DAQ Backbone Communication Protocol

Version 9.01
December 7, 2000
Editors: M. J. LeVine, T.J. Ljubicic and J.M. Nelson

1. Introduction

1.1 Communication needs

The protocol described here addresses communication among *entities*, which are processes that are *globally visible* within DAQ. The protocol is intended to meet the needs of all detector broker-event builder-global level 3 communication once a stable configuration is established. The principal need addressed is for intra-event synchronization. The required communications are detailed in the following paragraphs.

Most of the communications fall into two broad categories. The detector brokers [DET, formerly SB] either *report unsolicited results* (e.g., event ready) or *respond to a command* issued by either the global broker [GB] or the event builder [EVB]. The GB and EVB, on the other hand, *issue commands* to the DETs, while the DETs, in turn, issue commands to the SL3s.

1.2 Detector broker – Sector L3 - Global L3, Global broker

The trigger system delivers a token through the Trigger Daq Interface. This is passed to GB, which chooses a GL3 CPU and a suite of SL3 CPUs which will be attached to DETs for this event. Each DET then uses an SL3 CPU to do tracking for its sector; after which the results are then announced by SL3 to the selected GL3. GL3 responds by commanding each SL3 to send its results. After a decision has been made by GL3, this decision is communicated to GB, the GB may tell the detector broker to release its buffers associated with the token.

1.3 Sector broker- event builder

When an event is accepted by GL3, GB communicates this decision to EVB, which commands the DETs to format their data for event building. The DETs, in turn, announce the size of their contributed fragment.

EVB responds by commanding each detector broker to send its fragment, or some portion of it. DET tells event builder when it has finished. The event builder then may tell the detector broker to release its buffers associated with the token.

9. EVB->DET: EVB Format data (token #, format)
10. DET->EVB: DET Announce data (token #, sizes)
11. EVB->DET: EVB Send data (token #, destination addresses)
12. DET->EVB: Confirm send (token #)

These commands will be issued sequentially; i.e., one command must be completed before the next command is issued to a given DET.

2. DAQ Protocol

This scenario is based on an implementation of message passing. The message data payload can be as large as 108 bytes, and must be built locally by the sender. The strategy described here attempts to minimize the number of interrupts to which the CPUs have to respond. This is done by removing all handshakes. Instead, the global broker keeps track of the progress of an event.

2.1 DAQ packet format

The packets to be used in DAQ for message delivery carry a payload of 108 bytes. The DAQ message packets look like the following:

31 0

<table>
<tr><td>DAQ cmd</td><td colspan="2">dst task</td><td>status (4)</td><td colspan="2">token (12)</td></tr>
<tr><td colspan="2">valid words (12)</td><td>domain (4)</td><td colspan="3">source ID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">command specific (up to 108 bytes)</td></tr>
</table>

The **dst task** (byte) identifies the destination software entity. The defined values are identical to those defined for src task. These two fields are intended to be the primary source of information used by an interrupt service routine to decide the appropriate destination or action on receipt of a packet, rather than the DAQ node ID.

The **valid words** field (12 bits) specifies the number of 32-bit words which follow, starting with the word containing *transaction #*. Meaningful values are [1,28]. A value of 0 is an error.

The domain and source ID values are defined in
`/RTS/include/rtsSystems.h`
and reproduced here. [This file is automatically included in **`/RTS/include/iccp.h.`**]
The **domain** (4 bits) have the following values::

0: use best transport
1: Ethernet
2: VME backplane
3: Myrinet
4: EVB_local
5: SB_local
6: MZ_local
7: Sector

The **source ID** (16 bits) has the following meaning assigned to its bit fields beginning with the MS 4 bits:

Detector (4 bits):
0: TPC
1: SVT
2: TOF
3: EMC
4: PMD
5: FTPC
6: Reserved
7: RICH
8: TRG

9: L3
10: Slow Controls
15: DAQ

Route (1 bit):
0: Normal routing
1: Alternate routing

Subtype (2 bits):

0: Main
1: Global Level 3; TRG_L1
2: Sector Level 3; TRG_L2
3: Reserved

Instance (8 bits)
identifies a particular instance of this detector

The **transaction number** (16 bits) is an incrementing number generated in software which allows a response to uniquely identify the message to which it applies.

The **src task** (byte) identifies the sending software entity. For task values which are currently defined, see

```
/RTS/include/tasks.h
```

The **DAQ cmd** field is 8 bits wide. These commands are described in detail in the following paragraphs. It should be noted that most commands issued by the global CPUs do not require an immediate response; the response comes when the activity requested by the command has completed, and is normally specific to that activity. For example, the EVB *Format data* command expects to receive a *DET Announce data* message.
Addresses used as arguments in commands will always be **expressed in the PCI address space** of the message originator. These addresses are always meant to be used as destinations for transporting data.

Intra-event messages

2.1 Initial token handling at the "software" level:

Steps in italics are optional.

