DEVELOPMENT STANDARD

QIC-CRF1
Revision J
15 Jun 95

COMMON RECORDING FORMAT WITH MULTICHANNEL CAPABILITY FOR USE WITH RLL 1,7 ENCODED RECORDING FORMATS

For use with the following development standards:

- QIC-3230-MC
- QIC-5010-DC
- QIC-5210-DC

Quarter-Inch Cartridge Drive Standards, Inc.

311 East Carrillo Street
Santa Barbara, California 93101
Telephone (805) 963-3853
Fax (805) 962-1541
www.qic.org

(See important notices on the following page)

Table Of Contents

Overview of Revision Changes

Rev. A — This is the first revision of the QIC-CRF 1 document. It is based upon the QIC 91-41 revision C

Rev. B — Incorporated changes approved at the June 1992 meeting of the QIC Technical Committee

Rev. C — The following pages have been changed from revision b:
1, 2, 5, 7, 10, 22, 29, 30, 32, 35, 36 and 37

Rev. D — Incorporated changes from QIC 93-13B. removed section 4.6 to accommodate new Compression page from QIC 93-14A

Rev. G — Introduction of Media Management Blocks according to QIC 93-101.

Document is converted to Words for Windows. A total restructuring of the document was necessary in order to add changes from rev D to G in a structured way. Sections 5 has been modified to clear up the difference between fixed, variable and, logical block. The terms "logical block" or "variable block" were previously used for contradictory purposes. To avoid confusion, the new terms Logical Host Block and Logical Tape Block are introduced. The terms Physical Block Address and Physical Block Number where previously used for the same purpose but also in conflict to each other. A clarification is now added in Section 5 and all erroneous references are corrected.

To simplify references the Block Type names "Limited 255 Block" and "Limited 511 Block" has been introduced.

Unused Definitions has been removed and new Definitions added.

Rev. H — Reintroduction of Cancel Marks according to QIC 94-25.

Modified Media Header

Improved End Of Track detection

Rev. I — Introduction of Logical Tape Block Headers to simplify read operation and reduce micro processor load.

Introduced Track ID blocks according to QIC 94-27

Improved handling for servo dropout; Media Management block concept replaced by WPC concept.

Modified controlbyte layout and introduced WPC to avoid mixing of old and new data in case of servo dropout.

Introduced EOD block with WPC for safe EOD detection.

Device Directory Modified to handle WPC and physical blockno that are continuous at track turn.

Rev. J — Introduction of ECC Mode 2 according to QIC-95-59

Modified Media Header to match QIC 3095 as much as possible.

1. INTRODUCTION

1.1. Scope

This Standard provides a common recording format standard for a magnetic tape in a cartridge to be used for information interchange between information processing systems, communication systems, and associated equipment utilizing a standard code for information interchange, as agreed upon by the interchange parties. The Standard provides for the use of both 1/4 inch (6.35 mm) and 0,315 inch (8mm) wide tape.

The standard opens for use of track following servo, reference bursts and Track ID as optional features. This Standard refers solely to formatting of data and incorporation of Reed-Solomon Error Correcting Code for data retrieval and interchange. Specific tape drive standards which specify magnetic recording specifications and track geometry will reference this standard.

In this standard, several references are made to QIC-121. However, this does not imply that a drive meeting this Standard must be compatible with QIC-121.

1.2. Introduction

This Standard defines the logical frame format requirements for data recorded on magnetic tape in a cartridge. The drive recording specification and physical track specifications are defined in other QIC development standards.

2. DEFINITIONS

For the purpose of this standard, the following definitions apply:

Appendable Point:	A point at which a write or erase operation is permitted. See section 3.4.10.
Bad Block:	A Physical block determined to be bad during the Read-While-Write operation, or later during a read operation.
Bit:	A single digit in the binary system.
Block:	Short form of Physical Block.
Block Marker:	A group of encoded bits following the preamble and marking the start of each physical block.
BOT (Beginning of Tape) Marker:	The BOT Marker is a set of two holes punched in the tape. There are four sets of holes provided, the innermost of which is used for identifying the storage position for the cartridge. The additional sets of holes are used to ensure reliability of detection. Note: In the storage position, all of the permissible recording area of the tape is wound on the supply hub and is protected by at least one layer of tape not used for recording data. Cartridges to be interchanged shall be rewound to the storage position prior to interchange.
Byte:	A group of 8 data bits operated on as a unit.
Cancel Mark:	The Cancel Mark acts as a "negative" File Mark or Set Mark. When a Cancel Mark follows as the first physical block in the next frame after a File Mark or Set Mark, the drive when reading the tape will logically ignore the Cancel Mark and the File Mark or Set Mark it follows
Control Block:	A physical block designated as a Control block. This Standard does not define the use of control blocks nor the contents of the data area of the control block.
Control Field:	A group of 8 bytes recorded before the data area in each block, containing information about block address, track number and block type.
CRC (Cyclic Redundancy Code):	The CRC is a group of 4 bytes recorded at the end of each block of data for the purpose of error detection.
CRC Erasures:	The CRC has detected an error in the data.
CRC Error:	There is an error in the data that was not detected by the CRC.
Data Block:	A physical block containing user valid data in its data field.
ECC (Error Correction Code):	Special drive generated information which may be used to correct bad blocks.
ECC Block:	A physical block containing drive generated ECC data in its data field and part of control field.

Encoding:	A method whereby a group of data bits is translated into a group of recording bits. In this standard, 2,4 or 8 data bits are translated into 3, 6 or 12 recording bits.
EOD (End of recorded Data) Marker:	The EOD marker is used to mark the end of the data area. The marker consists of a minimum of 64K recordings of a 2-byte postamble pattern.
EOT (End of Tape) Marker:	The EOT Marker is a single hole punched in the tape to indicate that the usable recording area of the tape has been exceeded, and that the physical end of the tape is approaching. There are three EOT holes to ensure reliable detection.
EW (Early Warning) Marker:	The EW Marker is a single hole punched in the tape to indicate the approaching end of the usable recording area in the forward direction.
File Mark Block:	A physical block designated as a File Mark.
Frame:	A group of 64 physical blocks forming a complete logical unit.
GBytes (GB):	This standard defines 1 GB to be equal to 10^9 bytes.
KBytes (KB):	This standard defines 1 KB to be equal to 1024 bytes.
LEW (Logical Early Warning):	A logical position used to give an early warning that the End Of Partition is approaching. Physical blocks behind the LEW position will have the LEW bit set.
Logical Tape Block (LTB):	A set of physical tape blocks that are used to hold from 0 to 64536 logical host blocks.
Logical Tape Block Group (LTBG):	A set of logical tape blocks that is needed to hold one large logical host block (size > 65536 bytes).
Logical Tape Block Header (LTBH):	A field in start of a logical tape block that contains information about the logical tape block like size, quantity and type.
LP (Load Point) Marker:	The LP Marker is a single hole punched in the tape to indicate the approaching start of the usable recording area in the forward direction.
Magnetic Tape Cartridge:	A cartridge containing 0.25 inch (6.3 mm) or 0.315 inch (8 mm) wide magnetic tape wound on two coplanar hubs with an internal drive belt to transport the tape between the hubs.
Max Rewrites:	The max. no. of times a block may be rerecorded during Read After Write (RAW) verification..
Max Servo Dropout Distance:	The max. legal length of a servo dropout
MBytes (MB):	This standard defines 1 MB to be equal to 10^6 bytes.
Media Header Block:	A unique physical block used to identify the Media Header of format being recorded.
Overrun or Buffer Overrun:	A condition which occurs when reading a tape and the host cannot accept data fast enough to permit the drive to stream.
Partition:	A logical storage volume within the tape physical storage volume.
Partition Table:	A table in the Device Directory that relates Partitions to contiguous Track Set pairs

Physical Block:	A group of 512 consecutive data bytes plus additional control bytes recorded as a unit.
Postamble:	A special sequence of bits recorded at the end of each block.
Preamble:	A special sequence of bits recorded at the beginning of each block.
RAT (Random Access Table):	A part of the Device Directory providing means for synchronizing logical addresses to physical addresses on the tape so that positioning can be accomplished quickly.
Read-After-Write:	A method whereby data being recorded is read and verified on a separate pass as when they are written.
Read-While-Write:	A method whereby data being recorded is read and verified on the same pass as they are written.
Reserved:	Reserved fields are to be written with zeros and ignored by firmware to facilitate future use by QIC.
RLL (Run Length Limited):	A data encoding method where data bits are encoded so that certain constraints are met with regard to the maximum and minimum distances between flux transitions.
Servo Dropout:	The track following servo signal is outside the tolerances.
Set Mark Block:	A physical block designated as a Set Mark.
Streaming:	A method of recording on magnetic tape that maintains continuous tape motion without the requirement to start and stop within an interblock gap.
Track:	A longitudinal area on the tape along which magnetic signals may be serially recorded.
Track ID:	A prerecorded signal at the beginning of each track required to verify correct track position of the head actuator prior to writing on a track.
Track Set:	A logical collection of N physical tracks which are written or read simultaneously. A track set can be viewed as a logical track that holds N times as much data as a physical track and can transfer data N times as fast as a physical track. A track set may consist of only one track, i.e. N = 1.
Underrun, or buffer Underrun:	When writing to tape, a condition developed when the host transmits data at a rate less than required by the device for streaming operation.
Vendor Specific:	Vendor Specific fields are assigned by QIC for vendors to implement unique features beyond the scope of this document.
Volume Directory:	A directory located at the tape that holds partitioning information and tables that will enable fast access to logical positions on the tape.
Wide Tape:	A tape which is 0.315 inches (8mm) wide.
WPC (Write Pass Count):	A field in the Control Field used to distinguish between old and new data.
1/4 Inch:	A tape which is 0.225 inches (6.35 mm) wide.

3. MEDIA FORMAT

3.1. General Information

3.1.1. Tracks and Track Sets

The physical tape is organized in horizontal tracks. One or more (N) tracks are accessed simultaneously as a Track Set. The no. of tracks pr track set is specified in the relevant QIC Format Standard. Each track set is recorded sequentially, starting with track set 0, then track set 1, and so on. All even numbered track sets shall be recorded in the forward direction (the direction from the BOT marker to the EOT marker). All odd numbered track sets shall be recorded in the reverse direction (the direction from the EOT marker to the BOT marker). Before recording, data is grouped into physical blocks and blocks are then grouped into frames.

3.1.1.1. Track ID

Some Crf1 formats requires a prerecorded Track ID field at the beginning of each track. The Track ID is used to verify correct track position of the head actuator prior to writing a data track. The Track ID consists of 64 consecutive recordings of the Track ID block, followed by an Elongated postamble. The use of Track ID is specified in the relevant QIC Format Standard

3.1.2. Frames and Blocks

This Document specifies the use of both frames and blocks. Section 3.2 provides a detailed description of frames, while section 3.3 provides a detailed description of blocks.

3.1.3. Media Header

The Media Header is a small logical volume at the start of Track Set 0. It contains information about the tape like the no. of recorded Partitions. The Media Header is generally invisible to the host system, but the information in the data area may be transferred to the host by special command sequences. The use and layout of the Media Header is described in section 6.

