CHAPTER 10

DIGITAL ON-BOARD RECORDER STANDARD

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page intentionally left blank.

CHAPTER 10

DIGITAL ON-BOARD RECORDER STANDARD

10.1 General

A large number of unique and proprietary data structures have been developed lately for specific data recording applications that require unique decoding software programs. Writing unique decoding software, checking the software for accuracy, and decoding the data tapes is extremely time consuming and costly. In addition, in the late 90s the test ranges started to see the implementation of non-tape-based, high-data-rate recorders. The most predominate of these recorders were solid-state memory devices. As high-data-rate digital recorders were fielded, with solid state on the horizon, the Telemetry Group (TG) formed an ad hoc committee for a computer-compatible digital data acquisition standard.

There is a need for a digital data acquisition and on-board recorder standard (see the functional layout at Figure 10–1) that supports a broad range of requirements, including:

a. Data download and interface
b. One or more multiplexed data streams
c. One or more single data streams
d. Read-after-write–
e. Data format definitions
f. Recorder control
g. Media declassification

Specifically, this digital on-board recorder standard shall be compatible with the multiplexing of both synchronous and asynchronous digital inputs such as pulse code modulation (PCM) and MIL-STD-1553 data bus, time, analog, video, ARINC 429, discrete and RS-232/422 communication data. This digital on-board recorder standard will allow use of a common set of playback/data reduction software to take advantage of emerging random access recording media.

10.1.1 Interface Levels.

The purpose of this chapter is to establish a common interface standard for the implementation of digital data acquisition and on-board recording systems by the organizations participating in the Range Commanders Council (RCC). This standard does not imply hardware architecture such as the coupling of data acquisition, multiplexing, and media storage. Five interface levels are contained in this standard (see a through e below). In addition, declassification requirements are discussed in section 10.8.

a. Data Download and Electrical Interface, which is the physical interface for data access, is defined in section 10.4.
b. Interface File Structure, which defines data access structure, is at section 10.5.
c. Data Format Definition, which defines data types and packetization requirements, is defined in section 10.6.

d. Recorder Control and Status, which defines command and control mnemonics, status, and their interfaces, is described in section 10.7.
e. IEEE 1394B Interface To Recorder media is defined in section 10.9.

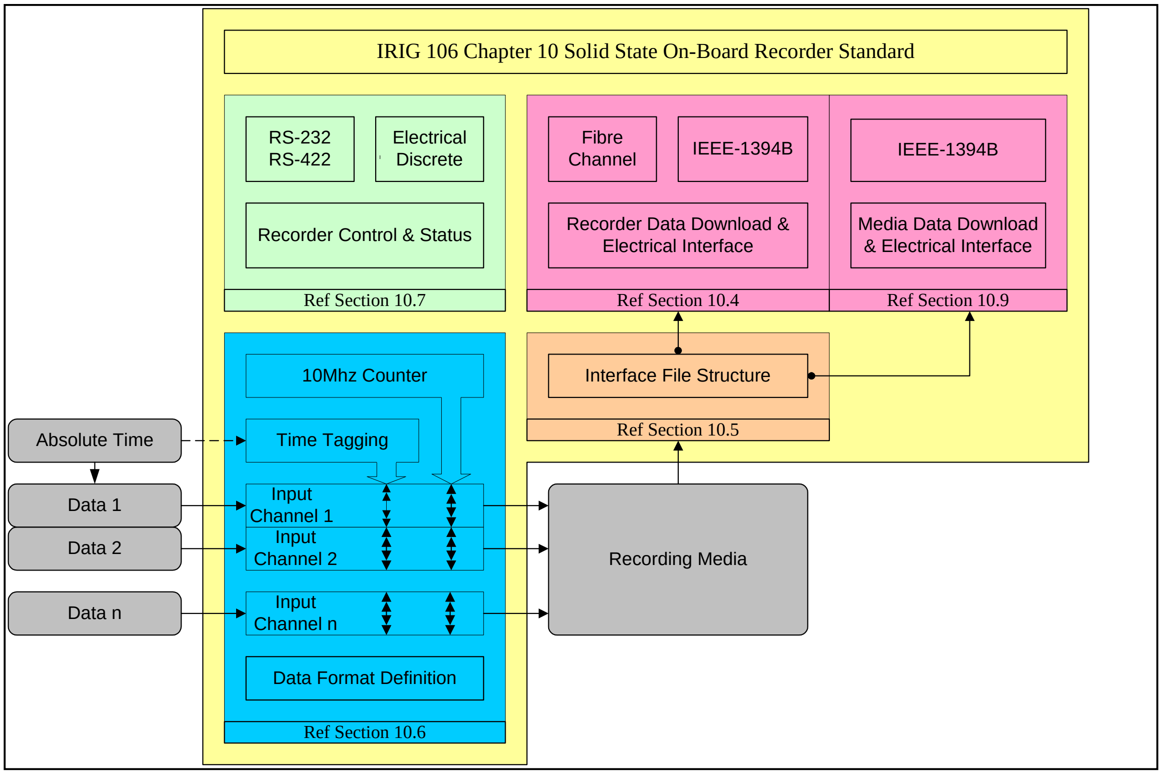


Figure 10-1. Functional layout of standard.

10.2 Definitions

Bad Block: A block, which has been determined to be unreliable for storing user data.

Bad Block Table: A table of bad block entries for a memory board. The data stored in the entry identifies the chip and block number of the bad block. The table entry also contains a flag field. The flag field is used to determine the circumstance in which the bad block was detected. It also provides a flag indicating whether the corresponding bad block has previously been "Secure Erased."

Block: A storage unit within the flash device. A block is the smallest unit of memory that can be erased.

Byte: A contiguous set of 8 bits that are acted on as a unit.

Channel ID: A unique value assigned to each channel in a system. Each channel must have a unique Channel ID (data channels and playback channels).

Channel Specific Data Word: A required word for each data type channel that has data specific information.

Checksum: The arithmetic sum of data bytes or words.

Erasing Flash: Performing an erase function on a flash device. Erasing a flash device sets all bits to a known logic state.

EVPD: Enable vital product data.

Intra-Packet Data Header: A header containing time and status information for the tagging of data inside a packet.

Long Word: A contiguous set of 32 bits that are acted on as a unit.

lsb: The "least significant bit" of a series of bits.

LSB: The "least significant byte" of a series of bytes.

LSW: The "least significant word" of a series of words.

LSLW: The "least significant long word" of a series of long words.

Magic Number: An identifier for the directory block. This identifier is a value chosen to support discovery of lost directory entries and directory reconstruction after a fault.

Memory Board: Printed circuit board containing flash memory devices used to store user data.

msb: The "most significant bit" of a series of bits.

MSB: The "most significant byte" of a series of bytes.

MSW: The "most significant word" of a series of words.

MSLW: The "most significant long word" of a series of long words.

Multiplexer: The entity that includes all the inputs, control interfaces, and functionality required to properly record data.

Non-volatile: Memory media that retains data when power is removed.

Packet: Encapsulates a block of observational and ancillary application data which is to be recorded.

Packet Header: Identifies the source and characteristics of the data packet and encapsulation environment.

Packet Secondary Header: Contains Packet Header time.

Page: Storage unit within the flash device. A page is the smallest storage unit that can be written.

Quad Word: A contiguous set of 64 bits that are acted on as a unit.

Recorder: The entity that includes the input and control interfaces, the Removable Memory Module (RMM), and the functionality required to properly record data.

Recording Session: Time interval from first data packet generated to end of the recording

Relative Time Counter: A free-running 10 MHz binary counter represented by 48-Bits common to all data channels. The counter shall be derived from an internal crystal oscillator and shall remain free running during each recording session. The applicable data bit to which the 48-bit value applies will be defined in each data type section.

Removable Memory Module (RMM): The element of the data recorder that contains the stored data.

Stream: A series of packets that are generated until the end of the recording session.

Word: A contiguous set of 16 bits that are acted on as a unit.

10.3 Operational Requirements

This section of the standard specifies the basic operation and required interfaces for the Data Storage and Download.

10.3.1 Required Configuration. A recorder, as a minimum, shall provide the following functionality:

a. Download port
b. Control/Maintenance port
c. External power port

The required download port interface shall be IAW section 10.4 This combination will allow data extraction and transfer from any recorder to any section 10.4 compliant intermediate storage unit.

10.3.2 Exclusions to Standard. The physical size, configuration and form factor for the recorder and/or the RMM are not controlled by this standard. Due to the variation in capacity/rate/cost requirements of the users, this standard does not specify the technology to be used in the RMM or the recorder.

10.3.3 Internal System Management. Any processing performed on the stored data by the ecorder (e.g. for the purposes of internal system management, error detection and correction (EDAC), physical frame formatting, etc.) shall be removed from the stored data when the stored data is downloaded or transferred from storage media.

10.3.4 Data Download. The data acquisition recorder may have an RMM capability or the recorder can be removed from the acquisition platform and taken to a ground station for data download. Reference paragraph 10.4.1 for recorder electrical interface and paragraph 10.9 for RMM interface.

10.3.5 Data Download File Extension. Upon data download to a host computing platform, all IRIG 106 Chapter 10 compliant recordings shall use the file extension ***.ch10 (or *.c10** extension for use on systems with a 3 character extension limit). The use of this standard extension will indicate that any file on a ground computing or storage platform is in compliance with this standard (defined in section 10.6).

10.3.6 Data Download Structure and Naming. Upon data download to a host computing platform all IRIG 106 Chapter 10 compliant recordings shall use the following structure and naming conventions unless host computing platform operating system imposes naming length limits. In this case the directory and file names are to be truncated after the last component, which completely fits within the name length limit:

a. Data Recording Directory Name. Each directory block from a RMM to be downloaded to ground computing or storage platform shall use Section 10.5, Table 10–2 VolName as the directory name where the Data Files will be placed. The directory name shall use lower case alpha characters.

   Note: If the VolName is empty (0x00) a default name or user-defined name shall be used. If used, the default name shall be ch10dirnnn, where nnn is the sequential directory block count.

b. Data File Name. Each Data File contained within a Directory Block on the RMM to be downloaded will be placed in the Directory identified in paragraph 10.3.6.1 and shall use the following naming convention. The data file name shall use lower case alpha characters:

   "filennnn"; where nnnn is the sequential RMM file count from each Directory Block File Entry (must be 8 alpha-numeric characters). Example "file0001", "file0002", …"file9999".

   If available "File Create Date", "File Create Time" and "File Close Time" from Section 10.5, Table 10–3, DDMMYYYY_HHMMSSss_HHMMSSss (8 numeric characters for File Create Date, 8 numeric characters for File Create Time separated by an underscore ASCII character code 0x5F, and 8 numeric characters for File Close Time). No spaces or other non-numeric characters allowed). Example 02092004_21302731_21451505.

   If the "File Create Date", "File Create Time" and "File Close Time" from Section 10.5, Table 10–3 values are not available and are filled with 0x30 then the system time from the host download platform will be used for "File Create Date" and "File Create Time" (DDMMYYYY_HHMMSS). "File Close Time" will not be used. "File Close Time" shall be replaced with "sys_time".

A structure example follows:

…\VolName\FileName_FileCreateDate_FileCreateTime_FileCloseTime
VolName not empty example:
…\<VolName>\file0001_02092004_21302731_21451505.ch10
VolName empty default example:
…\ch10dir001\file0001_02092004_21302731_21451505.ch10
VolName empty user defined example:
…\<User Defined>\file0001_02092004_21302731_21451505.ch10
Date/Time not available (0x30 fill) example:
…\file0001_02092004_213027_sys_time.ch10

The use of this standard recording and file naming convention will indicate that any file on a ground computing or storage platform is in compliance with this standard.

10.3.7 IEEE-1394b Interface to Recorder Media. Serial Interface to recorder media shall be accomplished utilizing IEEE-1394b interface. A nine-pin IEEE-1394b interface shall be provided on the media to allow direct download of data to host computer or storage device.

10.4 Data Download and Electrical Interface

At a minimum, the required recorder download port interface (see paragraph 10.3.1) shall be Fibre Channel or IEEE1394B. The physical, signaling, and command protocols contained in paragraphs 10.4.1 and 10.4.2 are a subset of, and adapted from the North Atlantic Treaty Organization (NATO) Military Agency for Standardization (MAS) Standardization Agreement (STANAG) NATO Advanced Data Storage Interface (NADSI) Number 4575, hereinafter referred to in this document as STANAG 4575. A second means of recorder download port interface can be provided via IEEE1394B interface (refer to section 10.9 of this standard).

10.4.1 Physical and Signaling. The interface shall comply with FC-PI (Physical Interfaces) and FC-FS (Framing and Signaling) as defined in Section 10.9, with configuration options as specified.

a. Physical Media. Fibre Channel copper interface will be utilized.
b. Signaling Rate. The transmission signaling rate shall be 1.0625 Gbaud.

10.4.2 Command Protocol. The interface shall conform to the requirements of the Fibre Channel Private Loop SCSI Direct Attach (FC-PLDA, ANSI NCITS TR19-1998) interoperability. Table 17 of FC-PLDA specifies a control protocol using a subset of commands, features and parameters defined for the Small Computer System Interface (SCSI-3). Table 17 of FC-PLDA also defines the command feature and parameter usage categories of “Required”, “Allowed”, “Invokable”, and “Prohibited” between the SCSI Initiator and Target. These definitions assume that the Target is a magnetic disk drive or equivalent device.

The control protocol must support a number of data storage media types. Only the minimum set of SCSI commands needed to download mission data from a memory cartridge are defined as “Required.” FC-PLDA SCSI commands, features and parameters not defined as

“Required” for this standard, are redefined as “Allowed” such that they may be implemented as appropriate. Table 10-1 provides the five “Required” SCSI commands and their features and parameter usage definitions.

TABLE 10-1. “REQUIRED” SCSI COMMANDS, FEATURES, AND PARAMETERS			
FEATURE (COMMAND)	**INITIATOR**	**TARGET***	**NOTES**
INQUIRY	I	R	
Standard INQUIRY data (bytes 0-35)	I	R	
EVPD = 1	I	R	
Vital Product Data page codes:			
hex’00’ (supported vital product pages)	I	R	
hex’80’ (unit serial number page)	I	R	
hex’81’ (implemented operations definition pg)	I	A	
hex’82’ (ASCII implemented operations def pg)	I	A	
hex’83’ (device identification page)	I	R	
READ (10)	I	R	
DPO = 0	I	A	1
DPO = 1	I	A	1
FUA = 0	I	A	2
FUA = 1	I	A	2
RelAdr = 0	R	R	
RelAdr = 1	P	P	3
READ CAPACITY	I	R	
RelAdr = 0	R	R	
RelAdr = 1	P	P	3
PMI = 0	I	R	
PMI = 1	I	A	
TEST UNIT READY	I	R	
REQUEST SENSE	I	R	

Notes:
1. The Disable Page Out (DPO) bit is associated with a device data caching policy.
2. The Force Unit Access (FUA) bit is associated with whether the device may or may not return the requested Read data from its local cache.
3. Relative Offset is prohibited, since this requires the use of linking, which is prohibited.

*__LEGEND__

P = Prohibited: The feature shall not be used between FC-PLDA compliant devices.
R = Required: The feature or parameter value shall be implemented by FC-PLDA compliant devices.
A = Allowed: The feature or parameter may be used between FC-PLDA compliant devices. The initiator determines if an Allowed feature/parameter is supported via a required discovery process or a minimal response by the recipient.
I = Invokable: The feature or parameter may be used between FC-PLDA compliant devices. The recipient shall support “Invokable” features or provide a response that it is not implemented as defined by the appropriate standard.

10.5 Interface File Structure Definition

The interface file structure definition in Section 10.5 are a subset of, and were adapted from STANAG 4575, Section 3, File Structure Definition. This file structure was selected to facilitate host computing platform independence and commonality. By incorporating an independent file structure, backward and forward compatibility is ensured for the life of the standard.

> **NOTE**
>
> This file structure definition does not define how data is physically stored on the recorder media but provides a standardized method for access of the stored data at the interface. Data can be physically organized in any way appropriate to the media, including multiple directories, as long as the file structure IAW paragraph 10.5 is maintained or seen at the interface (section 10.4).

10.5.1 Data Organization. A data recording can contain a single file, which is composed of one (1) or more types of packetized data, or multiple files, in which one (1) or more types of data are recorded simultaneously in separate files. For a recording file to be in compliance with this standard it must contain as a minimum the following:

a. Computer Generated Packet, Setup Record Format 1 IAW Section 10.6.7.2 as the first packet in the recording.
b. Time Data Packet(s) IAW Section 10.6.3 as the first dynamic packet after the Computer Generated Packet Set Record.
c. One (1) or more data format packets IAW Section 10.6.

Multiple recordings may reside on the media, and each recording may contain one (1) or more compliant files.

10.5.1.1 Data Hierarchy. The structures used to define the data stored according to this standard shall have the following relationships (highest to lowest):

a. Directory. One or more directory blocks of data comprising a list of all Data Files located under the guidance of this Standard. Also contains supporting data that may be of interest to those manipulating the Data Files. The list of files is made up from “File Entries.” The Directory shall always start at logical address zero of each directory block.

b. Directory Block. A memory block containing file entries and other metadata.

c. Directory Block File Entry. A fixed length data structure used to describe files. It contains the name, the starting address, the number of blocks of data assigned to the Data File, the total number of bytes contained in the file, and the file’s creation date and time. It also contains a reserved field for future growth and file close time.

d. Data Files. Data files are comprised of user data, presented at the interface in monotonically increasing contiguous logical addresses per file. Thus if a file starts at logical address X, the next location containing file data must be at the next logical address, X+1, and the next location after that must be at the next logical address, X+2, etc.

10.5.2 Directory Definition. The name and location information for all files is recorded in a Directory (see Figure 10–2). The Directory is composed of one or more directory blocks as shown in Figure 10–3. At least one Directory Block is required and it must be located at SCSI logical block address 1. Logical block address 0 is reserved.-

a. Directory Fixed Fields. The fixed fields within a directory block are used to name the volume of data, identify the number of entries, and to provide pointers to other addresses that contain additional directory blocks. The forward and backward link to the next address for the next Directory Block (if any) as well as back to the preceding Directory Block (if any). This allows for directory expansion beyond a single block and does not limit the placement of directory information.

b. Block Size. The media types used to implement this standard have varying block lengths. Some will have blocks as small as 512 bytes; others may have blocks as large as 64K bytes or larger. The block size used by a given media can be determined via the SCSI Read Capacity Command and is not defined here.

c. Directory to Data File Link. Each Data File on the media has a directory entry within a Directory Block that describes the file, as shown in Table 10–2. The directory entry for a Data File, as shown in Table 10–3, contains a link to the starting location of the data contained in each file and the total number of blocks assigned for the storage of data. This standard does not define the meaning of the data recorded within these Data File blocks.

d. File Entry Name. Each file entry in a directory shall have a unique name (See Naming Restrictions in Section 10.5.3.2). Default file name is a BCS numeric value incrementally increasing, starting at value “1”.

e. File Entry Singularity. Multiple File entries are not permitted to refer to the same regions of memory, partially or completely.

f. Directory Entries and Fields. Directory block fields and entries shall be logically contiguous.

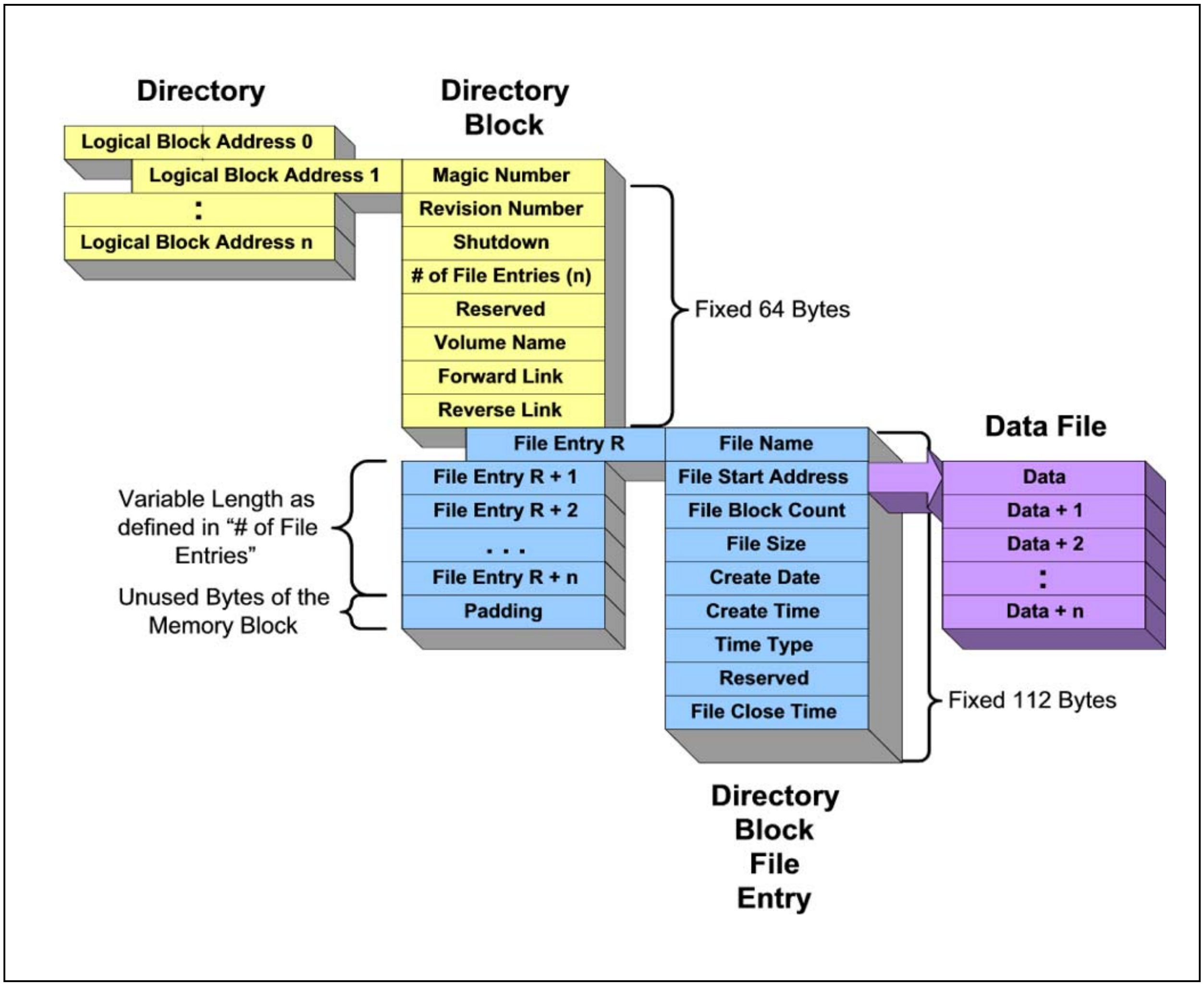


Figure 10-2. Directory structure.

