

**ITESRE**

**AGILE**

**DOCUMENT TYPE:** INTERFACE CONTROL DOCUMENT

**TITLE:** INTERFACE CONTROL DOCUMENT OF THE AGILE MINICALORIMETER SCIENCE CONSOLE FOR THE TEST BEAM CAMPAIGN (NOVEMBER 2000) AT CERN

**DOCUMENT No.** AGILE-ITE-ICD-002 **PAGE:** I of v,48

**PROJECT Ref.:** AGILE GSE

**ISSUE No.:** 01 **DATE:** December 2000  
Te.S.R.E. Report 281/00

**PREPARED BY:** MASSIMO TRIFOGLIO  
FULVIO GIANOTTI  
JOHN B. STEPHEN

**CHECKED BY:** MASSIMO TRIFOGLIO

**DISTRIBUTION LIST**

<b>NAME</b>	<b>DEPT.</b>	<b>e-mail address</b>	
Labanti Claudio	ITE	labanti@tesre.bo.cnr.it;	
Prest Michela	ITS	Michela.Prest@trieste.infn.it;	
Gianotti Fulvio	ITE	gianotti@tesre.bo.cnr.it;	
John B. Stephen	ITE	stephen@tesre.bo.cnr.it;	
Trifoglio Massimo	ITE	trifoglio@tesre.bo.cnr.it;	

**CHANGE RECORD**

Issue Draft 00/A                      First draft circulated within TESRE, on 6 November 2000.

Issue 01                                      First issue circulated within TESRE, on December 2000.

Note:    After the first issue, text modifications respect to the previous version are identified by a vertical bar on the right side

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>1</b>
1.1	PURPOSE AND SCOPE.....	1
1.2	REFERENCE DOCUMENTS.....	2
<b>2</b>	<b>TEST SCENARIO.....</b>	<b>3</b>
2.1	CsI BARS STANDALONE TESTS.....	3
2.1.1	Test setup.....	3
2.1.2	Data acquisition and management.....	5
2.2	BEAM TESTS.....	9
2.2.1	Test set up.....	9
2.2.2	Data acquisition and management.....	10
<b>3</b>	<b>INFN TE PACKET FORMAT.....</b>	<b>14</b>
3.1	PSEUDO TM PACKET FORMAT.....	14
3.1.1	Packet header.....	14
3.1.2	Packet Data Field.....	15
3.2	PSEUDO TC PACKET FORMAT.....	16
3.2.1	TC Packet header.....	16
3.2.2	TC Packet Data Field.....	17
<b>4</b>	<b>SCIENCE CONSOLE ARCHIVING.....</b>	<b>18</b>
4.1	ON-LINE ARCHIVING.....	18
4.1.1	Raw archiving.....	18
4.1.2	Raw Data Processing.....	19
4.1.3	FITS Archiving.....	19
4.2	THE PERMANENT ARCHIVE ON CD-ROM.....	20
4.2.1	Move off-line and initialise the on-line Archive.....	23
4.2.2	Remove Runs from the off-line Archive.....	24
4.2.3	Create the Run List in the off-line Archive.....	25
4.2.4	Copy the off-line Archive to the CD Area.....	27
4.2.5	Create the Run Lists in the CD Area.....	30
4.2.6	Verify and close the off-line Archive.....	31
4.2.7	Burn the CDs from the CD Area.....	33
5.2.5	Backing up the off-line Archive.....	34
5.2.6	Generate the Test Campaign log files.....	34
	<b>ANNEX A INFN TE EVENT DATA FORMAT.....</b>	<b>36</b>
	<b>ANNEX B SCIENCE CONSOLE RAW DATA PROCESSING CONFIGURATION.....</b>	<b>37</b>
B.1	PROCESSOR AND SHARED MEMORY CHANNEL ASSIGNMENT.....	37
B.2	Quick Look shared memory data format.....	37
	IT IS NOTED THAT ONLY A SUBSET OF THE EVENT DATA WILL BE PROCESSED BY THE QUICK LOOK.....	39
	<b>ANNEX C SCIENCE CONSOLE FITS FILE FORMATS.....</b>	<b>40</b>
C.1	LABEN TE AND TESRE TE.....	40
C.1.1	FITS format for the LABEN TE Data.....	40

C.1.2 FITS format for the TESRE TE Data ..... 42  
C.2 INFN TE ..... 45  
C.2.1 FITS format for the INFN TE Data ..... 45

## 1 Introduction

### 1.1 Purpose and Scope

The scope of the present note is the test campaign to be carried out at CERN in November 2000 in the framework of the AGILE instrument prototyping.

Its purpose is to define the data formats and the protocols which are relevant to the Minicalorimeter Science Console (MCAL SC), provided by ITESRE/CNR, i.e.:

- the data in input to the MCAL SC;
- the data which are generated by the processing performed by the MCAL SC itself.

As summarised in the Test scenario presented in Section 2, during the November 2000 CERN campaign, the MCAL SC could be interfaced with two of the following systems at a time:

- (a) the LABEN version of the CsI TE Host Computers (LABEN TE Host Computer);
- (b) the TESRE version of the CsI TE Host Computers (TESRE TE Host Computer);
- (c) the Main Console of the Test Equipment provided by INFN – Trieste (INFN TE);

As far as the **data in input** to the MCAL SC are concerned, this document:

- refers to [2] for the data generated by (a);
- refers to [3] for the data generated by (b);
- defines the data generated by (c), namely:
  - in Section 3 and Annex A, defines the structure and the contents of the Data Packets to be produced by the INFN TE;

Concerning the **output data** generated by the MCAL SC with the data received from the systems listed in (a), (b) and (c), this document:

- in Section 4 summaries the archiving operations giving details on the various directory structures forming the temporary and the permanent archives related to the test campaign;
- in Annex B provides an overview of the configuration related to the processing of the raw data;
- in Annex C, details the format for the FITS files produced by the raw data processing.

The proposed formats take into accounts further enhancements which will be required in the future in order to support the AGILE instrument development.

As well, the proposal allows the re-usage of some of the test software already developed by ITESRE for other instruments (EPIC-XMM, IBIS-INTEGRAL, PICsIT-IBIS).

## 1.2 Reference Documents

- [1] “Packet Telemetry Standard”, ESA PSS-04-106 Issue 1, January 1988.
- [2] “M.Trifoglio, F.Gianotti, J.B. Stephen, “User software requirements for the Host Computer of the AGILE Minicalorimeter *CAL-CSI* Test Equipment”, AGILE-ITE-SR-001.
- [3] “M.Trifoglio, F.Gianotti, M.Malaspina, J.B. Stephen, “User software requirements for the Host Computer of the AGILE Minicalorimeter *CSI Bars* TESRE Test Equipment”, AGILE-ITE-SR-002.

## 2 Test Scenario

Two main test scenarios are foreseen:

- the **Standalone Tests**, where the MCAL Science Console shall receive the instrument data from one or from both the CsI Bars Test Equipment available, i.e.:
  - the LABEN TE
  - the TESRE TE;
- the **Beam Tests**, where only one of the above CsI Bars Test Equipment shall be connected to the instrument, and shall be used in order to provide, in parallel, the CsI Bars PD signals to INFN TE, which shall acquire the data from the remaining instrument included in the test set up, and forward a subset of this data to the MCAL Science Console.

In both cases, the TEs shall forward to the MCAL Science Console the instrument data using the ESA Packet Telemetry Standard. The following APIDs shall be assigned to the different systems in order to identify the various TM Packet streams:

System	APID
INFN TE	1285
LABEN TE	1293
TESRE TE	1398

### 2.1 CsI Bars Standalone Tests

#### 2.1.1 Test set up.

The diagram shown below in fig 2.1-1 sketches the configuration of the test set up foreseen for the standalone functional tests and calibration (e.g. with pulse generators or radioactive sources) to be carried out before and after the calibration with the CERN beam.



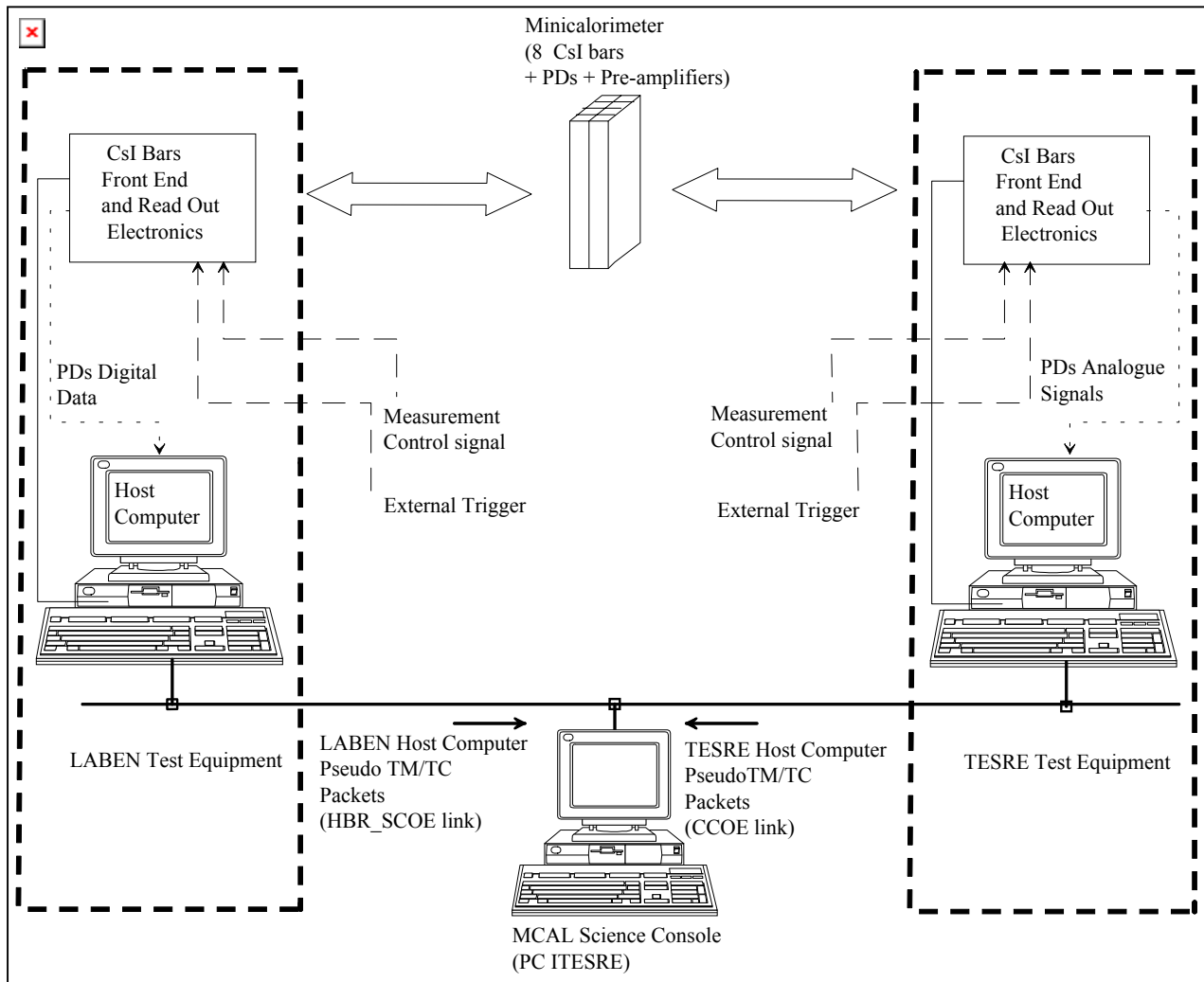


fig. 2.1-1 Set up configuration for CsI Bars standalone tests

With this set up, the MCAL Science Console shall be able to acquire the data either from only one or from both the LABEN Test Equipment and the TESRE Test Equipment.

Two external TTL signals could be provided to each Test Equipment, i.e.:

- the Measurement Control signal shall allow the synchronisation of the measurement time periods on the two Test Equipment, and consequently on the MCAL Science Console.