3.1.4. Partitions

This Document specifies a mean for dividing the Physical Storage Medium into several Partitions. The Partitions are created by grouping a no. of track set pairs as one logical volume. This is controlled by the Partition Table in the Volume Directory (see Section 6.2.2).

Each Partition is an integral number of track set pairs. The minimum Partition is a contiguous track set pair with the even track set at the beginning of partition, and the odd track set as the end of partition. Unless otherwise specified in the relevant QIC Format Standard, the maximum number of partitions is one-half the number of track sets. PSUM, or minimum partition granule as defined in SCSI-2 and QIC-

121, is the capacity corresponding one track set pair. See QIC-121 for details regarding creation and SCSI support of multiple partitions.

3.2. Frames

Each track set consists of sequentially recorded frames as shown in Figure 3.1.

Frame N	Frame N+1	Frame N+2	Frame N+3	Frame N+4	Frame N+5	Frame N+6

Figure 3.1 General Track Layout.

Frames are organized and employed to implement Reed-Solomon error correction (see Section 8). The frame operation is controlled by the recording drive and is normally invisible to the host.

Frames are identified using the 26 most significant bits of the Physical Block Address (see Section 3.4.1).

Underrun is not allowed in the middle of a frame, regardless of the frame type. If insufficient data is available to complete the frame, filler blocks shall be substituted with the restriction that filler blocks can never precede valid data blocks within a frame. Filler and control blocks shall not be written interior to logical blocks.

Frames may be overwritten with new data frames or EOD markers. To take advantage of this overwrite capability, append operations may begin at EOD (End of recorded Data) or at any Appendable Point, provided the conditions of section 4.5 are met.

3.2.1. Frame Layout

In every frame, data is recorded in 512 data byte blocks called physical blocks, to differentiate from the logical tape blocks, which may span multiple physical blocks. Each frame contains 64 physical blocks, of which 52 are data blocks, information blocks (file marks, set marks, media header, control or filler blocks), and 12 are ECC blocks.

The general layout of a frame is shown in Figure 3.2.

Data Block 0	Data Block 1	Data Block 2	....	Data Block 51	EEC Block 0	....	EEC Block 11

Figure 3.2 General Frame Layout.

3.2.2. Frame Types

There are 4 different types of frames:

* Data Frame.
* MH (Media Header) Frame.
* Track ID Frame.
* EOD Frame.

Data Frame

The Data Frame contains Data and Information blocks in addition to the normal ECC blocks.

Media Header Frame

The MH (Media Header) Frame contains only Media Header blocks and ECC blocks. The MH frames are only used to record the Media Header as the first four frames on Track Set 0. The Media Header blocks contain configuration information about the recording drive, and the Volume Directory as described in section 6.

Track Id Frame

The Track Id Frame is recorded at the beginning of each track as a part of the formatting process of the cartridge. These frames can easily be distinguished from other frame types because they always has WPC of zero

The Track ID is prerecorded as 64 consecutive blocks of 512 data byte called Track ID blocks. The Track ID is written in all available channels with the same TrackSet information in all channels. Even though the Track ID has the same size as a normal frame, it is different as all blocks have the same contents and since it contains no ECC blocks. The Track ID blockno. starts with 0 for channel 0 and 64 for channel 1. Each succeeding block shall have its blockno. increased by one over its predecessor. The relevant QIC Format Standard specifies if Track ID should be used.

End of Data Frame

The Eod Frame is written under firmware control each time the drive goes into underrun. The purpose is to give safe detection of EOD. The EOD Frame is written as 64 consecutive EOD blocks all having the same blockno. The blockno. should be the no. of the first block in the next frameset as if no underrun had occurred. In dual channel mode, the same blockno. should be used for EOD in both channels.

3.3. (Physical) Blocks

There are 9 different types of blocks:

```
Data Block            --------
Media Header Block            |
File Mark Block               |
Set Mark Block                | -----   Information Blocks
Cancel Mark Block             |
Filler Block          --------
ECC Block
EOD Block
Track ID Block
```

Information in the Frame Control Byte determines the type of block being recorded, except for ECC blocks which are recognized by their block numbers (least significant physical block number byte xx11 0100 through xx11 1111), and Track ID blocks which are recognized by their WPC being zero.

The Data Field of all physical blocks contain 512 bytes.

3.3.1. Data Block

The Data Block contains user data. A full Data Block contains 512 bytes, but limited physical data blocks may contain from 1 through 511 valid data bytes. Information about the number of user data bytes available in each data block is recorded in the Frame Control byte of the block. See Sections 3.4 and 5. Data Blocks are normally written in groups to build a Logical Tape Block. See Section 5.

3.3.2. Media Header Block

This block type is used to write the Media Header as the first four frames on track set 0 (see section 6). The Media Header blocks are generally invisible to the host system, but the information in the data area may be transferred to the host by special command sequences.

3.3.3. Filemark Block

Filemark Blocks are physical blocks written to tape in response to a host "Write Filemarks" command. A Filemark Block may contain from one to 65536 Filemarks. The no. of Filemarks and their logical block address is specified in the LTB Header (Logical Tape Block Header) which is hold in first bytes in the data area. See Section 5.3. The rest of the data bytes are not specified in this Standard and may contain vendor unique data.

3.3.4. Setmark Block

Setmark Blocks are physical blocks written to tape in response to a host "Write Setmarks" command. A Setmark Block may contain from one to 65536 Setmarks. The no. of Setmarks and their logical block address is specified in the LTB Header (Logical Tape Block Header) which is hold in first bytes in the data area. See Section 5.3. The rest of the data bytes are not specified in this Standard and may contain vendor unique data.

3.3.5. Cancelmark Block

Cancelmark Blocks are physical blocks written on tape under firmware control. Cancelmarks are used to "cancel" or negate one Filemark or Setmark when overwritten by a host command. The Cancelmark must be the first block in the frame following the frameset containing the Filemark or Setmark it cancels. A Filemark or Setmark may be canceled only if there are no data blocks between it and EOD. The correct resulting logical block count is recovered from the LTB Header (Logical Tape Block Header) which is hold in first bytes in the data area. See Section 5.3. The rest of the data bytes are not specified in this Standard and may contain vendor unique data.

3.3.6. Filler Block

A Filler Block contains no valid information in the data area. Filler Blocks are used to complete frames at the termination of a write process when insufficient host data is available. Filler Blocks may not appear interior to a logical block, and within a frame, filler blocks may never precede a data, filemark or setmark block, unless they are rewritten blocks belonging to the previous frame.

3.3.7. ECC Block

The ECC Block contains error correction bytes which may be used during a subsequent read operation to correct one or more data blocks which cannot be read correctly. The ECC blocks are recognized by the 6 least significant bits of the physical block number (Range 11 0100 through 11 1111).

3.3.8. EOD Block

The EOD Blocks are written under firmware control each time the drive goes into underrun. Their purpose is to give safe detection of EOD. The EOD blocks are written as 64 consecutive blocks all having the same blockno. The blockno. should be the no. of the first block in the next frameset as if no underrun had occurred. In dual channel mode, the same blockno. should be used for EOD in both channels. See Section 4.3 for more details.

3.3.9. Track ID Block

Some Crf1 formats requires Track ID blocks at the beginning of each trackset. The Track ID Block has a special layout (see Section 3.4.4.3) and are recognized by their WPC being 0. They are preformatted on the tape and are used for track verification when starting to write on a new track.

3.4. Physical Block Layout

3.4.1. General Layout

All physical blocks have the basic layout shown in Figure 3.4:

Preamble	Block Marker	Control Field	Data	CRC	Postamble

Figure 3.4 Layout of a block

All sections of a physical block are recorded continuously without any erased gaps between the sections. All blocks within a frame are also always recorded continuously without any erased interblock gaps. Frames are also recorded continuously, except during append operations or when a servo dropout occurs. When appending, a short area with erased or damaged recording may occur between the end of the postamble of one frame and the preamble of the next frame due to the write current turn on time. This area shall always be shorter than the length of the recording of one byte of data. When servo dropout occurs longer portions of the tape may be left not overwritten. See Appendix D.

No underrun or append operations are allowed in the middle of a frameset.

In order to provide the host with a flexible numbering system, this document provides for both a physical block number for every new block recorded on the tape and a logical host block number which may span more than one physical block. The physical block number is recorded in every physical block while the logical host block number is recorded in the LTB header in the datafield of the first physical block in a Logical Tape Block.

3.4.1.1. Physical Block Address

The Physical Block Address relates directly to each recorded block on the tape. Each new block, regardless of its contents, is given a unique physical address. However, a bad block being rewritten further down the tape keeps its original physical block address, regardless of track set number. .

The Physical Block Address consists of two parts; the Track Set Number and the Physical Block Number. The Physical Block Number begins from 0 at the beginning of each partition and the following physical blocks are numbered sequentially. Frames are numbered using the 26 most significant bits of the Physical Block Number.

3.4.1.2. Logical Host Block Address

Logical numbering does not relate to the blocks physically recorded on the tape, but to the block numbering system used by the host. Very often, a host system operates with logical blocks of a different size than the fixed 512 byte physical blocks recorded on the tape. These host blocks may be either smaller or larger than the physically recorded blocks. Host blocks may also be either fixed or variable. Fixed host blocks contain the same number of data bytes in every host block, while variable host blocks may contain a different number of data bytes for each host block. See section 5.1.

The logical numbering system starts with 0 for the first valid host data block, filemark, or setmark recorded on the partition, and is then incremented by one for each new host data block, filemark, or setmark recorded, regardless of track set number.

3.4.2. Preamble

The decoding system must be able to distinguish between a preamble and any possible combinations of data patterns. The Preamble patterns are specified in the relevant QIC Development Standard.

There are three types of preambles: NORMAL, ELONGATED and LONG.

Normal Preamble The Normal Preamble shall be used to synchronize the phase locked loop or similar circuit to the frequency and phase of the data signal. It may also be used to measure the average signal amplitude. The Normal Preamble may be recorded as the only preamble at the beginning of a block or in combination with other types of preambles.

Elongated Preamble An Elongated Preamble is not recorded alone, but will always be followed by a Normal Preamble. The transition between the Elongated Preamble and the following preamble shall be continuous without any erased or destroyed gaps.
An elongated preamble shall be recorded at the beginning of the first block in a frame which is appended to already existing data on a track or the first block in a frame after an underrun situation

Long Preamble The Long Preamble is not recorded alone, but always followed by a Normal Preamble. For formats using Track ID blocks, the Long Preamble is used in front of the Track ID blocks. For formats not using Track ID, the Long Preamble is recorded at the beginning of every trackset..

See Table 3.6 for summary.

3.4.3. Block Marker

The Block Marker, as defined in the relevant QIC Format Standard, marks the start of a new frame or block.

3.4.4. Control Field

The control field consists of 8 bytes numbered 0 to 7. These bytes contain information pertaining to the block they precede, or to the frame that block resides in. The organization of the control bytes are different for the first 52 blocks of each frame than for the ECC blocks.

Control Bytes 0-7 are encoded according to the rules in the relevant QIC Development Standard. Byte 7 is recorded first, followed by byte 6 and so on. The most significant encoded bit in each byte is recorded first.

