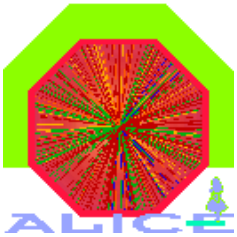
EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
European Laboratory for Particle Physics



CERN

CH-1211Genève 23
Switzerland

Internal Note/
ALICE reference number
ALICE-INT-2005-019 Version 2
Institute reference number
[-]
Date of last change
2005-09-15

Start of data and end of data events in ALICE

Authors:

F. Carena / CERN-PH, W. Carena / CERN-PH, D. Evans / Birmingham,
P. Jovanovic / Birmingham, A. Jusko / Birmingham, R. Lietava / Birmingham,
J.C. Marin / CERN-PH, K. Schossmaier / CERN-PH, P. Vande Vyvre / CERN-PH,
O. Villalobos Baillie / Birmingham

Abstract:

This note proposes a way to orderly perform the start-of-run and the end-of-run in ALICE. The proposal is to make the Trigger system to trigger the generation of the *start of data* and the *end of data* events using the software trigger option. The generation of minimum *start of data* and *end of data* events shall be mandatory for the readout electronics of all the detectors.

1. Introduction

The general concepts of the control of the ALICE online systems have been presented in the Trigger, DAQ, HLT, Control Technical Design Report [1]. This note proposes a way to perform well orchestrated start-of-run and end-of-run in ALICE.

At the end of a data-taking period, all the programs handling the DDLs (DATE *readout*) have to be stopped in a coordinated way in order to prevent some of them still generating fragments of events while others have already been halted. The failure to do so would produce incomplete events, causing the event builders to generate error messages at the end of run. This difficulty arises because the *readout* program does not have any information about the last event which has been triggered by the Central Trigger Processor (CTP).

A possible solution consists of defining an end-of-run timeout: during the timeout, all the *readout* programs would continue to receive the data, but they would stop when the timeout expires. The problem is that it is always tricky to properly set a timeout: either it is too short and the timeout expires too early, or it is too long and time is lost waiting for its completion. This is not relevant if the wait time is only used in abnormal working conditions, but this is not the case here. In normal conditions, all *readout* programs would wait for the entire timeout period. This would definitely be a waste of time during the installation and commissioning period when the data taking is frequently started and stopped.

A more elegant solution is to use a special end-of-file (or *end of data*) event that will unambiguously mark the end of dataflow. When this special event appears in the data flow, the *readout* program could immediately start the end-of-run procedure without any further wait. A timeout interlock has to be set in order to detect the case where the end-of-file marker is missing, but this would only happen in abnormal working conditions.

Similarly, the presence of a start-of-file (or *start of data*) event generated at the beginning of run, even before the calibration procedure, would allow a quick check that all the data flow chains are working properly.

The proposal is to use the trigger system as the central source triggering the generation of *start of data* and *end of data* events. The *start of data* and *end of data* events will be generated by the readout electronics of all the detectors and transmitted over DDL to the DAQ and the HLT as normal events. The generation of such events will be based on the existing hardware foreseen for the trigger, the detector readout, the DAQ, and the HLT. Some modification of the software will be required; they can be completed before the beginning of detector integration at Point 2 in November 2005.

2. The start-of-run and end-of-run sequences

The Start-Of-Run (SOR) and End-Of-Run (EOR) sequences involve several systems (TRG, DAQ, HLT, and DCS) and are therefore controlled by the Experiment Control System (ECS). The sequence of the start of run shall be as follows:

- The ECS initiates and configures the TRG, DAQ, HLT, and DCS.
- The ECS sends a command to the trigger system requesting the generation of a special software trigger defined as the SOR trigger.
- The trigger system will wait for all detectors to be ready to receive a trigger and will then issue a software trigger. This trigger would in turn provoke the readout systems of all detectors in the partition to generate a *start of data* event fragment. The generation of this special software trigger shall be supported for global partitions and in standalone mode.
- All these *start of data* event fragments shall then be sent to DAQ and HLT marking the start of all data streams.