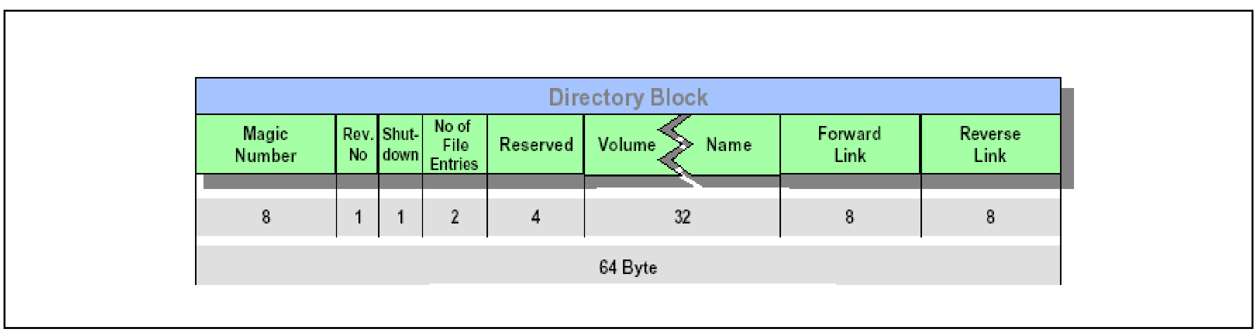


Figure 10-3. Directory block.

TABLE 10-2. DIRECTORY BLOCK FORMAT			
FIELD NAME	**BYTES**	**DESCRIPTION**	**DATA TYPE**
Magic Number	8	An identifier for a directory block. The value is ASCII "FORTYtwo" (Hex – 0x464F52545974776F)	BCS
Revision Number	1	Revision number of the standard compiled by the recording system. Example first version 00000001.	Unsigned Binary
Shutdown	1	Flag, if cleared to a 0x00 indicates that the volume was not properly dismounted, if seen on power-up is an indication that the directory chain may be faulty. If set = 0xFF, then file system properly shutdown. This field is only valid in the first directory block; other directory blocks set to 0xFF.	Unsigned Binary
Number of File Entries	2	Defines the number of file entries that follow in this block.	Unsigned Binary
Reserved	4	Fill with 0xFF. Bytes in this region are reserved for future growth.	Unsigned Binary
VolName	32	Volume name. Fill with all 0x00 for no name and for unused volume name characters.	BCS
Forward Link	8	Block address of the directory block pointing to this block. Set equal to address of this block if this is the end of the chain.	Unsigned Binary
Reverse Link	8	Block address of the directory block pointing to this block. Set equal to the address of this block if this is the start of the chain.	Unsigned Binary
(n File Entries)	112 *n	One entry for each file. Total number of directory entries for this directory block as defined by field "Number of Directory Entries." The value of "n" is dependent upon media block size.	See Table 10–3
Padding	Varies with n and Block Size	Pad to block boundary. Default value is 0xFF	Unsigned Binary
Note: 64 Bytes in fixed fields.			

TABLE 10-3. DIRECTORY ENTRY FORMAT			
FIELD NAME	**BYTES**	**DESCRIPTION**	**DATA TYPE**
Name	56	File name, see character set for restrictions. Fill any unused FileName Byte Positions with 0x00.	BCS
FileStartAdd	8	Zero based address of the first block reserved for data associated with this file. Fill with FFh for unused directory entries.	Unsigned Binary
FileBlkCnt	8	One based number that is the count of consecutive address blocks reserved for data for this file including the block pointed to by the FileStartAdd field.	Unsigned Binary
FileSize	8	The actual number of Bytes contained in this file. This file size will be equal to or less than the FileBlkCnt multiplied by the blocksize. This is an optional entry and will be filled with 0xFF if not used.	Unsigned Binary
File Create Date	8	DDMMYYYY BCS Character values, with no embedded spaces or other formatting characters, representing the numeric date on which the file was created (e.g. BCS codes for the decimal digits 02092000 → 0x30323039 0x32303030 represents 2 September 2000). Fill with 0x2D if a value for the field is not available, or for portions of the field where data is not available.	BCS
File Create Time	8	HHMMSSss Character values, with no embedded spaces or other formatting characters, representing the numeric time at which the file was created. HH is the number of the 24 hour based hour, MM is the number of minutes after the hour, SS is the number of seconds after the minute, and ss is the hundredths of seconds after the second. Fill with 0x2D if this value is not available, or for portions of the field where data is not available, e.g, "ss" is not available.	BCS
Time Type	1	A numeric code that qualifies the time and date values recorded in the "Create Date" and "Create Time" and "Close Time" fields. 0x00 = Coordinated Universal Time (Zulu) 0x01 = System Time 0x02 - 0xFE = Reserved 0xFF = Time Data Packet	Unsigned Binary
Reserved	7	Bytes in this region are reserved for future growth. Fill with 0xFF.	Unsigned Binary
File Close Time	8	HHMMSSss Character values, with no embedded spaces or other formatting characters, representing the numeric time at which the file was closed. HH is the number of the 24 hour based hour, MM is the number of minutes after the hour, SS is the number of seconds after the minute, and ss is the hundredths of seconds after the second. Fill with 0x2D if this value is not available, or for portions of the field where data is not available, e.g, "ss" is not available.	BCS
Note: 112 Bytes in fixed fields.			

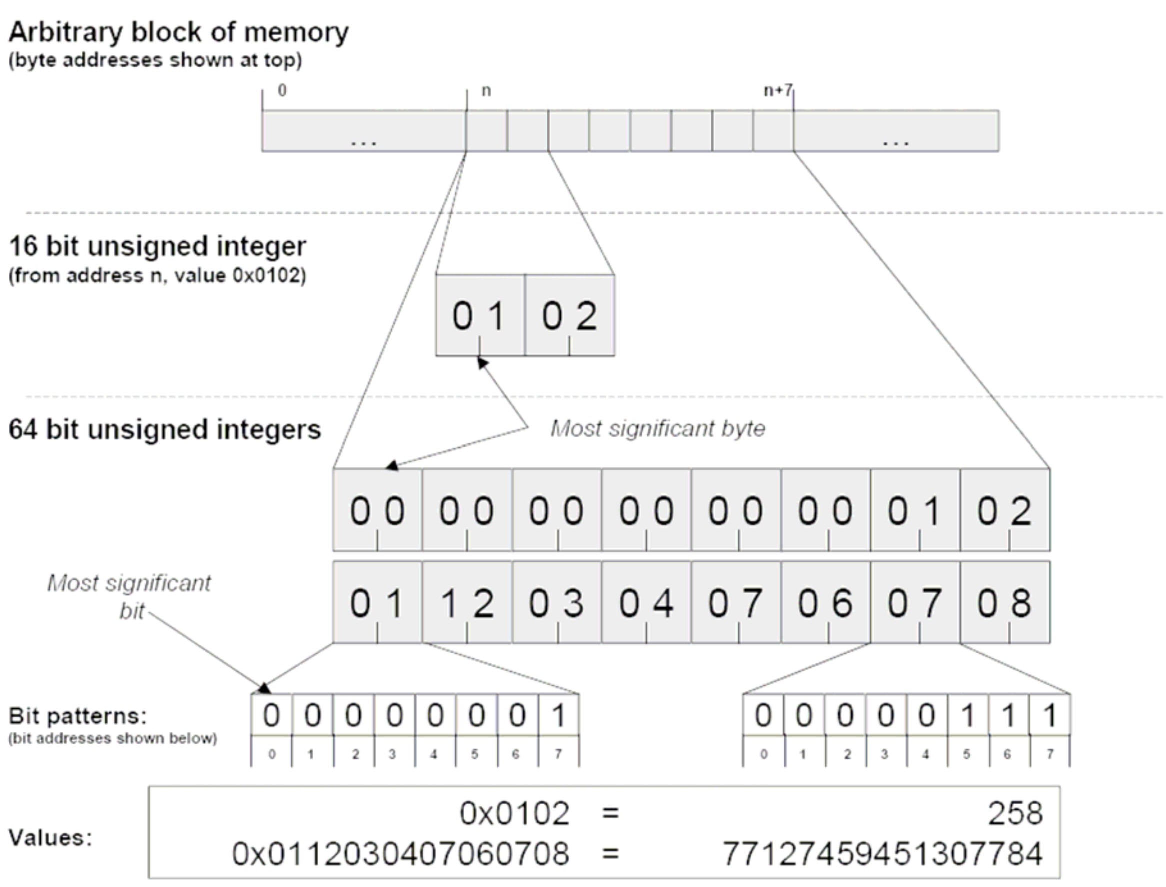
10.5.3 Data Definitions.

10.5.3.1 Directory Byte Order. The directory structures described in this section 10.5 of the standard are defined to have the following bit and byte orientation. The most significant byte of any multi-byte structure is byte 0. The most significant bit of each byte is bit 0. This ordering is commonly referred to as **"Big Endian."** This ordering is illustrated in Table 10-4 below.

TABLE 10-4. DIRECTORY BYTE ORDER (CORRESPONDENCE OF BITS TO BYTES)

	Bit 0	1	2	3	4	5	6	7
Byte [0]	31	30	29	28	27	26	25	24
Byte [1]	23	22	21	20	19	18	17	16
Byte [2]	15	14	13	12	11	10	9	8
Byte [3]	7	6	5	4	3	2	1	0

Bit and Byte Patterns: An Example



10.5.3.2 Data Format Byte Order. The data format structures described in section 10.6 of this standard are defined to have the following bit and byte orientation. The least significant byte shall be transmitted first, the least significant bit of each byte shall be transmitted first, and data is read from the lowest logical address first. This ordering is commonly referred to as **"Little Endian**." This ordering is illustrated in Table 10-5 below.

TABLE 10-5. DATA FORMAT BYTE ORDER (CORRESPONDENCE OF BITS TO BYTES)								
	Bit 7	6	5	4	3	2	1	0
Byte [3]	31	30	29	28	27	26	25	24
Byte [2]	23	22	21	20	19	18	17	16
Byte [1]	15	14	13	12	11	10	9	8
Byte [0]	7	6	5	4	3	2	1	0

10.5.3.3 Character-set. The character-set for all character fields is based on ISO/IEC 10646-1, the Universal Multiple-Octet Coded Character Set (UCS). The NIIA limits characters to a subset rather than allowing all characters. The subset will be single octets and be referred to as the Basic Character Set (BCS).

10.5.3.4 Naming Restrictions. The following rules shall be applied when forming names in order to assure the highest degree of interchange among other operating systems.

a. Characters. Characters from the first 127 common BCS characters (00h through 7Eh) may be used in names except for specific prohibited characters.

Any BCS character code value smaller than 20h is prohibited, except where the <00h> is used to terminate the name.
The other prohibited characters with their hexadecimal representation are defined in Table 10–5A.

<table>
<tr><th colspan="4">TABLE 10-5A. PROHIBITED CHARACTERS
(HEXADECIMAL REPRESENTATION)</th></tr>
<tr><th>Forbidden Characters In Names</th><th>Hexadecimal Value</th><th>Forbidden Characters In Names</th><th>Hexadecimal Value</th></tr>
<tr><td>”</td><td>0x22</td><td>=</td><td>0x3D</td></tr>
<tr><td>‘</td><td>0x27</td><td>></td><td>0x3E</td></tr>
<tr><td>*</td><td>0x2A</td><td>?</td><td>0x3F</td></tr>
<tr><td>/</td><td>0x2F</td><td>\</td><td>0x5C</td></tr>
<tr><td>:</td><td>0x3A</td><td>]</td><td>0x5D</td></tr>
<tr><td>;</td><td>0x3B</td><td>[</td><td>0x5B</td></tr>
<tr><td><</td><td>0x3C</td><td>|</td><td>0x7C</td></tr>
</table>

b. Names. Names used for this interface will observe the following rules:

1) Upper and lowercase characters are considered to be different within file names.
2) Leading and trailing spaces are not permitted.
3) Leading periods are not permitted.
4) Names shall fill their field with byte 0 (per section 10.5.3.1) and be terminated with a 0x00. Unused Name charters shall be filled with 0x00. Names may utilize the full length of the field, in which case the terminating 0x00 must be omitted.

File Name examples are provided in Table 10-5b.

TABLE 10-5B. FILE NAME EXAMPLES

File Name Byte Address	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	//	50	51	52	53	54	55
Host Provided File Name Example	R	E	C	O	R	D	I	N	G	1	S	E	N	S	O	R	2	0X 00	0X 00	//	0X 00	0X 00	0X 00	0X 00	0X 00	0X 00
Default File Name Example	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	// 0	0	0	0	0	0	1

10.6 Data Format Definition

10.6.1 Common Packet Elements. Data shall have three required parts, a Packet Header, a Packet Body, a Packet Trailer, and an optional part if enabled, a Packet Secondary Header. Single or multiple channel recordings will always conform to the structure outlined in Figure 10–4.

A packet has the basic structure shown in Figure 10–5. Note that the width of the structure is not related to any number of bits. This table is merely to represent relative packet elements and their placement within the packet. See Figure 10–6 for a diagram of the generic packet format. This figure does not depict the bit lengths of each field. Word sizes of 8-bit, 16-bit and 32-bit are used depending on the data type.

To further clarify the packet layout, Figure 10–6 shows the generic packet in a 32-bit, little-endian format, and assumes 16-bit data words and data checksum.

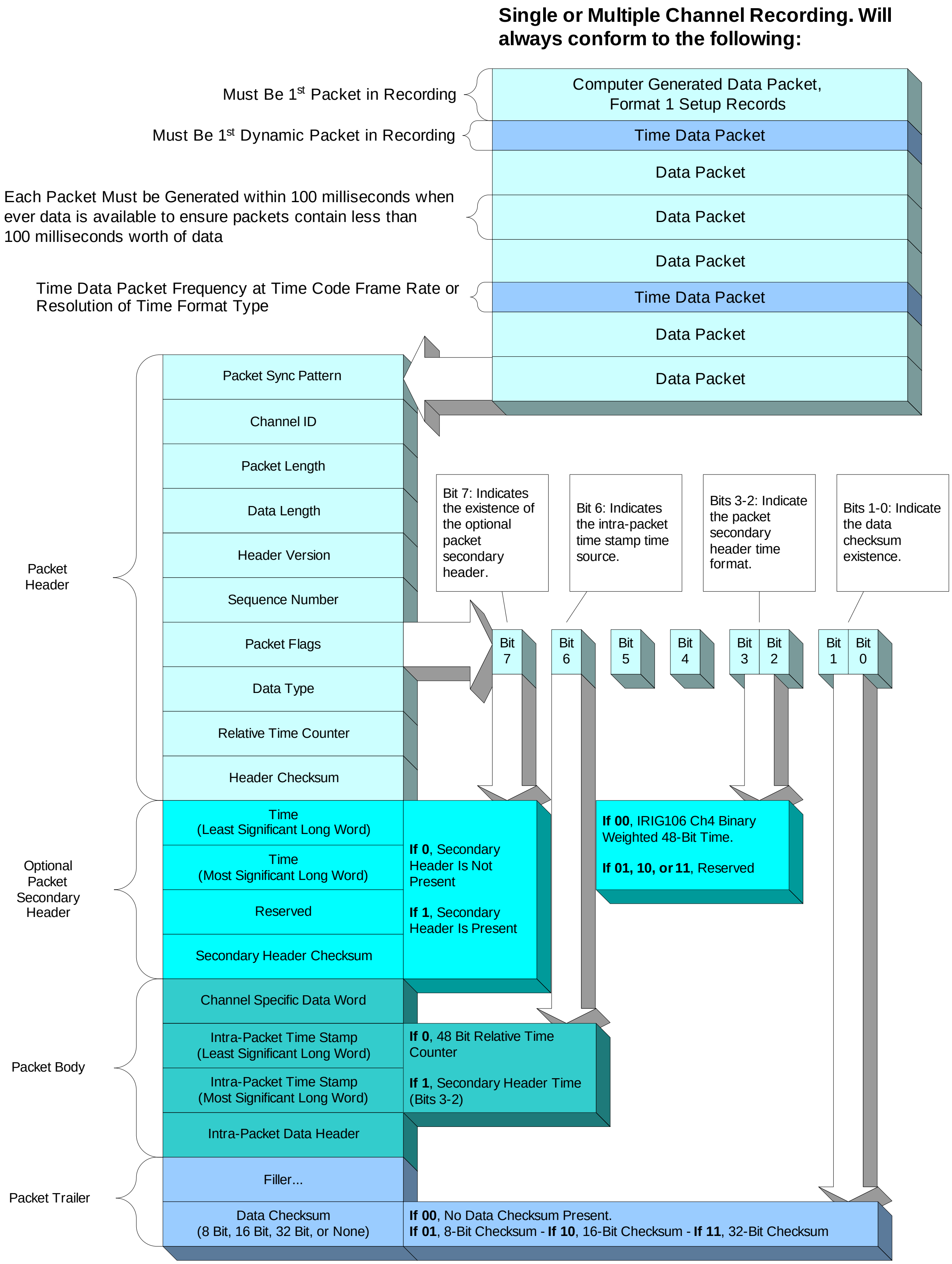


Figure 10-4. Data recording structure.

PACKET SYNC PATTERN	Packet Header
CHANNEL ID	
PACKET LENGTH	
DATA LENGTH	
HEADER VERSION	
SEQUENCE NUMBER	
PACKET FLAGS	
DATA TYPE	
RELATIVE TIME COUNTER	
HEADER CHECKSUM	
TIME	Packet Secondary Header (Optional)
RESERVED	
SECONDARY HEADER CHECKSUM	
CHANNEL SPECIFIC DATA	Packet Body
INTRA-PACKET TIME STAMP 1	
INTRA-PACKET DATA HEADER 1	
DATA 1	
:	
INTRA-PACKET TIME STAMP n	
INTRA-PACKET DATA HEADER n	
DATA n	
DATA CHECKSUM	Packet Trailer

Figure 10-5. General packet format.

msb 31	16	15	lsb 0	
CHANNEL ID		PACKET SYNC PATTERN		Packet Header
PACKET LENGTH				
DATA LENGTH				
DATA TYPE	PACKET FLAGS	SEQUENCE NUMBER	HEADER VERSION	
RELATIVE TIME COUNTER				
HEADER CHECKSUM		RELATIVE TIME COUNTER		
TIME (LSLW)				(Optional) Packet Secondary Header
TIME (MSLW)				
SECONDARY HEADER CHECKSUM		RESERVED		
CHANNEL SPECIFIC DATA				Packet Body
INTRA-PACKET TIME STAMP 1				
INTRA-PACKET TIME STAMP 1				
INTRA-PACKET DATA HEADER 1				
DATA 1 WORD 2		DATA 1 WORD 1		
DATA 1 WORD n		:		
INTRA-PACKET TIME STAMP 2				
INTRA-PACKET TIME STAMP 2				
INTRA-PACKET DATA HEADER 2				
DATA 2 WORD 2		DATA 2 WORD 1		
DATA 2 WORD n		:		
:				
INTRA-PACKET TIME STAMP N				
INTRA-PACKET TIME STAMP N				
INTRA-PACKET DATA HEADER N				
DATA N WORD 2		DATA N WORD 1		
DATA N WORD n		:		
[FILLER]				
DATA CHECKSUM				Packet Trailer

Figure 10-6. A 32-Bit packet format layout.

Depending on the data type, the size of the Data Checksum can be 16-bits, 32-bits, 8-bits, or left out entirely. For a 32-bit Data Checksum, the packet trailer would be as shown in Figure 10–7.

msb 7 lsb 0	
[FILLER]	Packet Trailer
DATA CHECKSUM (LSB)	
DATA CHECKSUM	
DATA CHECKSUM	
DATA CHECKSUM (MSB)	

Figure 10-7. Packet trailer for 32 bit data checksum.

For an 8-bit Data Checksum, the packet trailer would be as shown in Figure 10–8.

msb 7 lsb 0	
[FILLER]	Packet Trailer
DATA CHECKSUM	

Figure 10-8. Packet trailer for 8-bit data checksum.

The size of a single Packet may be a maximum of 524,288 bytes as shown in Table 10–5C. This includes the Packet Header, Packet Body, Packet Trailer, and optional Packet Secondary Header if enabled. The only exception to the packet size limit is the Computer Generated Data Packet, Format 1 Setup Record which may be a maximum of 134,217,728 bytes. Any Packet which requires more than 524,288 bytes may do so by utilizing the packet sequence counter and generating multiple packets. Some packet types allow a single data set to span multiple packets if the data set size or time does not fall under packet maximums. Consult the individual data types for the mechanisms that allow packet data spanning for a given type.

	With the exception of Computer Generated Packets all other packets shall be generated within 100 milliseconds whenever data is available. This requirement ensures that a packet shall contain less than or equal to 100 milliseconds worth of data, and that a packet containing any data must be generated within 100 milliseconds from the time the first data was placed in the packet. This strategy will assure packet granularity but save bandwidth by not forcing or marking empty/idle packets.

	Packets can not contain only filler or can not be idle or empty. All packets that are generated shall contain data.

NOTE

All reserved bit fields in packet headers or channel specific data words must be set to zero (0).

NOTE

With the exception of Computer Generated Data Packets all other packets must be committed to a Stream within 10,000,000 counts of the 10 MHz Relative Time Counter contained in the packet header.

TABLE 10-5C. PACKET REQUIREMENTS

PACKET TYPE	REQUIRED	MAXIMUM PACKET SIZE	REQUIRED PACKET LOCATION
Computer Generated Data Packet, Format 1 Setup Record	Yes	134,217,728 bytes	First Packet in Recording.
Time Data Packet	Yes	524,288 bytes	First Dynamic Data Packet Following Setup Record Packet(s). Reference the Time Data Packet Description for packet rate.
PCM, Mil-Std-1553, Analog, Message, Discrete, ARINC 429, Video, Image, UART, Computer Generated Data Packet, Format 2 Recording Events, Computer Generated Data Packet, Format 3 Recording Index (Node Index)	No	524,288 bytes	After First Time Data Packet and before the first Computer Generated Data Packet Format 2, Recording Index (Root Index) if enabled.
Computer Generated Data Packet, Format 3 Recording Index (Root Index)	Yes, if Recording Events are Enabled. No, if Recording Events are Disabled.	524,288 bytes	If Recording Index Packets are enabled the Root Index Packet Type will be the last packet(s) in a recording.

10.6.1.1 Packet Header. The length of the packet header is fixed at 24 bytes (192-bits). The Packet Header is mandatory and shall consist of the ten fields, positioned contiguously, in the following sequence:

a. Packet Sync Pattern. (2 Bytes) contain a static sync value for the every packet. The Packet Sync Pattern value shall be 0xEB25.

b. Channel ID. (2 Bytes) contain a value representing the Packet Channel ID. All channels in a system must have a unique value (data channels). Channel value 0x0000 is reserved, and is used to insert computer-generated messages into the composite data stream. Channel values 0x0001 thru 0xFFFF are available.

When a distributed Multiplexer system is used, the setup record will contain a recorder attribute that identifies the number of most significant bits used from the Channel ID field that identifies the channels from each multiplexer. The least significant bits of the Channel ID will provide the unique Channel ID for each data source acquired by a given multiplexer.

c. Packet Length. (4 Bytes) contain a value representing the length of the entire packet. The value shall be in bytes and is always a multiple of four (bits 1 and 0 shall always be zero). This Packet Length includes the Packet Header, Packet Secondary Header (if enabled), Channel Specific Data, Intra-Packet Data Headers, Data, Filler and Data Checksum.

d. Data Length. (4 Bytes) contain a value representing the valid data length within the packet. This value shall be represented in bytes. Valid data length includes Channel Specific Data, Intra-Packet Data Headers, Intra-Packet Time Stamp, and Data but does not include filler and Data Checksum.

e. Header Version. (1 Byte) contains a value representing the version of the Packet Header for Chapter 10 packets. The value shall be represented by the following bit patterns:

0x00 = Reserved
0x01 = Initial Release Header Version
0x02 = TG-78
0x03 = thru 0xFF = Reserved

f. Sequence Number. (1 Byte) contains a value representing the packet sequence number for each Channel ID. This is simply a counter that increments by n + 0x01 to 0xFF for every packet transferred from a particular channel and is not required to start at 0x00 for the first occurrence of a packet for the Channel ID.