To this purpose:

- this signal is set to 5V upon a measurement period is started and is reset to 0V once the measurement period is completed;
- this signal is acquired every time the TE acquires the event data, and the correspondent Measurement Control flag is generated and included by the TE in the event data;

- only the event data having the Measurement Control flag set to 1, are archived and processed by the TE;
- all the event data having this flag set to 0 are discarded by the TE.
- the External Trigger signal to be used in the external trigger mode in order to command the peak and hold and the ADC conversion of the PDs analogue lines, to be performed by the CsI Bars TE.

The Host Computers and the MCAL Science Console are connected to a 100BaseT LAN through a Mini HUB having at least 4 100BaseT RJ ports.

### 2.1.2 Data acquisition and management

In order to be activated, each acquisition chain requires that:

- the MCAL Science Console waits, as a Server, in order to establish the related TCP/IP connection, namely:
  - the *HBR\_SCOE* link for the LABEN TE (port 9003);
  - the *CCOE* link for the TESRE TE (port 9002);
- as a Client, the corresponding Host Computer requires and establishes the TCP/IP connection;

Each Data Link shall be kept along the whole test session.

In case the Data Link drops, automatically the Server restarts waiting for a new connection request from the Client.

Through a dedicated I/F, each TE is able to acquire the signals from the CsI Bars PDs, namely:

- in the LABEN TE case, the Host Computer:
  - is able to operate either in external trigger mode on all the 16 PDs, or in auto trigger mode on only one fixed PD;
  - acquires, for each detected event, the digital data generated by the ADC included in the CsI Bars FEE.
  - with PHA digital values, creates a *CsI Bars event data* format (as detailed in Annex A of [2]).
- in the TESRE TE case, the Host Computer
  - is able to operate either in external trigger mode on all the 16 PDs, or in auto trigger mode on all the 16 PDs;

- acquires, for each detected event, the CsI Bars analogue signals, which are then ADC converted by the DAQ I/F card hosted on the Host Computer itself;
  - with PHA digital values, creates a *CsI Bars event data* format (as detailed in Annex A of [3]).
- in both cases:
- with the *CsI Bars event data* format, the Host Computer prepares a TM packet of fixed length having the structure complying with of the ESA Telemetry Standard [1]; each Data Packet shall be capable of containing a fixed number of event data; further details on the format are given in Annex A of [2], and Annex A of [3], for the LABEN TE and the TESRE TE case respectively.
  - as soon as a Data Packet is filled with the event data or the *time-out* is expired since the last packet has been sent, the Host Computer shall send the Data Packet (if not empty) to the MCAL Science Console through the TCP/IP link established on the LAN.

On both the Test Equipment, the running of the acquisition task of the instrument data is commanded by the operator through the Host Computer console.

As already mentioned, only the acquired data having the Measurement Control signal enabled shall be actually archived and processed by the TE. This mechanism shall allow the operator to carry out consecutive measurement in automatic mode, by changing the Measurement Control signal.

This signal, when sent in parallel to both the LABEN TE and the ITESRE TE, shall allow also the synchronisation of the measurement periods on the two Host Computers and consequently on the Science Console.

Indeed, once the operator changes the Measurement Control signal from 0V to 5V in order to start the filing of a new measurement the following actions shall be performed automatically:

- **on the LABEN TE side,**
  - upon the **next** trigger is generated, the Host Computer shall detect that change and:
    - shall increment the RUN ID and open the measurement file using the file naming defined in [2];
    - shall generate and send on the HBR\_SCOE link with the Science Console a **start** measurement TC packet; consequently the Science Console:
      - shall close the *idle* files of the current RUN\_ID;
      - shall open the *measurement* files of the current RUN\_ID, where shall be archived the forthcoming CsI TE packets which shall be sent by the LABEN TE

Host Computer on the HBR\_SCOE link.

- **on the TESRE TE side,**
  - upon the **next** trigger is generated, the Host Computer shall detect that change and:
    - shall increment the RUN ID and shall open the measurement file using the file naming defined in [3];
    - shall generate and send on the CCOE link with the Science Console a **start** measurement TC packet; consequently the Science Console:
      - shall close the *Idle* files of the current RUN\_ID;
      - shall open the *measurement* files of the current RUN\_ID, where shall be archived the forthcoming CsI TE packets which shall be sent by the TESRE TE Host Computer on the CCOE link.

Once the operator will **stop** the current measurement (either from the Host Computer console or by resetting the Measurement Control signal to 0V):

- **on the LABEN TE side,** upon either the console command is detected or the change in the Measurement Control signal change is detected (when the next trigger is generated), the LABEN TE Host Computer:
  - shall close the file of the current measurement;
  - shall generate and send a **stop** measurement TC packet on the HBR\_SCOE link with the Science Console; consequently the Science Console:
    - shall increment the RUN\_ID for the next measurement;
    - shall open the *idle* files of the next measurement.
- **on the TESRE TE side,** upon either the console command is detected or the change in the Measurement Control signal change is detected (when the next trigger is generated), the LABEN TE Host Computer:
  - shall close the file of the current measurement;
  - shall generate and send a **stop** measurement TC packet on the CCOE link with the Science Console; consequently the Science Console:
    - shall increment the RUN\_ID for the next measurement;
    - shall open the *idle* files of the next measurement.

It is noted that, in order to allow the Science Console to complete the filing operations, a new start

measurement could non occur before 20 s since the last stop command.

In addition, it is noted that the Measurement Control flag is included in the event data. Hence, in **automatic mode**, some external trigger must be generated also during the Idle period (e.g. at least at the beginning of the Idle period) in order to allow the MCAL Science Console to acquire data having the Measurement Control signal set to off, and hence to identify that the current measurement is completed and a new set of TM archive files must be created.

The MCAL Science Console shall acquire and archive the Data Packets in one set of files for each measurement, by keeping separated the two packets streams.

By using local commands and without interfering with the Host Computers, at any time the Science Console operator will be able to close all the Science Console files related to the current measurement and to open a new set of Science Console files where all the forthcoming packets will be archived as a new measurement.

The Quick Look software running on the MCAL Science Console, shall allow the user to select one of the two data streams (either TESRE TE or the LABEN TE) on which to provide, in near real time a graphical representation on a percentage of the data.

## 2.2 Beam Tests

### 2.2.1 Test set up.

The Test Equipment set up for the Beam Test Campaign at CERN is depicted in figure 2.2-1 below.

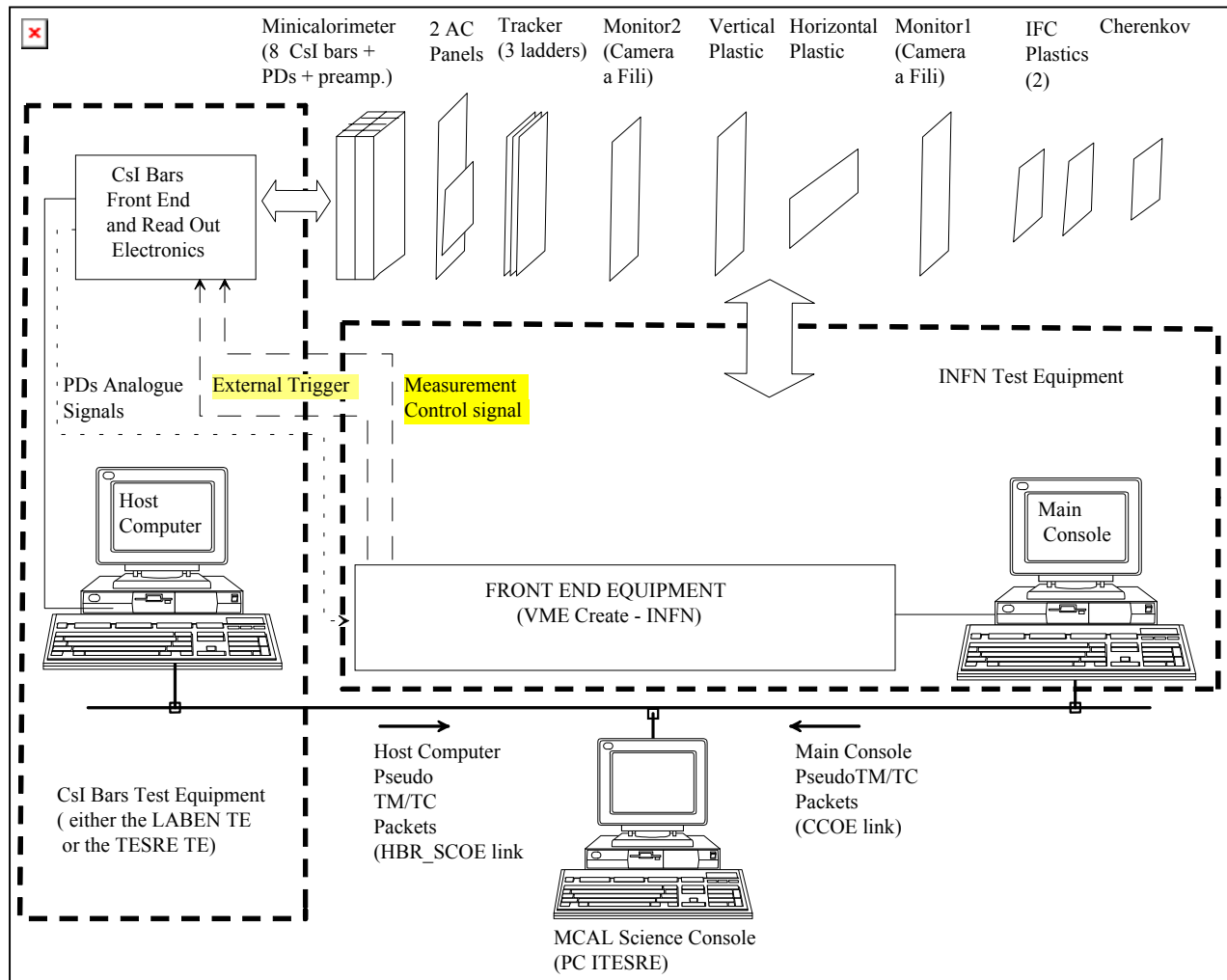


fig. 2.2-1 Test Equipment set up configuration during CERN Beam Tests

As in the case of the CsI Bars standalone tests described in the previous section, from the MCAL Science Console point of view two separate acquisition chains are identified:

- **on the left side:**
  - either the LABEN TE or the TESRE TE could be installed in order to acquire the CsI Bars data and to forward them to the MCAL Science Console, as in the standalone configuration;

to this purpose, the LABEN TE and TESRE TE provides the same external interfaces, i.e.:

- the same hardware interfacing with the INFN TE for the acquisition of the PDs analogue signals, i.e.:
  - the INFN TE shall generate an external trigger signal in order to command the peak and hold of the PDs signals;
  - for each 128 clocks received from the Main Console, the CsI TE multiplexer shall made available one signal; only the first 16 shall contain the PDs signal, and the remaining shall be meaningless.
- the same protocol and data structure interface with the MCAL Science Console for the acquisition of the TM packets.
- **on the right side**, the TESRE TE shall be replaced by the INFN TE, where:
  - the Main Console, through a dedicated I/F, is connected to the FEE which allows the PC to acquire the data produced by **all** the instruments (included the CsI Bars);
  - The Main Console sends to the MCAL Science Console a subset of the instrument data through a TCP/IP link established on the LAN.

The Host Computer, the Main Console and the MCAL Science Console are connected to a 100BaseT LAN through a Mini HUB having at least 4 100BaseT RJ ports.

### 2.2.2 Data acquisition and management

In order to be activated, each of the above acquisition chains requires that:

- the MCAL Science Console waits, as a Server, in order to establish the related TCP/IP connections, namely:
  - the HBR\_SCOE link with the either the LABEN TE or the TESRE TE (port 9003);
  - the CCOE link with the INFN TE (port 9002);

Each Data Link shall be kept along the whole test session.

