Table of Contents

LHCb data reduction (stripping)

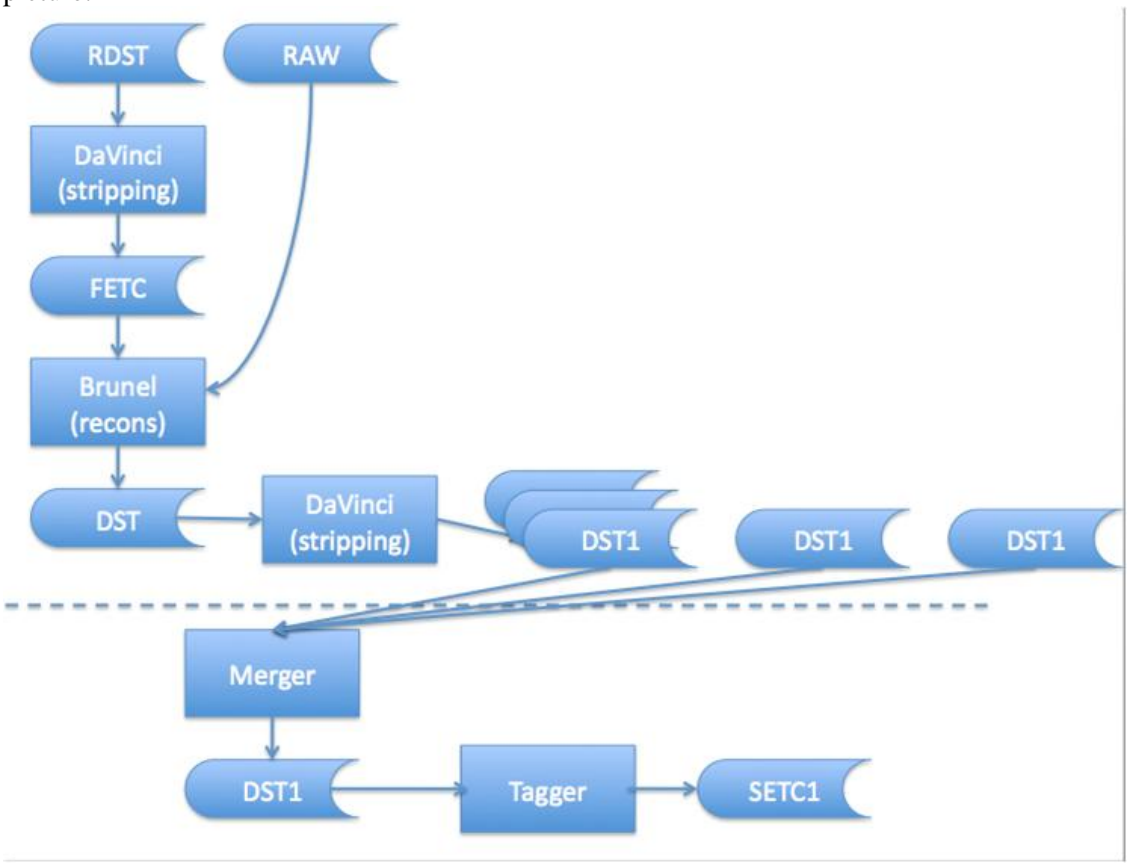
The LHCb computing model foresees that not all events recorded by the data acquisition are made available for physics analysis. After an initial reconstruction of all events, an event selection step (stripping) is executed - this is an analysis job that selects events based on "stripping selections" provided by the physics groups. Only events passing one or more of these selections are made available for further analysis.

Stripping framework

The stripping framework is a framework based on DaVinci within which the stripping selections are defined. More details are available here

Stripping workflow

The sequence of steps needed to produce a stripped DST ("stripping workflow") is shown in the following picture:



MC09 stripping workflow

What follows describes the workflow used to strip MC09 data, it refers to the picture above

In MC09, the RAW data was produced by Boole v18r1 and the RDST data by Brunel v34r7. In MC production, these datasets are not stored in separate files, they are combined in a single DST file which is the output of the Brunel production step.

Step 1: DaVinci (stripping)

This step reads the RDST from the MC09 production (taken from the MC09 DST files), it runs the default stripping and produces a full event tag collection (FETC): it stores the result of the stripping selection for each event read from the RDST.

Program version:	DaVinci v24r2p1
Options	AppConfig/v3r4/options/DaVinci/DVStrippingETC-MC09.py
Database:	SQLDDDB v5r11
Database tags:	SIMCOND MC09-20090402-vc-md100
	DDDB MC09-20090602

The output is an FETC. This output is not stored permanently; it is only used as input to Step 2

Open questions

- How can the stripped dataset be normalised? In the past, the FETC was used to normalise the stripped dataset to the number of input events. If we do not save the FETC, and in the presence of failed jobs, this normalisation will be lost. This problem will eventually be solved by using the File Summary Records (FSR), but this is not ready for this stripping. Marco will check whether the FETC can be saved this time, or whether an alternative method exists using the book-keeping. In any case this should not be a showstopper for this production.