3.4.4.1. Control Field, Data and Information blocks

Table 3.1 shows the general layout of the control field for Data and Information blocks. Control Byte 0 is the only control byte covered by ECC protection. Logical and physical blocks are treated in detail in Section 5

BYTE	Bit 7	6	5	4	3	2	1	0
7	Physical Block Number next LSB							
6	Physical Block Number LSB							
5	Physical Block Number MSB							
4	Physical Block Number next MSB							
3	WPC (Write Pass Count) MSB							
2	WPC (Write Pass Count) LSB							
1	Track Set Number							
0	Comp	LEW	BLTB	ELTB	Block Type Code			

Table 3.1 Layout of Control Field. First 52 blocks.

Comp The Comp (Compression) bit indicates whether or not this block was compressed. When set to a 1, it indicates this block was compressed. When 0, no compression was used on this block. See Section 7.

LEW The LEW (Logical Early Warning) bit shall be set to 1 for all blocks recorded past Logical Early Warning as defined by the drive. A zero in this field indicates a block recorded prior to the Early Warning for the partition in which the block is recorded.

BLTB, ELTB The BLTB (Beginning of Logical Tape Block) and ELTB (End of Logical Tape Block) are used to mark the beginning and the end of logical tape blocks. BLTB is set to 1 in the first physical block of the logical tape block. In the same way, ELTB is set to 1 in the last physical block containing data from the same logical tape block. BLTB and ELTB shall be set to 0 for all other physical blocks within the same logical tape block and for all other blocks than Data Blocks. Note that filemarks, setmarks and cancelmarks counts as logical tape blocks, and that both BLTB and ELTB should be set for those blocks. See Table 3.2 and Section 5

BLTB	ELTB	Comments
1	0	This physical block marks the beginning of a logical tape block and the presence of a logical tape block header. The logical tape block header is occupying the first bytes in the datafield. while the remaining bytes are used for host data.
0	0	This physical block is neither the first nor the last physical block within a logical tape block. This block, its immediate predecessor, and its successor are part of the same logical tape block. This block contains 512 valid data bytes.
0	1	This physical block is the last physical block within a logical tape block. It may contain a maximum of 512 valid data bytes.
1	1	This physical block marks both beginning and the end of a logical tape block, and the presence of a logical tape block header. The logical tape block header is occupying the first bytes in the datafield. while the remaining bytes may be used for host data if this is a datablock, or unused if it is a filemark, setmark or cancelmark block.

Table 3.2 Logical Tape Block Indicator

Block Type Code

The block type code specifies what type of information that is represented by the physical block. See Table 3.3 and Section 5

Code bits 3210	Block Type	Comments
0000	Full Data Block	This block contains 512 bytes of valid data. This block may be part of a logical tape block.
0001	Limited 255 Block	This block contains from 1 to 255 bytes of valid data. This block will end a logical tape block.
0010	Limited 511 Block	This block contains from 256 to 511 bytes of valid data. This block will end a logical tape block.
0100	File Mark	Contains at least a logical tape block header. The rest of the bytes may contain vendor unique information.
0101	Set Mark	Contains at least a logical tape block header. The rest of the bytes may contain vendor unique information.
0110	Cancel Mark	Contains at least a logical tape block header. The rest of the bytes may contain vendor unique information.
1000	Filler Block	May contain vendor unique information.
1001	EOD Block	May contain vendor unique information.
1010	Media Header Block	Contains special drive and host data. See Section 6.

Table 3.3 Encoding of Block Type Control Bits

Track Set Number

The Track Set Number, which is also the most significant byte of the Physical Block Address.

WPC The WPC (Write Pass Count) is set to 2 the first time a tape is written from BOP (WPC of 1 is reserved for erase function). For each new write from BOP the WPC should be incremented. The purpose of this field is to enable differentiation between new and old data. (Mixture of old and new data is likely if a servo dropout occurred when the tape was written. See Section 4.6). Only blocks with the expected WPC count as valid. If a tape has been erased and repartitioned the WPC should be set to the increment of the previous highest used WPC.

Physical Block Number The Physical Block Number in Control Bytes 4-7 are used to specify the physical block number. Note that the least 2 significant bytes of physical block number are recorded first followed by the 2 most significant bytes. This strange arrangement is a compromise that will allow present hardware to support the format!

Block numbering begins with 00 00 00h at the beginning of each partition.

3.4.4.2. Control Field, ECC blocks

Table 3.4 shows the general layout of the control field for ECC blocks.

BYTE	Bit 7	6	5	4	3	2	1	0
7	Physical Block Number next LSB							
6	Physical Block Number LSB							
5	Physical Block Number MSB							
4	Physical Block Number next MSB							
3	WPC (Write Pass Count) MSB							
2	WPC (Write Pass Count) LSB							
1	Track Set Number							
0	EEC for byte 0 of data blocks							

Table 3.4 Layout of Control Field, ECC blocks

ECC byte 0 of data blocks Control byte 0 of the ECC blocks hold the ECC syndromes for control byte 0 of the Data and Information blocks in the frame. See Section 8.

Track Set Number The Track Set Number, which is also the most significant byte of the Physical Block Address.

WPC The WPC (Write Pass Count) is set to 2 the first time a tape is written from BOP. For each new write from BOP the WPC should be incremented. The purpose of this field is to enable differentiation between new and old data. (Mixture of old and new data is likely if a servo dropout occurred when the tape was written. See Section 4.6). Only blocks with the expected WPC count as valid. If a tape has been erased and repartitioned the WPC should be set to the increment of the previous highest used WPC.

Physical Block Number The Physical Block Number in Control Bytes 4-7 are used to specify the physical block number. Note that the least 2 significant bytes of physical block number are recorded first followed by the 2 most significant bytes. This strange arrangement is a compromise that will allow present hardware to support the format!

Block numbering begins with 00 00 00h at the beginning of each partition.

3.4.4.3. Control Field, Track ID blocks

Table 3.5 shows the general layout of Track ID blocks.

BYTE	Bit 7	6	5	4	3	2	1	0
7	Physical Block Number next LSB							
6	Physical Block Number LSB							
5	Track Set Number							
4	Track Set Number							
3	WPC (Write Pass Count) MSB							
2	WPC (Write Pass Count) LSB							
1	Track Set Number							
0	Track Set Number							

Table 3.5 Layout of Track ID blocks

Track Set Number The Track Set Number is written in 4 positions to enable redundancy so that CRC errors can be trapped (block is only valid if all 4 bytes are equal). Channel 0 and channel 1 shall have the same Track Set Number. The tape is always formatted in dual channel mode, and there is therefore no change in the Track ID field if the tape later is used in single channel mode.

WPC The WPC (Write Pass Count) is set zero for Track ID blocks. This assures that Track ID blocks never can be read as data blocks.

Physical Block Number The Physical Block Number in Control Bytes 6-7 begins with 0 for channel 0 and 64 for channel 1. The number is incremented for each succeeding block.

3.4.5. Data Field

The Data Field contains 512 bytes of data, encoded according to the rules in the relevant QIC Development Standard. The content of the data field depends upon the type of block being recorded:

Data Block: The datafield may contain a logical tape block header for the first block in the logical tape block. The rest of the block is available for user data. If no logical tape block header is present, all 512 bytes are available for user data.

Media Header Block: Contains drive and/or host valid information in its data field.

Filemark/Setmark/ Cancelmark: The data field contains a logical tape block header. The rest of the block is not defined by this standard and may contain vendor unique information.

Filler Block: The data field is not defined by this standard and may contain vendor unique information

Control Block: The data field is not defined by this standard, and may contain vendor unique information

ECC Block: Contains error correction characters generated by the drive.

EOD Block: The data field is not defined by this standard, and may contain vendor unique information.

Track ID Block: The first 24 bytes of the data field contains 3 ASCII fields of 8 bytes each. The first field is the Cartridge Manufacturer ID. The 2nd. field may contain a Production Date Code (option). The 3rd field may contain the Cartridge Serial Number (option). The following bytes are not defined by this standard, and may contain vendor unique information.

3.4.6. CRC Field

The CRC (Cyclic Redundancy *Code*) field consists of 4 bytes calculated over the whole data block area and control field area, starting with the most significant bit of byte 7 in the Control Field ending with the least significant bit of byte in the Data Field.

All calculations are done prior to the data randomizing and encoding. All 32 bits in the CRC character shall be set to ONE prior to the start of the CRC calculation. The generating polynomial shall be:

$$x^{32} + x^{28} + x^{26} + x^{19} + x^{17} + x^{10} + x^{6} + x^{2} + 1$$

The four bytes shall be encoded according to the rules in the relevant QIC Development Standard prior to the recording. No randomizing shall take place on the CRC characters. The most significant bits of the most significant byte shall be recorded first. It is a possibility that the two last bits of the last CRC byte both are 0's. In this case 01 shall be appended to the two 0's to enable correct encoding. After decoding, these two extra bits have no significance. See the relevant QIC Development Standard.

There is a possibility that the last two bits of the CRC are both 0 in one channel of a multiple channel format, but not in the other channel(s). This would cause problems to occur later when the channels are, per this standard, brought into synchronization on the next frame boundary. Consequently, if a 01 is appended to the last byte of the CRC on any channel, 01 shall simultaneously be appended to the last byte of the CRC on all channels. It is permissible to always append 01 to the last byte of the CRC in all channels, irrespective of the actual result of the CRC computation(s). After decoding, these extra bits have no significance and should be ignored.

3.4.7. Postamble

The postamble consists of either the fixed Normal Postamble pattern or a combination of this pattern with the Low Frequency Postamble pattern, both patterns are defined in the relevant QIC Format Standard.

The decoding system must be able to distinguish between a postamble and any possible combination of data patterns. The Postamble is recorded at the end of each block immediately following the CRC bytes.

There are three types of postamble, NORMAL, ELONGATED, and MEDIA HEADER.

Normal Postamble A Normal Postamble is defined in the relevant QIC Format Standard and shall always be recorded at the end of each block. It may be followed by either an Elongated Postamble, or it may be followed by a new preamble.

Elongated Postamble An Elongated Postamble consists of between 6600 and 7200 recordings of the Low Frequency Preamble pattern as defined in the relevant QIC Format Standard. The Elongated Postamble will always follow a Normal Postamble.

The Elongated Postamble shall be recorded before and after the EOD blocks and if possible at the end of each track

Media Header Postamble The Media Header Postamble shall consist of the high frequency postamble pattern defined in the relevant drive standard. It shall be written following the four ID Frames on Trackset 0 to either the Media Header End Position or to the Partition 0 Start Point, as defined in the relevant QIC Format Standard. See Section 6.3

Table 3.6 shows a summary of the use of preambles and post-ambles for different types of recording situations.

Recording Type	Preamble	Postamble
Normal Data Block in a fixed frame	Normal	Normal
Underrun or write termination	Normal	Normal + Elongated
Append operation	Elongated + Normal	Normal
End of servo dropout	Elongated + Normal	Normal
Last block on a track	Normal	Normal + Elongated
Media Header area	Elongated + Normal	Media Header

Table 3.6 Summary of Preambles/Postambles Recordings

4. EXCEPTION HANDLING

4.1. Block Rewrites

The relevant QIC Format standard specifies if Read-While-Write (RWW) checking of the data is required. For QIC Formats that requires Read-While-Write check, the block rewrites shall be performed as described below. QIC formats not requiring this feature may still perform RWW checking.