In case the Data Link drops, automatically the Server restarts waiting for a new connection request from the Client.

During the Beam Tests, the CAL CsI TE electronics shall be operated only in **external** trigger mode which shall be provided by the INFN TE by using:

- either the output of the Monitor Detectors and the Horizontal and Vertical Plastics;

- or the FEF start signal generated by the Tracker for the GRID events;

As in the Standalone configuration, on the Test Equipment, the running of the acquisition task of the instrument data is commanded by the operator through the Host Computer console.

Also in this case, only the acquired data having the Measurement Control signal enabled are actually archived and processed by the TE. This mechanism shall allow the INFN TE to command the measurements in automatic mode.

Hence, during a test session, the synchronisation of the measurement periods in the two chains shall be under control of the INFN TE by means of the Measurement Control signal and the start/stop measurement TC packets, as already used in the standalone configuration.

Indeed, once the INFN Main Console operator will **start** a new measurement:

- **on the CCOE link**, the Main Console shall send to the Science Console a **start** measurement TC packet, consequently the Science Console:
  - shall close the *idle* files of the current RUN\_ID;
  - shall open the *measurement* files of the current RUN\_ID, where shall be archived the forthcoming INFN TE packets which shall be sent by the Main Console on the CCOE link;
- **on the CsI Bars TE side**, the Main Console shall notify that a new measurement is being started by using the Measurement Control signal, i.e.:
  - the INFN TE shall change from 0V to 5V the Measurement Control signal, consequently:
    - upon the **next** external trigger is generated, the CsI TE Host Computer shall detect that change and:
      - shall increment the RUN ID and open the measurement file using the file naming defined in [2] and in [3] for the LABEN TE case and the TESRE TE case, respectively;
      - shall generate and send on the HBR\_SCOE link with the Science Console a **start** measurement TC packet; consequently the Science Console:
        - shall close the *idle* files of the current RUN\_ID;
        - shall open the *measurement* files of the current RUN\_ID, where shall be archived the forthcoming CsI TE packets which shall be sent by the CsI TE Host Computer on the HBR\_SCOE link.

Once the Main Console operator will **stop** the current measurement:



- **on the CCOE link**, the Main Console shall send to the Science Console a stop measurement TC packet, consequently the Science Console:
  - shall close the *measurement* files of the current RUN\_ID;
  - shall increment the RUN\_ID for the next measurement;
  - shall open the *Idle* files of the next measurement.
  
- **on the CsI Bars TE side**, the Main Console shall notify that the current measurement is being closed by using the Measurement Control signal, i.e.:
  - the Measurement Control signal shall be reset 0V, consequently:
    - upon the next external trigger is generated, the CsI TE Host Computer shall detect the change and:
      - shall close the file of the current measurement;
      - shall generate and send a **stop** measurement TC packet on the HBR\_SCOE link with the Science Console; consequently the Science Console:
        - shall increment the RUN\_ID for the next measurement;
        - shall open the *idle* files of the next measurement.

It is noted that, in order to allow the Science Console to complete the filing operations, a new start measurement could non occur before 20 s since the last stop command.

In addition, it is noted that the Measurement Control flag is included in the event data. Hence, in any case some external trigger must be generated also during the Idle period (e.g. at least at the beginning of the Idle period) in order to allow the MCAL Science Console to acquire data having the Measurement Control signal set to off, and hence to identify that the current measurement is completed and a new set of TM archive files must be created.

For each detected event:

- the INFN TE shall acquire a set of data from all the instruments, namely:
  - Cherenkov: 1 bit flag;
  - Monitor 1: (X,Y);
  - Monitor 2: (X,Y);
  - CsI Bars PDs: 16 PHA;
  - Silicon Trackers data ( 3 Ladder with 384 PHA each)
  
- the CsI Bars TE shall acquire:

- CsI Bars PDs: 16 PHA;

From each of this data set:

- the Main Console of the INFN TE:
  - will extract from the digital data generated by the ADCs of the INFN TE, the data which are relevant to the Minicalorimeter data processing and creates an *INFN TE event data* format (as given in Annex A);
  - with the *INFN TE event data*, the Main Console will prepare a *INFN TE TM Packet* of fixed length having the structure complying with of the ESA Telemetry Standard [1]; each Data Packet will be capable of containing a fixed number of event data; further details on the format are given in section 3.
  - as soon as an Data Packet is filled with the event data or the *time-out* is expired since the last packet has been sent, the Main Console shall send the Data Packet (if not empty) to the MCAL Science Console.
- the CAL CsI Host Computer shall generate and send the TM packets to the MCAL Science Console with the same modalities and formats foreseen for the Standalone Tests.

The Science Console shall acquire and archive the Data Packets in one set of files for each measurement, by keeping separated the two packets streams.

The Quick Look software running on the Science Console, shall allow the user to select one of the two data stream (either INFN TE or the CsI Bars TE) on which to provide, in near real time a graphical representation on a percentage of the data.

### 3 INFN TE Packet Format

#### 3.1 Pseudo TM packet format

With the data acquired from the instrument chain, the Main Console generates packets having the same basic structure of the ESA Packet TM.

Each packet is of fixed length of bytes and consists of a Packet Header (6 bytes) followed by a Packet Data field (512 bytes).

Each packet is sent to the SC in one TCP/IP message containing an additional prefix of two bytes which contain the total number of bytes contained in the packet, specified in binary format (big-endian). I.e. 0x0206 (518 decimal).

Data words belonging to the same event cannot be split into two different packets.

The part of the Source Data Field which does not contain event data is filled with dummy bytes.

The value of the dummy byte is configurable; by default it will be set to 0xFF.

##### 3.1.1 Packet header

The Packet Header will have the following format:

###### Packet Header (3x16-bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB														LSB	
Version Number			Type	DHFH	APID										
SF		Source Sequence Counter													
Packet Length															

Where:

###### First word:

Version number: must be set to '100' binary

Type: must be set to '0' binary

DFHF: Data Field Header Flag, must be set to '1' binary

APID: Application Process Identifier, must be set to 0x505 (decimal 1285 corresponding to the Science Console shared memories channel 14)

**Hence: this word must be set to the fixed value: 0x8D05**

**Second word:**

SF: Sequence Flag, must be set to '11' bin  
Source Sequence Counter: counts the packets of the above APID

**Third word:**

Packet Length: [number of octets in Packet Data Field] – 1;  
**Hence: this word must be set to the fixed value: 0x01FD (511 decimal).**

**3.1.2 Packet Data Field**

The Packet Data Field will consists of the Data Field Header (6 byte) and the Source Data Field (506 byte).

The former will have the following format:

**Data Field Header (4 x 16-bit word)**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB								LSB							
Time Tag second (32 bits integer signed MSB)															
Time Tag second (32 bits integer signed LSB)															
Time Tag millisecond (16 bit unsigned)															
TT		Bsize -1						Nblocks -1							

Where:

Time Tag: CPU System Time expressed as second and millisecond since 00:00 UTC January 1,1970 (i.e. as given by the C routine *ftime*) corresponding to the creation of the packet.

The Source Data Field will be have the following format:

**Source Data Field (12 x 21 16-bit word record)**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB								LSB							
Source Data Field ( word 1)															
...															
Source Data Field (word 252)															

I.e. the Source Data Field will be capable of containing up to 12 MCAL event Data, each having the format detailed in Annex A.

### 3.2 Pseudo TC Packet Format

In order to notify to the Science Console when a new measurement is starting/stopping, the Host Computer shall generate and send on the Data link, packets having the same basic structure of the ESA Packet TC.

Each packet is of fixed length of 10 bytes and consists of a Packet Header (6 bytes) followed by a Packet Data field (4 bytes).

Through the same socket connection established for the TM Packets, each packet is sent on the Data Link in one TCP/IP message containing an additional prefix of two bytes which contain the total number of bytes contained in the packet, specified in binary format (big-endian). I.e. 0x000A (10 decimal).

Further details on the TC Packet format are given below.

#### 3.2.1 TC Packet header

The Packet Header will have the following format:

##### Packet Header (3x16-bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB														LSB	
Version Number				Type	DHFF	APID									
SF		Source Sequence Counter													
Packet Length															

Where:

##### First word:

Version number: must be set to '0001' binary

Type: must be set to '1' binary

DFHF: Data Field Header Flag, must be set to '1' binary

APID: Application Process Identifier, must be set to 0x101 (decimal 257)

**Hence: this word must be set to the fixed value: 0x1D01**

##### Second word:

SF: Sequence Flag, must be set to '11' bin

Source Sequence Counter: counts the packets of the above APID; to be reset at the begin of each Test Session.

##### Third word:

Packet Length: [number of octets in Packet Data Field] – 1;

**Hence: this word must be set to the fixed value: 0x0003 (3 decimal).**

### 3.2.2 TC Packet Data Field

The Packet Data Field will consists of the Data Field Header (2 bytes) and the Source Data Field (2 bytes).

The former will have the following format:

#### Data Field Header (1 x 16-bit word)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
Spare		Checksum type		ACK				Packet Type				Packet subtype			

Where:

Spare: must be set to '00' binary.

Checksum type: must be set to '00' binary

ACK: must be set to '00' binary.

Packet type: must be set to 0x5.

Packet subtype: must be set to 0x5.

**Hence: this word must be set to the fixed value: 0x0055**

The Source Data Field will be have the following format:

#### Source Data Field (1 x 16-bit word record)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSB															LSB
TC Application Data															

Where:

TC Application Data: **must be set to the fixed values:**

- **0x0200**, for the "Start measurement" command;
- **0x0000**, for the "Stop measurement" command.

## 4 Science Console Archiving

### 4.1 On-line archiving

All the acquired TM packets and the files generated on-line by the MCAL Science Console are automatically saved on the on-line Archive, which resides on the `/users/archive` directory tree located on a local magnetic disk.

As depicted below, this directory tree is structured in three main subdirectories (raw, erdf, , log) which are devoted to the raw data, the fits data, and the log data, respectively.

```
/users/archive
|-- erdf
|   |-- hk
|   |   |-- 0461
|   |   |-- science
|   |   |-- 0461
|-- log
`-- raw
    |-- hk
    |   |-- 0461
    |   |-- science
    |   |-- 0461
```

Fig. 4.1-1 The Science Console on-line Archive structure

This directory tree is unique for all the data acquired in parallel from the two TCP/IP links of the various test set up presented in section 2.

As detailed below, the identification of the link is provided by using different file name extensions.

#### 4.1.1 Raw archiving

The Science Console provides the archiving of all the raw TC and TM packets of each measurement, which is identified by a **run id** number generated automatically by the software at the start of a new observation. The raw files contain the packet byte stream sequence as received by the Science Console during each measurement.

After verification of the APID and the SSC, each acquired packet is written into two files, one containing all the TC/TM packets, and the other containing only the TC and the HK packets.

These files are archived in the *archive* directory tree under the *raw* subdirectory tree, where a subdirectory is devoted to each decade of run IDs. The files are univocally associated to a given measurement by including in their names the campaign ID **ccc**, the run ID **nnnnn** and the date of creation **yymmdd**, i.e:

```
ccnnnnn_yymmdd_.ext
ccnnnnn_yymmdd_.ext
ccnnnnn_yymmdd.ext
```

One underscore character appended to the name identifies the files related to the idle period between two consecutive measurements. Two of these characters are used for the first idle period in the test session.

In the **ext** suffix:

- the first character identifies the TCP/IP link from which the data are collected, namely:
  - **h** is for HBR\_SCOE link;
  - **c** is for CCOE link;
  - **e** is for ECOE link [not used here];
- the second and the third characters identify:
  - either the file containing all the TC/TM packets (**rt**);
  - or the file containing only the TC and the HK packets (**hk**);

The SC software limits automatically each measurement to a maximum of 400000 TM packets (corresponding to the maximum size of ~ 600 Mbytes available on a CD-ROM). Indeed once this limit is reached, the software closes all the files related to the current measurement and opens a new set of files, after having incremented the measurement RUN ID.