NOTE

Sequence Number counter will repeat (roll over to 0x00) if more that 256 packets are transferred in a given recording per channel.

g. Packet Flags. (1 Byte) contain bits representing information on the content and format of the packet(s).

Bit 7: Indicates the presence or absence of the Packet Secondary Header.

0 = Packet Secondary Header is not present.
1 = Packet Secondary Header is present.

Bit 6: Indicates the Intra-Packet Time Stamp Time Source.

0 = Packet Header 48-Bit Relative Time Counter.
1 = Packet Secondary Header Time (Bit 7 must = 1).

Bit 5: Relative Time Counter Sync Error.

0 = No Relative Time Counter sync error.
1 = Relative Time Counter sync error has occurred.

Bit 4: Indicates the Data Overflow Error.

0 = No data overflow.
1 = Data overflow has occurred.

Bits 3-2: Indicate the Packet Secondary Header Time Format.

00 = IRIG 106 Chapter 4 binary weighted 48-bit time format. The two LSB's of the 64-bit Packet Secondary Header Time and Intra-Packet Time Stamp shall be zero filled.
01 = Reserved
10 = Reserved
11 = Reserved

Bits 1-0: Indicate Data Checksum existence.

00 = No data checksum present
01 = 8-bit data checksum present
10 = 16-bit data checksum present
11 = 32-bit data checksum present

h. Data Type. (1 Byte) contains a value representing the type and format of the data. All values not used to define a data type are reserved for future data type growth:

Packet Header Value	Data Type Name	Data Type Description
0x00	Computer Generated Data, Format 0	(User Defined)
0x01	Computer Generated Data, Format 1	(Setup Record)
0x02	Computer Generated Data, Format 2	(Recording Events)
0x03	Computer Generated Data, Format 3	(Recording Index)
0x04 – 0x07	Computer Generated Data, Format 4 – Format 7	(Reserved for future use)
0x08	PCM Data, Format 0	(Reserved for future use)
0x09	PCM Data, Format 1	(IRIG 106 Chapter 4)
0x0A – 0x0F	PCM Data, Format 2 – Format 7	(Reserved for future use)
0x10	Time Data, Format 0	(Reserved for future use)
0x11	Time Data, Format 1	(IRIG/GPS/RTC)
0x12 – 0x17	Time Data, Format 2 – Format 7	(Reserved for future use)
0x18	MIL-STD-1553 Data, Format 0	(Reserved for future use)
0x19	MIL-STD-1553 Data, Format 1	(Mil-Std-1553B Data)
0x1a – 0x1F	MIL-STD-1553 Data, Format 2 – Format 7	(Reserved for future use)
0x20	Analog Data, Format 0	(Reserved for future use)
0x21	Analog Data, Format 1	(Analog Data)
0x22 – 0x27	Analog Data, Format 2 – Format 7	(Reserved for future use)
0x28	Discrete Data, Format 0	(Reserved for future use)
0x29	Discrete Data, Format 1	(Discrete Data)
0x2A – 0x2F	Discrete Data, Format 2 – Format 7	(Reserved for future use)
0x30	Message Data, Format 0	(Generic Message Data)
0x31 – 0x37	Message Data, Format 1 – Format 7	(Reserved for future use)
0x38	ARINC 429 Data, Format 0	(ARINC429 Data)
0x39– 0x3F	ARINC 429 Data, Format 1 – Format 7	(Reserved for future use)
0x40	Video Data, Format 0	(MPEG-2 Video)
0x41	Video Data, Format 1	(ISO 13818-1 MPEG-2 Bit Stream)
0x42 – 0x47	Video Data, Format 2 – Format 7	(Reserved for future use)
0x48	Image Data, Format 0	(Image Data)
0x49 – 0x4F	Image Data, Format 1 – Format 7	(Reserved for future use)
0x50	UART Data, Format 0	(UART Data)
0x51 – 0x57	UART Data, Format 1 – Format 7	(Reserved for future use)
0x58	IEEE-1394 Fire Wire Data, Format 0	(IEEE-1394 Data)
0x59 – 0x5F	IEEE-1394 Fire Wire Data, Format 1 – Format 7	(Reserved for future use)
0x60	Parallel Data, Format 0	(Parallel Data)
0x61 – 0x67	Parallel Data, Format 1 – Format 7	(Reserved for future use)

i. Relative Time Counter. (6 Bytes) contain a value representing the Relative Time Counter.

> **NOTE**
>
> This is a free-running 10 MHz binary counter represented by 48-Bits common to all data channels. The counter shall be derived from an internal crystal oscillator and shall remain free running during each recording session. The applicable data bit to which the 48-bit value APPLIES unless defined in each data type section shall correspond to the first bit of the data in the packet body.

j. Header Checksum. (2 Bytes) contain a value representing a 16-bit arithmetic sum of all 16-bit words in the header excluding the Header Checksum Word.

10.6.1.2 Packet Secondary Header (Optional). The length of the Packet Secondary Header is fixed at 12 bytes (96-bits). The Packet Secondary Header is optional and when enabled shall consist of the three fields, positioned contiguously, in the following sequence:

a. Time. (8 Bytes) contain the value representing Time in the format indicated by bits 2 and 3 of the Packet Flags in section 10.6.1.1g.

b. Reserved. (2 Bytes) are reserved and shall be zero filled.

c. Secondary Header Checksum. (2 Bytes) contain a value representing a 16-bit arithmetic sum of all Secondary Header bytes excluding the Secondary Header Checksum Word.

10.6.1.3 Packet Body. The format of the data in the packet body is unique to each channel type. Detailed descriptions of the type-specific data formats found in packet bodies are described in subsequent sections of this document.

a. Channel Specific Data. (Variable Bytes) contain the number and contents of the Channel Specific Data field(s) depending on the Data Type field in the Packet Header. Channel Specific Data is mandatory for each data type and channel. The occurrence of Channel Specific Data is once per packet and precedes packet channel data.

b. Intra-Packet Time Stamp. (8 Bytes) contain Time in either 48-bit Relative Time Counter format (plus 16 high-order zero bits) or 64-bit absolute format as specified in the Packet Flags in the Packet Header. The Intra-Packet Time Stamps are only mandatory where defined by the data formats.

c. Intra-Packet Data Header. (Variable Bytes) contain additional status and format information pertaining to the data items that follow. The Intra-packet Data Headers are only mandatory where defined by the data formats.

d. Data. (n Bytes) contain valid data from a particular channel as defined within the data formats contained within this standard.

> **NOTE**
>
> The Intra-Packet Time Stamp and the Intra-Packet Data Header are collectively called the Intra-Packet Header. In some cases an Intra-Packet Header may only have a Time Stamp (zero-length Data Header), while in other cases, the Intra-Packet Header only has a Data Header (zero-length Time Stamp). Some data types have no Intra-Packet Header. The Intra-Packet Header requirements are specified separately for each Data Type.

10.6.1.4 Packet Trailer. The packet trailer may contain filler, a data checksum, both filler and a data checksum, or neither filler nor a data checksum. In the latter case, the packet trailer has zero length. The reason a packet trailer would have a zero length is best explained by understanding the reason for inserting filler. The purpose of the filler is twofold:

a. Keep all packets aligned on 32-bit boundaries (i.e., make all packet lengths a multiple of 4 bytes), and
b. Optionally keep all packets from a particular channel the same length.

If both of the above requirements are already met without adding filler, then filler will not be added.

The inclusion of the data checksum is optional as well and is indicated by the Packet Flags setting. When included, the packet trailer contains either an 8-bit, 16-bit or 32-bit Data Checksum. Depending on the Packet Flags option selected, the Data Checksum is the arithmetic sum of all of the bytes (8-bits), words (16-bits) or long words (32-bits) in the packet excluding the 24 bytes of Packet Header Words, Packet Secondary Header (if enabled) and the Data Checksum Word. Stated another way, the Data Checksum includes everything in the packet body plus all added filler.

c. Filler. (n Bytes/Bits) All filler shall be set to 0x00 or 0Xff

d. 8-Bit Data Checksum. (1 Byte) contains a value representing an 8-bit arithmetic sum of the bytes in the packet (includes Channel Specific Data, Data and Filler). Only inserted if Packet Flag bits 0 and 1 = 01.

e. 16-Bit Data Checksum. (2 Bytes) contain a value representing a 16-bit arithmetic sum of the words in the packet (includes Channel Specific Data, Data and Filler). Only inserted if Packet Flag bits 0 and 1 = 10.

f. 32-Bit Data Checksum. (4 Bytes) contain a value representing a 32-bit arithmetic sum of the long words in the packet (includes Channel Specific Data, Data and Filler). Only inserted if Packet Flag bits 0 and 1 = 11.

10.6.2 PCM Data Packets.

10.6.2.1 PCM Data Packets Format 0. Reserved.

10.6.2.2 PCM Data Packets Format 1. A packet with PCM data has the basic structure shown in Figure 10–9. Note that the width of the structure is not related to any number of bits. This table is merely to represent relative placement of data in the packet.

PACKET HEADER
CHANNEL SPECIFIC DATA
(Optional) INTRA-PACKET TIME STAMP
(Optional) INTRA-PACKET DATA HEADER
MINOR FRAME DATA
(Optional) INTRA-PACKET TIME STAMP
(Optional) INTRA-PACKET DATA HEADER
MINOR FRAME DATA
(Optional) INTRA-PACKET TIME STAMP
(Optional) INTRA-PACKET DATA HEADER
MINOR FRAME DATA
(Optional) INTRA-PACKET TIME STAMP
(Optional) INTRA-PACKET DATA HEADER
MINOR FRAME DATA
:
(Optional) INTRA-PACKET TIME STAMP
(Optional) INTRA-PACKET DATA HEADER
MINOR FRAME DATA
PACKET TRAILER

Figure 10-9. General PCM data packet, format 1.

The user may separately enable or disable word unpacking on each active PCM channel. Word unpacking will force the least significant bit of each word to be aligned on a 16-bit boundary. High-order filler bits are added to words as necessary to force alignment.

The user may separately enable or disable frame synchronizing on each active PCM channel. This provides a Throughput Mode that will transfer data to the packet without frame synchronization. Throughput Mode essentially disables all setup and packing/unpacking options for the packet, and just puts data in the packet as it is received.

a. PCM Packet Channel Specific Data. The packet body portion of each PCM packet begins with the channel specific data, which is formatted as shown in Figure 10–10.

msb 31	30	29	28	27 24	23 18	17 lsb 0
R	IPH	MA	MI	LOCKST	MODE	SYNCOFFSET

- Sync Offset (SYNCOFFSET). (Bits 17-0) contain an 18-bit binary value representing the Word offset into the major frame for the first data word in the packet. Not valid for Packed or Throughput Mode.

- Mode (MODE). (Bits 23-18) indicate the data packing mode.

  Bits 23-22: Reserved.

  Bit 21: Alignment Mode.
  0 = 16 Bit Alignment Mode enabled.
  1 = 32 Bit Alignment Mode enabled.

  Bit 20: Indicates Throughput Data Mode.
  0 = Throughput Data Mode not enabled.
  1 = Throughput Data Mode enabled.

  Bit 19: Indicates Packed Data Mode.
  0 = Packed Data Mode not enabled.
  1 = Packed Data Mode enabled.

  Bit 18: Indicates Unpacked Data Mode.
  0 = Unpacked Data Mode not enabled.
  1 = Unpacked Data Mode enabled.

- Lock Status (LOCKST). (Bits 27-24) indicate the lock status of the frame synchronizer. Not valid for Throughput Mode.

  Bits 27-26: Indicates Minor Frame Status.
  00 = Reserved.
  01 = Reserved.
  10 = Minor Frame Check (after losing Lock).
  11 = Minor Frame Lock.

  Bits 25-24: Indicates Major Frame Status.
  00 = Minor Frames Only.
  01 = Reserved.
  10 = Major Frame Check (after losing Lock).
  11 = Major Frame Lock.

(Continued on next page)

(Figure 10-10 Continued)

- Minor Frame Indicator (MI). (Bit 28) indicate if the first word in the packet is the beginning of a minor frame. Not valid for Throughput Mode.

  0 = First word is not the beginning of a minor frame.
  1 = First word is the beginning of a minor frame.

- Major Frame Indicator (MA). (Bit 29) indicates if the first word in the packet is the beginning of a major frame. Not valid for Throughput Mode.

  0 = First word is not the beginning of a major frame.
  1 = First word is the beginning of a major frame.

- Intra-Packet Header (IPH). (Bit 30) indicates if Intra-Packet Headers (Intra-Packet Time Stamp and Intra-Packet Data Header) are inserted before each minor frame. Intra-Packet Headers are only optional because of the mode selection. This determines whether Intra-Packet Headers are included or omitted.

  0 = Intra-Packet Headers are omitted for Throughput Mode.
  1 = Intra-Packet Headers are required for Packed Data and Unpacked Data modes.

- Reserved. (Bit 31) is reserved.

Figure 10-10. PCM packet channel specific data format.

b. PCM Packet Body. After the Channel Specific Data, the PCM data and Intra-Packet Headers are inserted in the packet in integral numbers of minor or major frames, unless the packet is in Throughput Mode. In Throughput Mode, there is no frame or word alignment to the packet data and no Intra-Packet Headers are inserted in the data. In both Packed and Unpacked Modes, minor frame alignment is dependent on the MODE field in the Channel Specific Data. In 16 Bit Alignment Mode, PCM minor frames begin and end on 16-bit boundaries. In 32 Bit Alignment Mode, PCM minor frames begin and end on 32-bit boundaries. In either case, Alignment Mode does not affect the format of PCM data words themselves. However, depending on perspective, word order is affected, and a zero-filled data word may be required to maintain alignment.

c. PCM Data in Unpacked Mode. In Unpacked Mode, packing is disabled and each data word is padded with the number of filler bits necessary to align the first bit of each word with the next 16-bit boundary in the packet. For example, 4 pad bits are added to 12 bit words, 6 pad bits are added to 10 bit words, etc. In 32 Bit Alignment Mode, a zero-filled 16 bit word is required to maintain alignment when an odd number of 16 bit words exist in the minor frame.

Minor frame sync patterns larger than 16-bits are divided into two words of packet data. If the sync pattern has an even number of bits, then it will be divided in half and placed in two packet words. For example, a 24-bit sync pattern is broken into two (2) 12-bit words with 4-bits of pad in each word. If the sync pattern has an odd number of bits, it is broken into two (2) words with the second word having one-bit more of the sync pattern. For example, if the minor sync pattern is 25 bits, then the first sync word is 12-bits of sync pattern plus 4-bits of pad, and the second sync word is 13 bits of sync pattern plus 3-bits of pad.

Minor frame sync patterns larger than 32 bits are divided into (Number of bits+15) / 16 words in 16 Bit Alignment Mode, or (Number of bits+31) / 32 in 32 Bit Alignment Mode. If the sync word doesn't fill the words completely, the first words shall contain less number of bits in the first words and one bit more in the later words similarly to split frame sync pattern words into two words as described above, e.g. a 35-bit sync word shall be split into 11+12+12 bit words in 16 Bit Alignment Mode, or 17+18-bit words in 32 Bit Alignment Mode.

Given PCM frames with a 24-bit minor sync pattern and n data words where the bit lengths of data words 1, 2, and 3 are 12, 16, and 8 respectively, the resultant PCM packets are as shown in Figure 10–11.

d. PCM Data in Packed Mode. In Packed Mode, packing is enabled and pad is not added to each data word. However, filler bits may be required to maintain minor frame alignment. The number of filler bits is dependent on the Alignment Mode, where N is either 16 or 32. If the number of bits in the minor frame is not an integer multiple of N, then Y PAD bits will be added to the end of each minor frame of bit length L. Either Y = N – MOD(L,N), or N minus the integer remainder when L is divided by N. In packed mode, the PCM stream is minor frame synchronized so the first data bit in the packet is the first data bit of a minor frame. If X = N – Y, then the resultant PCM packets are as shown in Figure 10–12.

<table>
<tr><td colspan="2">msb
15</td><td>lsb
0</td></tr>
<tr><td colspan="3">PACKET HEADER</td></tr>
<tr><td colspan="3">CHANNEL SPECIFIC DATA (BITS 15-0)</td></tr>
<tr><td colspan="3">CHANNEL SPECIFIC DATA (BITS 31-16)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 15-0)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 31-16)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 47-32)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 63-48)</td></tr>
<tr><td colspan="3">INTRA-PACKET DATA HEADER (BITS 15-0)</td></tr>
<tr><td colspan="3">INTRA-PACKET DATA HEADER (BITS 31-16)
(32 Bit Alignment Mode ONLY)</td></tr>
<tr><td>4-BITS PAD</td><td colspan="2">12-BITS SYNC (BITS 23-12)</td></tr>
<tr><td>4-BITS PAD</td><td colspan="2">12-BITS SYNC (BITS 11-0)</td></tr>
<tr><td>4-BITS PAD</td><td colspan="2">12-BITS WORD 1 DATA</td></tr>
<tr><td colspan="3">16-BITS WORD 2 DATA</td></tr>
<tr><td colspan="2">8-BITS PAD</td><td>8-BITS WORD 3 DATA</td></tr>
<tr><td colspan="3">:</td></tr>
<tr><td colspan="3">WORD n DATA BITS + PAD IF NEEDED</td></tr>
<tr><td colspan="3">ZERO FILLED PAD WORD IF NEEDED
(32 Bit Alignment Mode ONLY)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 15-0)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 31-16)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 47-32)</td></tr>
<tr><td colspan="3">INTRA-PACKET TIME STAMP (BITS 63-48)</td></tr>
<tr><td colspan="3">INTRA-PACKET DATA HEADER (BITS 15-0)</td></tr>
<tr><td colspan="3">:</td></tr>
<tr><td colspan="3">REPEAT FOR EACH MINOR FRAME</td></tr>
<tr><td colspan="3">:</td></tr>
<tr><td colspan="3">PACKET TRAILER</td></tr>
</table>

Figure 10-11. PCM Data – unpacked mode sample packet.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
INTRA-PACKET TIME STAMP (BITS 15-0)	
INTRA-PACKET TIME STAMP (BITS 31-16)	
INTRA-PACKET TIME STAMP (BITS 47-32)	
INTRA-PACKET TIME STAMP (BITS 63-48)	
INTRA-PACKET DATA HEADER (BITS 15-0)	
DATA (BITS 15 – 0)	
DATA (BITS 31 – 16)	
DATA (BITS 47 – 32)	
:	
Y FILLER BITS	
Y FILLER BITS	X DATA BITS
INTRA-PACKET TIME STAMP (BITS 15-0)	
INTRA-PACKET TIME STAMP (BITS 31-16)	
INTRA-PACKET TIME STAMP (BITS 47-32)	
INTRA-PACKET TIME STAMP (BITS 63-48)	
INTRA-PACKET DATA HEADER (BITS 15-0)	
:	
REPEAT FOR EACH MINOR FRAME	
:	
PACKET TRAILER	

Figure 10-12. PCM Data – packed mode sample packet.

e. PCM Data in Throughput Mode. In Throughput Mode, the PCM data are not frame synchronized so the first data bit in the packet can be any bit in the major frame. The resultant PCM packets are as shown in Figure 10–13a . Only Bit 20 of the Channel Specific Data word is set to one (1), indicating Throughput Mode. All other bits of the Channel Specific Data word are undefined and shall be set to zero (0).

msb 15   0   lsb
PACKET HEADER
CHANNEL SPECIFIC DATA (BITS 15-0)
CHANNEL SPECIFIC DATA (BITS 31-16)
DATA (BITS 15 – 0)
DATA (BITS 31 – 16)
DATA (BITS 47 – 32)
:
PACKET TRAILER

Figure 10-13a. PCM Data – Throughput mode sample packet.

f. PCM Data Word Order in 32 Bit Alignment Mode. When recording in 32 Bit Alignment Mode, the resultant data word ordering will differ from 16 Bit Alignment Mode. The serial PCM data stream is shifted into 32 bit words from right to left, with bit 31 on the left, bit 0 on the right, and addresses ascending from top to bottom. Word order is affected depending on the reader's addressing perspective. For example, 16-bit data words when addressed as 32 bit words appear in order when read from left to right, and top to bottom. However, when addressed as 16 bit words each pair of data words will appear swapped. Figure 10–13b and Figure 10–13c depict the anomaly of perspective.

msb 31 16		lsb 15 0		addr
byte 3	byte 2	byte 1	byte 0	
DATA WORD 1		DATA WORD 2		1
DATA WORD 3		DATA WORD 4		2
:				
DATA WORD N-1		DATA WORD N		N/2-1

Figure 10-13b. 32 Bit alignment mode example, 16-bit data words (32 bit word addressing).

msb 15	lsb 0	addr
byte 1	byte 0	
DATA WORD 2		0
DATA WORD 1		1
DATA WORD 4		2
DATA WORD 3		3
:		:
DATA WORD N-1		N

Figure 10-13c. 32 Bit alignment mode example, 16-bit data words (16 bit word addressing).

g. PCM Intra-Packet Header. When recording in Packed or Unpacked mode, all PCM minor frames shall include an Intra-Packet Header containing a 64-bit Intra-Packet Time Stamp and a 16 or 32 bit Intra-Packet Data Header, as indicated by MODE in the Channel Specific Data. This header is inserted immediately before the minor frame sync pattern. Depending on Alignment Mode, the length of the Intra-Packet Header is either 10 or 12 bytes (80 or 96 bits) positioned contiguously, as depicted in Figure 10–14. In 16 Bit Alignment Mode, the Intra-Packet Data Header length is fixed at 2 bytes. 32 Bit Alignment Mode requires a 4 byte Intra-Packet Data Header, and the two most significant bytes are two zero-filled.

msb 31	16	15	12	11	lsb 0
TIME (LSLW)					
TIME (MSLW)					
zero filled		LOCKST		RESERVED	

Figure 10-14. PCM Intra-packet header.

1) Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the PCM minor frame. Not valid for Throughput Mode.

First Long Word Bits 31-0 and Second Long Word Bits 31-0 indicate the following values:

- The 48-bit Relative Time Counter that corresponds to the first data bit of the minor frame with bits 31 to 16 in the second long word zero filled or;
- Absolute Time, if enabled by bit 6 in the Packet Flags (section 10.6.1.1.g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1.g) and to the first data bit of the minor frame.

2) Intra-Packet Data Header

- Reserved. (Bits 11-0) are reserved.
- Lock Status (LOCKST). (Bits 15-12) indicate the lock status of the frame synchronizer for each minor frame.

Bits 15-14: Indicates Minor Frame Status.

00 = Reserved.
01 = Reserved.
10 = Minor Frame Check (after losing Lock).
11 = Minor Frame Lock.

Bits 13-12: Indicates Major Frame Status.

00 = Minor Frames Only.
01 = Reserved.
10 = Major Frame Check (after losing Lock).
11 = Major Frame Lock.

- 32 Bit Alignment (32 Bit Alignment Mode ONLY). Bits 31-16 are zero filled

10.6.3 Time Data Packets.

10.6.3.1 Time Data Packets, Format 0. Reserved.

10.6.3.2 Time Data Packets, Format 1. Time is treated like another data channel. If a time source other than "none" is used, the time packet will be generated at a minimum rate of 1000 milliseconds.

a. IRIG Time Type Formats. The 48-bit relative time counter shall be captured for insertion into the Time Packet Data header IAW IRIG 200 Serial Time Code Formats; On-Time Reference Marker definition.

b. All Non-IRIG Time Type Formats. The 48-bit relative time counter shall be captured for insertion into the Time Packet Data header at an integer multiple of 10 milliseconds.

NOTE

A Time Data Packet shall be the first dynamic data packet at the start of each recording. Only static Computer Generated Data packets may precede the Time Data Packet in the recording.