For write operation, a Read-While-Write test may be performed on the newly written data. Any block determined to be bad during the Read-While-Write verify operation shall be rerecorded after completion of the block being recorded when the error is detected, unless that particular block has previously been verified as good. When rewriting the bad blocks, following blocks already recorded shall also be rewritten whether they have been verified as good or not. The rewrite process must begin before any block with a physical address more than 7 higher than the block to be rewritten has been recorded on the tape.

Figure 4.1 shows typical formats resulting from rewrite operations. In 4.1A, block N-1 is bad, and this block plus blocks N, N+1 and N+2 must be rewritten (The number of blocks which must be rewritten, depends on the distance between the write and the read gap).

In figure 4.1C the bad block is N-2. When it is rewritten, the rewrite operation fails and a second rewrite must take place.

Figure 4.1E shows a case where a bad block N is rewritten due to a rewrite operation of block N-2. As both blocks N-2 and N are determined good during the last write operation, no further rewrite operations on these two blocks are required.

A bad block may be rerecorded, if necessary, up to a maximum of Max Rewrites times following the last good recording of the previous block. Max Rewrites is specified in the relevant QIC Format standard.

Blocks being rewritten shall be identical to the original blocks with the same block address and data contents etc.

During read operation, two or more good blocks with the same block address may therefore be detected by the drive. Should this happen, the drive may use the data contents of any of these valid blocks. Note, however, that the last occurrence of a block is not guaranteed to be a "good" block, as was the case in earlier formats.

It is permissible to rewrite consecutive blocks with the same block number as shown in figure 4.1H. In this circumstance, it is not necessary to rewrite a block that has previously been verified as "good".

A:

	N-3	N-2	N-1 X	N	N+1	N+2	N-1	N	N+1	N+2	N+3	N+4	N+5	N+6	

B: (different read/write head spacing)

	N-3	N-2	N-1 X	N	N+1	N+2	N+3	N-1	N	N+1	N+2	N+3	N+4	N+5	

C:

	N-3	N-2 X	N-1	N	N+1	N-2 X	N-1	N	N+1	N-2	N-1	N	N+1	N+2	

D:

	N-3	N-2 X	N-1	N	N+1	N-2	N-1	N X	N+1	N+2	N+3	N	N+1	N+2	
E:															
	N-3	N-2 X	N-1	N	N+1	N-2 X	N-1	N X	N+1	N-2	N-1	N	N+1	N+2	
F:															
	N-3	N-2 X	N-1	N X	N+1	N-2	N-1	N	N+1	N+2	N+3	N+4	N+5	N+6	
G:															
	N-3	N-2 X	N-1 X	N X	N+1	N-2	N-1 X	N	N+1	N+2	N-1	N	N+1	N+2	
H:															
	N-3	N-2	N-1	N	N+1	N+2	N+2	N+2	N+2	N+2	N+3	N+4	N+5	N+6	

Figure 4.1 Examples of possible format layout variations due to block rewrite operations. X denotes a bad block.

The writing of blocks in a new frame may start before all blocks from the previous frame have been verified. However, if one or more blocks have to be rewritten from the first frame, the rewrite operation shall follow the rules already discussed in this section and shown in Figure 4.2 See section 4.5 for rules on writing around corners and section 4.6 for rules on rewriting after servo dropouts.

At the end of a write operation (including an underrun) the last ECC block shall be rewritten until it is verified prior to writing the final postamble (Normal plus EOD Marker).

4.2. Frame Synchronization

In a tape drive which supports parallel track set recording (i.e. N > 1) each frame on each track MUST start at the same time. The implication is that if one frame of the parallel frames has a block rewrite, and comrade(s) do not, the last block of the respective frames will have to be rewritten on the other tracks to provide the synchronized start for the next frame group.

If a block in a previous frame is found to be in error after a new frame has started, the rewrite shall be handled normally in the next frame unless end of track is encountered (see section 4.5 and figure 4.2.

Channel	58	59	60	61	60	61	62	63	0	63	0	1	2	3
0			X					X						
Channel	58	59	60	61	62	63	63	63	0	1	2	3	4	5
1														

X denotes a bad block.

Figure 4.2 Example of format layout variations due to block rewrite operations at frame boundaries.

When re-writing the final block of a frame for synchronization, a data block may fail the read-while-write test. In this case the requirement to re-write may optionally be waived if the block has previously been written and verified.

4.3. Underrun

Underrun is allowed to occur only at frameset boundaries. When an underrun situation occurs, Block 63 of the last frame(s) shall be rewritten enough times to permit read-while-write verification of all blocks within the frame(s). These rewrites shall be performed in a manner which is consistent with section 4.1 of this document. After the final block has been verified, data shall be terminated with a Normal Postamble and an Elongated Postamble should be started.

If more data is available at a point between the first 12 to 5000 2-byte recordings of the Elongated Postamble, the new data may be appended on the fly by aborting the Elongated Postamble and start the Elongated Preamble for the new data. The so resulting Elongated Gap will then have the same length as if the tape actually was stopped and restarted. See section 4.4.

If no new data is available at a point between the first 12 to 5000 2-byte recordings of the Elongated Postamble, an Elongated Preamble followed by an EOD frame should be appended on the fly by aborting the Elongated Postamble and start the Elongated Preamble. The resulting Elongated Gap will then have the same length as if the tape actually was stopped and restarted. The EOD blocks should then be followed by a Normal + an Elongated Postamble (see Figure 4.3).

Last Block at end of Valid Data	Resulting Elongated Gap	EOD frame	Normal + Elongated Postamble

Figure 4.3 Elongated ambles and EOD frame

If the end of track is detected during the EOD procedure, a new EOD sequence shall be restarted at the next trackset. If however more data is available when starting at the next trackset those data can be written instead.

If End-Of-Partition is encountered, the Elongated Postamble which normally occurs at the end of a track shall be sufficient to mark the End Of recorded Data.

4.4. Append Operations at EOD

When a data frame is appended at EOD, e.g. after an underrun, the recording shall begin at a point between the first 12 to 5000 2-byte recordings of either the Elongated Postamble in front of the EOD blocks, or the Elongated Postamble after the EOD blocks. If it can be assured that all blocks belonging to the last frame are passed, the append operation should preferably start before the EOD blocks. If however servo dropouts have resulted in rewritten frames right in front of the EOD blocks it may be difficult to determine if all blocks of the last rewritten frame are passed. In this case it is safer to append after the EOD blocks. The physical block number of the first appended block will then be identical to the block number of the EOD blocks, but this is a valid situation! The append operation shall start with an Elongated Preamble followed by a Normal Preamble. See Figure 4.4

Last Block (data or EOD block)	Normal Postamble Marker	Residue of Old Elongated Postamble	Elongated Preamble	Normal Preamble

<--- Te --->

---> <--- To

Te = Minimum 12 Low Frequency Elongated Postamble Residue not overwritten by the Elongated Preamble.

To = Maximum 5000 Low Frequency Elongated Postamble Residue not overwritten by the Elongated Preamble.

Figure 4.4 Postamble/Preamble overlap after append operation.

4.5. End Of Track Operation

When EW is detected in forward or LP in reverse direction, writing should be terminated after present block. An Elongated Postamble should then preferably follow.

A frame set which cannot be completed and verified on one track set shall be rewritten in its entirety at the beginning of the following track set.

4.6. Servo Dropout Operation

The CRF1 formats relies on a servo system in order to position the head within the tolerances to the recorded track. If the servo signal drops below a certain limit, it can not be assured that the head is positioned correctly. If this happens at the time of recording, writing must be stopped immediately or neighbor tracks may be overwritten, at least partially. Special procedures are therefore necessary:

4.6.1. Servo Dropout procedure at the time of writing

1. Whenever the servo is lost, writing should be terminated immediately.

2. If the servo is locked again within the length of a physical block the write circuitry should be reenabled and no further handling is necessary since the normal rewrite operation will rewrite the interrupted block.

3. If the dropout exceeds the length of one physical block, writing should be restarted when the servo is locked again and correct trackset position is verified. In this case the writing should start with an elongated preamble and then the last not verified frameset should be rewritten.

The distance from the servo dropout to the point where a rewritten block (with valid WPC) has been verified after the dropout should be measured. If this distance exceeds the Max Servo Dropout Distance specified in the relevant QIC Format standard, a hard write error should be reported and writing should be stopped. Note that a series of repeated servo dropouts may count as one if no rewritten blocks where verified between the dropouts.

4.6.2. Servo Dropout procedure at the time of reading

In general Servo Dropouts that only occurs at the time of reading are not a problem due to the Track Set qualification of all read blocks. Blocks from adjacent Track Sets will therefore not be accepted. Normal Reread off-track procedure will assure that all recorded data can be read.

When reading a tape that had a servo dropout at the time of writing, the old data in the position of the servo dropout will normally be readable. Those datablocks will however not be accepted due to illegal WPC. When blocks are read with correct WPC reading can continue.

If no new blocks with valid WPC could be detected within the Max Servo Dropout Distance specified in the relevant QIC Format standard, a reread procedure should be started.

4.7. End Of Data detection

For non servo QIC formats, the reading of the EOD block (with valid WPC) is a safe EOD detection. For QIC formats using servo, the EOD block alone can not assure a safe EOD detection. The reason is that it might happen that a previous EOD block could not be overwritten at the time of appending, due to a servo dropout. If this was the case, writing would have been started later, but at least within the Max Servo Dropout Distance specified in the relevant QIC standard. To assure safe EOD detection the following should be done:

If an EOD block with valid WPC is read, the tape should be investigated for at least a Max Dropout Servo Distance.

If a block with valid WPC is found within the Max Servo Dropout Distance, reading should continue.

If no blocks with valid WPC is found within the Max Servo Dropout Distance, valid EOD is found.

4.8. Erase

The Crf1 format supports both logical and physical erase. Logical erase means that the data is not actually erased, but tagged as erased in the Volume Directory. It is therefore possible to undo the logical erase by turning the erase tag back again. Physical erase means that the data is actually overwritten. In order to maintain trackset information on the tape, the data will not be overwritten with constant frequency, but with non-informational blocks. Those so called Erase Filler Blocks, must have the WPC set to 1, the trackset field set to the present trackset and the type set to Filler Block. The physical blockno. does not have to be updated, and there is no need for ECC. The datafield of the physical blocks should however be filled with 00(hex). Due to the special WPC of one, those blocks can never be read as ordinary data.

4.8.1. Logical Erase from BOP

If a security erase is not essential, the fastest way of erasing the tape is to use the logical erase, or short erase as it is called in SCSI terms. This will cause only a bit to be updated in the Volume Directory (see section 6.2.2) meaning that the selected partition is erased.

4.8.2. Physical Erase from BOP

When a data should be physically erased from BOP, the Partition Entry in the Volume Directory should be updated to reflect BOP. The WPC field in the Partition Entry should however not be changed. This will assure that the next recording will use correct WPC. The tracksets for the entire partition should then be overwritten the special Erase Filler blocks with WPC set to 1.

4.8.3. Physical Erase from EOD

When a data should be erased from EOD, the recording of the special Erase Filler blocks with WPC set to 1, shall begin at a point between the first 600 to 12150 2-byte recordings of the Elongated Postamble after the EOD block. The append operation shall start with an Elongated Preamble followed by a Normal Preamble. See Figure 4.4. All the remaining tracksets in the partition should then be overwritten with the Erase Filler Block.

