BELL TELEPHONE LABORATORIES
INCORPORATED

Saltzer

SUBJECT: Standard Escape Sequences for Bell System Data Terminals - (Revision) - Case 39065

DATE: FEB 5 1968

FROM: D. A. Kerr

MEMORANDUM FOR FILE

ABSTRACT

This memorandum supersedes the earlier memorandum on the same subject dated October 24, 1967. The changes are as follows:

1. The detailed structure of the basic escape sequence doctrine has been modified. The new structure is consistent with a more recent position of the USASI group working on code extension standardization.
2. Certain errors in the list of specific sequences already assigned have been corrected.

The USA Standard Code for Information Interchange (ASCII) contains a character known as Escape, whose purpose is to serve as a prefix to sequences of characters used to represent additional control functions. This memorandum recommends a standard doctrine for the utilization of such sequences in general-purpose Bell System data terminals, and makes specific assignments for certain functions which are to be introduced in the immediate future. The relationship of this doctrine to the work in the field of the USA Standards Institute is also described.

HO-3142

Copy to
All Supervision in Center 314

Messrs. G. L. Baldwin - MH
A. B. Brown - HO
A. A. Currie - HO
C. A. Dahlbom - HO
R. W. Downing - IH
R. E. Drummond - MH
R. E. Edwards - MH
G. E. Fessler - HO
M. O. Fichter - HO
F. E. Froehlich - HO
R. B. Gibson - HO
P. K. Giloth - IH
J. R. Harris - HO
R. T. Herbst - IH
B. D. Holbrook - MH
H. G. Kienzle - IH
R. F. Lane - IH
R. C. Lipton - HO
N. A. Martelloto - IH
R. C. Matlack - HO
F. R. Michael - HO
J. F. Ossanna, Jr. - MH
L. R. Pamm - HO
G. W. Phipps - HO
W. C. Ridgway, III - WH
G. W. Smith - IH
A. E. Spencer - HO
E. W. Weber - IH
R. L. Wexelblat - HO
W. R. Young - HO

THIS COPY FOR

BELL TELEPHONE LABORATORIES
INCORPORATED

SUBJECT: Standard Escape Sequences for Bell System Data Terminals - (Revision)

DATE: FEB 5 1968

FROM: D. A. Kerr

MEMORANDUM FOR FILE

(This memorandum supersedes the memorandum of the same title dated October 24, 1967. It presents a revised "intermediate-final partition" for Escape sequences which is consistent with a more recent position of USASI Task Group X3.2.4 in their work on the standardization of code extension, and it corrects errors in the list of specific sequences presented on page 4 of the original memorandum. This memorandum is, however, complete, and may be utilized without reference to the earlier document.)

Introduction

In the development of the USA Standard Code for Interchange (ASCII), decisions were made as to the number and nature of control characters to be provided. It was recognized, however, that the repertoire provided in the standard code would not be adequate for many specific applications, applications which however could benefit from a high degree of code conformity with ASCII. For this reason, the developers of ASCII devoted one control character position to the character Escape (ESC), whose current standard definition is:

> Escape - A control character intended to provide code extension (supplementary characters) in general information interchange. The Escape character itself is a prefix affecting the interpretation of a limited number of contiguously following characters.

In any given data system, or arrangement of interrelated systems, it is attractive to establish a certain degree of standardization of the usage of such escape sequences, even though the control functions thereby provided may be highly specialized and limited in their realm of interchange. Prominent among the justifications for such standardization is the need for any code-sensitive device which may be exposed to an escape sequence to be able to isolate the sequence; that is, to be able to ascertain its length in order to suppress any interpretation of the individual characters of a sequence, since such characters taken individually lose their normal meanings.

In order to investigate the criteria for the establishment of a Bell System standard doctrine for the use of escape sequences, meetings have been held periodically since 1962 among various interested parties from BTL, Teletype, and AT&T. Recently, a BTL-Teletype ad hoc group has devised such a doctrine, which is presented below. It supersedes the earlier version described in the memorandum of October 24, 1967. The principal participants in the ad hoc group were Mr. D. A. Kerr of BTL and Messrs. F. D. Biggam, J. B. Booth and D. E. Huffman of Teletype Corporation. It is recommended that, wherever practical, requirements for the use of escape sequences in Bell System data terminals should be devised pursuant to this doctrine.

Standard Doctrine

This section will briefly define the standard doctrine.

Control functions not already suitably represented as a control character of ASCII should each be represented by an escape sequence.

An escape sequence consists of the following components, in the order given:

1. The ASCII character Escape (ESC).
2. (Optional) Some number of intermediate characters. (see below)
3. One final character. (see below)

Thus, escape sequences may potentially be of any length from two characters on.

The intermediate character(s), if used, are always chosen from column 2 of the ASCII code table. They comprise SP (space) and a number of "special symbols".