1.	TDI->GB	ANNOUNCE_TOKEN	0x08
2.	GB->GL3	ANNOUNCE_TOKEN	0x08
3.	GL3->GB	SEND_SUMMARY	0x0A
4.	GB->GL3	CONFIRM_SEND	0x09
5.	GL3->SL3	GL3_ANNOUNCE_TOKEN	0x74
6.	SL3->DET	ANNOUNCE_NODEID	0x05
7.	DET->SL3	DET_ANNOUNCE_RESULTS	0x90
8.	SL3->DET	SL3_SEND_RESULTS	0xA0
9.	DET->SL3	DET_START_SL3	0x92
10a.	*SL3->GL3*	*SEND_SUMMARY*	*0x0A*
10b.	*GL3->SL3*	*CONFIRM_SEND*	*0x09*
11.	SL3->GL3	SL3_ANNOUNCE_RESULTS	0xA1
12.	GL3->SL3	GL3_SEND_RESULTS	0x70
13.	SL3->GL3	SL3_CONFIRM_SEND	0xA2
14.	GL3->GB	GL3_BUILD_DECISION	0x73

2.2 If GL3 decision is ACCEPT, then

15.	GB->EVB	GB_BUILD_EVENT	0x80
16.	EVB->DET/GB/GL3	EVB_FORMAT_DATA	0x51
17.	GB->EVB	GB_ANNOUNCE_DATA	0x81
17.	GL3->EVB	GL3_ANNOUNCE_DATA	0x72
17.	DET->EVB	DET_ANNOUNCE_DATA	0x93
18.	EVB->DET/GB/GL3	EVB_SEND_DATA	0x52
19.	DET/GB/GL3->EVB	CONFIRM_SEND	0x09
20.	EVB->GB	EVENT_DONE	0x04
21.	GB->TDI	EVENT_DONE	0x04

2.3 If GL3 decision is REJECT, then

22.	GB->DET/GL3	RELEASE_TOKEN	0x07
23.	GL3/DET>GB	CONFIRM_RELEASE_TOKEN	0x06
24.	GB->TDI	EVENT_DONE	0x04

2.4 If there is no TRG in the setup, then the DETs must announce the trigger to the TDI simulator which takes over the role of TDI

0.	DET->TDI_SIM	ANNOUNCE_TOKEN	0x08
1.	TDI_SIM->GB	ANNOUNCE_TOKEN	0x08

then follows the sequence above until 24:

24.	GB->TDI_SIM	EVENT_DONE	0x04

2.5 If there is no L3 processing in an event then:

1.	TDI->GB	ANNOUNCE_TOKEN	0x08
2.	GB->DET	ANNOUNCE_NODEID	0x05

GB will then make the build/reject decision and the sequence under 2.4 and 2.5 follows except there is no interaction with GL3

Comments

ANNOUNCE_NODEID

DETs have to be informed of the NodeId of the SL3 they are to use. When L3 processing in active, SL3 will issue this command. Otherwise it will be issued by GB to inform DETs that L3 processing is inactive as well as the status of L2.5 abort.

DET_ANNOUNCE_RESULTS

When DET uses an SL3 for work, it informs the SL3 what type of detector DET is. (TPC or SVT etc). SL3 will later inform GL3 of the type so that GL3 knows how to process the data SL3 sends it.

DET_ANNOUNCE_TOKEN

7 words of Event Descriptor. The event descriptor may or may not be used by TDI_SIM.

ANNOUNCE_TOKEN

Event Descriptor (7 words)
Size (in words) of trigger summary
Size (in words) of trigger data block
Number of SL3/DET Node ID pairs to follow

Array (short) of SL3 Node IDs (ie. two per word) followed by array of DET Node IDs

SEND_SUMMARY

NPRE (MS 16 bits), NPOST(LS 16 bits)
Address for receipt of trigger summary (if non-zero)
Address for receipt of trigger data blocks (if non-zero)

NPRE and NPOST are the number of pre- and post-trigger data blocks required to be transmitted. These numbers must not exceed the corresponding values supplied in the event descriptor. If they do exceed the maximum, then the limit set in the event descriptor will be used and the status word in the message packet will be set to 1 if NPRE is exceeded, 2 if NPOST is exceeded and 3 if both are exceeded. If NPRE=NPOST=0 then only the principal trigger data block will be transmitted (if the relevant address is non-zero). Otherwise, the order of transmission of trigger data blocks will be (in a single transmission):
a) principal trigger data block
b) NPRE trigger blocks
c) NPOST trigger blocks

CONFIRM_SEND

After data suppliers have confirmed that their data have been sent, they should release all internal structures for that token. A RELEASE_TOKEN command will not be issued.