5. LOGICAL BLOCK TYPES

Historically QIC Format standards has used different terms for the same item, or same term for different items! The valid terms are explained in this section.

5.1. Fixed and Variable (Host) Blocks

Fixed and Variable blocks are terms that are used in SCSI documents (ref. QIC 121). Both Fixed and Variable Blocks are Logical Host Blocks. Using Fixed Blocks command is an effective way of reducing command overhead as several logical host blocks may be transferred with one command, provided they all have the same (fixed) size. When using Variable Block mode each Logical Host Block typically has a different size from the previous block, and a new SCSI command will be necessary for each new Logical Host Block.

5.2. Logical Host Blocks

For the tape format it does not matter if the host is using Fixed or Variable Block commands, the result is always Logical Host Blocks. I.e. it is not possible or desired to tell from a written tape if Fixed or Variable Block mode has been used at the time of recording!

Logical Host Blocks can not be recorded directly on the tape. They have to converted to one or more Logical Tape Blocks which again are recorded as Physical Blocks.

5.2.1. Logical Host Blocks of Size less or equal to 65536

When the logical host block size is less or equal to 64 K (65536) bytes, it will have space in one Logical Tape Block. When writing same sized logical host blocks they may at the drives discretion be grouped (and eventually compressed) and recorded as one Logical Tape Block on the tape. The total (uncompressed) size of those blocks may however not exceed 64 K (65536) bytes.

If writing a logical tape block that has a different size from the previous or the next logical host block, it is normally recorded as one Logical Tape Block.

5.2.2. Logical Host Blocks of Size greater than 65536

Logical host blocks having a transfer length greater than 64 K (65536) bytes or the maximum supported for one Logical Tape Block, shall be divided into multiple Logical Tape Blocks, referred to as a Logical Tape Block Group. Special link bits, BLBG and ELBG, provides a mean to connect the Logical Tape Blocks logically together.

5.3. Logical Tape Blocks

The Logical Tape Block (LTB) is used to hold one ore more Logical Host Blocks or partial Logical Host Blocks if their size exceeds 64536 bytes. The data may be compressed or uncompressed. The Logical Tape Block is recorded as one or up to 129 Physical Blocks. The First Physical Block will always contain a Logical Tape Block Header in the first 18 bytes in the Data Field. This header informs about the contents of the LTB like Logical Host Block size and quantity (See section 5.3.1).

5.3.1. Logical Tape Block Header

The Logical Tape Block Header consists of 18 bytes of uncompressed data placed at the beginning of each Logical Tape Block. The LTB Header occupies the first bytes in the data field, while the remaining 494 bytes are free for host data. All fields that spans more than one byte are recorded with the most significant byte as the first byte, and the least significant byte as the last byte. The LTB Header is organized as shown in Table 5.1

BYTE	Bit 7	6	5	4	3	2	1	0
0	UCMP	BLBG	ELBG	Header Length				
1	QIC compression algorithm ID							
2-3	Logical Host Block length							
4-5	Logical Host Block quantity							
6-9	Logical Host Block Address							
10-11	Net Logical Tape Block length							
12-15	Filemark Count							
16-17	Setmark Count							

Table 5.1 Logical Tape Block Header Structure

UCMP The UCMP (Uncompressed) bit indicates that the data in the Logical Tape Block is uncompressed.

BLBG The BLBG (Beginning of Logical tape Block Group) is set to 1 to indicate if this is the first or only Logical Tape Block in a Logical Tape Block Group. Linking of LTB's into Logical Tape Block Groups, can be necessary if Logical Host Blocks of size larger than 65536 are used.

ELBG The ELBG (End of Logical tape Block Group) bit is set to 1 to indicate if this is the last or only Logical Tape Block in the Logical Tape Block Group. Linking of LTB's into Logical Tape Block Groups, can be necessary if Logical Host Blocks of size larger than 65536 are used.

Header Length The Header Length indicates the length of the Logical Tape Block Header. This field is presently 18 decimal. This information should be used when reading to get the start position of the Net LTB. This will assure compatibility with future extensions of the LTB Header.

QIC Compression Algorithm The QIC Compression Algorithm contains the registered compression algorithm identifier. Refer to QIC-121 for additional details. This field is zero if no datacompression is used

Logical Host Block Length This field contains the uncompressed size, in bytes, of the host block(s) in the Logical Tape Block. If BLBG or ELBG is cleared this field contains the size of the uncompressed Partial Host Block. NOTE: a number 0 means 65536 bytes!

Logical Host Block Quantity	This field contains the number of logical host blocks recorded in one Logical Tape Block. If a logical host block is represented by multiple Logical Tape Blocks (either BLBG or ELBG is 0), the Logical Host Block Quantity is set to 1. If multiple Filemarks or Cancelmarks are recorded with one LTB, LHB Quantity specifies the no of Tapemarks. For Cancelmarks this field specifies the no. of Tapemarks that should be canceled. After the Cancelmark both the Logical Block Address and the Filemark, event. Setmark count should be reduced by the count LHB Quantity
Logical Block Address	This field contains the Logical Host Block Address of the first block in the Logical Tape Block. If a logical host block is split across multiple Logical Tape Blocks, then the Logical Block Address will be the same for all the Logical Tape Blocks within the Logical Tape Block Group.
Net Logical Tape Block Length	This field contains the number of valid bytes in the Logical Tape Block following the Logical Tape Block Header. NOTE: a number 0 means 65536 bytes for datablocks. For Filemarks, Setmarks and Cancelmarks this field is set to 1.
Filemark Count	Contains the number of Filemark recorded from the beginning of the partition up to the present position. Note: This count will be zero for the first Filemark!
Setmark Count	Contains the number of Setmark recorded from the beginning of the partition up to the present position. Note: This count will be zero for the first Setmark!

Example:

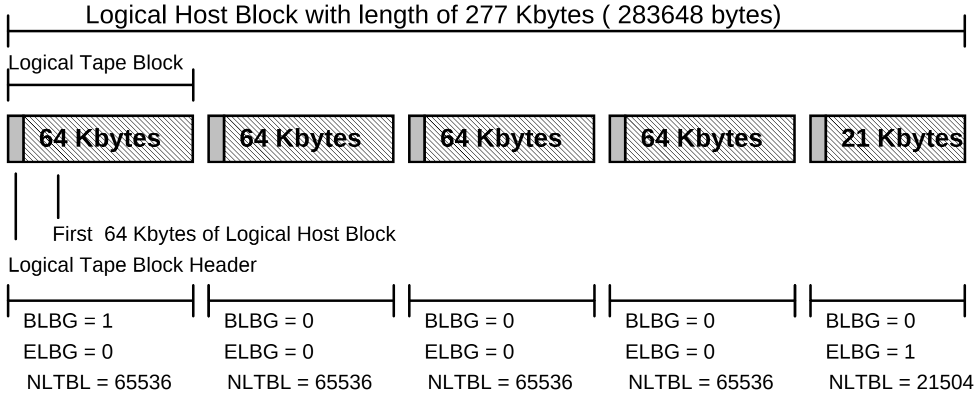

Figure 7.1 Encoding of BLBG, ELBG and Netto Logical Tape Block Length (NLTBL) in Logical Tape Block Header

5.3.2. Recording Logical Tape Blocks as Physical Blocks

A Logical Tape Block used for filemarks, setmarks or cancelmark should using the appropriate block type. Only one physical block will be needed. The first 18 bytes in the datafield will contain the LTB Header, the rest is unused.

A Logical Tape Block used for data can including the LTB Header, have any length between 19 and 65555 bytes. The physical block always contain 512 data bytes. Since the LTB normally don't fit into a complete no. of physical blocks the last block will need some padding bytes. If this is the case it will be written as a Limit 255 or Limit 511 blocktype. Otherwise the blocktype Full Data Block will be used. Special link bits, BLTB and ELTB, provides a mean to connect the physical blocks logically together. See also Section 3.4.4.1. The Physical block always contain 512 data bytes. Since the LTB normally don't fit into a complete no. of physical blocks the last block will need some padding bytes. The following sub sections illustrates by means of some examples how the Logical Tape Blocks are recorded as physical blocks.

5.3.2.1. Logical Tape Blocks, 512 Data Bytes

In this case, the logical tape block size (inclusive the header) is the same as the size of a physical block. The Block Type is set to "Full Data Block". (Bits 0-3 in Control Byte 0 are set to 0h.) All data bytes in the recorded block are valid. BLTB and ELTB in Control Byte 0 will be set to 1 for every physical block recorded, to indicate both the start and the end of a logical tape block.

5.3.2.2. Logical Tape Blocks, <256 Data Bytes

In this case, the logical tape block size (inclusive the header) is less than the size of a physical block. The Block Type is set to Limited 255 Block. (Bits 0-3 in Control Byte 0 are set to 1h.) The number of valid data bytes in the block is specified by the Valid Byte Counter, byte 511 of the data area. This byte contains a number from 1 to FFh (255Dec) depending upon the number of valid data bytes. The valid data bytes are always recorded first in the data field, followed by filler bytes which have no specified value. The Valid Byte Counter is the last byte in the data field. See Figure 5.1. BLTB and ELTB in Control byte 0 will be set to 1, to indicate the start and end of a logical tape block.

Data Field 512 Bytes		
Valid Data Bytes	Filler Bytes Unspecified	Valid Byte Counter (Byte 511)

Figure 5.1 Layout of Data Field, Limited Physical Data Blocks.

5.3.2.3. Logical Tape Blocks, 256 - 511 Data Bytes

As in the case above, the logical tape block size (inclusive the header) is still less than the size of a physical block. The Block Type is set to Limited 511 Block. (Bits 0-3 in Control Byte 0 are set to 2h.) The number of valid data bytes in the block is specified as 256 plus the number specified by the Valid Byte Counter, byte 511 of the data area. This byte contains a number from 0 to FFh (255Dec) depending upon the number of valid data bytes. The valid data bytes are always recorded first in the data field, followed by filler bytes which have no specified value. The Valid Byte Counter is the last byte in the Data Field. See Figure 5.1.

A logical tape block containing 392 bytes of valid data will therefore be recorded with Block Type Limited 511 Block and byte 511 of the data field set to 88h (136Dec). This indicates 256 + 136 = 392 valid data bytes in the block (i.e. valid data bytes from 0 to 391 in the data area). BLTB and ELTB in Control byte 0 will be set to 1, to indicate the start and end of a new host (logical) block.

5.3.2.4. Logical Tape Blocks, >512 Data Bytes in Multiples of 512.

In this case, the logical tape block size (inclusive the header) is greater than the physical block size, but the logical tape blocks are specified as Nx512 where N is an integer. Therefore all the physical blocks recorded to cover one logical tape block contain only valid data bytes.

As an example, assume a logical tape block of 2048 bytes. This will therefore require 4 physical blocks. BLTB in Control Byte 0 is set to 1 for the first physical block and 0 in the next three blocks. ELTB is set to 0 for the first 3 blocks and to 1 for the last one. The Block Type is set to Full Data Block in all four physical blocks. See Figure 5.3.

