcc62-opt  # use CVMFS with the correct gcc version
 cd /afs/cern.ch/work/X/USERNAME/
 git clone ssh://git@gitlab.cern.ch:7999/lhcb-HLT/trigger-dev.git workspace
 cd workspace

# Clone PRConfig
 mkdir DBASE && cd DBASE
 git lb-clone-pkg PRConfig
 cd ..

# Build and run the test
 export LC_ALL=C
 make # do not add -j
 Moore/run python DBASE/PRConfig/python/MooreTests/Moore_RateTest.py --inputdata=2018_Hlt1_0x11751801 --L0TCK=0x1801 --evtmax=1000 --input_rate=110.e3 | tee Moore_RateTest.log
```

Example: Testing the rate of your development HLT2 line with the production version of Moore, and HLT1 filtered input data.

This is the default example for testing the rate of your development HLT2 line for 2018 datataking. This example for 2017 was tested in LBHLT-138 and a log file is attached there in case you want to compare with your test.

```
#lb-dev --list Moore
lb-dev --name=RateTest Moore/prod
 cd RateTest
 #the following would be if you want to make some changes, e.g. to the settings or lines
 #git lb-use Hlt
 #git lb-checkout Hlt/2018-patches Hlt/HltSettings
 #git lb-checkout Hlt/2018-patches Hlt/Hlt2Lines
 mkdir DBASE && cd DBASE
 git lb-clone-pkg PRConfig
 cd ..
 make -j 4
 ./run python DBASE/PRConfig/python/MooreTests/Moore_RateTest.py --inputdata=2018_Hlt1_0x11751801 --L0TCK=0x1801 --evtmax=1000 --input_rate=110.e3 | tee Moore_RateTest_orig.log
```

Example: Measure the rate w.r.t. L0 filtered data while tuning some HLT2 lines with the Moore v26r0 release.

```
lb-dev Moore v26r0
 cd MooreDev_v26r0
 git lb-use Hlt
 git lb-checkout Hlt/master Hlt/HltSettings
 git lb-checkout Hlt/master Hlt/Hlt2Lines
 getpack PRConfig head
 make -j 4
 ./run python PRConfig/python/MooreTests/Moore_RateTest.py --evtmax=10000 --inputdata=2016NB_25ns_0x11751801 --inputrate=110.e3 | tee Moore_RateTest_orig.log
```

List of example instructions
Example: Run straight from a nightly or release build without lb-dev

```
make -j 4
./run python PRConfig/python/MooreTests/Moore_RateTest.py --evtmax=10000 --inputdata=2016NB_25ns_4
```

Example: Measure signal efficiencies

First of all, we need to prepare a suitable signal MC sample.

1. It should be juggled to look like raw data. Here is an example script: jugglestep.py.txt. For changing the example script to your data you might have to change the input version (2.0 can be used for most MC) together with the input location.
2. Then you should rerun L0 using L0App with
3. Then you should run the Moore_RateTest script with the --mode='efficiency' option.

There is a sample of juggled $D^0 \rightarrow D^0\pi, D^0 \rightarrow K^\ast 2015$, Sim08, MC in the testfile db under "Juggled_MC_2015_27163003_DstToD0pip_D0ToKmPip". Steps 2 and 3 are tested using a small number of events from this sample in the nightlies. If you encounter any problems with this script, please first look here to see if the nightly test is OK. Aside: If you are developing Hlt code, then it is expected that you regularly follow the nightly tests.

Please note: Not all MC samples contain the full raw event information which is needed to run L0 and Moore. A description of all file types can be found here.

dl-dev --nightly-cvmfs --nightly lhcb-head Moore/HEAD --name MooreDev_eff
cd MooreDev_eff
# if needed
# git lb-use ...
# git lb-checkout ...
make -j 8
# then you need to get the scripts for step 2
./run cp '$MOOREROOT/tests/options/Moore_RateTest/RedoL0Step.py Moore_RateTest_MC_RedoL0.py' .
./run cp '$PRCONFIGROOT/python/MooreTests/Moore_RateTest.py' .
# edit the RedoL0 script to take your juggled MC sample as input
# then run it
./run gauridrun.py Moore_RateTest_MC_RedoL0.py 2>&1 | tee Moore_RateTest_MC_RedoL0.log
# this should produce a new file called rerunL0.dst
# then you run Moore_RateTest over it in efficiency mode
./run python '$PRCONFIGROOT/python/MooreTests/Moore_RateTest.py' \
  --evtmax=500 \ 
  --mode=efficiency \ 
  --DisableRunStampCheck \ 
  --inputdata=rerunL0.dst \ 
  --TFDB=Juggled_MC_2015_27163003_DstToD0pip_D0ToKmPip 2>&1 | tee Moore_RateTest.log
# it uses the testfileDB entry to get the DataType, tags, Simulation=True/False etc...

Please note that you can also use conddb and simdb tags directly, as noted by Mika:

You can replace TFDBForTags=XYZ by settings the tags explicitly
https://svnweb.cern.ch/trac/lhcb/browser/DBASE/trunk/PRConfig/python/MooreTests/Moore_RateTest.py#L135
https://svnweb.cern.ch/trac/lhcb/browser/DBASE/trunk/PRConfig/python/MooreTests/Moore_RateTest.py#L129

Log files from this example are here

- For the RedoL0 step,
- And the interesting part for the Moore_RateTest step (Full log).

Example: Measure the rate w.r.t. L0 filtered data while tuning some HLT2lines with the Moore v26r0 release.
Example: Measure signal efficiencies