GL3_BUILD_DECISION

MS 16 bits: 1 (accept)/0 (reject)
LS 16 bits: 1 (Build)/0 (No build). Accept/reject is used for bookkeeping only. BUILD is used by GB to determine whether to build the event.
L3 status (4 words)

GB_BUILD_EVENT

MS 16 bits: 1 (accept)/0 (reject)
LS 16 bits: 1 (Build)/0 (No build)
Event Descriptor (7 words)
L3 status (4 words)
L2 status (1 word)
L1 status (1 word)

3. DAQ messages

3.1 General messages

Command: **Ping** PING

31 0

<table>
<tr><td>Command=0x01</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Required response: ACK.
Field **token number**: 12 bits

Command: **Acknowledge** ACK

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x03</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Field **token number**: 12 bits

Command: **Event done** EVENT_DONE

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x04</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Used by GB or EVB to inform a recipient that an event is complete and that structures may be released. No reply is required.

Required response: none.

Field **token number**: 12 bits

Command: **Announce nodeID** ANNOUNCE_NODEID

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x05</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">SL3 NodeID</td><td colspan="2"></td><td>L2.5 Abort</td></tr>
</table>

Used by GB or SL3 to inform DET of the NodeId of the SL3 it is required to use. If GB uses this command, then either no L3 processing is required, or a L2.5 abort is in progress. In both cases, SL3 NodeID will be zero. The L2.5 Abort byte is zero in all cases, except if GB is signalling an abort, when this field is 1. SL3 will duplicate its sourceID in the SL3 NodeId field.

Required response: none.

Field **token number**: 12 bits

Command: **Release token** RELEASE_TOKEN

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x07</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Used by GB or EVB to command a task to release buffers associated with the specified token number

Required response: Confirm release token [see below]

Field **token number**: 12 bits

Command: **Confirm release token** `CONFIRM_RELEASE_TOKEN`

31 0

<table>
<tr><td>Command=0x06</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Field **token number**: 12 bits

Command: **Announce token** `ANNOUNCE_TOKEN`

31 0

<table>
<tr><td>Command=0x08</td><td colspan="3">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">12 + (node_Id Pairs)</td><td colspan="2">domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="4">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="7">reserved for use by GL3</td></tr>
<tr><td colspan="7">Event descriptor (word 1)</td></tr>
<tr><td colspan="7">•••</td></tr>
<tr><td colspan="7">Event descriptor (word 7)</td></tr>
<tr><td colspan="7">trigger summary size</td></tr>
<tr><td colspan="7">TRG detector data size</td></tr>
<tr><td colspan="7">Number of SL3/DET Node Id pairs to follow</td></tr>
<tr><td colspan="3">SL3 NodeID 1</td><td colspan="4">DET NodeID 1</td></tr>
<tr><td colspan="3">SL3 NodeID 2</td><td colspan="4">DET NodeID 2</td></tr>
<tr><td colspan="3">••••</td><td colspan="4">••••</td></tr>
<tr><td colspan="3">SL3 NodeID n</td><td colspan="4">DET NodeID n</td></tr>
</table>

Used by the TDI to announce event to GB. Also used by the GB to announce to the GL3 that the trigger data are available. The GL3, in turn, forwards this message to the SL3 CPUs listed in the list of SL3 node Ids. [The GL3 may forward a truncated version of this message, i.e., minus the list of SL3 node Ids, after adjusting the *valid words* field.] The SL3 CPU is expected to issue a SEND_SUMMARY command to get any part of the summary record it needs.

It is expected that the trigger summary data will be valid until the token is released.

Field **event descriptor**: (7 words) TRG event descriptor

Field **trigger summary size**: (32 bits) number of words in TRG summary

Field TRG **detector data size**: (32 bits) number of words in TRG detector data block for a single bunch crossing.

Field: **Number of SL3/DET node Id pairs** (32 bits): Each SL3 is linked to the respective DET. This command has a variable length depending on the originator. TDI does not include information about SL3/DET pairs, while GB does.

Command: **Confirm send**

`CONFIRM_SEND`

31 0

<table>
<tr><td>Command=0x09</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Field **status**: (4 bits) non-zero indicates inability to complete shipment of SL3/GL3 results.
In the case of SEND_SUMMARY, the requestor should check the event descriptor for a discrepancy with the NPRE, NPOST requested.
Field **token number**: 12 bits

Command: **Send summary**

`SEND_SUMMARY`

31 0

<table>
<tr><td>Command=0x0A</td><td colspan="3">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">6</td><td colspan="2">domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="4">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">NPRE</td><td colspan="4">NPOST</td></tr>
<tr><td colspan="7">trigger summary destination address (upper 32 bits)</td></tr>
<tr><td colspan="7">trigger summary destination address (lower 32 bits)</td></tr>
<tr><td colspan="7">TRG detector destination address (upper 32 bits)</td></tr>
<tr><td colspan="7">TRG detector destination address (lower 32 bits)</td></tr>
</table>

Field **token number**: 12 bits
Fields **destination address:** (64 bits) 4-byte aligned. A value of 0 is to be interpreted as lack of authorization to ship the data referenced. Supplied as 64-bit address to match requirement for Linux destinations.
Field **NPRE**: (16 bits) Number of TRG detector data blocks requested for beam crossings preceding the main crossing.
Field **NPOST**: (16 bits) Number of TRG detector data blocks requested for beam crossings following the main crossing.
Data shall be sent in the following order: principal block, NPRE blocks of preceding crossing data, NPOST blocks of following crossing data.

