DRAFT 4/25/67

COMMUNICATIONS MESSAGE FORMAT

BY

A. K. BHUSHAN and R. H. STOTZ

The message format described herein is compatible with the American Standard Code for Information Interchange (ASCII) and follows the recommended standard given in "Control procedures for data communications - An ASA Progress Report," published in the February 1966 issue of the Communications of the ACM.

The message format is recommended for data communication between computing machines using serial synchronous binary transmission facilities similar to the Bell System 201, 301, and 303 series data set stations. The same format is applicable to asynchronous transmission as well. The message format may be used to communicate to GE Multics system, IBM System/360 and various other computers and time-sharing systems with suitable communication adapters and interfaces.

BASIC MESSAGE FORMAT

A typical message is illustrated in Figure 1.

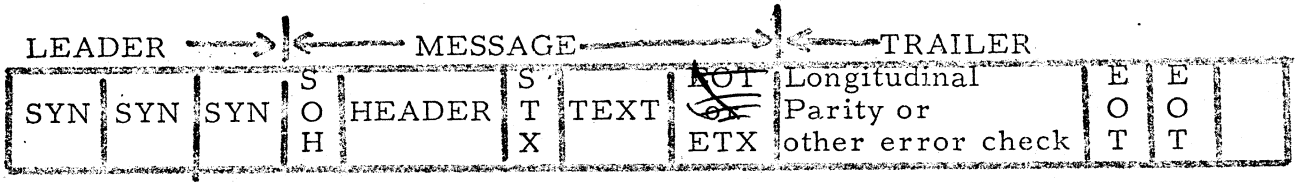

Figure 1. Basic Message Format.

will use FS + GS for binary / text separator

The following comments apply to the basic message.

1) Each character in the above message is a 7 bit ASCII character followed by one bit of odd parity. The least significant bit is transmitted first and the parity bit is transmitted last. SYN, SOH, STX, ETX, and EOT are standard ASCII control characters.

2) The leader and the trailer are always automatically supplied by communication adapter module on transmission and stipped from the message on reception. The funttion of the leader is to stabilize the line after turn on and provide the data set with correct synchronization. The trailer usually will contain longitudinal block parity (even) or other error check, and signal the end of transmission.

3) An STX may never appear without an SOH and a correct header. The STX signals the end of the header information and beginning of the user's text. If message does not contain text then either an EOT or an ETX character ends the message and header. Either an EOT or an ETX character also ends the message and text after an STX.

4) Whenever an SOH appears an EOT must appear to complete the message. After the first EOT or ETX appears after an SOH, the next character is read as a longitudinal block parity check or some other known form of error check (such as cyclic redundancy check for IBM machines). Any thing after this is not read at all till such a time that a next SOH appears.

Header Information

Information in the header of the message is machine sensible address or routing information used only to aid the communications programs to efficiently transmit, receive, sequence and route the messages. The header information is never seen by the user, and all information regarding the contents of the text portion of the message (e. g. should the ASCII characters in the text be interpreted as ASCII or binary) is contained in the text portion of the message.

The ASCII characters in the header are separated into two categories for convenience in transmission of information. The first category of characters is those which have a 'zero' as their most significant bit. These characters (octal 000 to 077) are the ASCII control characters, punctuation characters and graphic symbols. The non-control ASCII characters in this category shall be predefined and referred to as "Key" characters. The second category of characters is the remaining ASCII characters which have a 'one' as their most significant bit (octal 100 to 177), and may be interpreted as ASCII or binary (the lower order six bits representing the binary information).

The key characters are predefined and may be followed by a string of characters from the second category called the argument of the key. The argument of a particular key is interpreted by the computer in accordance with the predefined meaning assigned to the particular key. The key completely defines as to how the information contained in the argument is to be interpreted.

The following comments apply to header control characters:

1) Header control characters may be any character from the first category (i. e. control or key character).

2) ASCII numerals may not be used as key characters.

3) A Key character followed by another key character assumes the meaning of a new key character (key key),

4) The key character 'space' may be used to separate consecutive key characters if these don't represent a new key character (key key). Thus key space key will be interpreted as two different key characters.

5) If a particular key character is not recognized by the computer software (or device hardware), it is ignored together with the argument which may follow it till the next recognizable key or control character appears.

6) Key characters are not control characters and should not be used for such a purpose.

7) Control characters (ASCII) may also have arguments from the second category.