#### 4.1.2 Raw Data Processing

On each active TCP/IP link, as the packets are read from the **rt** file of the raw archive, they are sorted by APID and are made available to the related unpacking processor which operates on two different shared memories as follows:

- on the **SHM** shared memory, it receives the packets sorted by APID;
- on the **SHMF** shared memory, it writes the unpacked data to be analysed by the quick look program based on IDL;

Both shared memories are organised in *channels* which are assigned to the different APIDs.

Details on the raw data processing configuration are given in Annex B.

#### 4.1.3 FITS Archiving

In near real time, the unpacking processor devoted to a given APID archives the information extracted from the raw packets on files written in FITS format. These files are located under the **erdf** subdirectory tree in the **archive** directory, where a subdirectory is devoted to each decade of run IDs., e.g.

```
/users/archive/erdf
|-- hk
|  |-- 0461
|  |-- 0462
|  |-- 0463
|-- science
|  |-- 0461
|  |-- 0462
|  |-- 0463
```



Fig. 4.1-2 The Science Console FITS subdirectory tree structure

The **hk** subdirectory contains the hk files, while the **science** subdirectory contains the event lists.

The files in the erdf directory tree are univocally associated to a given measurement by a naming similar to the one used for the raw files, i.e.:

**ccc\_nnnnn\_yymmdd\_hhmmss\_\_ext**  
**ccc\_nnnnn\_yymmdd\_hhmmss\_\_ext**  
**ccc\_nnnnn\_yymmdd\_hhmmss.ext**

In this case, the name includes also the time of creation of the file, in the format:

**hhmmss**    hours minutes seconds (2+2+2)

The campaign ID **ccc**=cer.

In the **ext** suffix:

- the first character identifies the TCP/IP link from which the data are collected, namely:
  - **h** is for HBR\_SCOE link;
  - **c** is for CCOE link;
  - **e** is for ECOE link [not used here];
  - **g** is for the files reproduced in playback mode;
- the second and the third characters identifies the fits file (**ft**).

The Science erdf files are tabular files with one row per event. Details on the actual implementation of these files is given in Annex C.

## 4.2 The permanent archive on CD-ROM

Purpose of this archive is to save on CD-ROM all the relevant data archived on the on-line Archive by the Science Console during a given test campaign.

This is accomplished in different steps:

### 1. initialisation:

- the current on-line Archive directory tree (i.e.: /users/archive) is renamed to the *setpro* subdirectory of the **off-line Archive** directory tree */users/archive\_offline*;
- in the *Report* directory, an auxiliary directory, a new *setpro* subdirectory is initialised in order to host the summary information which shall be produced during the archiving of the off-line Archive files;
- in the *CD Area*, a temporary disc area, a new *setpro* subdirectory is initialised in order to host the off-line Archive files organised in a directory tree suitable for the creation of the CD-ROMs;

- a new on-line Archive directory tree is initialised for the forthcoming measurements;
2. cleaning and copy the off-line Archive *setpro* subdirectory tree on a temporary area (the CD Area):
    - meaningless measurement files are removed;
    - the measurement files are copied on the CD Area, which consists of a directory tree capable of containing the whole off-line Archive and where the measurements file are saved into subdirectories having a maximum size of about 600MB;
    - the copy operations are verified;
  3. close the off-line Archive:
    - the *setpro* subdirectory of the off-line Archive directory is renamed to *setxxx*;
    - the *setpro* subdirectory of the Report directory is renamed to *setxxx*;
    - the *setpro* subdirectory of the CD Area is renamed to *setxxx*;
  4. burning the CD-ROMs:
    - for each subdirectory contained in the *setxxx* subdirectory of the CD Area:
      - the CD image is created;
      - the master copy of the CD is burned;
      - the master copy is verified against the CD image;
      - a second copy of the CD is burned;
      - the CD image is removed.

The above steps shall be iterated until all the data produced during the Test Campaign are saved on the CD-ROMs.

As shown in fig. 4.2-1 below, the Report directory tree shall contain a subdirectory tree for each Test Campaign, plus a *conf* subdirectory devoted to some configuration information.

For each Test Campaign, the Report directory tree shall contain:

- the *doc* subdirectory, where to save the Test Campaign log file;
- the *setpro* directory related to the current off-line Archive,
- the *setxxx* directories, created by renaming the *setpro* subdirectory of the previous off-line Archives, which have been already closed by the operator.

```
/users/agile_report
|-- cernnov2000
|  |-- doc
|  |-- set000
|  `-- setpro
`-- conf
```

Fig. 4.2-1 The Report directory tree structure

As shown in fig. 4.2-2 below, the CD Area directory tree shall consist of:

- the *setpro* subdirectory, where the current off-line Archive files shall be copied by splitting the *archive* tree in one subdirectory *cdyyyy*, for each CD-ROM to be created;
- the *setxxx* subdirectories, created by renaming the *setpro* subdirectory of the previous off-line Archives, which have been already closed by the operator, but have not been burned yet on CD ROM.

The CD Area must be located on a computer equipped with a CD writer, and it must be capable of containing the files of the whole off-line Archive.

This area could either be hosted on a local Science Console disk (e.g. */data1/cd\_area/setpro*), or located on a remote computer and mounted on the Science Console through NFS (e.g.: */data1/remote\_cdarchive/setpro*).

```
/data1/remote_cdarchive/  
|-- set000  
    |-- cd0007  
        |-- archive  
    |-- cd0008  
        |-- archive  
|-- setpro
```

Fig. 4.2-2 The CD-Area directory tree structure

The off-line Archive directory tree shall consist of:

- the *setpro* subdirectory, where are located all the files of the off-line Archive which is being processed;
- the *setxxx* subdirectories, created by renaming the *setpro* subdirectory of the previous off-line Archives, which have been already closed by the operator, but have not been backed up on DAT yet.

As detailed in the following sections, most of the above operations are automated by means of shell scripts which are executed by the operator on the Science Console through the Archive Control widget shown below, which is obtained by typing *archive* to run Idl, and then typing *arc* to run the required Idl procedure.

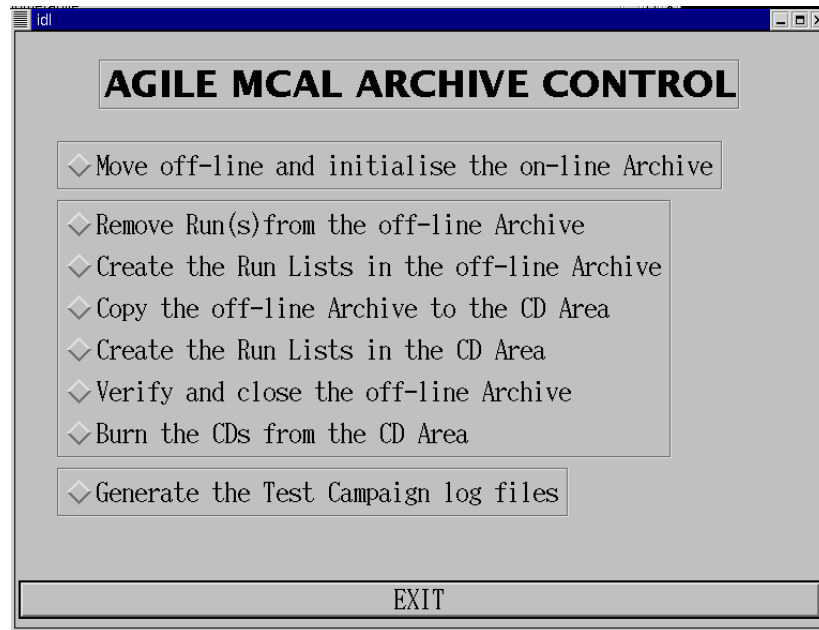


Fig. 4.2-3 AGILE Archive Control widget on the Science Console.

#### 4.2.1 Move off-line and initialise the on-line Archive

This is the first step to be performed in order to start the archiving operations of the measurement already performed without interfering with the data acquisition of the forthcoming measurements.

To this purpose, the operator selects the “*Move off-line and initialise the on-line Archive*” widget list item (see fig 4.2-3 above), which:

1. checks whether the archiving operations on the current off-line Archive has been completed and the off-line Archive has been closed (see below), and in the negative case terminates;
2. renames the current directory tree of the on-line Archive, which becomes the off-line Archive tree to be saved on CD ROM;
3. in the Report directory, creates a new *setpro* subdirectory (e.g.: */users/agile\_report/cernnov2000/setpro*), where to save the summary information of the off-line Archive, which shall be produced in the next steps;
4. in the CD Area, creates a new *setpro* subdirectory, where to copy the off-line Archive files organised in a directory tree which is suitable for the creation of the CD ROM;
5. creates and initialise a new directory tree for the on-line Archive.

E.g.:

The */data1/archive* subdir will be renamed to: */data1/archive\_offline/setpro*  
and a new *setpro* subdirectory shall be created in both  
the Report directory and in the CD Area

Would you like to continue ? (y/n) [n] y

... renaming the directory which was linked to /users/archive

/data1/archive moved to /data1/archive\_offline/setpro

... creating the Report setpro subdirectory  
/users/agile\_report/cernnov2000/setpro created

... creating the CD Area setpro subdirectory  
/data1/remote\_cdarchive/setpro created

... re-creating an empty on-line Archive directory /data1/archive for the link /users/archive

/data1/archive created

... initialising the on-line Archive directory /data1/archive

... /data1/archive/raw created  
... /data1/archive/erdf created  
... /data1/archive/log created

The required operation has been successfully completed

As shown above, a default value is proposed by the script for each parameter. Some of these defaults can be changed by the user, while others require the modification of the script, in the default setting section listed below.

```
# Main
#
#
#--- site default setting (start)
#
filesys_link=/users/archive
topdir=/data1
subdir=archive
topsubdir_offline=archive_offline
subdir_new_name=${topsubdir_offline}/setpro
reportdir=/users/agile_report/cernnov2000/setpro
destfilesys=/data1
destsubdir=remote_cdarchive/setpro

#
#--- site default setting (end)
#
```

As shown above, a default value is proposed by the script for each parameter. Some of these defaults can be changed by the user, while others require the modification of the script, in the default setting section listed below.

#### 4.2.2 Remove Runs from the off-line Archive

This step is performed in order to remove from the off-line Archive any file related to measurements which are meaningless;

To this purpose, the operator selects the “*Remove Run(s) from the off-line Archive*” widget list item (see fig 4.2-3 above) and provides the minimum and maximum Run ID of the set of measurements to be removed, e.g.:

```
Campaign id ? [cer]:  
the Campaign id is set to: cer  
Would you like to continue ? (y/n) [n]: y  
  
Data archive top directory ? [/data1/archive_offline/setpro]:  
the Data archive top directory is set to: /data1/archive_offline/setpro  
Would you like to continue ? (y/n) [n]: y  
  
the scoeids are set to HBRS, OLD  
Would you like to continue ? (y/n) [n]: y  
  
Prompt the user before removing each file (y/n) ? [y]: y  
the user will be prompt before removing each file  
Would you like to continue ? (y/n) [n]: y  
first runid to be removed ? 1  
last runid to be removed ? 10  
  
All files found in /data1/archive_saved for runids from 1 to 10 will be removed  
Would you like to continue ? (y/n) [n]:
```

It is noted that the *Data archive top directory* must be set to the directory path of the off-line Archive directory being saved on CD ROM.

As shown above, a default value is proposed by the script for each parameter. Some of these defaults can be changed by the user, while others require the modification of the script, in the default setting section listed below.

```
# Main  
#  
#  
#--- site default setting (start)  
#  
camp=cer  
source_top_dir=/data1/archive_offline/setpro  
#  
#####  
# Playback ECOE CCOE HBR_SCOE OLD #  
# ----- #  
# scoeid [not used] ECOE CCOE HBRS OLD #  
#####  
#  
scoeid1=HBRS  
scoeid2=OLD  
#  
#--- site default setting (end)
```

### 4.2.3 Create the Run List in the off-line Archive

With this step the whole off-line Archive is processed in order:

1. to identify the run which total size does not fit into a CDROM (600 Mbytes); in this case the

script:

- creates a file consisting of the copy of the header of the FITS elf file containing the data;
- compresses the elf file, and if necessary, compresses the raw file.