Command: **Reboot**

`REBOOT`

31 0

<table>
<tr><td>Command=0x0B</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

The command causes each task immediately reset. On VxWorks nodes this corresponds to rebooting the processor. On Unix nodes, the specific meaning is left to the implementation of the task.

3.2 Messages originating in the DET

Command: **DET announce data** `DET_ANNOUNCE_DATA`

31 0

<table>
<tr><td>Command=0x93</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">n+2</td><td>Domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">size (long words) for sector header</td></tr>
<tr><td colspan="6">size (long words) for fragment 1</td></tr>
<tr><td colspan="6">size (long words) for fragment 2</td></tr>
<tr><td colspan="6">●●●</td></tr>
<tr><td colspan="6">size (long words) for fragment n</td></tr>
</table>

Used by the DET to communicate with the EVB. Here n= the number of data fragments (not including the sector header) to be transmitted. The message with accompanying sector header is obligatory even in the absence of other data.
Field **status**: (4 bits) non-zero indicates inability to format data as requested. This signifies that, due to limited resources on the mezzanine board(s), the requested format cannot be prepared.
Field **token number**: 12 bits
Field **size**: 32 bits

Command: **DET announce results** `DET_ANNOUNCE_RESULTS`

31 0

<table>
<tr><td>Command=0x90</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">n+3</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">reserved</td><td colspan="2">detector type</td><td>sector ID</td></tr>
<tr><td colspan="6">Sector header size (words)</td></tr>
<tr><td colspan="6">size (long words) for fragment 1</td></tr>
<tr><td colspan="6">size (long words) for fragment 2</td></tr>
<tr><td colspan="6">●●●</td></tr>
<tr><td colspan="6">size (long words) for fragment n</td></tr>
</table>

Used by the DET to communicate with the SL3. Here n= the number of results fragments (not including the sector header) to be transmitted. The message with accompanying sector header is obligatory even in the absence of other results. DET informs SL3 of its detector type via *detector type* field, so that detectors may eventually access SL3s from a general pool.
Field **status**: (4 bits) non-zero indicates inability to complete cluster-finding as requested.
Field **token number**: 12 bits
Field **detector type**: 8 bits

0. TPC
1. SVT
2. TOF
3. EMC
4. PMD
5. FTPC

6. reserved
7. RICH
8. TRG
9. L3
10. Slow Controls
15: DAQ

Field **Sector ID**: (8 bits) Detector sector number [0,…,23]
Fields **size**: 32 bits

Command: **DET start SL3** `DET_START_SL3`

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x92</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">reserved</td><td colspan="2">detector ID</td><td>sector</td></tr>
</table>

DETs send this command to the selected SL3, in order to initiate processing. This command also serves to notify the SL3 that the requested results from DET have been delivered.

Command: **DET announce token** `DET_ANNOUNCE_TOKEN`

<table>
<tr><td>31</td><td></td><td></td><td></td><td></td><td>0</td></tr>
<tr><td>Command=0x91</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">8</td><td>Domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">Size of sector’s L3 contribution (words)</td></tr>
<tr><td colspan="6">Event descriptor (word 1)</td></tr>
<tr><td colspan="6">●●●</td></tr>
<tr><td colspan="6">Event descriptor (word 7)</td></tr>
</table>

Used by the DET to communicate with TDI_SIM. This will only be used when Trigger is detached from DAQ.
Field **token number**: 12 bits
Field **event descriptor**: (7 32 bit words) DET fills in those fields it can from the fiber header, and fills the remaining fields with 0’s.

3.3 Messages originating in SL3

Command: **SL3 Send Results**

SL3_SEND_RESULTS

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0xA0</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2N+2</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">reserved</td><td colspan="2">detector ID</td><td>sector</td></tr>
<tr><td colspan="6">SL3 header destination address (upper 32 bits)</td></tr>
<tr><td colspan="6">SL3 header destination address (lower 32 bits)</td></tr>
<tr><td colspan="6">SL3 results fragment 1 destination address (upper 32 bits)</td></tr>
<tr><td colspan="6">SL3 results fragment 1 destination address (lower 32 bits)</td></tr>
<tr><td colspan="6">•••</td></tr>
<tr><td colspan="6">SL3 results fragment N destination address (upper 32 bits)</td></tr>
<tr><td colspan="6">SL3 results fragment N destination address (lower 32 bits)</td></tr>
</table>

This message is sent by SL3 to DET to request delivery of previously announced DET found clusters.
Field **status**: (4 bits) non-zero indicates inability to complete SL3.
Field **token number**: 12 bits
Value **N**: taken from DET Announce results.
Field **SL3 destination address**: (64 bits) Address, 4-byte aligned, to which DET cluster results are to be delivered. A value 0 signifies denial of authorization to send fragment. Address of 64 bits required for Linux destinations.