NOTE

If the Time Data Packet Source is None at least one Time Data Packet is required IAW the previous note.

A packet with time data has the basic structure shown in Figure 10–15. Note that the width of the structure is not related to any number of bits. This drawing is merely to represent relative placement of data in the packet. Time Packets do not have Intra-Packet Headers.

PACKET HEADER :
CHANNEL SPECIFIC DATA
TIME DATA
: PACKET TRAILER

Figure 10-15. General time data packet, format 1.

- Time Packet Channel Specific Data. The Packet Body portion of each Time Data Packet begins with a Channel Specific Data word formatted as shown in Figure 10–16a (next page).

msb 31	12	11 8	7 4	3 lsb 0
RESERVED		DATE	FMT	SRC

- Time Source (SRC). (bits 3-0) indicates the source of the time in the payload of each time packet.

  0x0 = Internal (Time derived from the Clock in the Recorder)
  0x1 = External (Time derived from a Clock not in the Recorder)
  0x2 = Internal from RMM (Internal Time derived from the Clock in the RMM)
  0x3 – 0xE Reserved
  0xF = None

- Time Format (FMT). (bits 7-4) indicate the Time Data Packet format. All bit patterns not used to define a time format type are reserved for future data type growth.

  0x0 = IRIG-B
  0x1 = IRIG-A
  0x2 = IRIG-G
  0x3 = Internal Real-Time Clock
  0x4 = UTC Time from GPS
  0x5 = Native GPS Time
  0x6 thru 0xE = Reserved
  0xF = None (time packet payload invalid)

- Date Format (DATE). (bits 11-8) indicate the Date format. All bit patterns not used to define a date format type are reserved for future growth.

  Bits 11-10: Reserved

  Bit 9: Indicates Date Format

  0 = IRIG day available
  1 = Month and Year available

  Bit 8: Indicates if this is a leap year

  0 = Not a leap year
  1 = Is a leap year

- Reserved. (bits 31-12) are reserved.

Figure 10-16a. Time packet channel specific data format.

c. <u>Time Packet Body</u>. After the Channel Specific Data word, the time data words are inserted in the packet in Binary Coded Decimal (BCD) format as shown in Figure 10–16b and Figure 10–16c.

<table>
<tr><td colspan="8">msb</td><td colspan="8">lsb</td></tr>
<tr><td>15</td><td></td><td></td><td>12</td><td>11</td><td></td><td></td><td>8</td><td>7</td><td></td><td></td><td>4</td><td>3</td><td></td><td></td><td>0</td></tr>
<tr><td>0</td><td colspan="3">TSn</td><td colspan="4">Sn</td><td colspan="4">Hmn</td><td colspan="4">Tmn</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">THn</td><td colspan="4">Hn</td><td>0</td><td colspan="3">TMn</td><td colspan="4">Mn</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="2">HDn</td><td colspan="4">TDn</td><td colspan="4">Dn</td></tr>
</table>

Figure 10-16b. Time data - packet format, day format.

<table>
<tr><td colspan="8">msb</td><td colspan="8">lsb</td></tr>
<tr><td>15</td><td></td><td></td><td>12</td><td>11</td><td></td><td></td><td>8</td><td>7</td><td></td><td></td><td>4</td><td>3</td><td></td><td></td><td>0</td></tr>
<tr><td>0</td><td colspan="3">TSn</td><td colspan="4">Sn</td><td colspan="4">Hmn</td><td colspan="4">Tmn</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">THn</td><td colspan="4">Hn</td><td>0</td><td colspan="3">TMn</td><td colspan="4">Mn</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>TOn</td><td colspan="4">On</td><td colspan="4">TDn</td><td colspan="4">Dn</td></tr>
<tr><td>0</td><td>0</td><td colspan="2">OYn</td><td colspan="4">HYn</td><td colspan="4">TYn</td><td colspan="4">Yn</td></tr>
</table>

Figure 10-16c. Time data - packet format, day, month, and year format

The following units of measure apply to Figure 10–16b and Figure 10–16c.

Tmn	Tens of milliseconds	TDn	Tens of days
Hmn	Hundreds of milliseconds	HDn	Hundreds of Days
Sn	Units of seconds	On	Units of Months
TSn	Tens of Seconds	TOn	Tens of Months
Mn	Units of minutes	Yn	Units of Years
TMn	Tens of minutes	TYn	Tens of Years
Hn	Units of hours	HYn	Hundreds of Years
THn	Tens of Hours	OYn	Thousands of Years
Dn	Units of Days	0	Always zero

10.6.4 MIL-STD-1553.

10.6.4.1 MIL-STD-1553 Bus Data Packets, Format 0. Reserved

10.6.4.2 MIL-STD-1553 Bus Data Packets, Format 1. MIL-STD-1553 BUS data is packetized in Message Mode, where each 1553 bus "transaction" is recorded as a "message." A four-item Intra-Packet Data Header is inserted prior to each transaction. A transaction is a BC-to-RT, RT-to-BC, or RT-to-RT word sequence, starting with the command word and including all data and status words that are part of the transaction, or a mode code word broadcast. Multiple messages may be encoded into the data portion of a single packet.

a. MIL-STD-1553 Packet Channel Specific Data. The Packet Body portion of each MIL-STD–1553 data packet begins with a Channel Specific Data word formatted as shown in Figure 10–17.

msb 31 30	29 24	23 lsb 0
TTB	RESERVED	MSGCOUNT

- Message Count (MSGCOUNT). (Bits 23-0) indicate the binary value of the number of messages included in the packet. An integral number of complete messages will be in each packet.

- Reserved. (Bits 29-24) are reserved.

- Time Tag Bits (TTB). (Bits 31-30) indicate which bit of the MIL-STD-1553 message the Intra-Packet Header time tags.

  00 = Last bit of the last word of the message
  01 = First bit of the first word of the message
  10 = Last bit of the first (command) word of the message
  11 = Reserved

Figure 10-17. MIL-STD-1553 Packet channel specific data format.

b. MIL-STD-1553 Packet Body. A packet with **n** MIL-STD-1553 messages has the basic structure shown in Figure 10–18. Note that the width of the structure is not related to any number of bits. This drawing is merely to represent relative placement of data in the packet.

PACKET HEADER
CHANNEL SPECIFIC DATA
INTRA-PACKET TIME STAMP FOR MESSAGE 1
INTRA-PACKET DATA HEADER FOR MESSAGE 1
MESSAGE 1
INTRA-PACKET TIME STAMP FOR MESSAGE 2
INTRA-PACKET DATA HEADER FOR MESSAGE 2
MESSAGE 2
:
INTRA-PACKET TIME STAMP FOR MESSAGE n
INTRA-PACKET DATA HEADER FOR MESSAGE n
MESSAGE n
PACKET TRAILER

Figure 10-18. MIL-STD-1553 Data packet, format 1.

c. MIL-STD-1553 Intra-Packet Header. After the Channel Specific Data, the MIL-STD-1553 data are inserted into the packet in messages. Each MIL-STD-1553 message is preceded by an Intra-Packet Header consisting of an Intra-Packet Time Stamp and an Intra-Packet Data Header.

1) MIL-STD-1553 Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the MIL-STD-1553 message as follows.

- The 48-bit Relative Time Counter that corresponds to the data bit indicated in the MIL-STD-1553 Channel Specific Data, Time Tag Bits (section 10.6.4.2a) with bits 31 to 16 in the second long word zero filled or;

- The Absolute Time, if enabled by bit 6 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the data bit indicated in the MIL-STD-1553 Channel Specific Data, Time Tag Bits (section 10.6.3.5.3).

2) MIL-STD-1553 Intra-Packet Data Header. The length of the Intra-Packet Data Header is fixed at 6 bytes (48-bits) positioned contiguously, in the following sequence (Figure 10–19)

msb 15	lsb 0
BLOCK STATUS WORD	
GAP TIMES WORD	
LENGTH WORD	

Figure 10-19. MIL-STD-1553 Intra-packet data header.

d. Block Status Word (BSW). (Bits 15-0) contain the Block Status Word for both the message type and whether any 1553 bus protocol errors occurred during the message transfer. The Block Status Word bit definitions are in Figure 10-20.

msb 15-14	13	12	11	10	9	8	7	6	5	4	3	2	1	lsb 0
R	BID	ME	RR	FE	TM	R	R	R	LE	SE	WE	R	R	R

- Reserved. (Bits 15-14) are reserved.

- Bus ID (BID). (Bit 13) indicates the bus ID for the message.
  0 = Message was from Channel A
  1 = Message was from Channel B

- Message Error (ME). (Bit 12) indicates a message error was encountered.
  0 = No message error
  1 = Message error

- RT to RT Transfer (RR). (Bit 11) indicates a RT to RT transfer; message begins with two command words.
  0 = No RT to RT transfer
  1 = RT to RT transfer

- Format Error (FE). (Bit 10) indicates a frame error
  0 = No format error
  1 = Format error

- Response Time Out. (Bit 9) indicates a response time out occurred.
  0 = No response time out
  1 = Response time out

- Reserved. (Bits 8-6) are reserved.

- Word Count Error (LE). (Bit 5) indicates a word count error occurred.
  0 = No word count error
  1 = Word count error

- Sync Type Error (SE). (Bit 4) indicates an incorrect sync type occurred.
  0 = No sync type error
  1 = Sync type error

- Invalid Word Error (WE). (Bit 3) indicates an invalid word error occurred.
  0 = No invalid word error
  1 = Invalid word error

- Reserved. (Bits 2-0) are reserved.

Figure 10-20. Block status word format.

NOTE

Response Time Word (Gap). The Response Time Words indicate remote terminal response times as defined by MIL-STD 1553. The resolution of the response time shall be in tenths of microseconds. A maximum of two Response Time Words can exist. Remote Terminal to Remote Terminal type messages shall have two Response Time Words if both terminals respond, all other messages will have one Response Time Word, or none for broadcast type messages or messages with no Remote Terminal response.

The Gap Times Word indicates the number of tenths of microseconds in length of the internal gaps within a single transaction. For most messages, only GAP1 is meaningful. It measures the time between the command or data word and the first (and only) status word in the message. For RT-to-RT messages, GAP2 measures the time between the last data word and the second status word. The Gap Times Word bit definitions are as shown in Figure 10–21.

msb	lsb
15 8	7 0
GAP2	GAP1

Figure 10-21. Response time word format.

NOTE

Gap measurements shall be made IAW MIL-STD-1553 response time measurements from the mid-bit zero crossing of the parity bit of the last word to the mid-zero crossing of the sync of the status word.

e. Length Word. The Length of the message is the total number of bytes in the message. A message consists of command words, data words, and status words.

f. <u>Packet Format</u>. Unless an error occurred, as indicated by one of the error flags in the Block Status Word, the first word following the Length should always be a command word. The resultant packets have the format shown in Figure 10–22.

msb	lsb
15	0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 15-0)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 63-48)	
INTRA-PACKET DATA HEADER FOR MSG 1 (BITS 15-0)	
INTRA-PACKET DATA HEADER FOR MSG 1 (BITS 31-16)	
INTRA-PACKET DATA HEADER FOR MSG 1 (BITS 47-32)	
COMMAND WORD	
COMMAND, STATUS, OR DATA WORD	
DATA OR STATUS WORD	
:	
DATA OR STATUS WORD	
INTRA-PACKET TIME STAMP FOR MSG 2 (BITS 15-0)	
INTRA-PACKET TIME STAMP FOR MSG 2 (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR MSG 2 (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR MSG 2 (BITS 63-48)	
INTRA-PACKET DATA HEADER FOR MSG 2 (BITS 15-0)	
INTRA-PACKET DATA HEADER FOR MSG 2 (BITS 31-16)	
INTRA-PACKET DATA HEADER FOR MSG 2 (BITS 47-32)	
COMMAND WORD	
COMMAND, STATUS, OR DATA WORD	
DATA OR STATUS WORD	
:	
DATA OR STATUS WORD	
:	
INTRA-PACKET TIME STAMP FOR MSG N (BITS 15-0)	
INTRA-PACKET TIME STAMP FOR MSG N (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR MSG N (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR MSG N (BITS 63-48)	
INTRA-PACKET DATA HEADER FOR MSG N (BITS 15-0)	
INTRA-PACKET DATA HEADER FOR MSG N (BITS 31-16)	
INTRA-PACKET DATA HEADER FOR MSG N (BITS 47-32)	
COMMAND WORD	
COMMAND OR DATA, WORD	
DATA OR STATUS WORD	
:	
DATA OR STATUS WORD	
PACKET TRAILER	

Figure 10-22. MIL-STD-1553 data packet, format 1.

10.6.5 Analog Data Packets

10.6.5.1 Analog Data Packets, Format 0. Reserved

10.6.5.2 Analog Data Packets, Format 1. The generic packet structure for analog data is illustrated in Figure 10–23.

An Analog Data Packet will contain a Channel Specific Data word for each subchannel of analog data sampled within that packet. This will be followed by at least one complete sampling schedule of data.

A sampling schedule is defined as a sampling sequence in which each subchannel, described by a Channel Specific Data word, is sampled at least once. In many cases, due to simultaneous sampling rules and varied sampling rates, a particular subchannel will be sampled more than once during a sampling schedule. In addition, multiple complete sampling schedules may be included in a single packet. For these reasons, the number of Channel Specific Data words will usually be less than the number of samples.

Figure 10–23 depicts the generic packet data structure for M data subchannels and a single sampling schedule that has a length N. Note that the width of the structure is not related to any number of bits and is merely to represent relative placement of words within the packet.

PACKET HEADER :
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2
*
*
*
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M
SAMPLE 1
SAMPLE 2
• • •
SAMPLE N
: PACKET TRAILER

Figure 10-23. Generic analog data packet, format 1.

NOTE

The Packet Header Time in an Analog Data packet shall correspond to the first data sample in the packet. There are no Intra-Packet Headers in Analog Data Packets.

a. Analog Packet Channel Specific Data. The Packet Body portion of each Analog packet begins with the Channel Specific Data word(s). Each subchannel that is sampled within the packet sampling schedule must have a Channel Specific Data word within the packet. If subchannels are sampled at the same sampling rate, only one Channel Specific Data word is required. Bit 28 of the channel specific data word shall be used to indicate same sampling data rate for subchannels.

Channel Specific Data words for Analog Data Packets are formatted as shown in Figure 10-24.

msb						lsb
31 29	28	27 24	23 16	15 8	7 2	1 0
RESERVED	SAME	FACTOR	TOTCHAN	SUBCHAN	LENGTH	MODE

- Mode. (Bits 1-0) indicate alignment and packing modes of the analog data Bit 0 is the packing bit, Bit 1 is the alignment bit. When TOTCHAN, defined, is more than 1, the Mode must be the same for all subchannels in a single packet.

  00 = data is packed
  01 = data is unpacked, lsb padded
  10 = Reserved for future definition
  11 = data is unpacked, msb padded

NOTE: Samples less than 8 bits go into a 16 bit word boundary.

- Length. (Bits 7-2) indicate a binary value representing the number of bits in the Analog to Digital converter (A/D).

  000000 = sixty-four bit samples
  000001 = one bit samples
  :
  001000 = eight bit samples
  :
  001100 = twelve bit samples
  :

- Subchan. (Bits 15-8) indicate a binary value representing the number of the analog subchannel.

When an Analog Packet contains data from more than one subchannel and the Channel Specific Data Words are not the same for all channels (see field Same, Bit 28), the Channel Specific Data words must be inserted into the packet in ascending subchannel number as identified by this Subchan field. The Subchan values in these Channel Specific Data words need not be contiguous (see Totchan), but they must be in ascending decimal numerical order with the

(*continued on next page*)

(Figure 10-24 continued)

exception that subchannel 0 (256) is last. If the "Same" bit is set, Subchan field shall be set to zero.

0x01 = Subchannel 1
0x02 = Subchannel 2
:
0x00 = Subchannel 256

- Totchan. (Bits 23-16) indicate the total number of analog subchannels in the packet (and the number of Channel Specific Data words in the packet.)

This Totchan field must be the same value in all Channel Specific Data words in a single packet. The Totchan value may be less than the largest Subchan value. This can happen when a multi-channel analog input device has some of its subchannels disabled (turned off) for a specific recording. For example, if an analog input device has eight subchannels and not all eight are active, an analog data packet may have three subchannels (Totchan=3) numbered 4, 7, and 8 (enabled Subchan = 4, 7, 8). The number of subchannels (Totchan) and the subchannel number for each active subchannel (Subchan) in a packet are identified in the accompanying TMATS (Computer Generated Data, Format 1) packet.

0x00 = 256 Subchannels
0x01 = 1 Subchannel
0x02 = 2 Subchannels
:

- Factor. (Bits 27-24) are the exponent of the power of 2 sampling rate factor denominator for the corresponding subchannel (described in 10.6.4.2.2) in the range 0 to 15. (The sampling rate factor numerator is always 1.)

0x0 = sampling rate factor denominator $2^0 = 1$ => factor = 1/1
0x1 = sampling rate factor denominator $2^1 = 2$ => factor = ½
0x2 = sampling rate factor denominator $2^2 = 4$ => factor = ¼
:
0xF = sampling rate factor denominator $2^{15} = 32768$ => factor = 1/32768

- Same. (Bit 28) specifies if this Channel Specific Data Word applies for all the channels included in the packet, or if each channel has its own Channel Specific Data Word.

0 = each analog channel has its own Channel Specific Data Word
1 = The Channel Specific Data Word is valid for all analog channels stored in this packet

- Reserved. (Bits 31-29) are reserved.

Figure 10-24. Analog data packet format.

b. Analog Samples. To preserve timing relationships and allow for accurate reconstruction of the data, a simultaneous sampling scheme shall be employed. The highest sampling rate required shall define the primary simultaneous sampling rate within the packet. The primary simultaneous sampling rate is identified in the Telemetry Attributes Transfer Standard (TMATS) file describing the attributes of the analog data packet. The rate at which the other subchannels are sampled is then defined by the sampling factor (1, ½, ¼, 1/8, 1/16, … 1/32768) for each subchannel. As an example, a sampling factor of ¼ would yield that subchannel being sampled at one-fourth the primary simultaneous sampling rate and a sampling factor of 1 would yield that subchannel being sampled at the primary simultaneous sampling rate.

Directly following the Channel Specific Data word(s), at least one complete sampling schedule shall be inserted in the packet. The samples, within the sampling sequence, may be inserted either unpacked, MSB Packed, or LSB Packed as described in section 10.6.5.2.b(1) and 10.6.5.2.b(2). In either case, one or more subchannels may be included in a single packet. When multiple subchannels are encapsulated into a single packet, the subchannel with the highest sampling rate requirement defines the primary simultaneous sampling rate. The rate at which the other subchannels are sampled is defined by the sampling factor (contained within the Channel Specific Data words). Sampling factors are defined as:

$$\left(\frac{1}{2^{K}}\right) * X \quad ;K = 0, 1, 2, 3, 4, 5, \ldots$$

of the Primary Simultaneous Sampling Rate, *X*.

The subchannels are then sampled and ordered such that:

- the highest sample rate $1 * X$ subchannel(s) appear in every simultaneous sample,
- the $\left(\frac{1}{2}\right) * X$ in every 2nd simultaneous sample,
- the $\left(\frac{1}{4}\right) * X$ in every 4th simultaneous sample,

… and so on until all the subchannels are sampled, resulting in a complete sampling schedule of all subchannels described by the Channel Specific Data words. In doing so, the total number of simultaneous samples (not the total number of samples) will equal the denominator of the smallest sampling factor and all subchannels are sampled in the last simultaneous sample.

For example, a packet with six subchannels with Sampling Factors ½, 1/8, 1, ½, 1, and 1/8 respectively will yield a sampling sequence within the data packet as follows:

Simultaneous Sample 1: Subchannel 3
Simultaneous Sample 1: Subchannel 5

Simultaneous Sample 2: Subchannel 1
Simultaneous Sample 2: Subchannel 3
Simultaneous Sample 2: Subchannel 4
Simultaneous Sample 2: Subchannel 5

Simultaneous Sample 3: Subchannel 3
Simultaneous Sample 3: Subchannel 5

Simultaneous Sample 4: Subchannel 1
Simultaneous Sample 4: Subchannel 3
Simultaneous Sample 4: Subchannel 4
Simultaneous Sample 4: Subchannel 5

Simultaneous Sample 5: Subchannel 3
Simultaneous Sample 5: Subchannel 5

Simultaneous Sample 6: Subchannel 1
Simultaneous Sample 6: Subchannel 3
Simultaneous Sample 6: Subchannel 4
Simultaneous Sample 6: Subchannel 5

Simultaneous Sample 7: Subchannel 3
Simultaneous Sample 7: Subchannel 5

Simultaneous Sample 8: Subchannel 1
Simultaneous Sample 8: Subchannel 2
Simultaneous Sample 8: Subchannel 3
Simultaneous Sample 8: Subchannel 4
Simultaneous Sample 8: Subchannel 5
Simultaneous Sample 8: Subchannel 6

Notice that the denominator of the smallest sampling factor defines the number of simultaneous samples within the packet (in this example 8). However, the total number of samples within the sampling schedule does not have to equal number of simultaneous samples (in this example 26). Also notice that all subchannels are sampled during the last Simultaneous Sample. The order of the subchannel samples in each simultaneous sample is ascending by subchannel number.

Any number of complete sampling schedules may be placed within a packet so that the maximum packet length is not exceeded. The TMATS file identifies the number of samples contained within each packet.

1) Unpacked Mode. In Unpacked Mode, packing is disabled and each sample is padded with the number of bits necessary to align each word with the next 16-bit boundary in the packet. Four (4) pad bits are added to 12 bit words, eight (8) pad bits are added to 8 bit words, etc. All pad bits shall be zero (0).

To illustrate msb padding, given M analog subchannels mapping into N samples for the special case of all samples having bit lengths of 12 bits, the resultant Analog packets with msb padding have the form shown in Figure 10–25.

msb 15	lsb 0
PACKET HEADER **:**	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 31-16)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 31-16)	
:	
:	
:	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M (BITS 31-16)	
4-PAD BITS	SAMPLE 1, 12-DATA BITS
4-PAD BITS	SAMPLE 2, 12- DATA BITS
4-PAD BITS	SAMPLE 3, 12- DATA BITS
:	
4-PAD BITS	SAMPLE N, 12- DATA BITS
: PACKET TRAILER	

Figure 10-25. Analog data packet - unpacked mode, msb padding.

To illustrate LSB Packing, given M analog subchannels mapping into N samples for the special case of all samples having bit lengths of 12 bits, the resultant Analog packets with LSB padding have the form shown in Figure 10–26.

msb 15	lsb 0
PACKET HEADER **:**	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 31-16)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 31-16)	
:	
:	
:	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL m (BITS 31-16)	
:	
SAMPLE 1, 12-DATA BITS	4-PAD BITS
SAMPLE 2, 12- DATA BITS	4-PAD BITS
SAMPLE 3, 12- DATA BITS	4-PAD BITS
SAMPLE N, 12- DATA BITS	4-PAD BITS
: PACKET TRAILER	

Figure 10-26. Analog data packet - unpacked mode, lsb padding.