2. to create the *Run List* file containing a list of summary information related to all the run contained in the archive.

To this purpose, the operator selects the “*Create Run List in the off-line Archive*” widget list item (see fig 4-1 above), and answers to the script prompt, e.g.:

```
Archive Type ? (mo|cd|hd) [hd]:  
The Archive Type is set to: hd  
Would you like to continue ? (y/n) [n]: y
```

```
Source File System for HD? [/data1]:  
Archive ID is fixed to: 0000  
The Source File System Subdir is set to: archive_off-line/setpro  
Would you like to continue ? (y/n) [n]: y
```

```
- The list of all runs contained in /data1/archive_offline/setpro  
will be written on file /data1/archive_saved/summary/hd0000list.create  
Would you like to continue ? (y/n) [n]: y
```

As shown above, a default value is proposed by the script for each parameter. Some of these defaults can be changed by the user, while others require the modification of the script, in the default setting section listed below.

```
# Main  
#  
#  
#--- site default setting (start)  
#  
tmpdir="users/agile/tmp"  
arctype=hd  
hdsourcesys=/data1  
hdsourcesubdir=archive_offline/setpro  
cd_dir=/data1/remote_cdarchive/setpro  
reportdir=/users/agile_report/cernnov2000/setpro  
compressdir=/data1/gzdir  
minimum=20  
arcid=0  
cdmaxsize_mb=600  
empty_name="empty"  
#  
#--- site default setting (end)
```

Once the parameters setting is completed, the script analyses the whole off-line Archive and generates the ASCII files containing the related summary information:

- under the *summary* subdirectory of the off-line Archive, the script creates the *Run List* file (e.g. /data1/archive\_offline/setpro/summary/hd00001list.create)

- the script copies the *Run List* file into the Report *setpro* subdirectory.

e.g:

```
Verifying /data1 status ...  
... /data1 is mounted  
... /data1/archive_offline/setpro/summary exists  
  
creating run list file /data1/archive_offline/setpro/summary/hd0000list.create  
04289 (95040 bytes)  
elf file name= ./erdf/science/0428/ite04289_001204_143128__.hft  
keyword value is LABEN T.E.  
keyword value is 4289  
keyword value is 2000-12-04  
keyword value is 14:31:30  
keyword value is 14:34:25  
keyword value is 1560  
  
.....  
... /users/agile_report/cernnov2000/setpro exists
```

The *Run List* file is an ASCII file containing one line for each measurement. Each line consists of the following fields (separated by “;”):

- Archive ID (always set to 0 for the HD tytpe)
- Host Computer ID
- Run ID;
- Measurement Date;
- Measurement start time;
- Measurement end time;
- Measurement total number of events;
- the total size in bytes of the files (Log, Raw, Erdf) related to the measurement

Host Computer, Run ID, Date, times, total number of events and packet type are copied from the header of the FITS file and are set to “unknown” in case this file is missing.

A zero value for the “Measurement total number of events” means that the measurement has been terminated abnormally and the NAXIS2 header keyword has not been updated to the correct value. In this case, if necessary, the FITS files can be reproduced in playback from the raw files.

#### 4.2.4 Copy the off-line Archive to the CD Area

This step is performed in order to copy all the files contained in the off-line Archive *setpro* subdirectory on the *setpro* subdirectory of the CD Area, organised as a directory tree having subdirectories smaller than the capacity of one CDROM (600MB), namely:

- the subdirectories will be named *cdxxxx*, where *xxxx* is an incremental number which will identify the CDs. This numbering has been started from 1 with the May 2000 CERN campaign and will be incremented for all the remaining Agile campaigns;



- in each subdirectory a maximum of 600 MB of measurement files will be copied having care of keeping on the same subdirectory all the files related to a given measurement (i.e. the files related to one measurement cannot be split among CDs).

By selecting the “*Copy the off-line Archive to the CD Area*” list item (see fig. 4.2-3 above), on the Science Console the operator runs the shell script which performs the above operations, i.e.:

- determines the ID of the next CD subdirectory to be created, i.e.:
  - gets the ID found in the *lastcdid\_file* configuration file and checks whether the *setpro* subdirectory of the CD Area contains any subdirectory;
    - if not, it assumes that the CD corresponding to that ID has already been burned, and continues with the next step;
    - if yes, it checks whether the highest subdirectory found in the CD Area corresponds to that ID;
      - if yes, continues with the next step;
      - if not, terminates with an error message;
- checks whether the file system hosting the CD Area does contain the *setpro* subdirectory, and is capable of containing the whole off-line Archive:
  - if yes, continues with the next step;
  - if not, terminates with an error message;
- by iterating the following steps, the script copies the whole *setpro* subdirectory of the off-line Archive to the *setpro* subdirectory of the CD Area:
  - based on the ID number *xxxx* determined in the previous steps, the script evaluates whether in the subdirectory *cdxxxx* there is enough space to host at least the files related to the first run to be copied. If there is not space left, or the subdirectory *cdxxxx* does not exist at all, the script increments the ID number *xxxx* and creates a new subdirectory *cdxxxx*.
  - the script copies the off-line Archive data to the *cdxxxx* subdirectory up to the maximum of 600 MB, having care of keeping together all the files related to a given measurement.
  - the script writes on the *summary* subdirectory of the *sourcedir* directory, the file *cdxxxxlist.copy*, containing the lists of the runs copied on the *cdxxxx* directory.
- increments the CD ID until the whole on-line Archive has been copied.

e.g.:

The CD ID to be assigned to the data to be copied is: 8  
where 550 MB are available

```
Would you like to continue ? (y/n) [n]: y
The Source File System Subdir is set to: archive_offline/setpro
Would you like to continue ? (y/n) [n]: y
Verifying /data1 status ...
... /data1 exists
... /data1/archive_offline/setpro exists
Verifying /data1 status ...
... /data1 exists
Verifying /data1/remote_cdarchive/setpro status ...
... /data1/remote_cdarchive/setpro exists
... /data1/remote_cdarchive/setpro/cd0008 created
... /data1/remote_cdarchive/setpro/cd0008/archive created
... /data1/remote_cdarchive/setpro/cd0008/archive/raw created
... /data1/remote_cdarchive/setpro/cd0008/archive/erdf created
... /data1/remote_cdarchive/setpro/cd0008/archive/log created
... /data1/remote_cdarchive/setpro/cd0008/archive/summary created
```

The list of all runs which will be copied from /data1/archive\_offline/setpro to /data1/remote\_cdarchive/setpro/cd0008/archive will be written on /data1/archive\_14\_feb\_01/summary  
Would you like to continue ? (y/n) [n]:

As they fit in the required space, all the files for run 04383 will be copied ....

copying file labccoe04383.log to /data1/remote\_cdarchive/setpro/cd0008/archive/log ..... copied  
sleeping 2 seconds before continuing ...

```
... /data1/remote_cdarchive/setpro/cd0008/archive/raw/science/0438 created
... /data1/remote_cdarchive/setpro/cd0008/archive/raw/hk/0438 created
... /data1/remote_cdarchive/setpro/cd0008/archive/erdf/science/0438 created
... /data1/remote_cdarchive/setpro/cd0008/archive/erdf/hk/0438 created
```

The above steps lead to generate in the CD Area a number of subdirectories where the whole *setpro* subdirectory of the off-line Archive is copied, e.g.:

```
$ ls /data1/remote_cdarchive/setpro
cd0007 cd0009 cd0011 cd0013 cd0015 cd0017 cd0019 cd0021
cd0008 cd0010 cd0012 cd0014 cd0016 cd0018 cd0020
```

The related Run\_list files are added to the same summary directory containing the Run List file created at the previous step for the whole *setpro* subdirectory of the off-line Archive, e.g.:

```
$ ls /data1/archive_offline/setpro/summary
cd0007list.copy cd0011list.copy cd0015list.copy cd0019list.copy
cd0008list.copy cd0012list.copy cd0016list.copy cd0020list.copy
cd0009list.copy cd0013list.copy cd0017list.copy cd0021list.copy
cd0010list.copy cd0014list.copy cd0018list.copy hd00000list.create
```

The default setting section of the script is listed below.

```
#####
#
# Main
```

```
#  
#  
#--- site default setting (start)  
#  
..tmpdir="/users/agile/tmp"  
sourcefilesys=/data1  
..sourcesubdir=archive_offline/setpro  
sourcearctype=data1  
destfilesys=/data1  
destsubdir=remote_cdarchive/setpro  
destarctype=cd  
cdmaxsize_mb=550  
minimum_mb=20  
empty_name="empty"  
confdir=/users/agile_report/conf  
# file containing the ID of the last CD, to be used in case no cdxxxx dir has been left  
# on destsubdir  
lastcdid_file=${confdir}/lastcdid  
#  
#--- site default setting (end)
```

#### 4.2.5 Create the Run Lists in the CD Area

Once the previous step has been completed, the list of the Runs contained in the CD Area could be obtained by running, on all the *cdxxxx* subdirectories of the CD Area, the same scripts which has been run on the off-line Archive.

To this purpose, on the Science Console the operator selects the “*Create all the Run Lists in the CD Area*” list item (see fig. 4.2-3 above) in order to run the shell script which performs the required operations, i.e.:

- runs the script *create\_all\_cds\_run\_lists*, which in turn runs the *Create Run List* in order to perform on each *cdxxxx* subdirectory found in the CD Area the same processing of the script which created the *hdnnnnlist.create* file in the off-line Archive.

This time, after having removed any previous version of *Run List* files, the script produces in each *cdxxxx/summary* subdirectory the file *cdnnnnlist.create* containing the same kind of information produced for the off-line Archive case.

The *cdnnnnlist.create* files are also copied to the Report *setpro* subdirectory.

E.g.:

```
The CD Archive directory is set to: /data1/remote_cdarchive/setpro  
Would you like to continue ? (y/n) [n]:y  
The Archive Type is set to: cd  
CD ID is set to: 0007  
The Source File System Subdir is set to: cd0007/archive  
The list of all runs contained in /data1/remote_cdarchive/setpro/cd0007/archive  
will be written on file /data1/remote_cdarchive/setpro/cd0007/archive/summary/cd0007list.create  
Verifying /data1/remote_cdarchive status ...  
... /data1/remote_cdarchive is mounted
```

```
.....  
... /users/agile_report/cernnov2000/setpro exists
```

Type the Return key to return to the main window

The default setting section of the script is listed below.

```
#  
# Main  
#  
# for each CD directory in cddir  
#  
#--- site default setting (start)  
#  
cd_dir=/data1/remote_cdarchive/setpro  
script_dir=/users/agile/Archive  
report_dir=/users/agile_report/cernnov2000/setpro  
#  
#--- site default setting (end)
```

#### 4.2.6 Verify and close the off-line Archive

This is the last step to be performed before starting with the burning of the CD ROM. This step:

1. verifies that the whole off-line Archive has been copied successfully to the CD Area; this is accomplished by comparing the Run Lists created with the above steps.
2. closes the current off-line Archive.

To this purpose, on the Science Console the operator selects the “*Verify and close the off-line Archive*” list item (see fig. 4.2-3 above) in order to run the shell script which performs the required operations, i.e.:

- runs the script *verify\_run\_lists\_and\_close*, which:
  - compares the Run Lists (on-line Archive vs CD Area), i.e.:
    - copies in the file *hd.db* the *hd0000list.create* file;
    - creates the file *hd.merge* by copying the file *hd.db* without the first column;
    - merges in the file *cd.db* all the *cdnnnnlist.create* files;
    - creates the file *cd.merge* by copying the file *cd.db* without the first column;
    - presents to the user any difference found by comparing byte to byte the content of the files *mo.merge* and *cd.merge*;

as already detailed in the previous section, one of the fields in the run lists gives the total size in bytes of the files related to each measurement; hence in case the *.merge* files are

identical, the copy was successful;

all the above ASCII files are copied to the Report *setpro* subdirectory.