- All the programs handling the dataflow become ready upon reception of the *start of data* event. The DATE *readout* program for example either becomes ready upon reception of the *start of data* event or requests the run control to stop the run with error if the first event fragment received is not a *start of data* event. At the system level, the DAQ for example becomes ready when all the *readout* programs of all LDCs have received a *start of data* event from all active DDLs.
- If the DATE *readout* program does not receive the *start of data* event after the prescribed timeout, it will also request the run control to stop the run with error.
- The ECS monitors this and decides when the run has been properly started.
- The ECS sends a command to the trigger system to enable physics triggers

The state-machine of the DATE *readout* program during this sequence is shown in Fig.1.

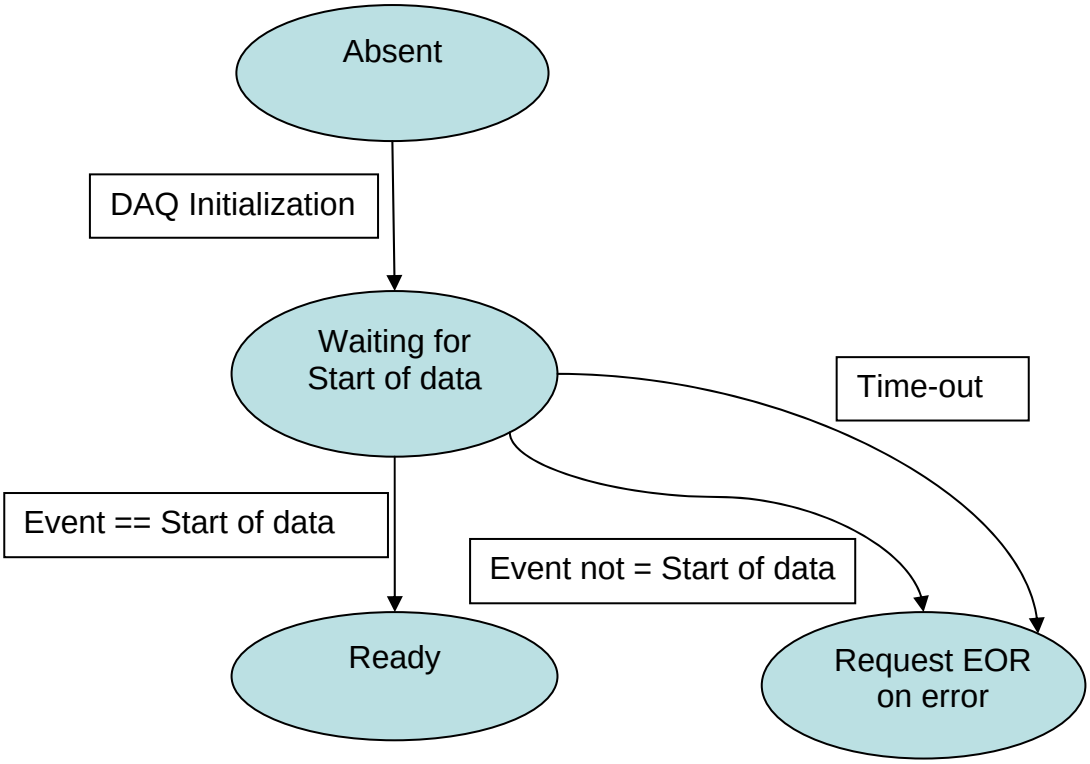


Figure 1: State-machine of the DATE readout program during the Start-Of-Run sequence.

The sequence of the end of run shall be as follows:

- The ECS sends a command to the trigger system to disable physics triggers. It also sends a command to DAQ and HLT to stop the data taking.
- The ECS requests the trigger system to send a special software trigger defined as the EOR trigger. The trigger system waits for all detectors in the partition to become available (not BUSY) before sending the trigger. This trigger provokes in all the detector readout systems the generation of an *end of data* event fragment. It might also be the occasion for the detector electronics to flush their buffers with any pending data block that would result from a malfunction.
- All these *end of data* event fragments shall then be sent to DAQ and HLT marking the end of all data streams.
- After reception of the end-of-run command, all the programs handling the dataflow will keep processing the events while waiting for an *end of data* event. The DATE *readout* program for example will continue to receive all the previously triggered

physics events from the DDL. As soon as it receives the *end of data* event, it will execute the end-of-run procedure and will then exit with a success condition. At the system level, the DAQ for example becomes ready when all the *readout* programs of all LDCs have received an *end of data* event from all active DDLs.