2) Packed Mode. In Packed Mode, packing is enabled and padding is not added to each data word. However, if the number of bits in the packet are not an integer multiple of 16, then Y filler bits will be used to msb fill the last data word to force alignment on a 16-bit boundary. Y is sixteen (16) minus the integer remainder of L, the total number of data bits in the packet, divided by 16 and is mathematically expressed as:

$$Y = 16 - (\text{MODULUS}\{L,16\}).$$

To illustrate msb padding, given M Analog subchannels mapping into N samples for the special case of all samples having bit lengths of 12 bits, the resultant Analog packets with padding bits at the end of the N^{th} sample have the form shown in Figure 10–27.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 1 (BITS 31-16)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL 2 (BITS 31-16)	
:	
:	
:	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M (BITS 15-0)	
CHANNEL SPECIFIC DATA WORD, SUBCHANNEL M (BITS 31-16)	
SAMPLE 2 (BITS 3-0)	SAMPLE 1 (BITS 11-0)
SAMPLE 3 (BITS 7-0)	SAMPLE 2 (BITS 11-4)
:	:
:	:
:	:
Y PADDING BITS	SAMPLE N (BITS 11-0)
:	
PACKET TRAILER	

Figure 10-27. Analog data packet - packed mode packet.

10.6.6 Discrete Data Packets.

10.6.6.1 Discrete Data Packets, Format 0. Reserved.

10.6.6.2 Discrete Data Packets, Format 1. A packet with Discrete data has the basic structure shown in Figure 10–28. Note that the width of the structure is not related to any number of bits. This drawing is merely to represent relative placement of data in the packet. One to 32 discrete states may be recorded for each event.

PACKET HEADER
CHANNEL SPECIFIC DATA
INTRA-PACKET HEADER FOR EVENT 1
EVENT 1 STATES
INTRA-PACKET HEADER FOR EVENT 2
EVENT 2 STATES
:
INTRA-PACKET HEADER FOR EVENT N
EVENT N STATES
PACKET TRAILER

Figure 10-28. General discrete data packet, format 1.

a. Discrete Packet Channel Specific Data Word. The Packet Body portion of each Discrete packet begins with the Channel Specific Data Word, which is formatted as shown in Figure 10–29.

msb 31 ... 8	7 ... 3	2 ... lsb 0
RESERVED	LENGTH	MODE

- Bits 2-0: Mode: indicate the mode of accessing the discrete data.

  Bit 0: indicates the Record State.
  0 = discrete data is recorded when the state changes
  1 = discrete data is recorded on a time interval basis

  Bit 1: indicates the alignment of the data.
  0 = lsb
  1 = msb

  Bit 2: reserved.

- (Bits 7-3) Length - indicates a binary value representing the number of bits in the event.
- (Bits 31-8) Reserved.

Figure 10-29. Discrete packet channel data word format.

b. Discrete Data. After the Channel Specific Data, the Discrete data (Figure 10–30) is inserted in the packet. Discrete data is described as Events. Each Event includes the Event State for each discrete input and the corresponding Intra packet Time. The Event State is a 32-bit word that provides the logical state of each discrete input.

msb 31	30		1	lsb 0
D31	D30		D1	D0

- Discrete Event Bits. (Bits 31-0) indicate the states of the discrete event bits.

  Bit 0: indicates Discrete 0 (DO) State.

  0 = discrete 0 is at state 0
  1 = discrete 0 is at state 1

  Bit 1: indicates Discrete 1 (D1) State.

  0 = discrete 1 is at state 0
  1 = discrete 1 is at state 1
  :
  Bit 31: indicates Discrete 31 (D31) State.

  0 = discrete 31 is at state 0
  1 = discrete 31 is at state 1

Figure 10-30. Discrete data format.

c. Discrete Event Intra-Packet Header. All discrete events shall include an Intra-Packet Header consisting of an Intra-Packet Time Stamp only, which is inserted immediately before the discrete event. The length of the Intra-Packet Header is fixed at 8 bytes (64-bits) positioned contiguously, in the following sequence as shown in Figure 10–31.

msb 31	lsb 0
TIME (LSLW)	
TIME (MSLW)	

Figure 10-31. Discrete event intra-packet data header.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the discrete event.

First Long Word Bits 31-0 and Second Long Word Bits 31-0 indicate the following values:

- o The Relative Time Counter that corresponds to the first data bit of the discrete event with bits 31 to 16 in the second long word zero filled or;
- o Time, if enabled by bit 7 in the Packet Flags (paragraph 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (paragraph 10.6.1.1g) and to the first data bit of the discrete event.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR EVENT 1 (BITS 15–0)	
INTRA-PACKET TIME STAMP FOR EVENT 1 (BITS 31–16)	
INTRA-PACKET TIME STAMP FOR EVENT 1 (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR EVENT 1 (BITS 63-48)	
STATES FOR EVENT 1 (BITS 15–0)	
STATES FOR EVENT 1 (BITS 31–16)	
:	
INTRA-PACKET TIME STAMP FOR EVENT n (BITS 15–0)	
INTRA-PACKET TIME STAMP FOR EVENT n (BITS 31–16)	
INTRA-PACKET TIME STAMP FOR EVENT n (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR EVENT n (BITS 63-48)	
STATES FOR EVENT n (BITS 15–0)	
STATES FOR EVENT n (BITS 31–16)	
PACKET TRAILER	

Figure 10-32. Discrete data - packet format.

10.6.7 Computer Generated Data Packets. Packets with Computer Generated Data have the basic structure shown in Figure 10–33. Formats 0, 1, 2 and 3 are used to add information packets to recorded data. This information contains annotation data, setup records, events or index information for the data that has been recorded. The width of the structure is not related to any number of bits. This drawing is merely to represent relative placement of data in the packet.

NOTE	Computer Generated Data is defined as non-external data or data generated within the recorder. After the Channel Specific Data Word, the Computer Generated Data is inserted in the packet. The organization and content of the Computer Generated Data is determined by the specific Format type.

PACKET HEADER
CHANNEL SPECIFIC DATA
INFORMATION PACKET CONTENTS
PACKET TRAILER

Figure 10-33. General computer generated data packet format.

10.6.7.1 Computer Generated Data Packets Format 0, User Defined. Format 0 enables the insertion of user-defined Computer Generated Data. ***Data can not be placed in this packet if the data type is already defined within this standard nor can data be inserted in this packet if it is generated from an external input to the recorder.***

Note: (reference Computer Generated Packets Format 0 Channel Specific Data Word). The Packet Body portion of each Format 0 Packet begins with the Channel Specific Data word, which is formatted as shown in Figure 10–34a.

msb 31	lsb 0
RESERVED	

Figure 10-34a. Computer generated format 0 specific data word format.

- Reserved. (Bits 31-0) are reserved.

10.6.7.2 Computer Generated Data Packets Format 1, Setup Records. Format 1 defines a setup record that describes the hardware, software, and data channel configuration used to produce the other data packets in the file. The organization and content of a Format 1 Setup Record is IAW with IRIG 106 Chapter 9 TMATS standard. It is mandatory for a TMATS record to be utilized to configure the recorder. A Format 1 Computer Generated Data Packet containing the TMATS record utilized to configure the recorder shall be the first packet in each data file.

Format 1 – Channel Specific Data Word. The Packet Body portion of each Format 1 Packet begins with the Channel Specific Data word, which is formatted as shown in Figure 10-34b.

msb 31	lsb 0
RESERVED	

Figure 10-34b. Computer generated format 1 channel specific data word format.

- Reserved. (Bits 31-0) are reserved.

10.6.7.3 Computer Generated Data Packets Format 2, Recording Event. Format 2 defines a recording event packet that contains the occurrence and information of one or more individual events that have been defined within the Format 1 Setup Record IAW "Recording Events" attribute. If the Recording Events information is larger than the maximum packet size of 512K bytes the Recording Events information may be contained in multiple packets using the Major Packet Header Packet Sequence Counter.

Events associated with the .EVENT command defined in section 10.7.8 can only be directly accessed from the recorder itself and are not contained within the recording session data. This does not preclude defining an event driven by the .EVENT command to also be defined within the Recording Event setup record information and placed in the appropriate event entry within an event packet. The .EVENT recorder command and the Recording Event packets will not be directly linked in this standard and any linking between the two will be an implementation of this standard within a recorder.

Note: It is not the intent for the event packets within the data to be directly coupled with recorder events per the .EVENT command in section 10.7.8.

a. Event Packet Location. Recording Event packets may be placed at any location within the recording after the first Time Data packet and before the first Root Index Packet. This may be at the time each event occurs, after multiple events have occurred or at an interval of time or packets. The complete event log of a recording (defined in Event Period of Capture section 10.6.7.3c) is constituted by the contents of all event packets in a recording concatenated in order of which the event(s) occurred in time.

Index Packets WILL BE ENABLED if Recording Event Packets are ENABLED.

b. Channel Specific Data Word. The Packet Body portion of each Format 2 Packet begins with the Channel Specific Data word, which is formatted as shown in Figure 10–34c.

msb 31 12	11 lsb 0
RESERVED	REEC

- Recording Event Entry Count (REEC). (Bits 11-0) are an unsigned binary that identifies the count of recording event entries included in the packet.
- Reserved. (Bits 31-12) are reserved.

Figure 10-34c. Computer generated format 2 channel specific data word.

c. Event Period of Capture. The period of capture (Figure 10–34d) so that Recording Events are relevant to a recording session is described as events occurring from the time a .RECORD command (section 10.7.8) is issued if it is the first recording session until the time a .STOP command (section 10.7.8) is issued. If there is a previous recording session the period of capture is described as events that occur from the previous recording's .STOP command until the .STOP command of the current recording session. This insures that any events that occurred in between recording sessions will be captured and contain special indicators that the event occurred in between .STOP and .RECORD commands.

Priority conditions and event limit counts of the Event Period of Capture are defined in the setup record attributes for each defined event. The ability to put finite limits on events during periods of capture precludes overflowing buffers or media capacities. These priority conditions and event limit counts are as follows:

- Priority 1: Defined event will always be captured during and in between recordings.
- Priority 2: Defined event will always be captured during recordings and up to a limit count in between recordings.
- Priority 3: Defined event will always be captured during recordings and not captured in between recordings.
- Priority 4: Defined event will be captured up to a limit count during recordings and in between recordings.
- Priority 5: Defined event will be captured up to a limit count for each defined event during recordings and not captured in between recordings.

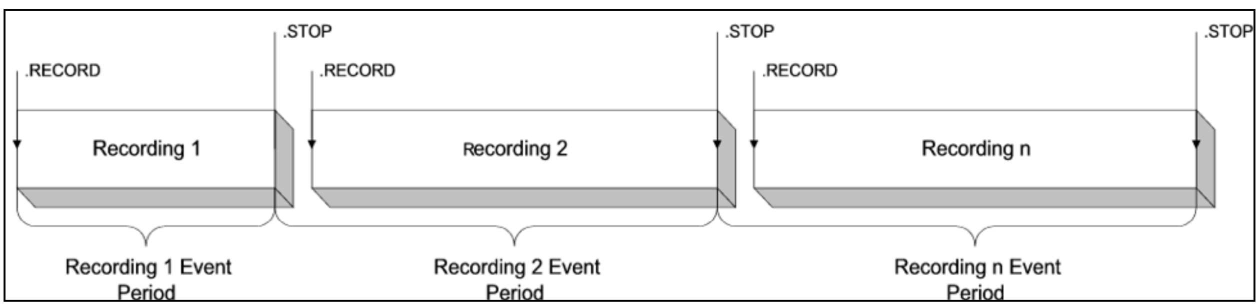


Figure 10-34d. Events recording period.

NOTE	If Event Type is MEASUREMENT DISCRETE or MEASUREMENT LIMIT the trigger measurement must be fully described using the appropriate Setup Record attributes for PCM, Bus, Analog, or Discrete channels. The trigger measurement source and name will also be identified in the Event Definition.

d. Recording Event Intra-Packet Time Stamp. (8 Bytes) indicates the time tag of the Recording Event Entry as follows:

- The 48-bit Relative Time Counter that corresponds to the Event Entry with bits 31 to 16 in the second long word zero filled. For Event Types that are MEASUREMENT DISCRETE or MEASUREMENT LIMIT the time tag will correspond to the data packet timing mechanism containing the trigger measurement i.e. the Packet Header Relative Time Counter value, or if enabled the Intra-Packet Time Stamp which ever provide the highest level of accuracy of when the event occurred or;

- The Absolute Time, if enabled by bit 6 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the Event Entry. For Event Types that are MEASUREMENT DISCRETE or MEASUREMENT LIMIT the time tag will correspond to the data packet timing mechanism containing the trigger measurement i.e. the Packet Secondary Header if enabled and using an absolute time value, or if enabled and using an absolute time value the Intra-Packet Time Stamp which ever provide the highest level of accuracy of when the event occurred.

e. Recording Event Intra-Packet Header. (8 Bytes) contains the absolute time of the event occurrence. The time source and format shall be derived from the Time Data Packet, Format 1. Depending on the Time Data packet time type un-used high-order bits will be zero filled.

msb **31**	**lsb** **0**
INTRA-PACKET TIME STAMP (LSLW)	
INTRA-PACKET TIME STAMP (MSLW)	
INTRA-PACKET DATA HEADER (LSLW)	
INTRA-PACKET DATA HEADER (MSLW)	

Figure 10-34e Recording event intra-packet header.

f. Event Packet Entry Format. (Figure 10–34f and Figure 10–34g).

g. Event Number. (Bits 11-0) An unsigned binary that identifies 4096 individual events types defined in the corresponding setup record Recording Event Number.

h. Event Count. (Bits 27-12) An unsigned binary that identifies the count of up to 65,535 occurrences of an individually defined event (as defined by Event Number in section 10.6.7.3g in the preceding paragraph). Event counts shall begin at 0x00 for the first occurrence of an individual event type (identifies by the event number). The event count can roll over to 0x00 and begin to count up again.

i. Event Occurrence (EO). (Bit 28) Indicates Event Occurrence State.

0 = Indicates the event occurred after the .STOP command and before the .RECORD command.

1 = Indicates the event occurred during the .RECORD command and the .STOP command.

j. Reserved. (Bits 31-29) Reserved for future growth and shall be zero filled.

PACKET HEADER
(Optional) PACKET SECONDARY HEADER
CHANNEL SPECIFIC DATA
INTRA-PACKET TIME STAMP FOR EVENT 1
INTRA-PACKET HEADER FOR EVENT 1
RECORDING EVENT 1
INTRA-PACKET TIME STAMP FOR EVENT 2
INTRA-PACKET HEADER FOR EVENT 2
RECORDING EVENT 2
:
INTRA-PACKET TIME STAMP FOR EVENT n
INTRA-PACKET HEADER FOR EVENT n
RECORDING EVENT n

Figure 10-34f. General recording event packet.

Msb **31 29**	**28**	**27 12**	**11** **lsb** **0**
RESERVED	EO	EVENT COUNT	NUMBER

Figure 10-34g. Recording event entry layout.

10.6.7.4 Computer Generated Data Packets Format 3, Recording Index. Defines an index packet for an individual recording session file used for direct access into the recording file. Recording Index packets will be enabled when Recording Event packets are enabled. There are two types of index packets:

NOTE Recording Index packets WILL BE ENABLED when Recording Event packets are enabled.

- Root Index Packets which contain zero based byte offset entries into the recording session file which are the beginning of Node Index packets. The last entry will be an offset to the beginning of the previous root index packet if there are more than one Root Index packets, or to the beginning of the Root Index packet itself, if this root index packet is the first root index packet of the recording session. Root Index packets will not contain Filler in the Packet Trailer and will contain a 32 Bit Data Checksum in the Packet Trailer.

- Node Index Packets which contain Node Items structures containing information about the location of data packets throughout the recording session.

NOTE Root Index packets will not contain Filler in the Packet Trailer and will contain a 32 Bit Data Checksum in the Packet Trailer

NOTE At a minimum, an index entry WILL EXIST for each Time Data Packet in the recording.

NOTE At a minimum, an index entry WILL EXIST for each Recording Event Packet in the recording.

NOTE If the Recording Index Type uses a Count rather than Time, the Time Data Packets and Computer Generated Data Packets are not included in the count interval.

NOTE Each recording session file must have at a minimum one (1) Root Index Type.

a. Recording Index Packet Location. If enabled the Root Index Type packet (Figure 10–34h) will be the last packet(s) in any recording session file. More than one (1) Root Index Type packet may be created once the first Root Index Type packet exceeds the packet size limitation defined in Table 10–5B. When more than one (1) Root Index Type packet is created the Packet Header sequence counter will be used starting from 0x00 to 0xFF and then roll over to 0x00. Node Index Types may be placed at any location within the recording after the first Time Data packet and before the Root Index packet. This may be at an interval of time or packets.

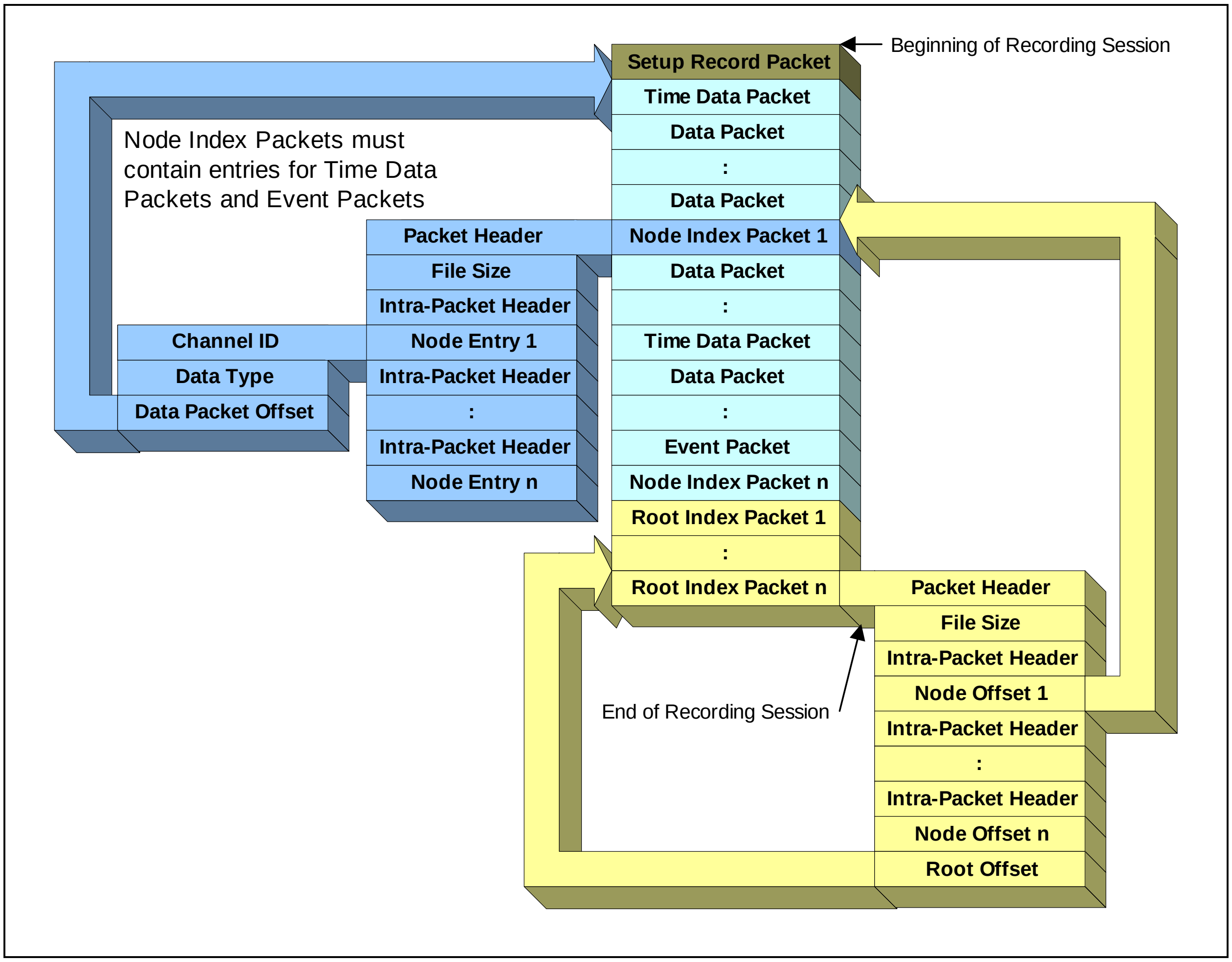


Figure 10-34h. Format showing root index packet.

b. Channel Specific Data Word. The Packet Body portion of each Format 2 Packet begins with the Channel Specific Data word, which is formatted as shown in Figure 10–34i.

Msb 31	30	29	16	15	lsb 0
IT	FSP	RESERVED		INDEX ENTRY COUNT	

- Index Entry Count. (Bits 15-0) indicate the unsigned binary value of the number of index entries included in the packet. An integral number of complete index entries will be in each packet.

- Reserved. (Bits 29-16) are reserved.

- File Size Present (FSP). (Bit 30) indicates if the file size at the time the index packet was created is present.
  0 = File Size Not Present
  1 = File Size Present

- Index Type (IT). (Bit 31) indicates the type of index packet.
  0 = Root Index
  1 = Node Index

Figure 10-34i. Channel specific data word format.

c. Recording Index Intra-Packet Time Stamp. (8 Bytes) indicates the time tag of the Recording Index Entry as follows:

- The 48-bit Relative Time Counter that corresponds to the Index Entry with bits 31 to 16 in the second long word zero filled. For Node Index Packets this corresponds to the first bit in the packet body of the packet associated with the Node Index Item or;

- The Absolute Time, if enabled by bit 6 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the Index Entry. For Node Index Packets this corresponds to the first bit in the packet body of the packet associated with the Node Index Item.

d. Recording Index Intra-Packet Header. (8 Bytes) contains the absolute time of the Index Entry. The time source and format shall be derived from the Time Data Packet, Format 1. Depending on the Time Data packet time type un-used high-order bits will be zero filled.

msb 31	lsb 0
INTRA-PACKET TIME STAMP (LSLW)	
INTRA-PACKET TIME STAMP (MSLW)	
INTRA-PACKET DATA HEADER (LSLW)	
INTRA-PACKET DATA HEADER (MSLW)	

Figure 10-34j. Recording index intra-packet header.

e. Root Index Packet Entry Format. A recording Root Index Packet contains a Node Index Offset entry or entries using the format shown in Figure 10–34k and Figure 10–34l.

- (Optional) Root Index File Size. (8 Bytes) An unsigned binary that identifies the current size of the file when the Root Index Packet was created. This value should be the same as the Root Index Offset. The file size is required when a recording is split across multiple media, individual or multiple channels are split from the original recording file or time slices are extracted from the original recording. In all cases the original recording file size will allow recalculation and or replacement of the index offsets when required.

- Node Index Offset. (8 Bytes) An unsigned binary that identifies the zero based byte offset from the beginning of the recording session file to the point in the file at which the Node Index Packet ***Sync Pattern (0xEB25)*** begins.

- Root Index Offset. (8 Bytes) An unsigned binary that identifies the zero based byte offset from the beginning of the recording session file to the point in the file at which the previous Root Index Packet begins if there are more than one (1) Root Index Packets or to itself if it is the first or only Root Index Packet.