The header control characters are divided into several categories to simplify communication to small machines.

Basic Class I header control characters are:

#id	id of the message is the argument (ASCII) that follows it.
ACK id	Acknowledgement receipt of message labelled id.
NAKid	Message labelled id not received
? id	Please repeat acknowledgement of message labelled id.
ENQ	Send "who are you " message.
*n	Send messages of size n only and use only the basic class I header control characters.
!	Communications equivalent to QUIT character.

A message of size n contains no more than 2**(6+n) characters with n ranging between 0 and 9 (ASCII numerals) this means that allowable message sizes falls between 64 and 32,768 characters. Since messages can be of variable length, n serves primarily as an upper bound on the maximum size of the message. This bound is needed in setting up receive buffers in the GIOC of the Multics System.

The number of characters in a message is equal to the sum of all characters between the leader and the trailer plus the receivable characters in the leader and the trailer.

If message contains any # id in the header , then it must be acknowlddged . There can be only one STX in a message and it can never appear without an SOH and # id header. multiple STX seperated by ETB and

STX - - - - -|ETB|BP|STX| - - - -|ETX|BP|EOT

Class II Header Control characters include:

/arg The argument is interpreted as a binary number equal to the total number of characters in the message.

, arg The argument is interpreted as a check sum (or any software error check) on the header. (binary)

& arg The argument is interpreted as the memory location (binary) where the text is going.

: arg The argument is interpreted as sender's status.

; arg The argument is interpreted as sender's interpretation of receiver's status.

Class II header control characters are provided to permit unbuffered transmission between machines (i. e. machine A can send data to machine B directly into its proper location in B). This is particularly important when a large time-sharing computer communicates with a small satellite computer. In general the satellite computer has a very limited amount of memory. It is therefore desirable not to have to provide a large message buffer in this machine. If a small message buffer is provided, it means multiple calls on the time-shared computer to get a large message across which is also undesirable.

Direct transmission means that the "where to" and "number of characters in this message" information must come before the text (i. e. in the header). In addition it is desirable to send an error check on the header to insure it is is received correctly before proceeding to store the text in the wrong location.

Class III Header control characters include:

.n id	Create buffer space for message id which will be of size n.
+ id	Space has been created for nid request.
-id	Space has not been created for nid request.
=n	Make the standard message buffer size n. The default value of n is 1 (i. e. 128 characters).
> n	= n request accepted.
< n	= n request rejected.

Class III header control characters are used primarily to obviate the need for "computer processor intervention after each message. The unassigned key characters (or key key characters) may be made use of to define supplementary header information as and when needed. But ASCII control characters should be assigned meaning only in accordance with ASA recommendations.

Text Information

The text of a message always consists of a string of ASCII characters even if binary information is being sent. Binary data is sent by first inserting into the text stream an SS* (octal 032) character which indicates that the binary transmission mode has been entered. The six least significant bits of each ASCII character which follows will be interpreted as transmitted binary information and the most significant bit will always be 'one'. The normal mode of transmissionis assumed to be "pure" ASCII until the first SS appears.

* SO character may have been more appropriate but this is already used in the Multics system for a red ribbon shift.

The ASCII stream is interpreted as binary information until an EM (octal 031) character appears which is the signal to leave binary mode. The positioning of SS and EM characters in text stream is completely independent of the message itself. Thus a single message may contain both binary and ASCII information , or any combination of the two.

Characters with 'zero' in their most significant bit may occur within binary mode of transmission and be interpreted usefully. These may be

(1) Communication control characters

(2) Other control characters

(3) Information separators

(4) Key characters.

The ASCII control characters are to be used in accordance with ASA recommendations. The information separators can be used any where in the text for hierarchical separation of information.

The key characters occurring within binary mode may be used to further define the binary information if it is so desired. The meaning assigned to the key characters in text may be different from that assigned in the header.

Thus key characters may be used to define the binary information that follows it and indicate what it is about (e.g. set point, line, etc.).

4/25/67

The enclosed draft is the description of a Communications Message Format recommended for data communication between computing machines. Your comments and suggestions are solicited before we go on to adopt the format as a standard for our systems.

A. K. Bhushan
R. H. Stotz

Comments may be addressed to

Abhay K. Bhushan
Rm 807, 545 Tech Square
M.I.T. Cambridge
Mass 02139
Tel No. (617) - 864-6900 X-5857.