EUROGAM PROJECT

Event-by-Event Data Formats

Edition 2.0

Date: May 1990

S. Kossionides

Institute of Nuclear Physics
N.C.S.R. Demokritos
153 10 Aghia Paraskevi, Greece

Data Formats for EUROGAM

S. Kossionides

1 Introduction

The first version of this document was transmitted by BITNET to a large group of people for comments. Only the Daresbury/Liverpool group responded and I thank them for it. Some of their comments have been worked into the specification. Some others are included in *italics* to be decided on during the meeting.

Since I will not be able to attend the May Meeting here are some general remarks:

(a) The IN2P3 standard does indeed have some redundancy, but it is the only standard covering ALL types of data that may be recorded from the EUROGAM system.

(b) It is true that the important information in the ‘ FILEH ’ record of IN2P3 is duplicated by the HDR1, HDR2 records of ANSI. The information not provided by ANSI is anyway redundant for EUROGAM. It could then be argued that the ANSI Standard should be adopted and IN2P3 dropped. I have left this decision to the Meeting, since I do not have information on the availability of Full ANSI standards for EXABYTE on systems to be likely used for off-line analysis.

(c) The two-word header of each event was accepted in the February Meeting as a safety for proper event identification. If the ANSI D Standard is accepted, each event could be made a logical record. In this case, the Standard requires a byte count as the first ‘word’ of the logical record. Then, the word count proposed here is redundant and it can be deleted.

(d) There is a need to write a ‘data terminator’ as the last logical record of a physical block. I know at least one system which, when writing ANSI Standard records that do not fill exactly a physical block, leaves garbage after the last logical record, creating problems to the play-back routines.

2 Design Criteria

In a distributed data collection and processing system the data blocks which belong to one data set (e.g. experiment, run) must have a fixed format. Some flexibility between data sets must be available to accomodate varying experimental needs.

The most probable error during the run is transmission error. Normaly this is corrected by a request to retransmit. With the rates anticipated for EUROGAM the loss of one data block may be less significant than the timing and sequencing problems connected with retransmission. A sequence number attached to each data block will be needed to control this error.

A policy of 'no-loss-acceptable' is suggested by the British team, but I still hold to my remark on the cost of retransmission.

Another error which may occur is accidental break of the data taking sequence (e.g. failure of part or all of the system) resulting in an open file on permanent storage (tape). A data set identifier attached to the block will allow off line detection of the end of valid data.

It can also be used to properly close the file with an off-line program in such ways as to allow writting further files to the tape if needed.

Off line data reduction will be performed on different computer systems. Data tapes must be protected from inadvertent overwrite and the user from the analysis of improper tapes. A standard form of tape labeling, recognized by all computer systems, will prevent these errors.

The off line analysis will be facilitated if parameters valid during data taking are written on the event tape. Integer parameters (e.g. trigger conditions) can be written in binary. Real parameters must be passed as character variables to accomodate the differing binary representations within each computer system.

3 Terminology and Conventions

It seems necessary to introduce some terms describing the data structure. The term CHANNEL is allready in use to describe the electronics associated with one Detector. Each Channel has several digitized outputs which we label as ITEMS. Each Item is encoded in 32 bits with the two high order bits as VALIDATION, the next 14 bits as IDENTIFICATION and the following 16 bits as DATA VALUE. We use the shorthand V, ID and DATA for the three parts of the ITEM. So far the following Channels and Items have been specified (the Item numbering is arbitrary):

CHANNEL	ITEM	Contents
GE	1	TAC
	2	Energy Converted in 20 MeV Range
	3	Energy Converted in 4 MeV Range
	4	Energy Converted in 800 keV Range (optional)
BGO	1	TAC
	2	Energy

Another term which will be needed is the GROUP. The Group contains associated Items from Various Channels. An obvious Group is one Ge-Detector and its associated BGO shield which may contain shared BGO-Detectors.

Data is transmitted in BLOCKS of length appropriate for the transmission medium. The Blocks cary a WORD COUNT which allways includes itself. It can thus be used as a relative pointer to the start of the next Block. The Following terms are introduced to better describe the structure of data in EUROGAM:

SUB-BLOCK	The Block of data transmitted by the ROC to the Event Builder.
EVENT BLOCK	The Block of data specifying one event.
DATA BLOCK	The Block of Data transmitted to systems downstream of the EB.