PACKET HEADER
(Optional) PACKET SECONDARY HEADER
CHANNEL SPECIFIC DATA
(Optional) ROOT INDEX FILE SIZE
INTRA-PACKET TIME STAMP FOR NODE INDEX 1
INTRA-PACKET DATA HEADER FOR NODE INDEX 1
NODE INDEX OFFSET 1
:
INTRA-PACKET TIME STAMP FOR NODE INDEX n
INTRA-PACKET DATA HEADER FOR NODE INDEX n
NODE INDEX OFFSET n
INTRA-PACKET TIME STAMP FOR ROOT INDEX
INTRA-PACKET DATA HEADER FOR ROOT INDEX
ROOT INDEX OFFSET

Figure 10-34k. General recording root index packet.

msb 31	lsb 0
(Optional) FILE SIZE (LSLW)	
(Optional) FILE SIZE (MSLW)	
INTRA-PACKET TIME STAMP FOR NODE INDEX 1 (LSLW)	
INTRA-PACKET TIME STAMP FOR NODE INDEX 1 (MSLW)	
INTRA-PACKET DATA HEADER FOR NODE INDEX 1 (LSLW)	
INTRA-PACKET DATA HEADER FOR NODE INDEX 1 (MSLW)	
NODE INDEX OFFSET 1 (LSLW)	
NODE INDEX OFFSET 1 (MSLW)	
:	
INTRA-PACKET TIME STAMP FOR NODE INDEX n (LSLW)	
INTRA-PACKET TIME STAMP FOR NODE INDEX n (MSLW)	
INTRA-PACKET DATA HEADER FOR NODE INDEX n (LSLW)	
INTRA-PACKET DATA HEADER FOR NODE INDEX n (MSLW)	
NODE INDEX OFFSET n (MSLW)	
NODE INDEX OFFSET n (MSLW)	
INTRA-PACKET TIME STAMP FOR ROOT INDEX (LSLW)	
INTRA-PACKET TIME STAMP FOR ROOT INDEX (MSLW)	
INTRA-PACKET DATA HEADER FOR ROOT INDEX (LSLW)	
INTRA-PACKET DATA HEADER FOR ROOT INDEX (MSLW)	
ROOT INDEX OFFSET (LSLW)	
ROOT INDEX OFFSET (MSLW)	

Figure 10-34l. Recording root index entry layout.

f. Node Index Packet Entry Format. A recording Node Index Packet contains an index entry or entries using the format shown in Figure 10–34l and Figure 10–34m.

- (Optional) Node Index File Size. (8 Bytes) An unsigned binary that identifies the current size of the file when the Node Index Packet was created. This value should be the same as the Node Index Offset. The file size is required when a recording is split across multiple media, individual or multiple channels are split from the original recording file or time slices are extracted from the original recording. In all cases the original recording file size will allow recalculation and or replacement of the index offsets when required.
- Channel ID. (2 Bytes) An unsigned binary that identifies a value representing the Packet Channel ID for the data packet associated with this Node Index Item.
- Data Type. (1 Byte) An unsigned binary that identifies a value representing the type and format of the data packet associated with this Node Index Item.
- Data Packet Offset. (8 Bytes) An unsigned binary that identifies the zero based byte offset from the beginning of the recording session file to the point in the file at which the Data Packet begins for this Node Index Packet item.

PACKET HEADER
(Optional) PACKET SECONDARY HEADER
CHANNEL SPECIFIC DATA
(Optional) NODE INDEX FILE SIZE
INTRA-PACKET TIME STAMP FOR NODE INDEX 1
INTRA-PACKET DATA HEADER FOR NODE INDEX 1
RECORDING NODE INDEX 1
INTRA-PACKET TIME STAMP FOR NODE INDEX 2
INTRA-PACKET DATA HEADER FOR NODE INDEX 2
RECORDING NODE INDEX 2
:
INTRA-PACKET TIME STAMP FOR NODE INDEX n
INTRA-PACKET DATA HEADER FOR NODE INDEX n
RECORDING NODE INDEX n

Figure 10-34m. General recording node index packet.

msb 31 24	23 16	15 lsb 0
RESERVED	DATA TYPE	CHANNEL ID
DATA PACKET OFFSET (LSLW)		
DATA PACKET OFFSET (MSLW)		

Figure 10-34n. Recording node index entry layout.

10.6.8 ARINC-429 Data Packets, Format 0. Data shall be packetized in Word Mode: each 32-bit word of an ARINC-429 bus shall be preceded by an Intra-Packet Header containing an Intra-Packet Data Header only with an identifier (ID Word) that provides type and status information. The Intra-Packet Header does not contain an Intra-Packet Time Stamp. The Packet Time in the Packet Header is the time of the first ARINC data word in the packet, and the time of successive ARINC data words is determined from the first word time using the gap times in the ID words that precede each of the data words. Multiple words of multiple ARINC-429 buses can be inserted into a single packet. The resultant packets shall have the following format as shown in Figure 10–35.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
ID WORD FOR DATA WORD 1	
ID WORD FOR DATA WORD 1	
ARINC-429 DATA WORD 1 (BITS 15-0)	
ARINC-429 DATA WORD 1 (BITS 31-16)	
ID WORD FOR DATA WORD 2	
ID WORD FOR DATA WORD 2	
ARINC-429 DATA WORD 2 (BITS 15-0)	
ARINC-429 DATA WORD 2 (BITS 31-16)	
:	
ID WORD FOR DATA WORD n	
ID WORD FOR DATA WORD n	
ARINC-429 DATA WORD n (BITS 15-0)	
ARINC-429 DATA WORD n (BITS 31-16)	
PACKET TRAILER	

Figure 10-35. ARINC-429 data packet format.

NOTE

Time tagging of ARINC-429 shall correspond to the first data bit of the packet.

a. ARINC-429 Packet Channel Specific Data Word. The Packet Body portion of each ARINC-429 data packet shall begin with a Channel Specific Data word formatted as shown in Figure 10–36.

msb **31**	**16**	**15**	**lsb** **0**
RESERVED		MSGCOUNT	

- Message Count (MSGCOUNT). (Bits 15-0) indicate the binary value of the number of ARINC-429 words included in the packet.

- Reserved. (Bits 31-16) are reserved

Figure 10-36. ARINC-429 packet channel specific data word format.

b. Intra-Packet Data Header. (Bits 31-0) contain the ARINC-429 ID Word. Each ARINC-429 bus data word is preceded by an identification word and the bit definitions are as shown in Figure 10–37.

msb 31 24	23	22	21	20	19 lsb 0
SUBCHANNEL-1	FE	PE	BS	RS	GAP TIME

- Gap Time. (Bits 19-0) contain a binary value which represent the gap time from the beginning of the preceding bus word to the beginning of this bus word in 0.1 microsecond increments. The gap time of the first word in the packet is GAP TIME=0. When the gap time is longer than 100 milliseconds, a new packet must be started.

- Reserved. (Bit 20) is reserved.

- ARINC-429 Bus (BS). (Bit 21) indicates which ARINC-429 bus the data is from.

  0 = Indicates Low-Speed ARINC-429 bus (12.5 kHz)
  1 = Indicates High-Speed ARINC-429 bus (100 kHz)

- Parity Error (PE). (Bit 22) indicates an ARINC-429 Parity Error.

  0 = No parity error has occurred
  1 = Parity error has occurred

- Format Error (FE). (Bit 23) indicates an ARINC-429 Format Error.

  0 = No format error has occurred
  1 = Format error has occurred

- Subchannel. (Bits 31-24) indicates a binary value which defines the ARINC-429 channel number belonging to the following data word. 0 means first channel. Maximum 256 ARINC-429 words can be placed in one packet.

Figure 10-37. Intra-packet data header format.

c. ARINC-429 Packet Data Words. ARINC-429 Data: The data words shall be inserted into the packet in the original 32-bit format as acquired from the bus.

10.6.9 Message Data Packets. Format 0. The data from one or more separate serial communication interface channels can be placed into a Message Data Packet (see Figure 10–38).

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 15–0)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 31–16)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR MSG 1 (BITS 63-48)	
INTRA-PACKET DATA HEADER FOR MSG 1 (BITS 15-0)	
INTRA-PACKET DATA HEADER FOR MSG 1 (BITS 31-16)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
:	
INTRA-PACKET TIME STAMP FOR MSG n (BITS 15–0)	
INTRA-PACKET TIME STAMP FOR MSG n (BITS 31–16)	
INTRA-PACKET TIME STAMP FOR MSG n (BITS 47-32)	
INTRA-PACKET TIME STAMP FOR MSG n (BITS 63-48)	
INTRA-PACKET DATA HEADER FOR MSG n (BITS 15-0)	
INTRA-PACKET DATA HEADER FOR MSG n (BITS 31-16)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
PACKET TRAILER	

Figure 10-38. Message data packet format.

a. Message Packet Channel Specific Data Word. The Packet Body portion of each Message Data Packet begins with a Channel Specific Data word. It indicates if the Packet Body contains several short messages (Type: Complete), or one segment of a long message (Type: Segmented).

b. Complete Message Channel Specific Data Word. The Channel Specific Data word is formatted for the Complete type of packet body as shown in Figure 10–39.

msb 31 18	17 16	15 lsb 0
RESERVED	TYPE	COUNTER

- Counter. (Bits 15-0) contain a binary value which represents the number of messages included in the packet.

- Type. (Bits 17-16) indicate the type of Serial Packet.

  00 = One or more complete messages
  01 = Reserved
  10 = Reserved
  11 = Reserved

- Reserved. (Bits 31-18) are reserved.

Figure 10-39. Complete message channel specific data word format.

c. Segmented Message Channel Specific Data Word. The Channel Specific Data word is formatted for the Segmented type of packet body as shown in Figure 10–40.

msb 31 18	17 16	15 lsb 0
RESERVED	TYPE	COUNTER

- Counter. (Bits 15-0) contain a binary value which represents the segment number of a long message. The number must start with 1 and must be incremented by one after each packet. The maximum length of a single long message can be 4GBytes (combined with the 16-bit Message Length field, see description at paragraph 10.6.9.1d (below Figure 10–42, Intra-packet data header format).

- Type. (Bits 17-16) indicate the type of Serial Packet.

  00 = Reserved
  01 = Packet is a beginning of a long message from a single source
  10 = Whole packet is the last part of a long message from a single source
  11 = Whole packet is a middle part of a long message from a single source

- Reserved. (Bits 31-18) are reserved.

Figure 10-40. Segmented message channel specific data word format.

d. Message Data Intra-Packet Header. After the Channel Specific Data, Message Data is inserted into the packet. Each Message is preceded by an Intra-Packet Header that has both an Intra-Packet Time Stamp and an Intra-Packet Data Header containing a Message ID Word. The length of the Intra-Packet Header is fixed at 12 bytes (96 bits) positioned contiguously, in the following sequence (shown in Figure 10–41).

msb **31**	**lsb** **0**
TIME (LSLW)	
TIME (MSLW)	
MESSAGE ID WORD	

Figure 10-41. Message data intra-packet header.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the Message Data.

First long word bits 31-0 and second long word bits 31-0 indicate the following values:

  - The Relative Time Counter that corresponds to the first data bit in the Message with bits 31 to 16 in the second long word zero filled or;

  - Time, if enabled by bit 7 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first data bit in the Message.

- Intra-Packet Data Header. The Intra-Packet Data Header is an identification word (Message ID Word) that precedes the message and is inserted into the packet with the format shown in Figure 10–42.

msb 31	 30	 29 16	 15 lsb 0
DE	FE	SUBCHANNEL	MESSAGE LENGTH

- Message Length. (Bits 15-0) contain a binary value representing the length of the message in bytes (n) that follows the ID Word. The maximum length of a message (complete) or a message segment (segmented) is 64K bytes.

- Subchannel. (Bit 29-16) contain a binary value that represents the subchannel number belonging to the message that follows the ID Word when the Channel ID in the packet header defines a group of subchannels. Zero means first and/or only sub-channel.

- Format Error (FE). (Bit 30) indicates a protocol error, such as out-of-sequence data or length errors.

  0 = No Format Error
  1 = Format Error encountered

- Data Error (DE). (Bit 31) indicates bad data bits as determined by parity, checksums, or CRC words received with the data.

  0 = No Data error has occurred
  1 = Data error has occurred

Figure 10-42. Intra-packet data header format.

10.6.10 Video Data Packet.

10.6.10.1 Video Packets, Format 0 (MPEG-2). Format 0 MPEG-2 encoding will be IAW Department of Defense (DoD) Motion Imagery Standards Profile (MISP) Standard 9701–Standard Definition Digital Motion Imagery, Compression Systems MPEG-2 Main Profile @ Main Level (MP@ML). The MPEG-2 format (see Table 10–6) will be Transport Streams (TS) per MISP Recommended Practice (RP) 0101. The TS will be encapsulated at a Constant Bit Rate (CBR) within the limits of MP@ML specifications for further standardization and telemeter/transmission requirements of the video. These MPEG-2 algorithm features are combined to produce an encoded video stream which will be encapsulated in Format 0 packets.

To promote interoperability, reliable system operation, and system development efficiency Format 0 limits the use of different MPEG-2 stream types in motion imagery applications and makes it possible to code an ITU-R 601 Rec. picture format without filtering processes before coding. This will eliminate the need for proprietary encoding/decoding (CODEC) filters which would violate the intent of an "open" standard and making decoding of the data difficult without specific knowledge of or access to the encoding process.

Transport streams are limited to a single program stream using Program Elementary Stream (PES) packets that share the same common time base. A transport stream must contain the Program Association Table (PAT) and Program Map Table (PMT) that define the Program ID (PID) for the Program Clock Reference (PCR) stream. Program streams also must contain at least one pack header.

TABLE 10-6. MPML@TABLE

Profile Table

–	–
B-frames	YES
chroma_format	4:2:0
scalability	NONE
Intra DC precision	8, 9, 10 bits

Level Table

–	–
Maximum Bit Rate	15 Mbps
Buffer Size	1835008 bits
Maximum Sample Density	720 samples/lines 576 lines/frame 30 frames/s
Luminance Sample Rate	10368000
Horizontal Vector Range	-512:+511.5
Vertical Vector Range (frame pictures)	-128:+127.5

A packet with Format 0 MPEG-2 Video data has the basic structure shown in Figure 10-43. Note that the width of the structure is not related to any number of bits. Figure 10-43 is merely to represent relative placement of data in the packet.

PACKET HEADER
CHANNEL SPECIFIC DATA
(Optional) INTRA-PACKET HEADER
188 BYTE TS DATA
(Optional) INTRA-PACKET HEADER
188 BYTE TS DATA
:
(Optional) INTRA-PACKET TIME HEADER
188 BYTE TS DATA
(Optional) INTRA-PACKET TIME HEADER
188 BYTE TS DATA
PACKET TRAILER

Figure 10-43. General MPEG-2 video packet, format 0

a. Video Packet Audio. When recording video using Format 0 if audio is present it will be inserted into the TS per ISO/IEC 13818-3. A separate analog channel to specifically record audio will not be required as MPEG-2 supports audio insertion into the TS. By combining video and audio recording bandwidth and memory capacity will be increased.

b. Video Packet Channel Specific Data Word. The packet body portion of each Format 0 packet begins with the Channel Specific Data word, which is formatted as shown in Figure 10–44a.

msb 31	30	29	28	27 lsb 0
ET	IPH	SRS	KLV	RESERVED

- Embedded Time (ET). (Bit 31) indicates if embedded time is present in the MPEG-2 video data.

    0 = No embedded time present
    1 = Embedded time is present.

- MPEG-2 stream embedded time if utilized will be IAW MISP Standard 9708 - Embedded Time Reference for Motion Imagery Systems and Standard 9715 - Time Reference Synchronization. Embedded time is used for the synchronization of core MPEG-2 data when extracted from the IRIG106 Chapter 10 domain, i.e. an export to MPEG-2 files.

- Intra-Packet Header (IPH). (Bit 30) indicate if Intra-Packet Time Stamps are inserted before each Transport Packet.

    0 = Intra-Packet Times Not Present.
    1 = Intra-Packet Times Present.

- SCR/RTC Sync (SRS). (Bit 29) indicates if the MPEG-2 SCR is RTC.

    0 = SCR is not synchronized with the 10 Mhz RTC.
    1 = SCR is synchronized with the 10 Mhz RTC.

Figure 10-44a. Video packet channel specific data word format.

Transport streams contain their own embedded time base used to facilitate the decoding and presentation of video and/or audio data at the decoder. Within a Program stream, all streams are synchronized to a single time source referred to as the System Clock Reference (SCR). Within a Transport stream, each embedded program contains its own PCR, requiring that each Format 0 encoded MPEG-2 Transport stream contain only a single program allowing each format to be treated in a similar manner using a single global clocking reference.

The 10 MHz RTC is for the purposes of synchronizing and time stamping the data acquired from multiple input sources. For input sources that don’t define an explicit timing model for data presentation, superimposing this timing model can be accomplished. For

MPEG–2, however, an explicit synchronization model based on a 27MHz clock is defined for the capture, compression, decompression and presentation of MPEG-2 data. In order to relate the two different timing models, the MPEG-2 SCR/PCR timestamps if enabled will be derived from the 10 MHz RTC timing reference source. The method in which this will be accomplished is to generate the 27 MHz MPEG-2 reference clock slaved to the 10 MHz RTC.

MPEG-2 defines the SCR/PCR timestamp as a 42-bit quantity, consisting of a 33 bit base value and a 9-bit extension value. The exact value is defined as:

$$SCR = SCR\_base * 300 + SCR\_ext$$

where:

$$SCR\_base = ((system\_clock\_frequency * t) / 300) \% 2^{33}$$

$$SCR\_ext = ((system\_clock\_frequency * t) / 1) \% 300$$

For recording periods of less than 26.5 hours, the SCR can be directly converted into the 10 MHz RTC using the equation:

10 MHz RTC time base = SCR * 10 / 27 (rounded to the nearest integer)

For recording periods longer than this, the Format 0 packet header time stamp can be used to determine the number of times the MPEG-2 SCR has rolled over and calculate the upper 8 bits of the free running counter's value.

- <u>KLV</u>. (Bit 28) indicates if KLV Metadata is present in the MPEG-2 video data

  0 = No KLV Metadata present
  1 = KLV Metadata is present.

MPEG-2 stream KLV Metadata if utilized will be IAW MISP Standard 9711 - Intelligence Motion Imagery Index, Geospatial Metadata, Standard 9712 - Intelligence Motion Imagery Index, Content Description Metadata (Dynamic Metadata Dictionary Structure and Contents), 9713 - Data Encoding Using Key-Length-Value, Recommended Practice 9717 - Packing KLV Packets into MPEG-2 Systems Streams, and Standard 0107 – Bit and Byte Order for Metadata in Motion Imagery Files and Streams.

- <u>Reserved</u>. (Bits 27-0) are reserved.

c. Intra-Packet Header. If enabled the Intra-Packet Header shall include a 64-bit Intra-Packet Time Stamp, which is inserted immediately before the TS sync pattern. The length of the Intra-Packet Header is fixed at 8 bytes (64-bits) positioned contiguously, in Figure 10–44b.

msb 31	lsb 0
TIME (LSLW)	
TIME (MSLW)	

Figure 10-44b. Intra-packet data header.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the individual Transport Stream packets.

First Long Word (LSLW) Bits 31-0 and Second Long Word (MSLW) Bits 31-0 indicate the following values:

  - o The 48-bit 10 MHz Relative Time Counter that will correspond to the first bit of the TS. Bits 31 to 16 in the second long word (MSLW) will be zero filled or;

  - o Time, if enabled by bit 6 in the Packet Flags (section). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first bit of the TS.

- Video Packet Data. A Format 0 packet shall contain in integral number of 188 Byte (1,504 bits) TS packets (Figure 10–45). Intra-Packet Headers can be inserted in Format 0 Video Data Packets. The 10MHz RTC Packet Header time is the time of the first bit of the first TS in the packet.

The CBR of the encoding will be user selectable and within the MP@ML specification. Per ISO/IEC 13818-1 the TS format will be fixed-length 188 byte (1,504 bits) frames containing an 8 bit sync pattern or "sync byte" (starting at bit 0 and ending at bit 7 of the TSF). The sync bytes value is 01000111 (0x47). The rest of the TS 187 data bytes will follow (Figure 10–46a).

msb 15	lsb 0
TS SYNC BYTE (BITS 7 TO 0)	TS DATA BYTE (BITS 15 TO 8)
TS DATA (BITS 31 TO 16)	
:	
TS DATA (BITS 1503 TO 1488)	

Figure 10-45. Format 0 MPEG 2 video frame sync and word format (example is 16 bit aligned).

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
(Optional) INTRA-PACKET TIME HEADER	
TS DATA (BITS 15 TO 0)	
TS DATA (BITS 31 TO 16)	
:	
TS DATA (BITS 1487 TO 1472)	
TS DATA (BITS 1503 TO 1488)	
(Optional) INTRA-PACKET TIME HEADER	
TS DATA (BITS 15 TO 0)	
TS DATA (BITS 31 TO 16)	
:	
TS DATA (BITS 1487 TO 1472)	
TS DATA (BITS 1503 TO 1488)	
:	
(Optional) INTRA-PACKET TIME HEADER	
REPEAT FOR EACH TS	
:	
PACKET TRAILER	

Figure 10-46a. Format 0 MPEG-2 Video Data-Sample Packet (Example is 16 Bit Aligned).

10.6.10.2 Video Packets, Format 1 (ISO 13818-1 MPEG-2 Bit Stream). Unlike Video Packets, Format 0 (MPEG-2) the Format 1 packets encapsulate the complete MPEG-2 ISO/IEC 13818-1:2000 bit streams for both Program and Transport with constant or variable bit rates. Also any of the Profiles and Level combinations as set forth by MPEG-2 ISO/IEC 13818-1:2000 may be utilized in the encoding process. Transport streams are limited to a single program stream using Program Elementary Stream (PES) packets that share the same common time base. A transport stream must contain the Program Association Table (PAT) and Program Map Table (PMT) that define the Program ID (PID) for the Program Clock Reference (PCR) stream. Program streams also must contain at least one pack header.

a. MPEG-2 Stream Packet Body. The Format 1 packet with **n** MPEG-2 packets has the basic structure shown in Figure 10–46b. Note that the width of the structure is not related to any number of bits. This drawing is merely to represent relative placement of data in the packet.

PACKET HEADER
CHANNEL SPECIFIC DATA
(Optional) INTRA-PACKET HEADER
MPEG-2 Packet 1
(Optional) INTRA-PACKET HEADER
MPEG-2 Packet 2
:
(Optional) INTRA-PACKET TIME HEADER
MPEG-2 Packet n
PACKET TRAILER

Figure 10-46b. General MPEG-2 video packet, format 1.

b. Video Packet Audio. When recording video using Format 0 if audio is present it will be inserted into the TS per ISO/IEC 13818-3. A separate analog channel to specifically record audio will not be required as MPEG-2 supports audio insertion into the TS or PS. By combining video and audio, recording bandwidth and memory capacity will be increased.

c. MPEG-2 Channel Specific Data Word. The Packet Body portion of each MPEG-2 bit stream begins with a Channel Specific Data word formatted as shown in Figure 10–46c.

Begin Figure 10-46c (Page 1 of 2). MPEG-2 channel specific data word format.

msb 31 22	21	20	19	18 15	14	13	12	11 lsb 0
RESERVED	KLV	SRS	IPH	EPL	ET	MD	TP	PC

- Packet Count (PC). (Bits 11-0) indicate the binary value of the number of MPEG-2 packets included in the Format 1 packet.

An integral number of complete packets will be in each Format 1 packet. If the MPEG-2 hardware implementation is unable to determine the value of this number, the value of 0 is used by default. If TYPE=0, then this number represents the number of Transport stream packets within the Format 1 packet. If TYPE=1, then this number represents of the number of Program stream packs within the Format 1 packet.

- Type (TP). (Bit 12) indicates the type of data the packetized MPEG-2 bit stream contains.
  0 = Transport data bit stream
  1 = Program data bit stream

- Mode (MD). (Bit 13) indicates whether the MPEG-2 bit stream was encoded using a variable or constant bit rate parameter setting.

  0 = Constant Bit Rate stream
  1 = Variable Bit Rate stream

- Embedded Time (ET). (Bit 14) indicates if embedded time is present in the MPEG-2 video data.