Command: **SL3 Announce Results**

SL3_ANNOUNCE_RESULTS

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0xA1</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">3</td><td>domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">number of tracks</td><td colspan="2">detector type</td><td>sector ID</td></tr>
<tr><td colspan="6">SL3 results size (long words)</td></tr>
</table>

SL3 notifies GL3 that it has results available.
Field **status**: (4 bits) non-zero indicates inability to complete SL3.
Field **token number**: 12 bits
Field **SL3 results size**: 32 bits. Size of SL3 contribution
Zero size for this field signifies contribution not available.
Field **detector type**: 8 bits

0. TPC
1. SVT
2. TOF
3. EMC
4. PMD
5. FTPC
6. reserved
7. RICH
8. TRG
9. L3

10. Slow Controls
15: DAQ

Field **Sector ID**: (8 bits) Detector sector number [0,…,23]
Fields *detector type, sector ID* identify the detector sector for which the SL3 performed the calculations.

Command: **SL3 Confirm send** SL3_CONFIRM_SEND

31 0

<table>
<tr><td>Command=0xA2</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">reserved</td><td colspan="2">detector ID</td><td>sector</td></tr>
</table>

Field **status**: (4 bits) non-zero indicates inability to complete shipment of SL3/GL3 results.
In the case of SEND_SUMMARY, the requestor should check the event descriptor for a discrepancy with the NPRE, NPOST requested.
Field **token number**: 12 bits

3.4 Messages generated by EVB

Command: **EVB Format data** EVB_FORMAT_DATA

31 0

<table>
<tr><td colspan="2">Command=0x51</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="3">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="4">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td>CS</td><td colspan="6">format type</td></tr>
</table>

EVB Format data is sent by EVB to the DETs to command them to format data, and to specify which data and which formats to use. The same command is used to request preparation of results from GB and GL3. [Note: EVB_FORMAT_SPECIAL_EVENT no longer exists; these formats are now part of EVB_FORMAT_DATA.]
Field **token number**: 12 bits
Field **CS:** 8 bits. If 1, require checksum calculation at source (debugging only)
Format type (24 bits) has the following bits defined **for DET**

0: ADC data, zero suppressed
1: ADC data, non zero-suppressed
2: ADC data Huffman coded
3: ASIC cluster pointer pairs
4: Cluster data
8: Pedestals
9: Pedestal RMS
10: Gain
11: Bad channel list
12: Configuration bank

At most one of these bits may be set to '1'. If none of these bits is set to '1' the formatting decision is left to the mezzanine CPU. The default contribution is ADC data only, if zero-suppressed, ADC data plus ASIC cluster-pointer pairs if not suppressed.
Format type (16 bits) has the following bits defined **for L3**:

0: Track information
1: Cluster information

2: reserved
3: reserved

Command: **EVB Send data** EVB_SEND_DATA

31 0

<table>
<tr><td>Command=0x52</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">n+1</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">destination address for fragment 1</td></tr>
<tr><td colspan="6">destination address for fragment 2</td></tr>
<tr><td colspan="6">•••</td></tr>
<tr><td colspan="6">destination address for fragment n</td></tr>
</table>

Sent to the DET, GB, or GL3 to authorize transmission of formatted data for event building. The valid words field = the number of authorized data fragments+1. The number of fragments includes the header(s), which is obligatory even in the absence of other data. The maximum number of data transfers is 13. All destination addresses must be aligned on 4-byte boundaries. The DET, GB, or GL3 responds only when *all specified transfers* have completed.
Field **token number**: 12 bits
Field **destination address**: (32 bits) Address, 4-byte aligned.

3.5 Messages issued by GL3 and GB

Command: **GL3 Announce Token** GL3_ANNOUNCE_TOKEN

31 0

<table>
<tr><td>Command=0x74</td><td colspan="3">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">12 + (node_Id Pairs)</td><td colspan="2">domain</td><td colspan="3">sourceID</td></tr>
<tr><td colspan="4">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="4">format</td><td colspan="2">detector ID</td><td>sector</td></tr>
<tr><td colspan="7">Event descriptor (word 1)</td></tr>
<tr><td colspan="7">•••</td></tr>
<tr><td colspan="7">Event descriptor (word 7)</td></tr>
<tr><td colspan="7">trigger summary size</td></tr>
<tr><td colspan="7">TRG detector data size</td></tr>
<tr><td colspan="7">Number of SL3/DET Node Id pairs to follow</td></tr>
<tr><td colspan="3">SL3 NodeID 1</td><td colspan="4">DET NodeID 1</td></tr>
<tr><td colspan="3">SL3 NodeID 2</td><td colspan="4">DET NodeID 2</td></tr>
<tr><td colspan="3">••••</td><td colspan="4">••••</td></tr>
<tr><td colspan="3">SL3 NodeID n</td><td colspan="4">DET NodeID n</td></tr>
</table>

Used by GL3 to announce event to SL3. Also used by the GB to announce to the GL3 that the trigger data are available. The GL3, in turn, forwards this message to the SL3 CPUs listed in the list of SL3 node Ids. [The GL3 may forward a truncated version of this message, i.e., minus the list of SL3 node Ids, after adjusting the *valid words* field.] The SL3 CPU is expected to issue a SEND_SUMMARY command to get any part of the summary record it needs.

