Table of Contents

How to create HLT TCKs

This page describes the pre-2016 procedure to make a single HLT1+HLT2 TCK. For datataking in 2016 onwards, we have split HLT1 and HLT2 TCKs. HLT piquets should go to these instructions for producing split TCKs for 2016 datataking

Creating a private TCK

To keep this tutorial simple and complete a separate config.tar file is used to store the private TCK, but the same steps apply when using config.tar from the TCK/HltTCK package.

The first thing which needs to be done is to ensure that a ThresholdSettings is available which contains all of the required features. This will generally involve getpacking Hlt/HltSettings, Hlt/Hlt{1,2}Lines and sometimes other packages and making the required changes. This is assumed to be completed and the settings are assumed to be named PrivateThresholdSettings.

Setting up the environment

Start by setting up the cmt environment for Moore and creating an empty file in which the TCK(s) will be stored; this needs to be done only once. A template script to use is attached as https://twiki.cern.ch/twiki/pub/LHCb/PrivateTCK/CreateTCK.py.txt; save it somewhere and modify it to suit the input MC DST you want to use. Make sure that the database tags and datatype are correct.

```
$> SetupProject --build-env Moore v24r0
$> cd $User_release_area/Moore_v24r0
$> mkdir TCKData
$> cp /path/to/CreateTCK.py.txt CreateTCK.py
```

Make sure that the TCKData property in the script points to the directory you just created.

Create the TCK

Logout and log back in (or create a new shell), make sure that the modified threshold settings and other packages which contain the changes you require are properly installed and run the script.

```
$> SetupProject Moore v24r0
$> cd $User_release_area/Moore_v24r0
$> gaudirun.py CreateTCK.py
```

You should now have the TCK dumped in a config.cdb file in the TCKData directory. If something goes wrong, delete the TCKData directory, make it again, fix the problem and run again.

Mapping the configuration to a TCK.

The following steps require the use of TCKsh; you can stay in the same shell.

To prevent possible confusion and errors, It is vital that you select a unique and unused TCK for your configuration. To see which TCKs are already in use, start TCKsh and look at the configurations. Search for the latest Moore version and find the highest TCK.

```
$> TCKsh
>>> listConfigurations()
>>> quit()
```

Once you have found a suitable TCK, it is time to map your configuration to it. Below 0x00900032 is used.

```
$> TCKsh
>>> cas_rw = ConfigAccessSvc( Mode = 'ReadWrite', File='/path/to/config.cdb' )
>>> printConfigurations( getConfigurations( cas = cas_rw ) )
starting appMgr @ pid = 720
MOORE_v12r7
    Physics_May2011
         <NONE> : 24e17751af7447fed73ec35b108218b6 : Test TCK
>>> id = '24e17751af7447fed73ec35b108218b6'
>>> createTCKEntries( { 0x00900032 : id }, cas = cas_rw )
creating mapping TCK: 0x00900032 -> ID: 24e17751af7447fed73ec35b108218b6
>>> quit()
```

The private TCK 0x00900032 is now created and stored in your own config.tar

Modifying a private TCK.

Instead of creating a new TCK from scratch, it can be easier, safer and faster to create a new TCK by modifying an existing one. This is described on the HLTTCK page.

Creating a DST using a private TCK

Once the private TCK has been generated, Moore needs to be run on the input DST using the private TCK.

The following lines insure the available TCKs are read from your private file:

```
from Configurables import ConfigCDBAccessSvc
ConfigCDBAccessSvc().File = 'config.cdb'
Moore().TCKData='/path/to/TCKData'
```

A template script is attached as https://twiki.cern.ch/twiki/pub/LHCb/PrivateTCK/runMoore_CreateDST.py.txt; save it somewhere.

```
$> SetupProject Moore v12r7
$> cd $User_release_area/Moore_v12r7/scripts
$> cp runMoore_CreateDST.py.txt runMoore_CreateDST.py
$> gaudirun.py runMoore_CreateDST.py
```

Running DaVinci on your DST with private TCK

Once the DST has been created, DaVinci can be run on it, but it needs to be told where to find the private TCK. This is done by adding the following options to the script:

```
from Configurables import ConfigAccessSvc
ConfigAccessSvc().File = '/path/to/your/config.tar'
```

This topic: LHCb > PrivateTCK
Topic revision: r17 - 2016-04-28 - MikaVesterinen