

Table of Contents

TCK Trigger Configuration Key.....	1
TCKsh.....	1

TCK Trigger Configuration Key

The HLT is configured via a unique key, named Trigger Configuration Key (TCK), that defines the sequence of algorithms, and the cuts (we refer to them as *filters of a selection*)

- TCK is a 32 bit value which labels configurations, lower 16 bits specify L0 config.
- The individual configurations are identified by 128 bit configuration IDs (think of this ID as the checksum of the HLT part of the configuration)
- for each TCK, you can find out the 'HltType' and 'Release' it corresponds to. The promise is that within the same (Release, HltType) pair, the 'flow' of algorithms **must** be the same
- things that **can** be different are thresholds and prescales... (due to be able to do fast run changes)

TCKsh

Is an interactive environment in which you can list configurations, compare them, query for properties of algorithms, etc. For example with Moore v28r3p1, do:

```
lb-run -c best Moore/v28r3p1 TCKsh
```

Command	Description
<code>getTCKInfo(0x00AB0046)</code>	shows the threshold settings and Moore version for a given TCK
<code>listConfigurations()</code>	shows all known configurations, grouped by release, and 'hltType'
<code>listAlgorithms(0x00AB0046)</code>	shows a list of configured algorithms given a TCK or configID
<code>dump(0x00AB0046)</code>	shows a list of properties given a TCK
<code>getHlt1Lines(0x00AB0046)</code>	returns a list of Hlt1 lines
<code>listProperties(0x00AB0046, '.*\/Hlt1Global')</code>	list properties of algorithms whose name matches the regex 'Hlt1Global'
<code>listProperties(0x803b0000, '.*\/Hlt1', 'Code FilterDesc')</code>	.. same but only for properties that match 'Code' or 'FilterDesc'
<code>getProperties(0x803b0000, '.*\/Hlt1', 'Code FilterDesc')</code>	conducts a regex search like listProperties but returns a dictionary of the form {algorithm name: {property: value}}
<code>dumpL0(0x00AB0046)</code>	returns the L0 configuration, with channels and thresholds
<code>getL0Prescales(0x00AB0046)</code>	returns the L0 prescales
<code>getRoutingBits(0x00AB0046)</code>	return a python list of Routing Bit definitions given a TCK or ConfigID
<code>listRoutingBits(0x00AB0046)</code>	list Routing bit definitions given a TCK or ConfigID
<code>diff(0x80350000, 0x00AB0046)</code>	shows items only in left, only in right, and differences of items in both
<code>updateProperties(...)</code>	allows to copy-and-modify existing configurations (eg. change prescales)
<code>createTCKentries({ .. : .. , })</code>	creates entries in mapping from TCK to configID, eg. update to new L0 TCK, but keep same HLT config

TCK < LHCb < TWiki

To analyse the sometimes lengthy output it may help to redirect it to a file using `TCKsh > out.txt` or use `sys.stdout=open("out.txt","w")` inside `TCKsh`

There is also an unofficial script for printing the selection tree of a given trigger line and [TCK](#).

This topic: LHCb > TCK

Topic revision: r19 - 2019-12-03 - RenatoQuagliani



Copyright &© 2008-2020 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](#)