Logical Tape Block, 2048 Data Bytes			
Physical Block No. N	Physical Block No. N+1	Physical Block No. N+2	Physical Block No. N+3
Start Logical Tape Block Full Data Block BLTB = 1 ELTB = 0	Partial Logical Tape Block Full Data Block BLTB = 0 ELTB = 0	Partial Logical Tape Block Full Data Block BLTB = 0 ELTB = 0	End Logical Tape Block Full Data Block BLTB = 0 ELTB = 1

Figure 5.3 Logical Tape Block, 2048 Bytes (inclusive the header).

5.3.2.5. *Logical Tape Blocks, >512 Data Bytes, not a multiple of 512.*

In this case, the logical tape block size (inclusive the header) is greater than the physical block size, but the logical tape blocks are specified as different from Nx512 where N is an integer. Therefore the last of the physical blocks recorded to cover the logical tape block contains less than 512 bytes of valid data bytes.

As an example, assume a logical tape block of 1027 bytes. Three physical blocks will be required. BLTB in Control Byte 0 is set to 1 for the first physical block and 0 in the next two.

ELTB is set to 0 for the first 2 blocks and 1 in the last one. The Block Type is set to Full Data Block in the first two physical blocks and to Limited 255 Block for the last one. See Figure 3.10.

Logical Tape Block, 1027 Data Bytes		
Physical Block No. N	Physical Block No. N+1	Physical Block No. N+2
Start Logical Tape Block Full Data Block BLTB = 1 ELTB = 0	Partial Logical Tape Block Full Data Block BLTB = 0 ELTB = 0	End Logical Tape Block Limited 255 Block Last Byte in data field is 03h (1027 - 512 - 512 = 3) BLTB = 0 ELTB = 1

Figure 5.3 Logical Tape Block, 1027 Bytes (inclusive the header) .

Figure 5.4 shows another example, with the block specified as 1417 bytes long (inclusive the header).

Logical Tape Block, 1417 Data Bytes		
Physical Block No. N	Physical Block No. N+1	Physical Block No. N+2
Start Logical Tape Block Full Data Block BLTB = 1 ELTB = 0	Partial Logical Tape Block Full Data Block BLTB = 0 ELTB = 0	End Logical Tape Block Limited 511 Block Last Byte in data field is 89h (1417- 512 - 512-256 = 137) BLTB = 0 ELTB = 1

Figure 5.4 Logical Tape Block, 1417 Bytes (inclusive the header).

6. MEDIA HEADER

The Media Header is a small rewritable volume recorded with Media Header Frames at the beginning of trackset 0. The Media Header consists of four frames that contains information about the tape. It has two main parts, the Identifier and the Volume Directory, but also reserved areas for future extensions.

The Media Header frames are written repeatedly at the beginning of Track Set 0, until the "Media Header End Position" (MHE) is reached. At this position the high frequency Media Header Postamble follows. The Media Header Postamble is written either to the "Start Partition 0 Position" (SP0) if writing shall continue, or to the "Media Header Update Position" (MHU) if only the Media Header should be updated (see Figures 6.1 and 6.2).

The Media Header End Position, Start Partition 0 Position and the Media Header Update Position are defined in the relevant QIC Format standard.

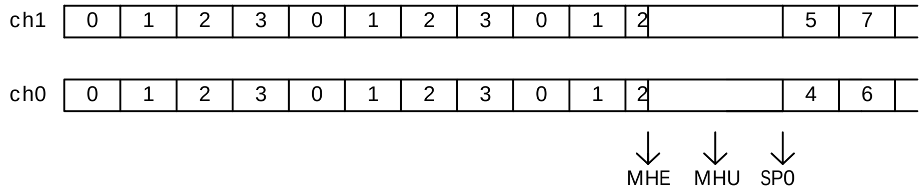

Figure 6.1: Media Header Frames and essential physical positions for dual channel format.

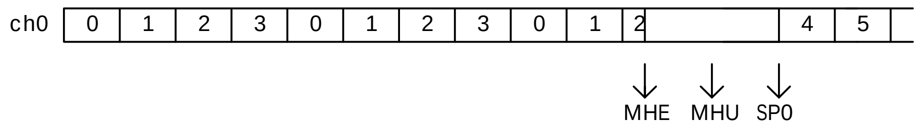

Figure 6.2: Media Header Frames and essential physical positions for single channel format

The contents of the four Media Header frames are as follows:

Identifier
(Frame 0)
The Identifier (ID) Frame. This frame contain 52 Media Header Blocks (plus the normal 12 ECC blocks). The MH blocks contain specific host, drive and vendor information and are used to specify the QIC standard recording format.

Reserved
(Frame 1)
Reserved for future extensions: Contains 00. Hex. in all bytes.

Volume Directory
(Frame 2)
The Volume Directory (VD) Field. This frame contains 52 MH blocks in which specific volume information is recorded (see Section 6.2);

Reserved
(Frame 3)
Reserved for future extensions: Contains 00. Hex. in all bytes.

6.1. Identifier

6.1.1. Media Header Block 0, Frame 0

Except for byte 64, all defined fields contain ASCII characters to describe the drive which last wrote to the tape. Unused bytes shall be filled with the ASCII space character.

The layout of Media Header Block 0 is shown in Table 6.1

BYTE	Bit 7	6	5	4	3	2	1	0
0-15	Format ID							
16-17	Format Revision							
18-19	CRF1 Revision							
20-27	Manufacturer ID							
28-43	Product Identification							
44-47	Firmware Revision							
48-63	Drive Serial Number							
64	Update	RWW	Reserved					
65-511	Reserved							

Table 6.1 Media Header Block 0 Contents.

Format ID Contain the ASCII code for the characters “QIC-XXXX-XC The actual contents is specified in the relevant QIC format standard.

Format Revision Contains the revision level of the Tape Format Standard the writing drive was designed to meet. If only one character is required, it shall be placed in byte 17 with an ASCII space character in byte 16.

CRF1 Revision Contains the revision level of the QIC-CRF1 Standard the writing drive was designed to meet. If only one character is required, it shall be placed in byte 17 with an ASCII space character in byte 18.

Manufacturer ID This field shall identify the WRITING DRIVE MANUFACTURER in accordance with table J-1 of the QIC-121 VENDOR IDENTIFICATION list.

Product Identification Product Name of the writing drive. Use of this field is at the discretion of the drive manufacturer.

Firmware Revision Firmware revision level of the writing drive. Use of this field is at the discretion of the drive manufacturer

Drive Serial Number Serial Number of the writing drive. Use of this field is at the discretion of the drive manufacturer

Update Set if the tape is recorded with firmware update information.

RWW	Set if the drive which last recorded information on the tape used Read-While-Write verification

6.1.2. Media Header Blocks 1-51, Frame 0

The data area of blocks 1 through 51 are reserved and shall be filled with 00hex.

6.2. Volume Directory

The Volume Directory keeps information that is needed by the drive to provide fast and efficient positioning of the tape. It may also contain information that will handle multiple partitions. The content of the Volume Directory is build up by the drive transparent for the host, and is not accessible for the host system.

The Volume Directory consists of four elements; the Header Field, Partition Table, Track Table, and Random Access Table. This data is recorded continuously. That is, the data is not specified as to which block contains what portion of the Volume Directory. The Header Field is recorded first, starting with byte 0 in Block 0. Following the Header Field, the Partition Table is recorded, then the Track Table, and finally, the Random Access Table. When the data field of block 0 is filled, recording continues in block 1, etc. All unused data bytes in the Volume Directory frame shall be filled with 00h. All fields that spans more than one byte are recorded with the most significant byte as the first byte, and the least significant byte as the last byte. All fields may not be supported by all formats.

Header Table (bytes 0-21)	Keeps information about Volume Directory and its organization.
Partition Table (variable size)	Keeps information about partition sizes and End Of Data.
Track Set Table (variable size)	Keeps the no. of valid RAT entries for each track set and start of track information.
Random Access Table (variable size)	Keeps relation between physical and logical tape positions.

Table 6.3 Volume Directory Layout

Note:

To avoid incompatibility with future revisions of the Volume Directory, all accesses to data within the Volume Directory should be made by indexing off the values stored in the Volume Directory Header Table.

6.2.1. Volume Directory Header Table

The Header Table is expandable and contains information about the Volume Directory and its organization. If not otherwise described, values are defined in the relevant QIC Format standard. The Header Table is organized as shown in Table 6.3.

BYTE	Bit 7	6	5	4	3	2	1	0
0-6	Key							
7	Revision							
8	Maximum Number of Partitions							
9	Active Number of Partitions							
10	Number of Channels per Trackset							
11	Start of Partition Table							
12-13	Start of Trackset Table							
14-15	Start of Random Access Table							
16	Partition Table Entry Size							
17	Trackset Table Entry Size							
18	Random Access Table Entry Size							
19	Maximum RAT Entries per Trackset							
20-21	Random Access Table Distance							

Table 6.3 Volume Header Structure

Key — This field is to authenticate the Volume Directory. If the proper key , "QIC DIR" is not present, the Volume Directory is not present.

Revision — This field is further authentication. Revision 00h indicates prototype level.

Maximum Number of Partitions — This standard supports multiple partitions. The maximum number of partitions supported is specified in this field

Active Number of Partitions — The number of partitions into which the tape is presently divided is specified in this field.

Number of Channels per Trackset — This field specifies the number of tracks in the trackset.

Start of Partition Table — This field specifies the number of bytes from the beginning of the Header Field to the first byte of the Partition Table.

Start of Track Set Table — This field specifies the number of bytes from the beginning of the Header Field to the first byte of the Track Set Table.

Start of Random Access Table — This field specifies the number of bytes from the beginning of the Header Field to the first byte of the Random Access Table.

Partition Table Entry Size This field specifies the number of bytes in each entry of the Partition Table.

Track Set Table Entry Size This field specifies the number of bytes in each entry of the Track Set Table.

Random Access Table Entry Size This field specifies the number of bytes in each entry of the Random Access Table. If the size is set to 0, this means that the RAT is not present

Maximum RAT Entries Per Track Set This field specifies the maximum number of RAT Entries pr. track set. The value is dependent on tape format and tape length used. If RAT is not supported this value shall be set to 0.

Random Access Table Distance The number of physical blocks between entry points in the RAT. This field specifies the number of the first physical block that has a RAT entry, and the distance between succeeding entries. The value to use may be dependent on tape format and tape length used. If RAT is not supported this value shall be set to 0.

6.2.2. Volume Directory Partition Table

Each partition is an integral number of track set pairs. The minimum partition is a contiguous track set pair with the even track set at the beginning of partition, and the odd track set as the end of partition. The maximum number of partitions is one-half the number of track sets. PSUM, or minimum partition granule as defined in SCSI-2 and QIC-121, is the capacity corresponding one track set pair. See QIC-121 for details regarding creation and SCSI support of multiple partitions.

Partition 0 always begins at the BOT end of trackset 0. Partition 1 begins at the BOT end of the first even trackset that is not part of Partition 0. Partitions are allocated sequentially. Note that logical addressing are independent for each partition. The Partition Table entry is organized as shown in Table 6.4.