  0 = No embedded time present
  1 = Embedded time is present.

MPEG-2 stream embedded time if utilized will be IAW MISP Standard 9708 - Embedded Time Reference for Motion Imagery Systems and Standard 9715 - Time Reference Synchronization. Embedded time is used for the synchronization of core MPEG-2 data when extracted from the IRIG106 Chapter 10 domain, i.e., an export to MPEG-2 files.

- Encoding Profile and Level (EPL). (Bits 18-15) indicate the MPEG-2 Profile and Level of the encoded bit stream.

0000 = SimpleProfile@MainLevel	1000 = HighProfile@MainLevel
0001 = MainProfile@LowLevel	1001 = HighProfile@High-1440Level
0010 = MainProfile@MainLevel	1010 = HighProfile@HighLevel
0111 = MainProfile@High-1440Level	1011 = 4:2:2Profile@MainLevel
0100 = MainProfile@HighLevel	1100 = Reserved
0101 = SNRProfile@LowLevel	1101 = Reserved
0110 = SNRProfile@MainLevel	1110 = Reserved
0111 = SpatialProfile@High-1440Level	1111 = Reserved

- Intra-Packet Header (IPH). (Bit 19) indicate if Intra-Packet Time Stamps are inserted before each Program or Transport Packet.

- SCR/RTC Sync (SRS). (Bit 20) indicates if the MPEG-2 SCR is RTC.

  0 = SCR is not synchronized with the 10 MHz RTC.
  1 = SCR is synchronized with the 10 MHz RTC.

(continued on next page)

(Figure 10-46c Continued)

Transport streams contain their own embedded time base used to facilitate the decoding and presentation of video and/or audio data at the decoder. Within a Program stream, all streams are synchronized to a single time source referred to as the System Clock Reference (SCR). Within a Transport stream, each embedded program contains its own PCR, requiring that each Format 0 encoded MPEG-2 Transport stream contain only a single program allowing each format to be treated in a similar manner using a single global clocking reference.

The 10 MHz RTC is for the purposes of synchronizing and time stamping the data acquired from multiple input sources. For input sources that don't define an explicit timing model for data presentation, superimposing this timing model can be accomplished. For MPEG-2, however, an explicit synchronization model based on a 27 MHz clock is defined for the capture, compression, decompression and presentation of MPEG-2 data. In order to relate the two different timing models, the MPEG-2 SCR/PCR timestamps if enabled will be derived from the 10 MHz RTC timing reference source. The method by which this is accomplished is to generate the 27 MHz MPEG-2 reference clock slaved to the 10 MHz RTC.

MPEG-2 defines the SCR/PCR timestamp as a 42-bit quantity, consisting of a 33 bit base value and a 9-bit extension value. The exact value is defined as:

$$SCR = SCR\_base * 300 + SCR\_ext$$

where:

$$SCR\_base = ((system\_clock\_frequency * t) / 300) \% 2^{33}$$

$$SCR\_ext = ((system\_clock\_frequency * t) / 1) \% 300$$

For recording periods of less than 26.5 hours, the SCR can be directly converted into the 10 MHz RTC using the equation:

10 MHz RTC time base = SCR * 10 / 27 (rounded to the nearest integer)

For recording periods longer than this, the Format 0 packet header time stamp can be used to determine the number of times the MPEG-2 SCR has rolled over and calculate the upper 8 bits of the free running counter's value.

- KLV. (Bit 21) indicates if KLV Metadata is present in the MPEG-2 video data

  0 = No KLV Metadata present
  1 = KLV Metadata is present.

MPEG-2 stream KLV Metadata if utilized will be IAW MISP Standard 9711 - Intelligence Motion Imagery Index, Geospatial Metadata, Standard 9712 - Intelligence Motion Imagery Index, Content Description Metadata (Dynamic Metadata Dictionary Structure and Contents), 9713 - Data Encoding Using Key-Length-Value, Recommended Practice 9717 - Packing KLV Packets into MPEG-2 Systems Streams, and Standard 0107 – Bit and Byte Order for Metadata in Motion Imagery Files and Streams.

- Reserved. (Bits 31-22) are reserved for future use.

Figure 10-46c. MPEG-2 channel specific data word format.

d. Intra-Packet Header. If enabled the Intra-Packet Header shall include a 64-bit Intra-Packet Time Stamp, which is inserted immediately before the MPEG-2 packet (transport or program). The length of the Intra-Packet Header is fixed at 8 bytes (64-bits) positioned contiguously, in the following sequence (Figure 10–46d):

msb 31	lsb 0
TIME (LSLW)	
TIME (MSLW)	

Figure 10-46d. Intra-packet data header.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the individual MPEG-2 packets (transport or program).

First Long Word (LSLW) Bits 31-0 and Second Long Word (MSLW) Bits 31-0 indicate the following values:

- The 48-bit 10 MHz Relative Time Counter that will correspond to the first bit of the MPEG-2 packet (transport or program). Bits 31 to 16 in the second long word (MSLW) will be zero filled or;

- Time, if enabled by bit 6 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first bit of the MPEG-2 packet (transport or program).

10.6.11 Image Packets.

a. Image Packets, Format 0. A Format 0 Image Packet (Figure 10–47) shall contain one or more fixed-length segments of one or more video images. The channel specific data word for an image packet identifies the number of segments in the packet and the portion of the image or images contained in the packet. If the optional Intra-Packet Header is not included with each segment, the Relative Time Counter in the packet header is the time of the first segment in the packet.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT 1 (BITS 15–0)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT 1 (BITS 31–16)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT 1 (BITS 47-32)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT 1 (BITS 63-48)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
:	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT n (BITS 15–0)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT n (BITS 31–16)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT n (BITS 47-32)	
OPTIONAL INTRA-PACKET HEADER FOR SEGMENT n (BITS 63-48)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
PACKET TRAILER	

Figure 10-47. Image packet, format 0

b. Image Packet Channel Specific Data Word. The Packet Body portion of each Image Packet begins with a Channel Specific Data word. It defines the byte length of each segment and indicates if the Packet Body contains several complete images or partial images, and whether or not the Intra-Packet Data Header precedes each segment (Figure 10–48).

msb 31 30	29 28	27	26 lsb 0
PARTS	SUM	IPH	LENGTH

- Length. (Bits 26-0) indicates a binary value which represents the byte length of each segment.

- Intra-Packet Header (IPH). (Bit 27) indicates that the Intra-Packet Header (Time Stamp) precedes each segment of the image.

  0 = Intra-Packet Header not enabled.
  1 = Intra-Packet Header enabled.

- Sum. (Bit 29-28) indicates if the packet contains a partial image, one complete image, multiple complete images, or pieces from multiple images.

  00 = Packet contains less than one complete image
  01 = Packet contains one complete image
  10 = Packet contains multiple complete images
  11 = Packet contains multiple incomplete images

- Parts. (Bit 31-30) indicates which piece or pieces of the video frame are contained in the packet.

  00 = Packet does not contains first or last segment of image
  01 = Packet contains first segment of image
  10 = Packet contains last segment of image
  11 = Packet contains both first and last segment of image

Figure 10-48. Image packet channel specific data word format.

c. Image Intra-Packet Header. After the Channel Specific Data, Format 1 Image Data is inserted into the packet. Each block of data is optionally preceded by an Intra-Packet Header as indicated by the IPH bit in the Channel Specific Data word. When included, the Intra-Packet Header consists of an Intra-Packet Time Stamp only. The length of the Intra-Packet Header is fixed at 8 bytes (64-bits) positioned contiguously, in the following sequence (Figure 10–49).

msb. 31 lsb 0
TIME (LSLW)
TIME (MSLW)

Figure 10-49. Image data intra-packet data header, format 0.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the Format 0 Image Data.

  First long word bits 31-0 and second long word bits 31-0 indicate the following values:

  - The Relative Time Counter that corresponds to the first data bit in the first byte with bits 31 to 16 in the second long word zero filled or;

  - Time, if enabled by bit 7 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first data bit in the Message.

10.6.12 UART Data Packets, Format 0. The data from one or more separate asynchronous serial communication interface channels (RS-232, RS-422, RS-485, etc…) can be placed into a UART Data Packet as shown in Figure 10–50.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART 1 (BITS 15–0)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART 1 (BITS 31–16)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART 1 (BITS 47-32)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART 1 (BITS 63-48)	
UART ID for UART 1 (BITS 15-0)	
UART ID for UART 1 (BITS 31-16)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
:	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART n (BITS 15–0)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART n (BITS 31–16)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART n (BITS 47-32)	
(OPTIONAL) INTRA-PACKET DATA HEADER FOR UART n (BITS 63-48)	
UART ID for UART n (BITS 15-0)	
UART ID for UART n (BITS 31-16)	
BYTE 2	BYTE 1
:	:
FILLER (IF n IS ODD)	BYTE n
PACKET TRAILER	

Figure 10-50. UART Data packet format.

a. UART Packet Channel Specific Data Word. The Packet Body portion of each UART Data Packet begins with a Channel Specific Data word as shown in Figure 10–51.

msb 31	30	lsb 0
IPH	RESERVED	

- Reserved. (Bits 30-0) are reserved.
- Intra-Packet Header (IPH). (Bit 31) indicates that the Intra-Packet Header is inserted before the UART ID words.

  0 = Intra-Packet Header not enabled.
  1 = Intra-Packet Header enabled.

Figure 10-51. UART Packet channel specific data word format.

b. UART Intra-Packet Header. After the Channel Specific Data, UART data is inserted into the packet. Each block of data is preceded by an Intra-Packet Header consisting of the Intra-Packet Time Stamp and a UART ID WORD Intra-Packet Data Header. The length of the Intra-Packet Header is fixed at 8 bytes (64-bits) positioned contiguously, in the following sequence (Figure 10–52).

msb 31	lsb 0
TIME (LSLW)	
TIME (MSLW)	
UART ID WORD	

Figure 10-52. UART Data intra-packet data header.

- Intra-Packet Time Stamp. (8 Bytes) indicate the time tag of the Format 1 Image Data.

First long word bits 31-0 and second long word bits 31-0 indicate the following values:

  - The Relative Time Counter that corresponds to the first data bit in the first byte with bits 31 to 16 in the second long word zero filled or;
  - Time, if enabled by bit 7 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first data bit in the Message.

- Intra-Packet Data Header. The Intra-Packet Data Header is an identification word (UART ID Word) that precedes the data and is inserted into the packet with the following format (Figure 10–53a).

msb 31	30	29 16	15 lsb 0
PE	RESERVED	SUBCHANNEL	DATA LENGTH

- Data Length. (Bits 15-0) indicate a binary value which represents the length of the UART data in bytes (n) that follows the UART ID word.

- Subchannel. (Bits 29-16) indicates a binary value which defines the subchannel number belonging to the data that follows the UART ID Word when the Channel ID in the packet header defines a group of subchannels. Zero means first and/or only sub-channel that the Intra-Packet Data Header is inserted before the UART ID words.

- Reserved. (Bit 30) is reserved.

- Parity Error (PE). (Bit 31) indicates a Parity Error.

  0 = No Parity Error
  1 = Parity Error

Figure 10-53a. Intra-packet data header format.

10.6.13 IEEE-1394 Data Packets, Format 0. This format applies to FireWire data as described by IEEE 1394-1995, IEEE 1394a and IEEE 1394b. FireWire data is packetized to encapsulate completed transactions between nodes. A packet may contain 0 to 65,536 transactions, but is constrained by the common packet elements size limits prescribed in section 10.6.1. FireWire packets have the basic structure shown in Figure 10–53b. Note that the width of the structure is not related to any number of bits. The drawing merely represents relative placement of data within the packet.

PACKET HEADER
CHANNEL SPECIFIC DATA WORD
(Optional) INTRA-PACKET HEADER
(Optional) TRANSACTION DATA
(Optional) INTRA-PACKET HEADER
(Optional) TRANSACTION DATA
:
(Optional) INTRA-PACKET HEADER
(Optional) TRANSACTION DATA
PACKET TRAILER

Figure 10-53b. IEEE-1394 Data packet, format 0.

a. Fire Wire Channel Specific Data Word. The packet body portion (Figure 10–53c) of each FireWire packet shall begin with a Channel Specific Data Word.

msb 31 29	28 25	24 16	15 lsb 0
PBT	SY	RESERVED	TC

- Transaction Count (TC). (Bits 15-0) indicate the binary value of the number of transactions encapsulated in the packet. An integral number of complete transactions shall be included in each packet. It is mandatory that Transaction Count be 0 for Type 0 packet bodies and 1 for Type 1 packet bodies.

- Reserved. (Bits 24-16) are reserved.

- Synchronization Code (SY). (Bits 28-25) indicate the IEEE-1394 Synchronization Code from the transaction. This field is mandatory for Type 1 packet bodies. Otherwise, it is reserved.

- Packet Body Type (PBT). (Bits 31-29) indicate the packet body type as follows:

  000 = Type 0
  001 = Type 1
  010 = Type 2
  011 = Reserved
  1xx = Reserved

Figure 10-53c. IEEE-1394 Channel specific data word.

b. Fire Wire Intra-Packet Header. Each Intra-Packet Header shall contain an 8 byte Intra-Packet Time Stamp only. The format of a FireWire Intra-Packet Header is described by Figure 10–53d.

msb 31 lsb 0
INTRA-PACKET TIME STAMP
INTRA-PACKET TIME STAMP

Figure 10-53d. IEEE-1394 Intra-packet header.

IEEE-1394 Intra-Packet Time Stamp. This frame (8 Bytes) indicates the time tag of the FireWire transaction that immediately follows it in the packet. Time is coded in accordance with all other Chapter 10 packet formats. Specifically, the first long word bits 31-0 and second long word bits 31-0 indicate the following values:

- The Relative Time Counter that corresponds to the first data bit of the transaction, with bits 15-0 in the second long word zero filled or;

- Time, if enabled by bit 7 in the Packet Flags (section 10.6.1.1g). Time format corresponds to the time format indicated by bits 2 and 3 in the Packet Flags (section 10.6.1.1g) and to the first data bit of the transaction.

c. IEEE-1394 Data Packet Body Types. Three packet body types are defined for the encapsulation of FireWire data. Regardless of type, each packet body shall begin with the UART Packet Channel Specific Data Word as described by Section 10.6.12.1a above. The packet body type is indicated within the Channel Specific Data Word. Depending on the packet body type, the Channel Specific Data Word is followed by 0 or more transactions. Also, dependent on packet body type, each transaction may be preceded by an Intra-Packet Header.

d. IEEE-1394 Packet Body Type 0: Bus Status. Type 0 packet bodies shall contain zero Intra-Packet Headers and zero transactions. The Channel Specific Data Word Transaction Count shall be zero. The packet body shall contain the Channel Specific Data Word immediately followed by a single 32 bit word.

Bus Status events shall be encapsulated by Type 0 packet bodies. The 32 bit word in the packet body shall contain an Event Data Word as indicated in Figure 10–53e.

msb 31	30 lsb 0
R	RESERVED

- Reserved. (Bits 30-0) Reserved.

- Reset. (Bit 31) when set, this bit indicates that a IEEE-1394 Bus Reset has occurred.

Figure 10-53e. IEEE-1394 Event data word format.

e. FireWire Packet Body Type 1: Streaming Data. Type 1 packet bodies shall encapsulate FireWire streaming data only. Type 1 packet body data is restricted to that from an IEEE-1394 packet with a Transaction Code of 0xA, be it from an Isochronous Channel or Asynchronous Stream. The packet body shall contain zero Intra-Packet Headers and one transaction. The Channel Specific Data Word Transaction Count shall be one.

The Channel Specific Data Word is immediately followed by a non-zero number of data bytes. The data bytes shall be exactly those of a single IEEE-1394 data block, excluding the IEEE-1394 packet header and Data Block CRC. Data recorded from the stream shall be known to be valid, insofar as, both the IEEE-1394 Header CRC and Data Block CRC tests have passed. In accordance with the definition of Packet Header Data Length, and accounting for the size of the Channel Specific Data Word, the number of data bytes shall be exactly four less than the value indicated in Data Length. Conversely, the value stored in the Packet Header Data Length shall be the number of bytes in the IEEE-1394 data block plus four. The Synchronization Code (sy) from the stream packet shall be indicated in the Channel Specific Data Word, and the Channel Number shall be indicated in the Packet Header Channel ID.

f. FireWire Packet Body Type 2: General Purpose. Type 2 packet bodies encapsulate complete IEEE-1394 packets, including header and data. Use of Type 2 packet bodies is unrestricted and may encapsulate streaming, non-streaming (IEEE-1394 packets with Transaction Codes other than 0xA), isochronous, and asynchronous data. Multiple IEEE-1394 packets, with differing Transaction Codes and Channel Numbers, may be carried within a single Type 2 packet body.

The Channel Specific Data Word shall be followed by a non-zero number of completed transactions as indicated by the Channel Specific Data Word Transaction Count. Each transaction shall be preceded by an Intra-Packet Header as defined above for FireWire Data Packets. Immediately following the Intra-Packet Header, the transaction shall be recorded in its entirety and must be a properly formed IEEE-1394 packet in accordance with the specification in use. The revision of the specification used shall be declared within the accompanying TMATS packet.

NOTE All IEEE-1394 packets contain a 4 bit Transaction Code field (tcode). This field indicates the IEEE-1394 specific format of the transaction.

10.6.14 Parallel Data Packets. Format 0. Parallel data packets are designed to record data from parallel interfaces (2-128 bit wide) including the industry de facto standard 8-bit DCRsi interface. A single packet can hold data words or special data structures as currently defined for the DCRsi scan format. The exact format selection is defined in the Channel Specific Data Word. The data recorded from a parallel interface shall be placed into a Parallel Data Packet Format 0 as shown in Figure 10–53f.

msb 15	lsb 0
PACKET HEADER	
CHANNEL SPECIFIC DATA (BITS 15-0)	
CHANNEL SPECIFIC DATA (BITS 31-16)	
DATA WORD 1	
:	
DATA WORD n	
PACKET TRAILER	

Figure 10-53f. Parallel data, format 0.

a. Parallel Packet Channel Specific Data Word. The Packet Body portion of each Parallel Data Packet begins with a Channel Specific Data word. The Channel Specific Data word is formatted as shown in Figure 10–53g.

msb 31	24	23	lsb 0
TYPE		RESERVED (0) OR SCAN NUMBER	

- Type. (Bits 31-24) indicate the data type stored

0-1:	Reserved
2-128:	Number of bits of the recorded data (parallel data word width in bits)
129-253:	Reserved
254:	DCRsi scan format, contains auxiliary data and DCRsi main data.
255:	Reserved

- Scan Number. (Bits 23-0) is reserved (0) for general purpose parallel data packets or contains the scan number of the first scan stored in the packet for DCRsi data.

Figure 10-53g. Parallel packet channel specific data word format.

b. General Purpose Parallel Data. General purpose parallel data packets, can contain any number of data bytes – as indicated in the Data Length field in the Packet Headers (Figure 10–53h).

- Please note, that to get the number of data words stored in the packet the Data Length must be divided with the number of bytes necessary to hold one Parallel Data word – see below.

- If the number of data bits is less than 9 bits, the word shall be padded to 8-bit bytes.

msb 15			lsb 0
PAD (0)	WORD 2	PAD (0)	WORD 1
:		:	
PAD (0)	WORD N, or PAD (0) IF LENGTH IS ODD	PAD (0)	WORD N-1

Figure 10-53h. Parallel data, up to 8-bit wide words.

If the number of data bits is between 9-16 the words shall be padded to one 16-bit word, as in Figure 10–53i.

msb 15	lsb 0
PAD (0)	DATA WORD 1
:	
PAD (0)	DATA WORD N

Figure 10-53i. Parallel data, 9-16 bit wide words

If the number of data bits is greater than 16 the words shall be padded to multiples of 16–bit data words. Figure 10–53j shows storing of 28-bit data words.

msb 15	lsb 0
DATA WORD1, LS BITS 15-0	
PAD (0)	DATA WORD 1, MS BITS 27-16
:	
DATA WORD N, LS BITS 15-0	
PAD (0)	DATA WORD N, MS BITS 27-16

Figure 10-53j. Parallel data, (Example: 28-bit wide words).

c. <u>DCRsi Parallel Data Packets</u>. The DCRsi data packets can contain any number of complete DCRsi Scans – containing nine auxiliary data and 4356 main data bytes. The number of the scans can be calculated from the Data Length field of the Packet Header. The structure of one DCRsi scan is in Figure 10–53k.

msb 15	lsb 0
AUXILIARY DATA 2	AUXILIARY DATA 1
PAD (0)	AUXILIARY DATA 3
AUXILIARY DATA 5	AUXILIARY DATA 4
PAD (0)	AUXILIARY DATA 6
AUXILIARY DATA 8	AUXILIARY DATA 7
PAD (0)	AUXILIARY DATA 9
DATA BYTE 2	DATA BYTE 1
DATA BYTE 4	DATA BYTE 3
:	:
DATA BYTE 4356	DATA BYTE 4355

Figure 10-53k. DCRsi Scan, 9-auxiliary data byte + 4356 bytes.

The length of the packet can be only N * (12 + 4356) + 4 bytes including the length of the Channel Specific data word.

DCRsi data without auxiliary data bytes can be stored also as 8-bit General Purpose Parallel Data as described in Paragraph 10.6.14.b.

10.7 Recorder Control And Status

10.7.1 Recorder Control. The recorder may be controlled by either discrete control/status lines and/or serial communication ports. The serial interface shall consist of both RS-232 and RS-422 full duplex serial communications.

10.7.2 Communication Ports. The RS-232 and RS-422 serial communication ports shall be functional simultaneously without requiring selection of either port. Status requested by either port shall be returned on both ports. Note that unexpected results may occur if commands are issues on both ports simultaneously.

10.7.3 RS-232/422 Port. An RS-232/422 port shall be available at the Download Port.

10.7.4 Commands. Commands received through the serial communication ports shall not override hardwire discrete controls.

10.7.5 Status Requests. Status requests received through the serial communication ports shall not interfere with hardwire controls.

10.7.6 Serial Status. Serial status shall be provided on either serial status request or discrete activation.

10.7.7 Default Interface. Default Interface with user equipment shall utilize the following ASCII serial communication protocol:

- 38400 baud
- One start bit
- 8 bit data
- No parity
- One stop bit

10.7.8 Serial Commands. The following commands are a subset of the Recorder Command and Control Mnemonics defined in IRIG Standard 106 Chapter 6 Section 8, where additional rules regarding command syntax and recorder operation are also specified, along with examples showing the use of each command. The commands are simple ASCII command strings delimited by spaces. All commands begin with an ASCII period (".") and, with the single exception of the .TMATS command, end with a carriage return and line-feed terminator sequence. Commands will not be case sensitive.