In describing the Event Format we will use the following notation:

- yy : Group number
- xx : Item number
- zzzz: 16-bit Data value
- wwww: 16-bit WC (counting 32-bit words)

4 Data Identification

There are 14 bits available for coding the Item identification. If the view to EUROBALL must remain open then 9 bits will be needed for the coding of the Ge-ID. With 10 BGO crystals around each Ge-detector it is clear that the remaining 5 bits are just enough to code the 24 Item-ID's. We propose the following coding scheme (bit 15 highest, bit 0 lowest order bit).

BITS	Contents
15,14	VALIDATION
13	Reserved for expansion
12...8	ITEM number
7...0	GROUP number

We propose to use the Ge-Detector number as Group number. It can then be quickly extracted as the least significant byte and used as a pointer into energy calibration tables, associated BGO tables etc. The BGO Channels are considered as Items of the Ge-Detector which they surround. Additional detectors (e.g. The RMS) obtain their own group numbers.

The ID will be created at the card level. The solution accepted is to include for each ADC a 14-bit writable register which can be preset during Experiment Setup.

PROGRAMMING NOTE: It is necessary to introduce intelligence into the Event Builder. The program should recognize conditions for rejecting events. This implies recognition of ID's. It is suggested that the relevant programs be written in the same language as the Sorting Programs. Therefore, commands to recognize and unpack ID's should be included in the Sorting Language.

This idea has been considered as to difficult to implement in EUROGAM. It is anyhow a point to be decided later.

4.1 Reserved Group Numbers

The general format of 32 bit data words, consisting of V//ID//DATA, should be kept as much as possible. Therefore, system information transmitted with the data should also have the same format. To this end the Group Number FF is reserved for system information. If the 63 Items thus made available prove insufficient, a second group number could be assigned. A word of all zeros should be avoided, therefore the Group number '00' is considered illegal.

5 ROC OUTPUT FORMAT

Each ROC, when requested, should send the data in a Sub-block. The Sub-block starts with a two word header consisting of:

	ID	Data
WORD 1	3FFF	Crate Number
WORD 2	2AFF	Word Count

The V-bits should be set to reflect a 'Valid Data' condition. The use of a two-word header in connection with fixed ID's and possibly a ROC Readout Sequence known to the Event Builder will allow detection and possibly correction of readout errors.

The question is raised, whether the ROC 'knows' the word count when the transmission starts. The current hardware specification is that all ROC's will start reading their respective crate upon the receiving of the Validation signal from the Main Trigger unit. It is also evident that a local buffer will be needed to hold the sub-block until the ROC is allowed to transmit to the EB. At this stage the word count can be generated.

A second question refers to the necessity of word counts. It was decided in the February Meeting that the redundancy of a Header+WC pair is necessary to allow for testing the data integrity.

The usefulness of the Crate Number as a Sub-Block Header has also been questioned. It is true that it provides no usefull information. It is also true that it is the easiest thing to implement in hardware, which will distinguish sub-blocks and add to the data integrity.

The need for the identification of the event to which the data Sub-block belongs has been asserted. The identification is important especially for pipelined operation. The solutions discussed so far are that:

a) The Trigger unit broadcasts an event number to all ROC's or

b) Each ROC keeps and updates an event counter.

In both cases the number is then attached to the Sub-block. An alternative solution is that each ROC, detecting an 'empty crate' condition when requested to read-out, should create an 'empty' Sub-block (with WC=1). This is a flexible solution allowing for pipelining and local ROC buffers holding more than one event. If this solution is adopted, there is no need for local event counting in the ROC and the Event Number can be kept only at the Trigger Unit.

With all schemes, event identification relies on the recognition by the ROC of the Validation signal. If the ROC misses this signal misidentification will occur. This problem must be considered by the Hardware Group.