BYTE	Bit 7	6	5	4	3	2	1	0
0	Erased	EOP	HWE	LEW	Reserved			
1	BOP Trackset Number							
2	EOP Trackset Number							
3	EOD Trackset Number							
4-7	EOD Physical Block Number							
8-11	EOD Logical Host Block Address							
12-13	Write Pass Count							
14-17	Filemark Count							
18-19	Setmark Count							

Table 6.4 Volume Directory Partition Table Entry Structure

Erased Bit set if partition is logically erased. Cleared if partition contains data.

EOP	Bit is set if End Of Partition is reached during write. The HWE bit will be set when this bit is set.
HWE	Bit set if a Hard Write Error has occurred during write. If this bit is set will EOD Phys. Block No hold the physical block number of the next block to be written after the Hard Write Error. EOD Log. Host Block No. is invalid when HWE is set.
LEW	Bit is set if Logical Early Warning is reached during write.
BOP Trackset Number	Track Set No. for Beginning of Partition
EOP Trackset Number	Track Set No. for End of Partition
EOD Trackset Number	Track Set No. for End Of Data
EOD Physical Block Number	Physical Block Number of last block in the last Data Frame before End Of Data.
EOD Logical Host Block Address	Physical Block Number of last block before End Of Data.
Write Pass Count	Write Pass Count for partition
Filemark Count	Total number of Filemarks recorded in the Partition..
Setmark Count	Total number of Setmarks recorded in the Partition..

6.2.3. Volume Directory Trackset Table

The Volume Directory Trackset Table contains information pertaining to each trackset. It consists of entries of at least 6-bytes for every track set. Some format may require up to 16 bytes. The actual supported no. of bytes is defined in the Partition Table Entry Size field in the Header Field. The Trackset Table entry is organized as shown in Table 6.5.

BYTE	Bit 7	6	5	4	3	2	1	0
0-1	Valid RAT entries from this Trackset							
2-5	Starting Physical Block Number							
6-9	Logical Host Block Address							
10-13	Filemark Count							
14-15	Setmark Count							

Table 6.5 Volume Directory Trackset Table Entry Structure

Valid Rat Entries from this Track Set	The entry represents the number of valid entries in the RAT for this trackset. A value of 0 indicates no data is recorded on the trackset.
Starting Physical Block Number.	Physical Block Number of the first block recorded on this trackset.

Logical Host Block Address

Logical Host Block Address of the next logical host block on tape. If the trackset turn is within a LTBG this field will hold the logical block number for the first block following the LTBG. This field may not be supported by all formats.

Filemark Count

The number of Filemarks recorded in this partition previous to this trackset. This field may not be supported by all formats.

Setmark Count

The number of Setmarks recorded in this partition previous to this trackset. This field may not be supported by all formats.

6.2.4. Volume Directory Random Access Table

The Random Access Table (RAT) allows for increased resolution over the Trackset Table into the relationships between Physical Addresses, Logical Addresses, and the location of Filemarks and Setmarks. A snapshot of the relationship between logical and physical addressing is taken every N physical blocks, where N is the RAT Distance defined in the Volume Directory Header Table. The RAT distance is chosen to optimize the relationship between minimizing the time required to position accurately to a logical block on tape, and the amount of memory space required to contain the RAT. Some formats may not require the Random Access Table, or not all the specified fields. The actual supported no. of bytes is defined in the Random Access Table Entry Size field in the Header Table. The structure of each RAT entry is defined in Table 6.6

BYTE	Bit 7	6	5	4	3	2	1	0
0-3	Logical Host Block Address							
4-7	Filemark Count							
8-9	Setmark Count							

Table 6.6 Volume Directory RAT Entry Structure

Logical Host Block Address

Logical Host Block Address of the next logical host block on tape. If the RAT entry is within a LTBG the entry will hold the logical block number for the first block following the LTBG.

Filemark Count

The number of Filemarks recorded in the partition up to but not including this RAT entry point.

Setmark Count

The number of Setmarks recorded in the partition up to but not including this RAT entry point.

The physical block number is implied by the entry number, the trackset number, and the RAT Distance.

1) Entry at start of Lhb :

```
BOP |----Lhb=0----|---Lhb=1---|--Lhb=2-FMK--|--Lhb=3--|---Lhb=4---|...
                  |
                  Entry : Lhb = 1, Fmk = 0, Smk = 0
```

2) Entry within Lhb :

```
BOP |----Lhb=0----|---Lhb=1---|--Lhb=2-FMK--|--Lhb=3--|---Lhb=4---|...
                       |
                       Entry : Lhb = 2, Fmk = 0, Smk = 0
```

3) Entry at start of one Ltb :

```
BOP |--Lhb=0,Q=100--|--Lhb=100,Q=200--|--Lhb=300,Q=1,FMK--|--Lhb=301,Q=5--|-Lhb=306..
                    |
                    Entry : Lhb = 100, Fmk = 0, Smk = 0
```

4) Entry within one Ltb :

```
BOP |--Lhb=0,Q=100--|--Lhb=100,Q=200--|--Lhb=300,Q=1,FMK--|--Lhb=301,Q=5--|-Lhb=306...
                           |
                           Entry : Lhb = 300, Fmk = 0, Smk = 0
```

5) Entry at Fmk :

```
...|--Lhb=100,Q=200--|--Lhb=300,Q=1,FMK--|--Lhb=301,Q=5--|-Lhb=306,Q=100-|-Lhb=406...
                     |
                     Entry : Lhb = 300, Fmk = 0, Smk = 0
```

6) Entry after Fmk :

```
...|--Lhb=100,Q=200--|--Lhb=300,Q=1,FMK--|--Lhb=301,Q=5--|-Lhb=306,Q=100-|-Lhb=406...
                                         |
                                         Entry : Lhb = 301, Fmk = 1, Smk = 0
```

7) Entry at Cancel Mark :

```
...|--Lhb=100,Q=200--|--Lhb=300,Q=1,FMK--|--Lhb=301,Q=1,CMK--|-Lhb=300,Q=100-|...
                                         |
                                         Entry : Lhb = 300, Fmk = 0, Smk = 0
```

Figure 6.7 Examples of RAT entries

6.3. Recording and Updating the Media Header

When a tape is recorded for the first time, there is no Media Header. Before any user data is recorded the tape should be initialized with an initial Media Header where all partitions are set to empty in the Volume Directory. Once data has been recorded on the tape, the Media Header must be updated prior to unloading the cartridge. This applies to all write or append operations. The requirements of Section 6.1 must be met.

When writing from BOP, the Media Header must be updated with "partition empty" information before the writing of the partition data starts.

The writing of the Media Header is terminated in two different ways. This is done to assure that media header blocks may be rewritten as any other blocks and that no conflict between old and new updated data may occur because of the rewriting. A safety zone is also specified to prevent overwriting of the start of partition 0 which always is following the Media Header on track set 0.

6.3.1. Writing from start of partition 0.

When recording from BOT, the Media Header should be appended (see section 4.4) after the Track ID blocks. When the Media Header End Position is passed, the Media Header Postamble (see Section 3.4.7) should follow. This amble shall be written until the Partition 0 Start Point, as defined in the relevant drive standard. The user data zone will begin at this point, and data will be recorded following an elongated (low frequency) preamble. See figures 6.1 and 6.2.

6.3.2. Updating the Media Header

When starting to write at other positions than EOD or the beginning of Partition 0, or following an append or write operation, the Media Header shall be updated (i.e., overwritten) so that the information about the new EOD and the search keys that was build up by the drive, can be saved on the tape.

The Media Header should be appended after the Track ID blocks the same way as described in section 6.3.1, except that writing of the Media Header Postamble shall be terminated at the Media Header Update Position as defined in the relevant drive standard. This will assure that all eventual rewritten blocks of the previous Media Header are overwritten and that writing is terminated before the start of the Partition 0. See figures 6.1 and 6.2.

7. DATA COMPRESSION

This Standard supports data compression as an optional feature. More than one method of data compression may be employed.

It is not a requirement in this Standard that the drive must support data compression to be compatible with the Standard. The Standard is designed such that drives that do support the general QIC-Drive and QIC-121 Standards but do not support data compression will be able to retrieve compressed data and transfer it to the host.

The rules for data compression are set forth below. The user should also refer to the QIC-121 for additional information.

7.1. General Rules

Only the data area in the data blocks that contain user data may contain compressed data. All other information is uncompressed. Other blocks and tape marks shall be uncompressed.

7.2. Using Data Compression

Compressed data will be grouped into Logical Tape Blocks in the same way as uncompressed data. The difference in the Logical Tape Block Header are:

The UCMP bit is cleared.

The QIC compression algorithm field will indicate the used HW compression type

The Netto Logical Tape Block Length will indicate the compressed length.

In addition the Comp bit will be set in Controlbyte 0 in all blocks in the Logical Tape Block. This is useful to determine if recovery is possible if the Physical Block containing the Logical Tape Block Header could not be read.

7.3. Configuration of drive for data Compression

According to QIC-121 MODE page 0Fh shall be used to select compression or decompression algorithm. The user should refer to QIC-121 for more details.

8. ERROR CORRECTION

The ECC blocks at the end of each frame may be used during the data read operation to reconstruct blocks in error.

Two different ECC modes are defined, ECC mode 1 and ECC mode 2. Both modes has the same errorcorrecting capabilities for statistical random errors. ECC mode 2 uses two tracks and will be able to handle larger systematic errors caused by logitudinal defects than ECC mode 1. Such defects are frequently occuring on streaming tape drives as a result of tape edge defects ore wear stripes.

Which mode that should be used is specified in the relevant tape format specification.

8.1. Mode 1: Single channel ECC

The ECC mode 1 uses frames of 64 blocks where the last 12 blocks are reserved for ECC. This mode may be used both for single - and multiple channel operation. The error correction system is based upon an interleave organization effectively separating each frame into two groups, one containing all the even numbered blocks, the other one containing all the odd numbered blocks. Each group has independent error correction. The group with the even blocks shall be referred to as the even interleave (a), while the group containing the odd blocks shall be referred to as the odd interleave (b). For each of these groups, the error correction system makes it possible to correct any combination of *s* blocks with CRC erasures (or pointers) and *t* blocks with CRC errors, as long as :

$s + 2t < 7$

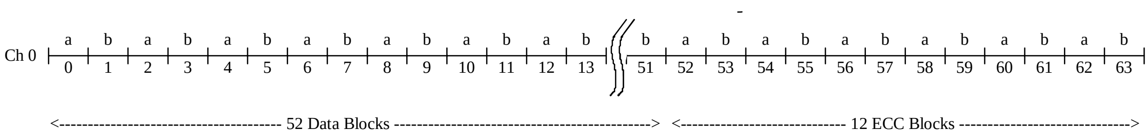

Figure 5.1 ECC Mode 1 Frame Interleave Group Format

8.2. Mode 1 Error Correction Matrix Format

A frame contains 64 blocks, 52 data blocks and 12 ECC parity blocks. Each block contains 512 data bytes and 1 Control Byte (Control Byte 0) which are covered by ECC control.

The bytes in the ECC frames are considered to be arranged in 32 blocks (rows) by 513 bytes as shown in Figure 5.2.