It is expected that the trigger summary data will be valid until the token is released.
Field **event descriptor**: (7 words) TRG event descriptor
Field **trigger summary size**: (32 bits) number of words in TRG summary
Field TRG **detector data size**: (32 bits) number of words in TRG detector data block for a single bunch crossing.
Field: **Number of SL3/DET node Id pairs** (32 bits): Each SL3 is linked to the respective DET. This command has a variable length depending on the originator. TDI does not include information about SL3/DET pairs, while GB does.

Command: **GL3 Send results** GL3_SEND_RESULTS

31 0

<table>
<tr><td>Command=0x70</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">Token number</td></tr>
<tr><td colspan="2">4</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">reserved</td><td colspan="2">detector ID</td><td>sector</td></tr>
<tr><td colspan="6">SL3 results destination address (upper 32 bits)</td></tr>
<tr><td colspan="6">SL3 results destination address (lower 32 bits)</td></tr>
</table>

GL3 sends this to SL3 to request shipment of results.
Field **token number**: 12 bits
Fields **destination address**: (64 bits) Destination addresses must be aligned on 4-byte boundaries. A null (0) address signifies unauthorized transfer of this fragment. Address of 64 bits required for Linux destinations.

Command: **GL3 Announce data** GL3_ANNOUNCE_DATA

31 0

<table>
<tr><td>Command=0x72</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">Transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">Size of L3 contribution (words)</td></tr>
</table>

Used by the GL3 to announce event to EVB..
Field **status**: 4 bits. If ≠0, GL3 has been unable to format data
Field **token number**: 12 bits

Command: **GL3 build decision** GL3_BUILD_DECISION

31 0

Command=0x73	dst task	not used	token number
6	domain	SourceID	
Transaction #	reserved	src task	
Accept	Build	GL3 Node ID	
L3 summary (word 1)			
L3 summary (word 2)			
L3 summary (word 3)			
L3 summary (word 4)			

Message sent by GL3 to GB.
Field **Accept**: (8 bits) 1 means event accepted by L3. **For bookkeeping purposes only**
Field **Build**: (8 bits) 1 means build event (regardless of accept). **This field determines GB action.**
Field **GL3 Node ID**: (16 bits) ID of GL3 CPU assigned to this token.
Field **L3 summary**: (4 words)
Field **token number**: 12 bits

Command: **GB Announce data** GB_ANNOUNCE_DATA

31 0

Command=0x81	dst task	status	token number
2	domain	SourceID	
Transaction #	reserved	src task	
Size of TRGD contribution (words)			

Used by the GB to announce event to EVB. EVB is expected to construct the (TRGP) header.
It is expected that the trigger summary data will be valid until the token is released.
Field **status**: 4 bits. If ≠0, TRG has cancelled event.
Field **token number**: 12 bits

Command: **GB Build event** GB_BUILD_EVENT

31 0

Command=0x80	dst task	not used	token number
15	domain	SourceID	
Transaction #	reserved	src task	
Accept	Build	GL3 Node ID	
event descriptor (7 WORDS)			
L3 summary (4 words)			
L2 summary			
L1 summary			

This command is used by GB to announce to EVB that an event needs to be built.
The body of the message contains a copy of the event descriptor passed by TRG. (For details see STAR Trigger-DAQ Interface Specification.)
No response is required.
Field **token number**: 12 bits
Field **Accept**: 8 bits
Field **Build**: 8 bits
Field **GL3 Node ID**: (16 bits) ID of GL3 CPU assigned to this token.
Field **Event descriptor**: 7 32 bit words
Field **L3 summary**: 4 32 bit words
Field **L2 summary**: 32 bits
Field **L1 summary**: 32 bits

3.6 *Messages issued by EVB to the Taper*

Command: **EVB tape request** **EVB_TAPE_REQUEST**

31 0

<table>
<tr><td>Command=0x53</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">Transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">Run Number</td></tr>
<tr><td colspan="6">Size of event</td></tr>
<tr><td colspan="6">Offset</td></tr>
<tr><td colspan="6">Event</td></tr>
<tr><td colspan="6">Event type</td></tr>
<tr><td colspan="6">Token</td></tr>
</table>

Field **status**: (4 bits) non-zero indicates inability to format data.
Field **token number**: 12 bits

Command: **Taper Done** **TAPER_DONE**

31 0

<table>
<tr><td>Command=0x54</td><td colspan="2">dst task</td><td>not used</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">1</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">Transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
</table>

Field **token number**: 12 bits

3.7 *RTS Manager commands*

Almost all of the RTS Manager commands are distributed to every major task. The tasks reply to the Run Control Handler by returning the same command with the status indicating whether the command was successfully accomplished. The following tasks receive Run Control commands:

All DET Brokers
GB

EVB
All SL3's
All GL3's
Trigger component tasks

Command: **RTS Run start** `RTS_RUN_START`

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x30</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">run number</td></tr>
</table>

The command starts a run. The task returns STAT_OK when it is ready to start accepting new tokens.