10.7.9 Required Commands. Table 10–7 summarizes the required commands. **Note: Commands will not be case sensitive.**

TABLE 10-7. COMMAND SUMMARY		
This table is for reference only. Refer to IRIG 106 Chapter 6, Section 8.		
COMMAND	**PARAMETERS**	**DESCRIPTION**
.BIT		Runs all of the Built-In-Tests
.CRITICAL	[*n* [*mask*]]	Specify and view masks that determine which of the .HEALTH status bits are critical warnings
.DECLASSIFY		Secure Erases the data in the canister(s)
.DISMOUNT		Disables and removes power from the canister(s)
.DUB	[location]	Same as .PLAY but with internal clock
.ERASE		Erases the data in the canister(s)
.EVENT	[text string]	Display event table or add event to event table
.FILES		Displays information about each recorded file
.FIND	[value [mode]]	Display current locations or find new play point
.HEALTH	[feature]	Display detailed status of the recorder system
.HELP		Displays table of “dot” commands
.LOOP		Starts record and play in read-after-write mode
.MEDIA		Displays memory usage summary
.MOUNT		Powers and enables the memory canister(s)
.PLAY	[location]	Reproduce recorded data starting at [location] using external clock
.RECORD	[filename]	Starts a recording at the current End Of Data
.RESET		Perform software initiated system reset
.SETUP	[*n*]	Displays or selects 1 of 16 (0…15) pre-programmed data recording formats
.SHUTTLE	[*endpoint* [*mode*]]	Play data repeatedly from current location to the specified endpoint location using external clock
.STATUS		Displays the current system status
.STOP	[mode]	Stops the current recording, playback, or both
.TIME	[start-time]	Displays or sets the internal system time
.TMATS	{mode} [*n*]	Write, Read, Save, or Get TMATS file

10.7.10 Required Discrete Control Functions. Required discrete control functions are noted in Figure 10–54.

Description
RECORD
ERASE
DECLASSIFY
ENABLE
BIT

Figure 10-54. Required discrete control functions.

- Control and Status Lines. Five contacts for discrete control and five lines for indicating status shall be provided. Grounding a control line (or causing the indicator line to go to ground) referenced to the recorders ground completes the circuit to activate a function as shown in Figure 10-55.

- Record Command. Activated by toggle switch (Normally closed position 0.55 Volts or less), this discrete commands the recorder to start recording. Recorder will remain in this mode until such time as the switch is set to normally open position.

- Erase Command. Activated by momentary switch (0.55 Volts or less, minimum duration of 100 ms), this discrete commands the recorder to erase its user data and file directory memory provided the enable switch is also activated.

- Declassify Command. Activated by momentary switch (0.55 Volts or less, minimum duration of 100 ms), this discrete causes the recorder to start the declassify procedure provided the enable switch is also activated.

- Command Enable. Activated by momentary switch (0.55 volts or less) for either ERASE or DECLASSIFY discrete to operate.

- Bit Command. Activated by momentary switch (0.55 Volts or less), this discrete commands the recorder to start the BIT procedure.

- Record Status. A Record indication (ON) shall be active low 0.55 volts or less. A Non-Record indication (OFF) will be an open circuit. Current limit of 60 milliamps required.

- BIT Status. A BIT indication (ON) shall be 0.55 volts or less. A Non-BIT indication (OFF) will be an open circuit. Current limit of 60 milliamps required.

- Fault Status. A Fault indication (ON) shall be 0.55 volts or less. A Non-Fault indication (OFF) will be an open circuit. Current limit of 60 milliamps required.

- Erase Status. An Erase indication (ON) shall be0.55 volts or less. A Non-Erase indication (OFF) will be an open circuit. Current limit of 60 milliamps required.

- Declassify Status. A Declassify indication (ON) shall be 0.55 volts or less. A Non-Declassify indication (OFF) will be an open circuit. No discrete control line shall be available at the Download port. Current limit of 60 milliamps required.

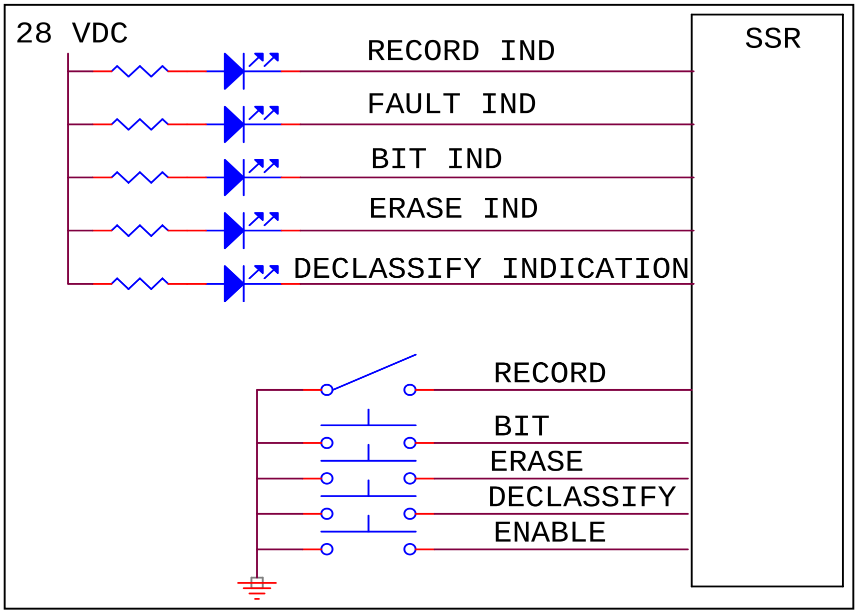


Figure 10-55. Discrete control and indicator functional diagram.

10.7.11 Voltage. 28-VDC auxiliary voltage output shall be provided from the discrete/control port (250 mA max, short circuit protection).

10.7.12 Status Querying. Status querying shall be limited to intervals not to exceed 2 seconds and not faster than one second.

10.8 Declassification Associated Documents.

Documents such as NSA-130-2, DOD 5200.28 (1972) and DCI-116 historically covered declassification guidelines/requirements. These documents focused on declassification of standard disk and other conventional memory technologies. With the advent of advanced, high-density memory technologies, additional guidance must be provided. A new document which addresses various solid state, hard disk, floppy disk, RAID and other storage media declassification, is being developed under NTISSP-9 working group for U.S. Policy.

10.8.1 Approach. The following approaches for declassification are currently recommended. The risk that proper declassification has been effectively implemented will reside ultimately with the user/customer/program manager. It is believed that the user is the most qualified to determine the declassification procedures for any program situation. It is the users responsibility to correctly apply the guidelines to the program in each location to optimize the cost/effect while providing appropriate protection for the data. The guidelines are planned to be available on the Internet at Defense Link.

10.8.2 Algorithm. The algorithm to erase secure data is described below. During the secure erase procedure, all blocks of memory shall be processed. No block in memory shall be excluded from secure erase processing for any reason.

a. First Erase. Every memory block on the board is erased. Any erase failures reported by memory chips will result in the corresponding chip/block being declared a bad block. In the event this bad block is not already in the corresponding board's bad block table, a new bad block entry will be appended onto the board's bad block table. Note that this new entry will not have the Secure Erase flag set.

b. First Write (0x55). Every memory chip location is recorded with the pattern 0x55. As each location is written, the data is read back to guarantee that all bits were written to the expected pattern. Any write failures reported by the chips, or any data errors will result in the corresponding chip/block being declared a bad block. In the event this bad block is not already in the corresponding board's bad block table, a new bad block entry will be appended onto the board's bad block table. Note that this new entry will not have the Secure Erase flag set.

c. Second Erase. Every memory chip shall be erased. Any erase failures reported by the memory chips will result in the corresponding chip/block being declared a bad block. In the event this bad block is not already in the corresponding board's bad block table, a new bad block entry will be appended onto the board's bad block table. Note that this new entry will not have the Secure Erase flag set.

d. Second Write (0xAA). Every memory chip location is recorded with the pattern 0xAA. As each location is written, the data is read back to guarantee that all bits were written to the expected pattern. Any write failures reported by the memory chips, or any data errors will result in the corresponding chip/block being declared a bad block. In the event this bad block is not already in the corresponding board's bad block table, a new bad block entry will be appended onto the board's bad block table. Note that this new entry will not have the Secure Erase flag set.

e. Third Erase. Every memory location is erased. Any erase failures reported by the memory chips will result in the corresponding chip/block being declared a bad block. In the event this bad block is not already in the corresponding board's bad block table, a new bad block entry will be appended onto the board's bad block table. Note that this new entry will not have the Secure Erase flag set.

f. Usable Secure Erased Blocks. All blocks that do not have an entry in the bad block table are now considered to be Secure Erased.

g. Unusable Secure Erased Blocks. If a bad block entry contains the flag indicating it has already been Secure Erased, this block has already been secure erased and requires no further processing, since it is known that this block was skipped during the previous recording.

h. Unsecure Bad Block Processing. A board's bad block table may contain bad block entries that have not previously been Secure Erased. If any such entries exist, the following steps are performed on each block.

- Write Zeros Loop. For each page in the block, a pattern of all zeros is written to the page, and the page is checked to determine if any unexpected ones (UOs) are found. If any UOs are found, the page is re-written to all zeros. This process is repeated up to 16 times. After all allowed re-writes, the board, chip, and block numbers of the block containing any remaining UOs are written to a Failed Erase Table.

- Write Ones Loop. For each page in the block, the page is erased (to all ones) and checked to determine if any unexpected zeros (UZs) are found. If any UZs are found, another erase command is issued to the block. This process is repeated up to 16 times. After all allowed erase operations, the board, chip, and block numbers of the block containing any remaining UZs are written to the Failed Erase Table.

i. Failed Erase Table Processing. Any remaining entries in the Failed Erase Table correspond to blocks that cannot be erased. These blocks may still contain user data and therefore are declared to have failed the secure erase.

A count of the number of bad blocks in the Failed Erase Table that have not been Secure Erased is returned as part of the secure erase results. A non-zero count indicates a secure erase failure of at least one block. A command will allow the user to retrieve the Failed Erase Table. A command will also allow a user to retrieve the data from such blocks and manually determine if these blocks can be designated as "Secure Erased". In most cases a single stuck bit will not compromise any user data and the offending block can be manually declared to be Secure Erased. If the results of manual inspection are indeterminate, the chip containing the failed block must be removed and destroyed, and the Secure Erase procedure must be repeated.

j. Secure Erase Completion. When all blocks are secure erased (no entries in the Failed Erase Table), a single file is written that completely fills the memory. The content of the file is the ASCII string "Secure Erase" repeated over and over. The name of the file in the file table is "SecureErase".

10.9 IEEE 1394B Interface to Recorder Media

This interface definition specifies the interface between the removable media and the host platform as IEEE 1394B. The selection of this protocol was adopted to facilitate a common interface between the media and the computing platform.

> **NOTE**
> This definition does not mandate the interface between the recorder and media.

10.9.1 Media Time Synchronization. In order to allow recorders to be synchronized to the same time without requiring platform modification or external time source provided to recorder. The Removable Media Cartridges shall maintain time allowing for time initialization of recorder. Removable media cartridges shall allow for a battery back-up real time clock device. Initialization of time shall be accomplished via IEEE-1394B interface.

10.9.2 Physical and Signaling. The interface shall allow control of Vendor Specific Solid State devices and Commercial Off The Shelf (COTS) Media as per Figure 10-56.

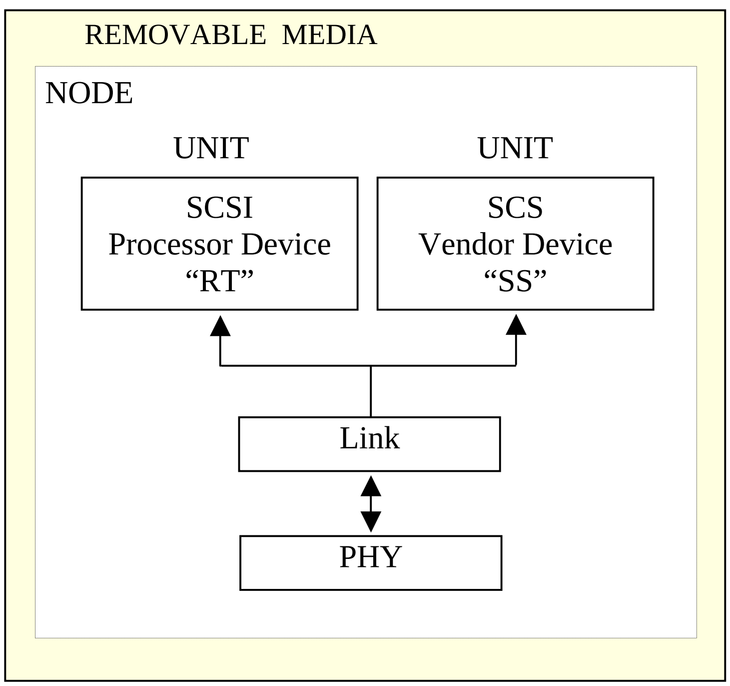


Figure 10-56. Removable media.

10.9.3 Removable Media Communication. The fundamental method of communicating shall be in accordance to the IEEE 1394b protocol. Packets sent and received shall be asynchronous transmissions. IEEE-1394b packets shall encapsulate Serial Bus Protocol (SBP-2) formatted packets for the transport of commands and data. Removable media devices are to use SCSI command set(s) and therefore SCSI commands and status shall be encapsulated in SBP-2 Operation Request Blocks (ORB).

> **NOTE** SBP-2 provides for the transport of 6-10- and 12-byte SCSI Common Descriptor Blocks within a command ORB.

10.9.4 Transport of Serial Commands. Removable devices shall implement the SEND and RECEIVE Processor Device SCSI-2 Commands. The IRIG Standard 106 Chapter 6 commands and data will be transported using these SCSI commands and the data buffers.

10.9.5 Mandated IEEE-1394b Interface Connector. The connector type for the removable media shall be a ““IEEE 1394b Bilingual Socket” connector. Power for the Removable Media shall be derived from the Bilingual interface connector.

10.9.6 Real Time Clock. Removable media configured with a real time clock shall allow for time to be preset in the media allowing for the transfer to the recorder. SCSI command set shall be utilized to set time on the cartridge.

- Real Time Clock Time Format. Time format shall be in accordance with paragraph IRIG 106 Chapter 6 section 6.8.4.23. The date format shall be in accordance with ISO 8601.

- Real Time Clock Logic Unit Number. The standard SCSI Media devices are using Logical Unit Number (LUN) = 0.

The Real Time Clock shall be assigned LUN =1.

10.9.7 Mandatory Commands for Processor Devices.

.BIT
.TIME
.DATE
.ERASE
.STATUS
.HEALTH
.CRITICAL

> **NOTE** The operation of these commands is described in Chapter 6, Section 8 Command and Control Mnemonics.

10.9.8 Time Setting Requirements. To set time, the .TIME commands should be used according to Chapter 6, Section 8, Recorder Command and Control Mnemonics, Paragraph 6.8.4.23. To guarantee and avoid uncontrolled delay the following algorithm shall be used:

a. The Host device puts a .TIME command with time parameter to be set in its SEND buffer and sends it at least 100 ms prior to the correct time to the Real Time Clock device. The delay is necessary to allow the Processor Device to be prepared for the exact time setting and to hold off enough in the Host to force a Doorbell with the next SCSI command. Without enough delay the Host will not be able to chain the next SCSI command together with the previous command. If the operating system demands it a delay greater than 100 ms can be used.

b. The Processor Device shall process this time and be prepared to set it at receipt of the Doorbell.

c. A SEND command shall be sent to the Real Time Clock with the message .TIME without parameters to ask back the time set.

10.9.9 Set Time. To set time the .TIME commands should be used according to IRIG 106 Chapter 6, Section 8 Recorder Command and Control Mnemonics, Paragraph 6.8.4.23.

10.9.10 Date Setting Requirements. A .DATE [start-date] command shall be utilized for setting or displaying date of the removable memory real time clock. The date shall be set in year-month-day format according to ISO 8601.

- Date Example.

```
.DATE
DATE 2002-12-31
*
```

10.9.11 Checking Battery Status. Verification of health of battery shall be accomplished with .CRITICAL and .HEALTH commands IAW IRIG 106 Chapter 6, Section 8. Recorder Command and Control Mnemonics, Paragraph 6.8.4.2 and Paragraph 6.8.4.10.

10.9.12 Vendor Specific Devices. The “Mandatory SCSI Command Set for Vendor Specific Devices” are as follows:

a. For random access devices:

INQUIRY
READ
READ CAPACITY
TEST UNIT READY
REQUEST SENSE

b. For sequential access devices:

INQUIRY
READ
REWIND
TEST UNIT READY
REQUEST SENSE

NOTE COTS media shall support as a minimum the SCSI command set to support data download in accordance with section 10.4 of this standard.

10.9.13 Mandatory ORB Formats for the Processor Device.

10.9.13.1 Minimum Operational Requirements. The time setting accuracy of the Real Time Clock device shall be better than 1 millisecond.

The short time accuracy of the Real Time Clock device must be at least 10 ppm in the temperature range 0-40C, and at least 50 ppm in the temperature range –40C - +85C.

10.9.13.2 ORB Format.

a. Login ORB format. The login ORB format is illustrated in Figure 10-57.

most significant						
password						
Login_response						
n	Rq_fmt	x	reserved	reconnect	function	LUN
password_length						login_response_length
Status_FIFO						
						least significant

Field and Bit Descriptions

password	32 bits	The password shall be "RTC." The password field shall contain the immediate data and the password_length shall be zero.
Login_response login_response_length	64 bits 16 bits	The login_response (32 bits) field and login_response_length (16 bits) field shall specify the address and size of a buffer (minimum of 12 bytes) allocated by the host for the return of the login response.
n	1 bit	The notify bit "n" shall be one.
Rq_fmt	2 bits	The rq_fmt shall be zero.
X	1 bits	The exclusive bit "x" shall be one.
reserved	4 bits	Reserved shall be zero (0).
reconnect	4 bits	The reconnect field shall specify the reconnect time as a power of 2 seconds. A value of zero shall mean one second.
function	4 bits	The Function shall be zero.
LUN	16 bits	The logical unit number shall be one (1).
Status_FIFO	64 bits	The Status_FIFO shall contain the address allocated by the host for the return of status for the LOGIN request and for the return of subsequent write and read buffer response(s) indicating success/failure of the operation

Figure 10-57. Login ORB format.

b. <u>Login Response</u>. The Login Response format is illustrated in Figure 10–58.

most significant	
length	login_ID
command_block_agent	
reserved	reconnect_hold
	least significant

Field and Bit Descriptions		
length	16 bits	The length, in bytes, of the login response data
login_ID	16 bits	Used in all subsequent requests to the SSMC's Management Agent.
command_block_agent	64 bits	Base address of agent's CSR
reserved	16 bits	Reserved shall be zero (0).
Reconnect_hold	16 bits	TBD

Figure 10-58. Login response format.

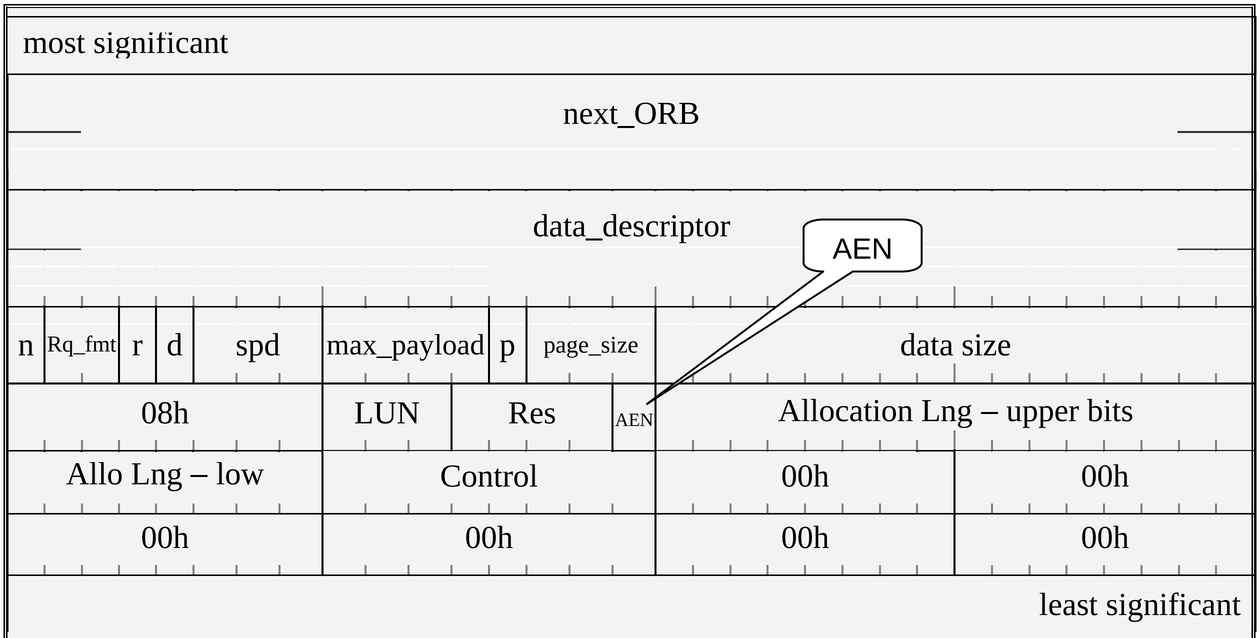
c. <u>Send</u>. The SEND command ORB format is illustrated in Figure 10–59.

most significant								
next_ORB								
data_descriptor								
n	Rq_fmt	r	d	spd	max_payload	p	page_size	data size
0Ah					LUN	Res	AEN	Xfer Lng – upper bits
Xfer Lng – low					Control			00h / 00h
00h					00h			00h / 00h
least significant								

Field and Bit Descriptions		
next_ORB	64 bits	ORB pointer format shall be in accordance with SBP-2 specifications.
data_descriptor	64 bits	Data_descriptor shall contain the address of the data buffer.
n	1 bits	Completion notification “n” shall be one. The target shall store a status block at the status_FIFO address at the address supplied in the login request.
rq_fmt	2 bits	Required format shall be zero.
r	1 bits	Reserved shall be zero (0);
d	1 bits	Direction bit “d” shall be zero.
spd	3 bits	Speed “spd” shall have a value of two.
max_payload	4 bits	The maximum data transfer length “max_payload” shall be nine.
p	1 bits	The removable media device must be prepared to handle the Page table bit p=0 and p=1 cases as the standard operating systems set this bit without influence of the application programs.
page_size	3 bits	Page size shall be zero if the p field is set to 0, otherwise this field shall be set to the valid page size.
data size	16 bits	The data size shall be 50h (80 characters).
LUN	3 bits	The logical unit number shall be one (1).
Res	4 bits	Reserved shall be zero (0).
AEN	1 bits	AEN shall be zero.
Xfer Lng	24 bits	Length of the Chapter 6 command string. Shall be 50h (80 characters).
Control	8 bits	Control shall be 80h.

Figure 10-59. Send command ORB format (8 quadlets).

d. RECEIVE command block ORB format. The Receive command block ORB format is illustrated in Figure 10–61.



Field and Bit Descriptions		
next ORB	64 bits	ORB pointer format shall be in accordance with SBP-2 specifications.
data descriptor	64 bits	Data_descriptor shall contain the address of the data buffer.
n	1 bit	Completion notification “n” shall be one. The target shall store a status block at the status_FIFO address at the address supplied in the login request.
Rq_fmt	2 bits	Rq_fmt shall be zero.
r	1 bit	Reserved: shall be zero (0).
d	1 bit	Direction bit “d” shall be one.
spd	3 bits	Speed “spd” shall have a value of two.
max_payload	4 bits	The maximum data transfer length “max_payload” shall be nine.
p	1 bit	The removable media device must be prepared to handle the Page table bit p=0 and p=1 cases as the standard operating systems set this bit without influence of the application programs.
page_size	3 bits	Page size shall be zero if the p field is set to 0, otherwise this field shall be set to the valid page size.
data size	16 bits	The data size shall be 50h (80 characters).
LUN	3 bits	The logical unit number shall be one (1).
Res	4 bits	Reserved shall be zero (0).
AEN	1 bit	AEN shall be zero.
Allocation Lng	24 bits	Allocation_Lng = length of the Chapter 6 response string.
Control	8 bits	Control shall be 80h.

Figure 10-60. Receive command block ORB format (8 quadlets).

****** END OF CHAPTER 10 ******