The even parity rows (i.e. blocks 52, 54, 56, 58, 60 and 62) shall be chosen so that each column of the even rows (blocks 0, 2, 4, ..., 48 and 50) of the matrix forms an independent Reed-Solomon codeword of redundancy six, with 8-bit characters, as shown in Figures 5.2. Similarly, the odd parity rows (i.e. 53,55, 57, 59, 61 and 63) shall be chosen so that each column of the odd rows (blocks 1, 3, 5, ..., 49 and 51) of the matrix forms an independent Reed-Solomon codeword of redundancy six. See Figure 5.3. Data shall be written on the tape row by row, starting with row 0, and within each row (i.e. block) the bytes shall be written starting with column 0. This implementation gives a very effective interleaving of the data regarding ECC, although the data itself is recorded in the normal order received from the host. This implementation ensures that the influence of any error spreading over two neighbor blocks is reduced.

	Control Byte 0	Data Bytes 1	2	...	511	512		
0							CRC	
2							CRC	
4							CRC	DATA
:								ROWS
:				:				
48							CRC	
50							CRC	
52				ECC0			CRC	
54				ECC2			CRC	
56				ECC4			CRC	PARITY
58				ECC6			CRC	ROWS
60				ECC8			CRC	
62				ECC10			CRC	

Figure 5.2 ECC Mode 1 Frame Format, Interleave a (Even Blocks)

	Control Byte 0	Data Bytes 1	2	...	511	512		
1							CRC	
3							CRC	
5							CRC	DATA
:								ROWS
:				:				
49							CRC	
51							CRC	
53				ECC1			CRC	
55				ECC3			CRC	
57				ECC5			CRC	PARITY
59				ECC7			CRC	ROWS
61				ECC9			CRC	
63				ECC11			CRC	

Figure 5.3 ECC Mode 1 Frame Format, Interleave b(Odd Blocks)

8.3. Mode 2: Dual channel ECC

The ECC mode 2 uses framesets of two frames. Each frame consists of 64 blocks where the last 12 blocks are reserved for ECC. The two frames may however be looked upon as one larger frame with 128 blocks where 24 blocks are reserved for ECC. This mode may only be used for multiple channel operation. The error correction system is based upon an interleave organization effectively separating each frameset into four groups. Each group has independent error correction. The groups are referred as group a, b, c and d.. For each of these groups, the error correction system makes it possible to correct any combination of *s* blocks with CRC erasures (or pointers) and *t* blocks with CRC errors, as long as :

$s + 2t < 7$

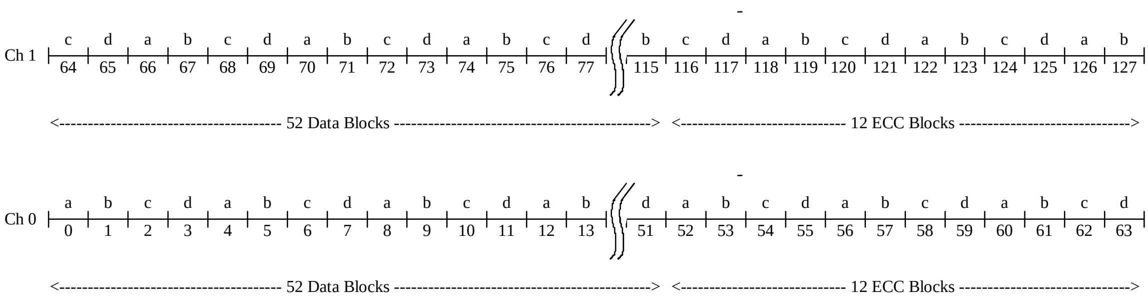

Figure 5.4 ECC Mode 2 Frameset Interleave Group Format

8.4. Mode 2 Error Correction Matrix Format

A frameset contains 128 blocks, 104 data blocks and 24 ECC parity blocks. Each block contains 512 data bytes and 1 Control Byte (Control Byte 0) which are covered by ECC control.

The bytes in the ECC frames are considered to be arranged in 32 blocks (rows) by 513 bytes as shown in Figure 5.5.

The interleave a rows (i.e. blocks 52, 118, 56, 122, 60 and 126) shall be chosen so that each column of the interleave a rows (blocks 0, 66, 4, ..., 48 and 114) of the matrix forms an independent Reed-Solomon codeword of redundancy six, with 8-bit characters, as shown in Figures 5.5. Similarly, the interleave b rows (i.e. 53,119, 57, 123, 61 and 127) shall be chosen so that each column of the interleave b rows (blocks 1, 67, 5, ..., 49 and 115) of the matrix forms an independent Reed-Solomon codeword of redundancy six. See Figure 5.6. Interleaves c and d are build in a similar way. Data shall be written on the tape two rows at a time in two channels in parallel, starting with row 0 and 1, and within each row (i.e. block) the bytes shall be written starting with column 0. This implementation gives a very effective interleaving of the data regarding ECC, although the data itself is recorded in the normal order received from the host. This implementation ensures that the influence of any error spreading over two neighbor blocks is reduced. Since the ECC information is spread over two tracks geographically spaced, this ECC mode also improves the errorcorrecting for longitudinal defects.

	Control Byte 0	Data Bytes 1	2	...	511	512		
0							CRC	
66							CRC	
4							CRC	
70							CRC	
8							CRC	
:								DATA
106							CRC	ROWS
44							CRC	
110							CRC	
48							CRC	
114							CRC	
52				ECC0			CRC	
118				ECC14			CRC	
56				ECC4			CRC	PARITY
122				ECC18			CRC	ROWS
60				ECC8			CRC	
126				ECC22			CRC	

Figure 5.5 ECC Mode 2 Frameset format, Interleave a

	Control Byte 0	Data Bytes 1	2	...	511	512		
1							CRC	
67							CRC	
5							CRC	
71							CRC	
9							CRC	
:								DATA
107							CRC	ROWS
45							CRC	
111							CRC	
49							CRC	
115							CRC	
53				ECC1			CRC	
119				ECC15			CRC	
57				ECC5			CRC	PARITY
123				ECC19			CRC	ROWS
61				ECC9			CRC	
127				ECC23			CRC	

Figure 5.6 ECC Mode 2 Frameset Format, Interleave b

	Control Byte 0	Data Bytes 1	2	...	511	512		
64							CRC	
2							CRC	
68							CRC	
6							CRC	
72							CRC	
:								DATA
42							CRC	ROWS
108							CRC	
46							CRC	
112							CRC	
50							CRC	
116				ECC12			CRC	
54				ECC2			CRC	
120				ECC16			CRC	PARITY
58				ECC6			CRC	ROWS
124				ECC20			CRC	
62				ECC10			CRC	

Figure 5.7 ECC Mode 2 Frameset Format, Interleave c

	Control Byte 0	Data Bytes 1	2	...	511	512		
65							CRC	
3							CRC	
69							CRC	
7							CRC	
73							CRC	
:								DATA
43							CRC	ROWS
109							CRC	
47							CRC	
113							CRC	
51							CRC	
117				ECC13			CRC	
55				ECC3			CRC	
121				ECC17			CRC	PARITY
59				ECC7			CRC	ROWS
125				ECC21			CRC	
63				ECC11			CRC	

Figure 5.8 ECC Mode 2 Frameset Format, Interleave d

8.5. Field Representation

GF(256) is the field consisting of 256 elements. Each field element "a" has the form :

$a = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

where each a_i is either 0 or 1. A field element "a" shall be represented by a byte as shown in figure 5.3.

MSB LSB

a7	a6	a5	a4	a3	a2	a1	a0

Figure 5.3 Bit Numbering Convention

Field math operations (addition, multiplication, division) are defined to be polynomial math modulo a_n irreducible binary polynomial of degree eight, f(x), where binary addition is the logical exclusive-or operation and binary multiplication is the AND operation. The irreducible polynomial used to generate the field GF(256) shall be:

$f(x) = x^8 + x^7 + x^2 + x + 1$.

8.6. Code Generator Polynomial

Let r be a root of f(x); in hex notation, r=2hex. The generator polynomial for the Reed-Solomon code shall be of the form:

$g(x) = (x+r^0)(x+r^1)(x+r^2)(x+r^3)(x+r^4)(x+r^5)$

$= x^6 + 3Fx^5 + 28x^4 + A6x^3 + 12x^2 + 56^x + F4$

where r = 2. This Reed-Solomon code contains the QIC-525 polynomial as a sub-code.

Encoding shall be accomplished as follows. In a full frame, each column contains data bytes d_0 to d_{63}, numbered as in figure 4.4. The parity bytes d_{52} to d_{63} in each column shall be chosen so that the two polynomials:

$$d_o(x) = \sum_{i=0}^{31} d_{62}\text{-}2ix^i$$

and

$$d_o(x) = \sum_{i=0}^{31} d_{63}\text{-}2ix^i$$

(_ = SUM)

are each divisible by g(x), using polynomial division over GF(256).

8.7. Examples of Codewords

The following columns of bytes are codewords for the polynomials defined in the preceding sections, using hex notation for the field elements.

Byte Number

Data Blocks Number	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.. 0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. 1	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. .	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
.. .	..	..	..	..	..	..	(all	zeroes)			..	..	..	..	..	..
.. 42	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. 43	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. 44	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. 45	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
.. 46	00	00	00	00	00	00	01	00	01	01	00	01	00	00	00	00
.. 47	00	00	00	00	00	00	00	01	01	00	01	01	00	00	00	00
.. 48	00	00	00	01	00	01	00	00	00	01	00	01	FF	00	FF	00
.. 49	00	00	00	00	01	01	00	00	00	00	01	01	00	FF	FF	00
.. 50	01	00	01	00	00	00	00	00	00	01	00	01	00	00	00	00
.. 51	00	01	01	00	00	00	00	00	00	00	01	01	00	00	00	00

Byte Number

Parity Blocks Number	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
.. 52	3F	00	3F	6F	00	6F	ED	00	ED	BD	00	BD	0A	00	0A	00
.. 53	00	3F	3F	00	6F	6F	00	ED	ED	00	BD	BD	00	0A	0A	00
.. 54	28	00	28	A2	00	A2	A9	00	A9	23	00	23	5E	00	5E	00
.. 55	00	28	28	00	A2	A2	00	A9	A9	00	23	23	00	5E	5E	00
.. 56	A6	00	A6	80	00	80	97	00	97	B1	00	B1	2C	00	2C	00
57	00	A6	A6	00	80	80	00	97	97	00	B1	B1	00	2C	2C	00
.. 58	12	00	12	D6	00	D6	4C	00	4C	88	00	88	77	00	77	00
.. 59	00	12	12	00	D6	D6	00	4C	4C	00	88	88	00	77	77	00
.. 60	56	00	56	7E	00	7E	53	00	53	7B	00	7B	33	00	33	00
.. 61	00	56	56	00	7E	7E	00	53	53	00	7B	7B	00	33	33	00
.. 62	F4	00	F4	E4	00	E4	CD	00	CD	DD	00	DD	C3	00	C3	00
.. 63	00	F4	F4	00	E4	E4	00	CD	CD	00	DD	DD	00	C3	C3	00

Table 5.1 Example of ECC moder 1 Codewords

9.1. APPENDIX A Write Equalization Suppression Characteristics

DATA FREQUENCY	SUPPRESSION
Fd	0dB
Fd/2	5.7dB
Fd/4	8.4dB

Table B-1 Suppression Characteristics