Command: **RTS Run stop** `RTS_RUN_STOP`

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x31</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">run number</td></tr>
</table>

The command stops a run. The task returns STAT_OK when the run is stopped. If the task is unable to stop the run because it is still processing tokens it indicates this by return a non-zero status
GB injects token 0 into system, returns to RC only when all tokens have been returned to it, including token 0.

Command: **RTS Run pause** `RTS_RUN_PAUSE`

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x32</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">run number</td></tr>
</table>

The command pauses a run. The meaning of a paused run is that no tokens will come from trigger, and that no tokens remain in the DAQ system. Eventually, some reconfiguration of DAQ, in particular of the EVB, may be allowed, but at present the only commands that will be received by paused tasks are DAQ_RUN_RESUME and DAQ_RUN_STOP.

Field **status**: (4 bits) Non-zero if there are still tokens in DAQ when command is received.

Command: **RTS Run resume**

RTS_RUN_RESUME

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x33</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">run number</td></tr>
</table>

The command resumes a run. This task returns STAT_OK when the task is ready to begin accepting tokens

Command: **RTS Send configuration**

RTS_SEND_CONFIG

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x34</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">3</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">Port</td></tr>
<tr><td colspan="6">Handler</td></tr>
<tr><td colspan="6">Payload Word N</td></tr>
</table>

See /RTS/include/RC_Config.h for specific details.

Command: **RTS Flush tokens**

RTS_FLUSH_TOKENS

<table>
<tr><td colspan="3">31</td><td colspan="3">0</td></tr>
<tr><td>Command=0x39</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">run number</td></tr>
</table>

The command causes each task to return to the ready state immediately.

Command: **RTS Query tokens** `RTS_QUERY_TOKENS`

31 0

<table>
<tr><td>Command=0x4B</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">2,..,14</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="3">Task Status</td><td colspan="3"># of Tokens</td></tr>
<tr><td colspan="6">DWORD 1</td></tr>
<tr><td colspan="6">DWORD 2</td></tr>
<tr><td colspan="6">….</td></tr>
<tr><td colspan="6">DWORD 12</td></tr>
</table>

The task receiving Query Tokens returns the number of active tokens and the tasks current status. It may also fill in up to twelve task dependent words. For the detector brokers these will correspond to the number of active tokens for each receiver board.

Command: **RTS Force Stop** `RTS_FORCE_STOP`

31 0

<table>
<tr><td>Command=0x39</td><td colspan="2">dst task</td><td>status</td><td colspan="2">token number</td></tr>
<tr><td colspan="2">N</td><td>domain</td><td colspan="3">SourceID</td></tr>
<tr><td colspan="3">transaction #</td><td colspan="2">reserved</td><td>src task</td></tr>
<tr><td colspan="6">Null terminated string – variable length payload up to 108 bytes</td></tr>
</table>

Can be used by a task to require RTS to stop a run. The payload must contain a string to provide a message for the user. The string should be inserted by calling MSG_ERR. See **/RTS/include/rtErrMsg.h** for details.
Field **status**: (4 bits) This will be non-zero. Meaning to be defined later.

4. List of commands (sorted by number)

0x01	PING	
0x03	ACK	
0x04	EVENT_DONE	
0x05	ANNOUNCE_NODEID	
0x06	CONFIRM_RELEASE_TOKEN	
0x07	RELEASE_TOKEN	
0x08	ANNOUNCE_TOKEN	
0x09	CONFIRM_SEND	
0x0A	SEND_SUMMARY	
0x0B	REBOOT	
0x30	RTS_RUN_START	
0x31	RTS_RUN_STOP	
0x32	RTS_RUN_PAUSE	
0x33	RTS_RUN_RESUME	
0x34	RTS_SEND_CONFIG	
0x39	RTS_FLUSH_TOKENS	