5.1 Trigger Sub-block

The Trigger Unit is a 'special crate' with a Crate Number = FFFF. It transmitts the first Sub-block to the Event Builder. Thus the first word of an Event Block is V//3FFF//FFFF and this is kept through to the output tape. It is allready agreed that the Trigger Unit will transmitt to the Event Builder a 32-bit Event Number and a 64-bit Time Stamp. It is also agreed that information on trigger type and/or trigger configuration should be transmitted. The format of the latter is not yet fixed. If the general format of V//ID//DATA should be maintened it would require a large number of ID's to be reserved, it would double the amount of data transmitted and it would require reconstruction of the information at the Event Builder.

It is proposed that this information is transmitted in 32-bit words without ID in a fixed order which also fixes the meaning. The WC, transmitted as the second word fixes the number of transmitted Items.

The following format is thus proposed:

WORD	Contents	Meaning
1	3FFFFFFF	Trigger Crate
2	2AFFwwww	Word Count
3	32-bit Integer	Event Number
4,5	64-bit Integer	Time Stamp
6,..	–	To be defined

A minimum WC of 4 must be found indicating that the Event Number and Time Stamp have been included. Filtering or rejection or reformating of this information will be left to the Event Builder.

5.2 STOP/PAUSE Control

In a distributed system STOP and PAUSE commands must 'ripple' through the subsystems to allow for orderly emptying of buffers and pointer reset, tape closures etc. It is proposed that the STOP/PAUSE command is directed to the Trigger Unit only. This unit will stop generating Trigger and Validation Pulses and may also broadcast to the crates an inhibit signal. It should also create special trigger Sub-blocks which, as they are detected by the subsystems downstream of the Trigger Unit, will cause their orderly shut-down. We propose the following assignement of Trigger Sub-blocks:

PAUSE Command	3FFFFFFF
	3CFF0000
STOP Command	3FFFFFFF
	3CFFAAAA

The STOP Command should cause a general shutdown of data taking, which includes Tape File Closure, Reseting of Counters, Releasing of Buffers etc.

The PAUSE Command should allow to change On-line processing parameters which do not affect tape output but may cause unpredictable results if executed in midstream during processing. Therefore, it is necessary to push all pending buffers through the processing and output tasks before affecting the changes.

6 Event Builder Formats

In early proposals a Concetrator was placed between the ROC's and the Event Builder input buffers consisting of HSM's. The current design for the NSF System, proposes a 'serial' readout of crates over a FERA bus into HSM units. These will be controlled by a dedicated CPU, acting as concentrator, and they will be switched between ROC read-in and event building. In both cases several events will be stored in each HSM before switching. The need arises, therefore, for a marker of the end of valid events. It is proposed that an 'empty trigger' Sub-block:

3FFFFFFF
2AFF0001

is placed at the end of valid events by the controlling CPU.

The Event Builder output consists of Data Blocks containing several events each. The block size should be a parameter which will be fixed for optimum speed of the transmitting medium. The Event Builder should receive the event Sub-blocks and join them into one Data Block. It should also contain intelligence to allow filtering of events and grouping of Items.

6.1 Group Format

As elaborated by John Cresswell, considerable rate reduction could be obtained by grouping, i.e. by transmitting only the Data parts of correlated Items in one group labeled by group ID. Roughly, grouping becomes more economic than the standard V//ID//DATA Format if the number of Items, $N > 2$. If the 32-bit word is maintened then padding with a 16-bit Zero will be necessary for odd N, making the limit $N > 3$. This condition will not arise for the Ge-group, which is the only one currently definable. In this case four Items (TAC, Energy1, Energy2, Energy3) are stored for each detector and they can be encoded in the two halves of a 32-bit word. We propose the convention that Item Number 0 (Zero) indicates a group. Thus, depending on the ID, the interpretation of the 16-bit Data portion of a 32-bit word will be:

ID	Data
xxyy	Item Data
00yy	Group WC

This convention implies a fixed order of Item data, packed into 16-bit halves of the 32-bit data words. It allows to extend the grouping to other kinds of correlated data structures especially, when the position of the data source, which is specified by the ID, is not important for the data evaluation.

A question is whether we will allow mixing of groupped and standard Data in one Event. In principle this will not cause any problems and we propose that it will be allowed.

6.2 System Information Group

With the introduction of grouping and the permission to mix Standard and Grouped Format data in one event, it becomes possible to transmitt out of the Event Builder 32-bit data like the Event Number. We propose to pack such data into a System Group, with ID = 00FF, which can be recognized by subsequent systems as containing 32-bit data.