- asks the operator whether the off-line Archive can be closed by:
  - renaming the *setpro* subdirectory of off-line Archive directory tree
  - renaming the *setpro* subdirectory of Report directory tree;
  - renaming the *setpro* subdirectory of CD Area directory tree.

It is noted that until the off-line Archive has not been closed, the operator shall **not** be able to start the archiving operations on the current On-line Archive.

E.g.:

```
The Report Dir is set to: /users/agile_report/cernnov2000/setpro
Would you like to continue ? (y/n) [n]: y
```

```
The hd and the cd merged files are identical
```

```
Close the off-line Archive ? (y/n): y
```

```
The /data1/remote_cdarchive/setpro dir shall be renamed to: /data1/remote_cdarchive/set000
```

```
Would you like to continue ? (y/n) [n] y
/data1/remote_cdarchive/setpro moved to /data1/remote_cdarchive/set000
```

```
The /users/agile_report/cernnov2000/setpro dir shall be renamed to: /users/agile_report/cernnov2000/set000
```

```
Would you like to continue ? (y/n) [n] y
/users/agile_report/cernnov2000/setpro moved to /users/agile_report/cernnov2000/set000
```

```
Type the Return key to return to the main window
```

The default setting section of the script is listed below.

```
#
# Main
#
#
#--- site default setting (start)
#
tmpdir="/users/agile/tmp"
topdir=/data1
topsubdir_offline=archive_offline
offline_dir=${topdir}/${topsubdir_offline}
cd_dir=/data1/remote_cdarchive/setpro
reportdir=/users/agile_report/cernnov2000/setpro
hd_merged_file=${reportdir}/hd.merged
hd_db_file=${reportdir}/hd.db
cd_merged_file=${reportdir}/cd.merged
```

```
cd_db_file=${reportdir}/cd.db
diff_file=${reportdir}/hd_cd.diff
#
#--- site default setting (end)
#
```

#### 4.2.7 Burn the CDs from the CD Area

Once the off-line Archive has been verified and closed as detailed in the previous sections, the corresponding CD Area subdirectory tree (e.g. */data1/remote\_cdarchive/set000*) contains a subdirectory *cdxxxx* for each of the CDROM which is ready to be burned.

The instructions to be followed in order to burn the CDROM are obtained by selecting the “*Burn the CDs from the CD Area*” list item (see fig. 4.2-3 above), i.e.:

On the PC equipped with the CD writer, login as root, and perform the following operations for each *cdxxxx* subdirectory:

- 1) open a first terminal and execute the commands:
  - cd /data1/cd\_area
  - ./create\_agile\_cd <xxxx> where xxxx identifies the CD to be burned  
this runs the scripts which:
    - ask you to confirm/change set setxxx CD Area subdirectory
    - creates the raw image /cd\_area/picsitcdxxxx.raw corresponding to the subdirectory *cdxxxx*;
    - burns the image on the CD; assigning the volume name *agilexxxx*;
- 2) open a second terminal from which verify the burned CD against the image, by using the *xcdroast* package; this CD will be kept as master copy.
- 3) on the first terminal, run again the above script:
  - ./create\_agile\_cd <xxxx>  
this time the scripts finds the raw image and only burns a second CD to be kept as working copy.
  - remove the raw image;
  - remove the subdirectory *cdxxxx*.

Type the Return key to return to the main window

The *create\_agile\_cd* script creates the raw image by using the command */usr/bin/mkisofs* (Version 1.12b5, 17 Feb 1998), and burns the image on the CD by using the command */usr/bin/cdrecord* (AUTHOR: Joerg Schilling, Seestr. 110 D-13353 Berlin Germany).

The default setting section of the *create\_agile\_cd* script is listed below.

```
#
#####
# Main
#
#
#--- site default setting (start)
#
cd_area=/data1/cd_area
agile_topdir=${cd_area}/agile
agile_subdir=set000
raw_pref="agilecd"
raw_ext="raw"
cd_pref="cd"
```

```
vol_pref="agile"  
#  
#--- site default setting (end)
```

The labels for the CDs are produced using the Fujifilm CD-Labeling System by Labelle™.

The *setxxx* subdirectory of the CD Area shall be removed by the operator once all the data contained in the subdirectory have been burned on CD-ROMs.

### 5.2.5 Backing up the off-line Archive

Before removal, the *setxxx* subdirectories of the off-line Archive directory tree must be backed up on DAT tape by using tar.

The following command are required (logged in as root):

- *cd* <off-line Archive top directory>                    e.g. *cd /data1/archived\_saved*
- *tar cvf /dev/rmt/0mn ./*

It is noted that the tape is not rewound by the tar command.

### 5.2.6 Generate the Test Campaign log files

As already mentioned, the Report *doc* subdirectory is used to collect the ancillary files related to the whole campaign.

Once the last set of CDROM of campaign has been burned, the following steps are required in order to create in the *doc* subdirectory the ASCII files containing the complete directory of the CDROMs to be included in the Test Report document.

To this purpose, on the Science Console the operator selects the “*Generate the Test Campaign log*” list item (see fig. 4.2-3 above) in order to run the shell script which performs the required operations, i.e.:

- runs the *create\_docs* script which in the above *doc* directory:
  - merges in the file *all\_hd.db* all the *hd.db* files found under all the *setnnn* directories;
  - merges in the file *all\_cd.db* all the *cd.db* files found under all the *setnnn* directories;

E.g.:

```
The Report Dir top is set to: /users/agile_report/cernnov2000  
Would you like to continue ? (y/n) [n]: y  
The file /users/agile_report/cernnov2000/doc/all_hd.db has been created  
The file /users/agile_report/cernnov2000/doc/all_cd.db has been created
```

Type the Return key to return to the main window

The default setting section of the script is listed below.

```
#  
# Main  
#  
#  
#--- site default setting (start)  
#  
tmpdir="/users/agile/tmp"  
reportdirtop="/users/agile_report/cernnov2000"  
setid="set"  
hd_db_file=hd.db  
cd_db_file=cd.db  
docdir=${reportdirtop}/doc  
all_hd_db_file=${docdir}/all_hd.db  
all_cd_db_file=${docdir}/all_cd.db  
#  
#--- site default setting (end)  
#
```

Eventually, in the Report *doc* subdirectory, the operator creates by hand the ASCII file *doc/all\_hd\_tar.db* which contains, for each Off-line Archive backed up as given in the previous section; the Off-line Archive ID, the DAT #, the File # within the DAT (separated by semicolon).



**Annex A INFN TE Event Data format**

For each detected pulse event, the Main Console adds 42 bytes in the Packet Data Field. The format of each event data set is shown below.

**Event Data Format (21x16 bit word)**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
MSB															LSB	
spare			MCAL - PD0 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD1 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD2 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD3 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD4 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD5 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD6 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD7 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD8 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD9 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD10 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD11 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD12 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD13 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD14 Pulse Height Amplitude (12 bits)													
spare			MCAL - PD15 Pulse Height Amplitude (12 bits)													
Monitor 1 - X coordinate (16 bits unsigned) [horizontal]																
Monitor 1 - Y coordinate (16 bits unsigned) [vertical]																
Monitor 2 - X coordinate (16 bits unsigned) [horizontal]																
Monitor 2 - Y coordinate (16 bits unsigned) [vertical]																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Flag

Where:

MCAL PD: contain the Pulse Height Amplitude as generated by the 10 bits ADCs of the INFN TE;

Monitors X,Y X,Y range is 0-20000 in 10 μm- units (i.e. 0-20 cm);

Flag Cherenkov flag, which gives the event type: 0=electron, 1=muone;

Note: The INFN TE 10 bits ADC values are fit into 12 bits.

The remaining are self-explaining.

## Annex B Science Console Raw data processing configuration

### B.1 Processor and shared memory channel assignment

In the AGILE Test Beam Campaign (November 2000) at CERN, the processor *proc\_2\_lnx* shall be able to process on the Science Console the TM packet data coming from either the INFN-TE, or the TESRE TE, or the LABEN TE.

One copy of the program shall be run on each of the two possible active TCP/IP link (HBR\_SCOE link, CCOE link).

On each active link, the Science Console shall receive the packet stream having the APID assigned to the TE connected on that link, and on the basis of the APID, shall perform the different processing required for that TE.

As shown in the table below, the various APIDs shall be assigned to different shared memory channels.

APID Nominal/ Redundant	Packet type/ subtype	Function name	SHM channel	Unpacking processor on the Science Console	SHMF Channel
1348 , 1476			0		0
1352 , 1480			1		1
1356 , 1484			2		2
1357 , 1485			3		3
1360 , 1488			4		4
1361 , 1489			5		5
1364 , 1492			6		6
1368 , 1496			7		7
1372 , 1500			8		8
1373 , 1501			9		9
1376 , 1504			10		10
1293 , 1421			11		11
1280 , 1408			12		12
1281 , 1409			13		13
1285 , 1413		INFN TE	14	<i>proc_2_lnx</i>	14
1286 , 1414			15		15
1397 , 1525			16		16
1398 , 1526		TESRE TE	17	<i>proc_2_lnx</i>	17
1399 , 1527			18		18
1293		LABEN TE	18	<i>proc_2_lnx</i>	18

Table B-1 APID and shared memories channels mapping

### B.2 Quick Look shared memory data format

The processor shall write the unpacked data on the Shared Memory SHMF using the **unique** format (i.e. APID independent) detailed hereafter.

The parameters defining the dimensions of the Quick Look shared memory (SHMF) and the buffer BUFFER to be written on it, shall be set as follows:

```
C
C from parameters.inc
C
C     SIZE_FRAME was set to 4224 to be multiply by:
C         100 ( buffers per SHMF channel)
C         21 ( total number of SHMF channels)
C         ---> total of 8.870.400 byte of RAM
C     N.B.: on 25/01/2000 changed from 100 a 300
C
C BLHEADER is the header length in bytes
C
C BLEVENT is the event length in bytes
C for the time being, for each event it is reserved the maximum
C space as required by the Compton events.
C
C MAXEV is the max number of events per frame which can be
C written in each SHMF buffer; this is also the number of events
C which are buffered in the FITS arrays.
C Note: in the SHMF buffer are written: 1 header+1frame+1header
C for Nov 2000, changed BLEVENT from 40 to 50
C     INTEGER*4 BLEVENT
C     PARAMETER (BLEVENT=50)
C
C     INTEGER*4 BLBUFFER, BLHEADER, MAXEV, MAXCELL
C     PARAMETER (BLBUFFER=4224, BLHEADER=40,
+     MAXEV=( (BLBUFFER-2*BLHEADER)/(BLEVENT) ) /2 ,
+     MAXCELL=848)
C
```

The following format is applied to the structure which contains the header and the events. Depending on the number of events, one or more of this structure can fit into the buffer BUFFER located in the SHMF shared memory.

```
C from PUTFRAME
C
C     INTEGER*2 HEADER(BLHEADER/2), BUFFER(BLBUFFER/2)
C
C     HEADER is the array where the header info are prepared
C
C     word 1: BLHEADER, header length in bytes
C     word 2: PKT_APID, APID
C     word 3: SUBTYPE
C     word 4: SHM channel
C     word 5: NEVENT, number of events (i.e. of Trigger Data Blocks) contained in the frame
C     word 6: BLEVENT, event length (i.e. length of Trigger Data Blocks) in bytes
C     word 7: SPARE
C     word 8-9: NFRAME, frame #
C     word 10: BLSIZE, number of PDs actually running (dei 16 previsti in ogni event format)
C     word 11: Trigger Type flag, 0 = internal trigger, 1 = external trigger
C     word 12: Data Source, 0 = CAL Csi LABEN TE, 1= Csi TESRE TE, 2=CERN INFN TE
C     word 13-16: spare
C     word 17-18: RUNID
C     word 19-20: spare
C
```