0x46	RTS_FORCE_STOP	
0x4B	RTS_QUERY_TOKENS	
0x51	EVB_FORMAT_DATA	
0x52	EVB_SEND_DATA	
0x70	GL3_SEND_RESULTS	
0x72	GL3_ANNOUNCE_DATA	
0x73	GL3_BUILD_DECISION	
0x74	GL3_ANNOUNCE_TOKEN	
0x80	GB_BUILD_EVENT	
0x81	GB_ANNOUNCE_DATA	
0x90	DET_ANNOUNCE_RESULTS	
0x91	DET_ANNOUNCE_TOKEN	
0x92	DET_START_SL3	
0x93	DET_ANNOUNCE_DATA	
0xA0	SL3_SEND_RESULTS	
0xA1	SL3_ANNOUNCE_RESULTS	
0xA2	SL3_CONFIRM_SEND	
0xB0	MZ_PROCESS_EVENT	
0xB1	MZ_EMUL_FIBER	
0xC0	RB_ANNOUNCE_RESULTS	Not defined here
0xC1	RB_ANNOUNCE_DATA	Not defined here
0xC2	RB_CONFIRM_SEND_SL3	Not defined here
0xD0	MYRI_QUASI_DIRECT_SEND	
0xE0	EVP_REQ_SYNC	
0xE1	EVP_REQ_EVENT	
0xE2	EVP_REQ_ACCEPT	
0xE3	EVP_REQ_CANCEL	
0xE4	EVP_SND_DONE	
0xE5	EVP_RSP_EVENT	
0xE6	EVP_RSP_CANCEL	
0xE7	EVP_RSP_ACCEPT	
0xE8	EVP_REQ_PING	
0xE9	EVP_RSP_PING	
0xEA	EVP_RSP_SYNC	

5. Code and include samples

Refer to:

/RTS/include/cmds.h.

`/RTS/include/tasks.h.`

Ver	Loc'n	Change	Reason
5.4	all	list field widths	
5.4	format data	rearrange last word, add checksum bit	
5.4	command response	add token number	ease of use for destination task
5.4	send data multiple	deleted	no longer needed
5.4	cancel command	add token number	ease of use for destination task
5.4	1.1.2	add release token following event build completion	allows for debugging using multiple formats of same event
5.4		add SB announce SL3 results	conform with naming convention
5.4		add GL3 release token	conform with naming convention
5.4		change name to EVB release token	conform with naming convention
5.4		add GL3 send SL3 results	conform with naming convention
5.4		command response now fixed length	no need for variable length
5.4	several	valid word changed	correction of errors
5.4	2.1	address space defined	
5.4	2.1	Acknowledge message added	ease of debugging
5.4	2.3	SB Event status add SB field	better diagnostics
5.4	2.4	added EVB Event status	reply to TM needed
5.4	2.4	added GL3 Event status	reply to TM needed
5.4	2.2	Added general message category	better organization
5.4	2.0	added warning on node ID	hardware "feature"
5.5	2.3	add SB confirm release	implementation of TM
5.5	2.3	add SB confirm formatted data	implementation of EVB
5.5	2.3	add SB confirm SL3 results	implementation of GL3
5.5	several	rearrange arguments: token	uniform command format
5.5		merge GL3/EVB Release token	simplicity
5.5	2.5	new category	
6.0	all	completely new message format	required by DMA implementaiton
6.0	2.4	add GL3 formats to EVB Format data	required by EVB
6.0	2.5	add command GL3 announce formatted data	required for EVB functionality
6.0	2.5	add command GL3 Confirm formatted data	required for EVB functionality
6.0	all	add mnemonic to all commands	improve document
6.0		add addresses to announce trigger summary	correct oversight in TRG-DAQ interface
6.0		add comment to Release token	clarify usage
6.0		generalize EVB send formatted data	used to send to both GL3 and SB
6.03	2.2	add PING command	

6.03	4.0	add command list	maintainability
6.04	2.4	send data: 16 byte aligned	
6.05		Renumber commands	Better organization
6.05		Add L3 announce results	Needed for L3-GL3 communication
6.05		Add EVB format results	Format bits different from SB’s
6.05		Change EVB format data	Add format bits
6.05		Change GL3 Event status	Add live time, worker node ID
6.05		Modify SB Announce SL3 results	New fields for header, cluster,,, sizes
6.05		Remove include file	Document maintainability
6.05		Modify GL3 Send SL3 results	Separate destinations for header,…
6.05a	2.6	added section with GL3-L3	
6.05a	2.6	added GL3 Announce SL3 results	
6.05b	2.4	added: EVB Format special event	
6.05c	2.8	added DAQ write end of event	
6.05c	2.4	modify ANNOUNCE_TRG_SUMMARY	remove struct description – belongs in TRG-DAQ interface document
6.05c	1.12	change protocol description	release token now implicit on build
8.0		Complete reorganization	
8.01		Destination addresses now 64 bits: SEND_SUMMARY, SEND_RESULTS, GL3_SEND_RESULTS	Needed for Linux destinations
8.02		Changed many command numbers	Track cmds.8.0.hh
8.03	2.1	Changed defn of valid words	Also fix valid words in ANNOUNCE_TOKEN, DET_ANNOUNCE_RESULTS, SL3_SEND_RESULTS, GB_BUILD_EVENT, DAQ_QUERY_TOKENS
9.01	all	New protocols to allow for a farm of SL3; absence of TM	Removal of unused commands