6.3 Event Format

The Event will start with the Event Header of the form:

ID	Data	Interpretation
3FFF	FFFF	Start of Event
2AFF	wwww	Event WC

Event data will follow either in the Standard Form:

xxyy	zzzz	

or in the Group Form:

00yy	wwww	Group WC
zzzz	zzzz	
..	..	
zzzz	zzzz	(or 0000 if odd number of Items)

Both forms may be mixed within one Event. We propose for additional safety that if, the System Group (with ID=00FF) is transmitted, should be placed immediately after the Event Header.

6.4 Data Block Format

Several events are packed into one Data Block. The Data Block starts with a 32-bit Block Number followed by a 32-bit WC. The Block number is generated and incremented by the Event Builder. It was proposed that the initial Block Number is generated as a random number to provide an additional form of Data set recognition. This procedure, however, may generate a Block Number of Zero somewhere in the sequence and should be avoided. We propose therefore, that the initial Block Number be allways 1 (ONE). We propose that the end of valid data is signaled by an 'empty trigger' of

3FFFFFFF
2AFF0001

for additional safety of the data.

The Data Block will be bracketed with Transmission protocol information and sent to the Sorting Engine.

The introduction of the 'empty trigger' may not be necessary if it is decided to transmitt each complete event as one logical record. Considering the relatively small size of the Event Block, it will still be necessary to pack several events into one block to reduce the influence of transmission protocol overhead to the total data rate. The proposed structure has the advantage of independence of protocol peculiarities.

7 Tape Format

Documentation was available on the old Daresbury format, the CERN EPIO and ZEBRA formats as well as the IN2P3 standard (Version 2, 24 May 89). The latter is best adapted to our needs. It is therefore recommended to adopt the IN2P3 standard with the some specializations.

7.1 Labels and File Format

Standard ANSI Level 1 labels should be used. HDR2 and EOF2 labels are not needed because the information they contain is available in the " FILEH " record of IN2P3. The option to write IBM Labels should only be provided if it is explicitly requested by collaborating laboratories. The File name should be limited to 15 characters to conform with its use later on.

Several files may be written per tape volume. This is especially necessary for EXABYTE. However, no file should extend over one volume. Each file should start with a " FILEH " record. There is enough space in this record for comments, therefore no " COMMENT" records need be supported. Any parameters that will be saved on tape should be written in a " PARAM " record immediately following the " FILEH ". Then the " EVENTH " record and the " EVENTD " records should follow. It is not recommended to write " SCALER ", " SPECTH " and " SPECTD " records in the same file with Event-by-Event data. It is better practice to write such data in a separate file or even a separate volume. If the user community insists, such records should be written at the end of the file.

If the IN2P3 Standard is dropped, Parameters should be written in an ANSI Standard text file using the " PARAM " format. It is suggested that this file be written before *the Event by Event file to facilitate batch processing of the data.*

7.2 The " EVENTH " record

A new event type " EUROGAM" should be declared to indicate the changes in the " EVENTD " record. The data type in bytes 6 to 9 should be fixed to " INT*4 ". The Run Name should be set equal to the File Name. Hence the limitation of the file name to 15 characters.

The Run Number should default to -1 if not set to a positive number greater than zero. It should be limited to the largest positive integer fitting in the data format declared in bytes 9 to 16.

7.3 The " EVENTD " record

The Tape Subsystem will receive standard Data Blocks, bracketed by protocol information. It should remove these 'Brackets' and insert the information described here. The record starts with the IN2P3 Standard header consisting of the ASCI coded word " EVENTD " followed by the record number. This is NOT the original Block Number provided by the event builder but an internal record number generated by the Tape subsystem and counting ALL records in the file. For additional safety this number is compared to the Block Number in the Data Block, which is provided by the Sorting engine and removed by the Tape Subsystem.

The first data word after the header is the run number, binary coded in an INT*4 word. It is followed by the 32-bit Data Block WC. Then the Event Blocks follow, delimited at the end of valid data by the 'empty trigger' block.

Information after this block to the end of the Tape record is meaningless. This form of recording allows reconstruction of the standard Data Block on input from the Tape.