C Each event consists of the following info:

C

C word 1-4: TIME (8 bytes)

C word 5: MCAL PD0 Energy (2bytes)

C word 6: MCAL PD1 Energy (2bytes)

C word 7: MCAL PD2 Energy (2bytes)

C word 8: MCAL PD3 Energy (2bytes)

C word 9: MCAL PD4 Energy (2bytes)

C word 10 MCAL PD5 Energy (2bytes)

C word 11 MCAL PD6 Energy (2bytes)

C word 12 MCAL PD7 Energy (2bytes)

C word 13 MCAL PD8 Energy (2bytes)

C word 14 MCAL PD9 Energy (2bytes)

C word 15 MCAL PD10 Energy (2bytes)

C word 16 MCAL PD11 Energy (2bytes)

C word 17 MCAL PD12 Energy (2bytes)

C word 18 MCAL PD13 Energy (2bytes)

C word 19 MCAL PD14 Energy (2bytes)

C word 20 MCAL PD15 Energy (2bytes)

C word 21: Monitor 1 - X (2bytes)

C word 22: Monitor 1 - Y (2 bytes)

C word 23 Monitor 2 - X (2bytes)

C word 24 Monitor 2 - Y (2bytes)

C word 25 Cherenkov Flag (2 bytes)

C

C

C BUFFER is the buffer where HEADER + events are

C saved to the Shared Memory SHMF when the current

C frame does not fit into it

C

**It is noted that only a subset of the event data will be processed by the Quick Look.**

## Annex C Science Console FITS file formats

### C.1 LABEN TE and TESRE TE

In both cases, each TM packet shall contain in its Data Field the number of trigger data blocks specified by the field *Nblocks* contained in its Data Field header. Each trigger data block shall contain the number of PD data specified by the field *Bsize*.

For each forthcoming TM packet, the Science Console shall extract each trigger data block in order to write them on the FITS binary table; using the format detailed in the following subheadings.

Each trigger data block shall be saved on one row of the binary table, namely:

- one column for the time given in the Header of the Data Field containing the block;
- one column for each of the PD PHA;

#### C.1.1 FITS format for the LABEN TE Data

##### C.1.1.1 Primary Header

```
SIMPLE =          T / file does conform to FITS standard
BITPIX =          16 / number of bits per data pixel
NAXIS =           0 / number of data axes
EXTEND =          T / FITS dataset may contain extensions
COMMENT FITS (Flexible Image Transport System) format defined in Astronomy and
COMMENT Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.
COMMENT Contact the NASA Science Office of Standards and Technology for the
COMMENT FITS Definition document #100 and other FITS information.
END
```

##### C.1.1.2 Extension Header

```
XTENSION='BINTABLE' / binary table extension
BITPIX =          8 / 8-bit bytes
NAXIS =           2 / 2-dimensional binary table
NAXIS1 =          40 / width of table in bytes
NAXIS2 =          15968 / number of rows in table
PCOUNT =          0 / size of special data area
GCOUNT =          1 / one data group (required keyword)
TFIELDS =         17 / number of fields in each row
TTYPE1 = 'TIME ' / label for field 1
TFORM1 = '1D ' / data format of field: 8-byte DOUBLE
TUNIT1 = 's ' / physical unit of field
TTYPE2 = 'MC_SIGNAL0' / label for field 2
TFORM2 = '1I ' / data format of field: 2-byte INTEGER
TZERO2 =          32768 / offset for unsigned integers
TSCAL2 =          1 / data are not scaled
TUNIT2 = 'PHA ' / physical unit of field
TTYPE3 = 'MC_SIGNAL1' / label for field 3
```

TFORM3 = 'II' / data format of field: 2-byte INTEGER  
TZERO3 = 32768 / offset for unsigned integers  
TSCAL3 = 1 / data are not scaled  
TUNIT3 = 'PHA' / physical unit of field  
TTYPER4 = 'MC\_SIGNAL2' / label for field 4  
TFORM4 = 'II' / data format of field: 2-byte INTEGER  
TZERO4 = 32768 / offset for unsigned integers  
TSCAL4 = 1 / data are not scaled  
TUNIT4 = 'PHA' / physical unit of field  
TTYPER5 = 'MC\_SIGNAL3' / label for field 5  
TFORM5 = 'II' / data format of field: 2-byte INTEGER  
TZERO5 = 32768 / offset for unsigned integers  
TSCAL5 = 1 / data are not scaled  
TUNIT5 = 'PHA' / physical unit of field  
TTYPER6 = 'MC\_SIGNAL4' / label for field 6  
TFORM6 = 'II' / data format of field: 2-byte INTEGER  
TZERO6 = 32768 / offset for unsigned integers  
TSCAL6 = 1 / data are not scaled  
TUNIT6 = 'PHA' / physical unit of field  
TTYPER7 = 'MC\_SIGNAL5' / label for field 7  
TFORM7 = 'II' / data format of field: 2-byte INTEGER  
TZERO7 = 32768 / offset for unsigned integers  
TSCAL7 = 1 / data are not scaled  
TUNIT7 = 'PHA' / physical unit of field  
TTYPER8 = 'MC\_SIGNAL6' / label for field 8  
TFORM8 = 'II' / data format of field: 2-byte INTEGER  
TZERO8 = 32768 / offset for unsigned integers  
TSCAL8 = 1 / data are not scaled  
TUNIT8 = 'PHA' / physical unit of field  
TTYPER9 = 'MC\_SIGNAL7' / label for field 9  
TFORM9 = 'II' / data format of field: 2-byte INTEGER  
TZERO9 = 32768 / offset for unsigned integers  
TSCAL9 = 1 / data are not scaled  
TUNIT9 = 'PHA' / physical unit of field  
TTYPER10 = 'MC\_SIGNAL8' / label for field 10  
TFORM10 = 'II' / data format of field: 2-byte INTEGER  
TZERO10 = 32768 / offset for unsigned integers  
TSCAL10 = 1 / data are not scaled  
TUNIT10 = 'PHA' / physical unit of field  
TTYPER11 = 'MC\_SIGNAL9' / label for field 11  
TFORM11 = 'II' / data format of field: 2-byte INTEGER  
TZERO11 = 32768 / offset for unsigned integers  
TSCAL11 = 1 / data are not scaled  
TUNIT11 = 'PHA' / physical unit of field  
TTYPER12 = 'MC\_SIGNAL10' / label for field 12  
TFORM12 = 'II' / data format of field: 2-byte INTEGER  
TZERO12 = 32768 / offset for unsigned integers  
TSCAL12 = 1 / data are not scaled  
TUNIT12 = 'PHA' / physical unit of field  
TTYPER13 = 'MC\_SIGNAL11' / label for field 13  
TFORM13 = 'II' / data format of field: 2-byte INTEGER  
TZERO13 = 32768 / offset for unsigned integers  
TSCAL13 = 1 / data are not scaled  
TUNIT13 = 'PHA' / physical unit of field  
TTYPER14 = 'MC\_SIGNAL12' / label for field 14  
TFORM14 = 'II' / data format of field: 2-byte INTEGER

TZERO14 = 32768 / offset for unsigned integers  
TSCAL14 = 1 / data are not scaled  
TUNIT14 = 'PHA ' / physical unit of field  
TTYPER15 = 'MC\_SIGNAL13' / label for field 15  
TFORM15 = 'II ' / data format of field: 2-byte INTEGER  
TZERO15 = 32768 / offset for unsigned integers  
TSCAL15 = 1 / data are not scaled  
TUNIT15 = 'PHA ' / physical unit of field  
TTYPER16 = 'MC\_SIGNAL14' / label for field 16  
TFORM16 = 'II ' / data format of field: 2-byte INTEGER  
TZERO16 = 32768 / offset for unsigned integers  
TSCAL16 = 1 / data are not scaled  
TUNIT16 = 'PHA ' / physical unit of field  
TTYPER17 = 'MC\_SIGNAL15' / label for field 17  
TFORM17 = 'II ' / data format of field: 2-byte INTEGER  
TZERO17 = 32768 / offset for unsigned integers  
TSCAL17 = 1 / data are not scaled  
TUNIT17 = 'PHA ' / physical unit of field  
EXTNAME = 'AGILE\_Binary' / name of this binary table extension  
ORIGIN = 'TESRE Science Console V.2'  
TELESCOP = 'Agile '  
INSTRUME = 'Grid '  
MODEL = 'Nov2000 '  
DETNAME = 'MCAL '  
NBARS = 8 / Number of CsI Bars under test  
HOSTCOMP = 'LABEN T.E.'  
APID = 1293  
TESTLEVE = 'FEE '  
DATATYPE = 'Diagnostic'  
RUNID = 4268  
CAMPAIGN = 'cer '  
FILENAME = '/users/archive/erdf/science/0426/ser04268\_001128\_183329.hft'  
DISCARD = 0 / Total number of discarded events  
DATE-OBS = '2000-11-28'  
TIME-OBS = '18:33:24'  
DATE-END = '2000-11-28'  
TIME-END = '18:59:39'  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
COMMENT  
END

## C.1.2 FITS format for the TESRE TE Data

### C.1.1.1 Primary Header

SIMPLE = T / file does conform to FITS standard  
BITPIX = 16 / number of bits per data pixel  
NAXIS = 0 / number of data axes

EXTEND = T / FITS dataset may contain extensions  
COMMENT FITS (Flexible Image Transport System) format defined in Astronomy and  
COMMENT Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.  
COMMENT Contact the NASA Science Office of Standards and Technology for the  
COMMENT FITS Definition document #100 and other FITS information.  
END

### C.1.1.2 Extension Header

XTENSION= 'BINTABLE' / binary table extension  
BITPIX = 8 / 8-bit bytes  
NAXIS = 2 / 2-dimensional binary table  
NAXIS1 = 40 / width of table in bytes  
NAXIS2 = 48576 / number of rows in table  
PCOUNT = 0 / size of special data area  
GCOUNT = 1 / one data group (required keyword)  
TFIELDS = 17 / number of fields in each row  
TTYPER1 = 'TIME' / label for field 1  
TFORM1 = 'ID' / data format of field: 8-byte DOUBLE  
TUNIT1 = 's' / physical unit of field  
TTYPER2 = 'MC\_SIGNAL0' / label for field 2  
TFORM2 = 'I' / data format of field: 2-byte INTEGER  
TZERO2 = 32768 / offset for unsigned integers  
TSCAL2 = 1 / data are not scaled  
TUNIT2 = 'PHA' / physical unit of field  
TTYPER3 = 'MC\_SIGNAL1' / label for field 3  
TFORM3 = 'I' / data format of field: 2-byte INTEGER  
TZERO3 = 32768 / offset for unsigned integers  
TSCAL3 = 1 / data are not scaled  
TUNIT3 = 'PHA' / physical unit of field  
TTYPER4 = 'MC\_SIGNAL2' / label for field 4  
TFORM4 = 'I' / data format of field: 2-byte INTEGER  
TZERO4 = 32768 / offset for unsigned integers  
TSCAL4 = 1 / data are not scaled  
TUNIT4 = 'PHA' / physical unit of field  
TTYPER5 = 'MC\_SIGNAL3' / label for field 5  
TFORM5 = 'I' / data format of field: 2-byte INTEGER  
TZERO5 = 32768 / offset for unsigned integers  
TSCAL5 = 1 / data are not scaled  
TUNIT5 = 'PHA' / physical unit of field  
TTYPER6 = 'MC\_SIGNAL4' / label for field 6  
TFORM6 = 'I' / data format of field: 2-byte INTEGER  
TZERO6 = 32768 / offset for unsigned integers  
TSCAL6 = 1 / data are not scaled  
TUNIT6 = 'PHA' / physical unit of field  
TTYPER7 = 'MC\_SIGNAL5' / label for field 7  
TFORM7 = 'I' / data format of field: 2-byte INTEGER  
TZERO7 = 32768 / offset for unsigned integers  
TSCAL7 = 1 / data are not scaled  
TUNIT7 = 'PHA' / physical unit of field  
TTYPER8 = 'MC\_SIGNAL6' / label for field 8  
TFORM8 = 'I' / data format of field: 2-byte INTEGER  
TZERO8 = 32768 / offset for unsigned integers  
TSCAL8 = 1 / data are not scaled  
TUNIT8 = 'PHA' / physical unit of field  
TTYPER9 = 'MC\_SIGNAL7' / label for field 9