- If the DATE *readout* program does not receive the *end of data* event after the prescribed timeout, it will also execute the end-of-run procedure but will then exit with an error condition.
- All the programs handling the dataflow could start the end-of-run processing.
- The ECS monitors this and realizes when the run has been properly finished.

3. Generation of *start of data* and *end of data* events

Both the ALICE CTP and Local Trigger Unit (LTU) include the provision for software triggers. These triggers are generated upon request to the CTP or the LTU control software.

The mechanism leading to the generation of the SOR and EOR triggers will slightly differ whether the ECS operates a global partition or a single detector in standalone mode. In the case of a global partition, the ECS Partition Control Agent (PCA) interacts with the CTP through the Trigger Partition Agent (TPA). For a detector in standalone mode, the ECS Detector Agent (DCA) controls the LTU through its control software. In both cases, the ECS issues similar requests for the generation of special software triggers defined as the SOR and EOR triggers. These two cases have to be completely identical and transparent for the detector electronics. The sequence of operation and the content of the messages exchanged at SOR and EOR must be identical in both cases. It must be possible to perform in standalone mode the same tests as the one performed in global mode.

In standalone mode, it shall be the task of the LTU control software to generate the *start of data and end of data* triggers using the CTP emulator.

The SOR and EOR triggers messages shall be identified by the presence of the Software Class bit in the L1 Message and L2 Message (L1SwC and L2SwC respectively) and the readout Control Bits (RoC):

- Bit 4 of Word 1 of the L1 Message (Software Class L1 trigger status asserted: L1SwC=1)
- Bits 6 to 9 of Word 1 of the L1 Message (Read-out Control bits used to distinguish SOR - RoC [4..1]=0xE - from EOR - RoC[4..1]=0xF). The RoC[4..1] contents 0xE and 0xF are among *reserved codes* (indicated with RoC[4] = 1) that detectors must be able to decode, but are not allowed to use for any other purpose.
- Bit 8 of the Word 4 of the L2a Message (Software Class L2 trigger status asserted: L2SwC=1).

Upon reception of the SOR and EOR software triggers, the detector readout shall send over the DDL an event fragment including at least a header following the standard format described in [2]. The format already includes the L1SwC and RoC fields. The presence of these 2 fields will allow the decoding and the processing of the *start of data* and *end of data* pseudo-events.

The HLT will also generate *start of data* and *end of data* events so that there is no difference between HLT and other detectors.

4. Other applications of the *start of data* and *end of data* events

The generation of an empty event (including a single header) shall be mandatory. However, the detector groups might decide to use the SOR and EOR events for other applications.

The detectors have to be fully initialized in order to be ready for data taking, which starts with a *start of data* event. During the *start of data* event the detectors shall verify that their configuration, stored in the internal configuration registers matches the one set by DCS/ECS.

The EOR trigger might be used to transfer calibrations data computed by the readout electronics or a summary of all the errors.

5. Conclusion

This note proposes a way to orderly perform the start-of-run and the end-of-run in ALICE. The proposal is to make the Trigger system to trigger the generation of the *start of data* and the *end of data* events using the software trigger option. The generation of minimum *start of data* and *end of data* events shall be mandatory for the readout electronics of all the detectors. The detector groups are allowed to use the *start of data* and *end of data* events for other applications such as control or transmission of monitoring or calibration data.

6. References

1. ALICE Technical Design Report on Trigger, Data Acquisition, High-Level Trigger and Control System, CERN-LHCC-2003-062.

2. Data format over the ALICE DDL, ALICE Internal Note, ALICE-INT-2002-010.