Step 2: Brunel

This step runs the reconstruction to produce a DST containing all the events selected by Step 1. The inputs are the FETC produced by Step 1 and the RAW from the MC09 production (taken from the MC09 DST files), the output is a DST containing the OR of all the stripping selections. There are two possibilities for the version of Brunel:

1. The same version used in the original MC09 production. This guarantees consistency between the stripping done in Step 1, and that to be done in Step 3.
2. A more recent version, using the same version of REC as the version of DaVinci to be used in Step 3. This guarantees consistency of the reconstruction (in particular track pattern recognition) between Steps 2 and 3, but means that some of the events selected in Step 1 will not be re-selected in Step 3. Conversely, any events that would have been selected by step 3 but were not selected by step 1 are lost.

For this stripping exercise it is decided to use option 1, since it has already been commissioned and is better tested. Note that option 1 is a pure computing exercise in the case of MC data: since the input is an MC DST and the output is also an MC DST produced by an identical programme, Steps 1 and 2 could be skipped in a future stripping of MC data that uses option 1.

Program version:	Brunel v34r7
Options	AppConfig/v2r9p1/options/Brunel/MC09-Stripping.py
Database:	SQLDDDB v5r11
Database tags:	SIMCOND MC09-20090402-vc-md100
	DDDB MC09-20090602

The output is a DST. This output is not stored permanently; it is used as input to Step 3.

Step 3 DaVinci (stripping)

This step reruns the stripping selections taking as input the DST produced in step 2, and produces several DSTs, one per stripping stream. The stripping result is stored as a DST object. It re-runs the L0 and replaces on the RawEvent the L0 decision RawBank produced by Boole. It also runs HLT1 and stores the result as an

additional RawBank on the RawEvent.

Program version:	DaVinci v24r2p3
Options	AppConfig/v3r7/options/DaVinci/DVStrippingDST-MC09.py
Database:	SQLDDDB v5r11
Database tags:	SIMCOND MC09-20090402-vc-md100
	DDDB MC09-20090602

Open questions

- Have the output DST and the new options been checked? Not yet. Juan will produce a DST using the new options and give it to Thomas who will check the content.

Answered questions

- Which is the the right DaVinci version? DaVinci v24r2p3 should be used, as it contains fixes to some Hlt lines.
- Do the options above re-run L0, run HLT1, and store the results in RawBank as required? Yes.
- Are the L0 and HLT1 that are run using a correct TCK and is the `L0Conf().TCK` option is still needed? Yes, as confirmed to Juan by Miriam and Gerhard.
- Does the information in the RawBanks contain all that is needed for people interested in TIS/TOS studies? In principle yes, since the same information is stored as in the real data. If something is missing this is the occasion to find out!
- Is the stripping result stored on the DST? Yes: there is a TES directory created for each stripping selection that passes, and this is saved to the DST
- Are all B candidates and intermediate resonances stored on the DST? Yes.
- What happens when different selections in a stream select the same B candidates? It gets copied twice, since the output locations are different for each selection.
- Is it possible to save other information (eg. PV refit)? If the ReFitPVs property of the relevant DVAlgorithms is set to true, this gets done automatically. If not, some options have ot be set up. Not needed for this stripping
- It is quite possible that some of the output DSTs will contain zero selected events, resulting in empty or non-existent output files. Is this a problem for DIRAC? Possibly, but this is likely to be very rare and will be fixed only if needed. This is a short term problem since in future even files with no events will contain at least an FSR.
- Are the names of the streams OK and interfaced to the book-keeping? Yes

The outputs of this step are one DST per stripping stream.

Step 4: Merger

This is a technical step, does not require any specific application, just gaudirun.py with standard DST merging options. For each stripping stream it combines the (small) DSTs of individual stripping jobs into a larger DST whose file size is optimized for storage.

Step 5: Tagger

This is a DaVinci job that, for each stripping stream, reads the merged DST and uses the stripping result stored on the DST to produce a selection ETC (SETC). This SETC contains the stripping result for each event on the stripped DST, and can be used by analysis jobs to select only events passing a given stripping selection in a given stripping stream.

Open questions

- Which DaVinci version and options should be used? This functionality has not yet been implemented.

Juan will give to Anton the stripped DST from the Step1 tests, who will then make the necessary changes to DaVinci. In any case this step is performed in a separate production from steps 1 to 3.

-- MarcoCattaneo - 2009-09-29

This topic: LHCb > StrippingWorkFlow
Topic revision: r6 - 2009-10-01 - unknown