TFORM9 = 'II ' / data format of field: 2-byte INTEGER  
TZERO9 = 32768 / offset for unsigned integers  
TSCAL9 = 1 / data are not scaled  
TUNIT9 = 'PHA ' / physical unit of field  
TTYPER10 = 'MC\_SIGNAL8' / label for field 10  
TFORM10 = 'II ' / data format of field: 2-byte INTEGER  
TZERO10 = 32768 / offset for unsigned integers  
TSCAL10 = 1 / data are not scaled  
TUNIT10 = 'PHA ' / physical unit of field  
TTYPER11 = 'MC\_SIGNAL9' / label for field 11  
TFORM11 = 'II ' / data format of field: 2-byte INTEGER  
TZERO11 = 32768 / offset for unsigned integers  
TSCAL11 = 1 / data are not scaled  
TUNIT11 = 'PHA ' / physical unit of field  
TTYPER12 = 'MC\_SIGNAL10' / label for field 12  
TFORM12 = 'II ' / data format of field: 2-byte INTEGER  
TZERO12 = 32768 / offset for unsigned integers  
TSCAL12 = 1 / data are not scaled  
TUNIT12 = 'PHA ' / physical unit of field  
TTYPER13 = 'MC\_SIGNAL11' / label for field 13  
TFORM13 = 'II ' / data format of field: 2-byte INTEGER  
TZERO13 = 32768 / offset for unsigned integers  
TSCAL13 = 1 / data are not scaled  
TUNIT13 = 'PHA ' / physical unit of field  
TTYPER14 = 'MC\_SIGNAL12' / label for field 14  
TFORM14 = 'II ' / data format of field: 2-byte INTEGER  
TZERO14 = 32768 / offset for unsigned integers  
TSCAL14 = 1 / data are not scaled  
TUNIT14 = 'PHA ' / physical unit of field  
TTYPER15 = 'MC\_SIGNAL13' / label for field 15  
TFORM15 = 'II ' / data format of field: 2-byte INTEGER  
TZERO15 = 32768 / offset for unsigned integers  
TSCAL15 = 1 / data are not scaled  
TUNIT15 = 'PHA ' / physical unit of field  
TTYPER16 = 'MC\_SIGNAL14' / label for field 16  
TFORM16 = 'II ' / data format of field: 2-byte INTEGER  
TZERO16 = 32768 / offset for unsigned integers  
TSCAL16 = 1 / data are not scaled  
TUNIT16 = 'PHA ' / physical unit of field  
TTYPER17 = 'MC\_SIGNAL15' / label for field 17  
TFORM17 = 'II ' / data format of field: 2-byte INTEGER  
TZERO17 = 32768 / offset for unsigned integers  
TSCAL17 = 1 / data are not scaled  
TUNIT17 = 'PHA ' / physical unit of field  
EXTNAME = 'AGILE\_Binary' / name of this binary table extension  
ORIGIN = 'TESRE Science Console V.2'  
TELESCOP = 'Agile '  
INSTRUME = 'Grid '  
MODEL = 'Nov2000 '  
DETNAM = 'MCAL '  
NBARS = 8 / Number of CsI Bars under test  
HOSTCOMP = 'TESRE T.E.'  
APID = 1398  
TESTLEVE = 'FEE '  
DATATYPE = 'Diagnostic'  
RUNID = 4268

```
CAMPAIGN= 'cer  '
FILENAME= '/users/archive/erdf/science/0426/er04268_001128_233529.cft'
DISCARD =      0 / Total number of discarded events
DATE-OBS= '2000-11-28'
TIME-OBS= '23:35:31'
DATE-END= '2000-11-29'
TIME-END= '00:21:37'
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
COMMENT
END
```

## C.2 INFN TE

Also in this case, each TM packet shall contain in its Data Field the number of trigger data blocks specified by the field *Nblocks* contained in its Data Field header. Each trigger data block shall contain the number of data specified by the field *Bsize*.

It is noted in this case, in addition to the 16 PDs, each trigger data block shall contain the two Monitors XY and the Cherenkov flag.

For each forthcoming TM packet, the Science Console shall extract each trigger data block in order to write them on the FITS binary table; using the format detailed in the following subheadings.

Each trigger data block shall be saved on one row of the binary table, namely:

- one column for the time given in the Header of the Data Field containing the block;
- one column for each of the PD PHA;
- one column for each of the Monitor X and Monitor Y;
- one column for the Cherenkov flag.

### C.2.1 FITS format for the INFN TE Data

#### C.2.1.1 Primary Header

```
SIMPLE =      T / file does conform to FITS standard
BITPIX =     16 / number of bits per data pixel
NAXIS  =      0 / number of data axes
EXTEND  =      T / FITS dataset may contain extensions
```

COMMENT FITS (Flexible Image Transport System) format defined in Astronomy and  
COMMENT Astrophysics Supplement Series v44/p363, v44/p371, v73/p359, v73/p365.  
COMMENT Contact the NASA Science Office of Standards and Technology for the  
COMMENT FITS Definition document #100 and other FITS information.  
END

### C.1.1.2 Extension Header

XTENSION= 'BINTABLE' / binary table extension  
BITPIX = 8 / 8-bit bytes  
NAXIS = 2 / 2-dimensional binary table  
NAXIS1 = 50 / width of table in bytes  
NAXIS2 = 9960 / number of rows in table  
PCOUNT = 0 / size of special data area  
GCOUNT = 1 / one data group (required keyword)  
TFIELDS = 22 / number of fields in each row  
TTYPER1 = 'TIME ' / label for field 1  
TFORM1 = '1D ' / data format of field: 8-byte DOUBLE  
TUNIT1 = 's ' / physical unit of field  
TTYPER2 = 'MC\_SIGNAL0' / label for field 2  
TFORM2 = '1I ' / data format of field: 2-byte INTEGER  
TZERO2 = 32768 / offset for unsigned integers  
TSCAL2 = 1 / data are not scaled  
TUNIT2 = 'PHA ' / physical unit of field  
TTYPER3 = 'MC\_SIGNAL1' / label for field 3  
TFORM3 = '1I ' / data format of field: 2-byte INTEGER  
TZERO3 = 32768 / offset for unsigned integers  
TSCAL3 = 1 / data are not scaled  
TUNIT3 = 'PHA ' / physical unit of field  
TTYPER4 = 'MC\_SIGNAL2' / label for field 4  
TFORM4 = '1I ' / data format of field: 2-byte INTEGER  
TZERO4 = 32768 / offset for unsigned integers  
TSCAL4 = 1 / data are not scaled  
TUNIT4 = 'PHA ' / physical unit of field  
TTYPER5 = 'MC\_SIGNAL3' / label for field 5  
TFORM5 = '1I ' / data format of field: 2-byte INTEGER  
TZERO5 = 32768 / offset for unsigned integers  
TSCAL5 = 1 / data are not scaled  
TUNIT5 = 'PHA ' / physical unit of field  
TTYPER6 = 'MC\_SIGNAL4' / label for field 6  
TFORM6 = '1I ' / data format of field: 2-byte INTEGER  
TZERO6 = 32768 / offset for unsigned integers  
TSCAL6 = 1 / data are not scaled  
TUNIT6 = 'PHA ' / physical unit of field  
TTYPER7 = 'MC\_SIGNAL5' / label for field 7  
TFORM7 = '1I ' / data format of field: 2-byte INTEGER  
TZERO7 = 32768 / offset for unsigned integers  
TSCAL7 = 1 / data are not scaled  
TUNIT7 = 'PHA ' / physical unit of field  
TTYPER8 = 'MC\_SIGNAL6' / label for field 8  
TFORM8 = '1I ' / data format of field: 2-byte INTEGER  
TZERO8 = 32768 / offset for unsigned integers  
TSCAL8 = 1 / data are not scaled  
TUNIT8 = 'PHA ' / physical unit of field  
TTYPER9 = 'MC\_SIGNAL7' / label for field 9

TFORM9 = 'II' / data format of field: 2-byte INTEGER  
TZERO9 = 32768 / offset for unsigned integers  
TSCAL9 = 1 / data are not scaled  
TUNIT9 = 'PHA' / physical unit of field  
TTYPER9 = 'MC\_SIGNAL8' / label for field 10  
TFORM10 = 'II' / data format of field: 2-byte INTEGER  
TZERO10 = 32768 / offset for unsigned integers  
TSCAL10 = 1 / data are not scaled  
TUNIT10 = 'PHA' / physical unit of field  
TTYPER10 = 'MC\_SIGNAL9' / label for field 11  
TFORM11 = 'II' / data format of field: 2-byte INTEGER  
TZERO11 = 32768 / offset for unsigned integers  
TSCAL11 = 1 / data are not scaled  
TUNIT11 = 'PHA' / physical unit of field  
TTYPER11 = 'MC\_SIGNAL10' / label for field 12  
TFORM12 = 'II' / data format of field: 2-byte INTEGER  
TZERO12 = 32768 / offset for unsigned integers  
TSCAL12 = 1 / data are not scaled  
TUNIT12 = 'PHA' / physical unit of field  
TTYPER12 = 'MC\_SIGNAL11' / label for field 13  
TFORM13 = 'II' / data format of field: 2-byte INTEGER  
TZERO13 = 32768 / offset for unsigned integers  
TSCAL13 = 1 / data are not scaled  
TUNIT13 = 'PHA' / physical unit of field  
TTYPER13 = 'MC\_SIGNAL12' / label for field 14  
TFORM14 = 'II' / data format of field: 2-byte INTEGER  
TZERO14 = 32768 / offset for unsigned integers  
TSCAL14 = 1 / data are not scaled  
TUNIT14 = 'PHA' / physical unit of field  
TTYPER14 = 'MC\_SIGNAL13' / label for field 15  
TFORM15 = 'II' / data format of field: 2-byte INTEGER  
TZERO15 = 32768 / offset for unsigned integers  
TSCAL15 = 1 / data are not scaled  
TUNIT15 = 'PHA' / physical unit of field  
TTYPER15 = 'MC\_SIGNAL14' / label for field 16  
TFORM16 = 'II' / data format of field: 2-byte INTEGER  
TZERO16 = 32768 / offset for unsigned integers  
TSCAL16 = 1 / data are not scaled  
TUNIT16 = 'PHA' / physical unit of field  
TTYPER16 = 'MC\_SIGNAL15' / label for field 17  
TFORM17 = 'II' / data format of field: 2-byte INTEGER  
TZERO17 = 32768 / offset for unsigned integers  
TSCAL17 = 1 / data are not scaled  
TUNIT17 = 'PHA' / physical unit of field  
TTYPER17 = 'MON1\_X' / label for field 18  
TFORM18 = 'II' / data format of field: 2-byte INTEGER  
TZERO18 = 32768 / offset for unsigned integers  
TSCAL18 = 1 / data are not scaled  
TUNIT18 = 'Micron\*10' / physical unit of field  
TTYPER18 = 'MON1\_Y' / label for field 19  
TFORM19 = 'II' / data format of field: 2-byte INTEGER  
TZERO19 = 32768 / offset for unsigned integers  
TSCAL19 = 1 / data are not scaled  
TUNIT19 = 'Micron\*10' / physical unit of field  
TTYPER19 = 'MON2\_X' / label for field 20  
TFORM20 = 'II' / data format of field: 2-byte INTEGER