The final character is always chosen from columns 0, 1 or 3 through 7 of the code table, potentially comprising the lower- and upper-case alphabetics, numerics, many special symbols, and the control characters. There are, however, restrictions on the use of certain control characters, as follows:

The following control characters shall not be used:

NUL	DLE
SOH	NAK
STC	SYN
ETX	ETB
EOT	CAN
ENQ	SUB
ACK	ESC

Use of other control characters should be made only with adequate caution regarding possible anomalous behavior should the sequence not be properly identified as such. DEL (Delete) should also not be assigned in a sequence.

Since all final characters have $b_7b_6 \neq 01$, and all intermediates have $b_7b_6 = 01$, the duration of an escape sequence can easily be delineated as commencing with ESC and extending through the next character in which $b_7b_6 \neq 01$. Should a second ESC inadvertently appear in a sequence, it should be deemed to end that sequence (since it is a final character), and initiate a new escape sequence.

In terminals which include code-sensitive devices such as printers, arrangements should be made to prevent the terminal from responding to any of the individual characters of an escape sequence, whether or not the specific sequence is itself recognized for action. This can be done by identifying the length of the sequence in accordance with the rule above, and taking appropriate action for this duration, such as:

- Diverting the characters of the sequence from the printer or other code-sensitive device.

- Placing the printer in a condition where it will not print, space, or respond to individual control characters.

It should be noted that the "non-print" condition implied by the latter method should not be confused with a "non-print" condition imposed upon a printer to suspend printing while leaving the printer's stuntbox operative. In this latter state, individual control characters not associated with printing continue to be recognized, while a printing-related function indicated by an escape sequence must be ignored.

Specific Assignments

The following specific assignments have been made by the ad hoc group for features planned for introduction into Bell System terminals in the near future:

ESC 1	HT Set
ESC 2	HT Clear
ESC 3	Print Red (or alternate graphic presentation)
ESC 4	Print Black (or normal graphic presentation)
ESC 5	VT Set
ESC 6	VT Clear
ESC 7	Reverse Line Feed
ESC 8	Reverse Half Line Feed
ESC 9	(Forward) Half Line Feed

ESC : Local Copy Off (FDX)
ESC ; Local Copy On (HDX)

USA Standards Institute Activities

USASI Subcommittee X3.2, Codes and Input-Output, through its Task Group X3.2.4, Code development, have been studying the standardization of escape sequence matters for over five years. A "proposed USA Standard Escape Sequence Procedures" has been submitted for letter ballot at the X3.2 level. A copy is attached. The proposed standard procedures are precisely consonant with the recommendations of this memorandum; however, no specific sequence assignments are included. However, there are still a number of critical and fundamental issues yet unresolved, and as a result the future course of this work is hardly clear. Prominent in the issues requiring clarification are:

- How should the use of escape sequences to represent additional functions be related to their direct representation in an 8-bit, 256-character code, if such a standard code were to evolve? (A proposed partial answer to this question is offered in the attachment.)

- Should certain functions be assigned standard sequences by the Institute, and how can this be administered?

- How does the use of SO (Shift Out) and SI (Shift In) relate to the use of ESC and to the relationship between 7- and 8-bit codes?

Extensive discussions are underway between BTL and Teletype members of X3.2 and their fellow members from other companies in an effort to determine what course of action seems most likely of acceptance by all concerned. It is, however, impossible at this time to predict the direction which this work will take or the length of time before a clear direction will emerge.

The basic principles of the doctrine proposed in this paper has borne up under extensive scrutiny both within the Bell System and within X3.2.4, and its adoption by the Bell System at this time seems entirely warranted. Success by us in its utilization may further aid its continued progress down the road to acceptance as an industry standard.

HO-3142-DAK-MR

D. A. KERR

Att.
As above

X3.2/622
X3.2.4/235
December 12, 1967

PROPOSED

USA STANDARD

ESCAPE SEQUENCE PROCEDURES

FOR USE WITH

THE USA STANDARD CODE

FOR INFORMATION INTERCHANGE

FOREWORD

(This foreword is not a part of the USA Standard Escape Sequence Procedures for Use With the USA Standard Code for Information Interchange, X3...)

The USA Standard Code for Information Interchange (ASCII - USASI X3.4-1967) provides coded representations for a set of graphic and control characters having general utility in information interchange. In some applications, it may be desirable to augment the standard repertoire of characters with additional graphic symbols or control functions.

The Code includes several special characters intended to facilitate the representation of such additional symbols or functions, a process known as code extension. Although the basic nature of code extension - providing for encoding of information beyond the standard - limits the degree to which it may be standardized, there are advantages to adherence to certain standard rules of procedure. These advantages include: minimized risk of conflict between systems required to interoperate, and the possibility of including advance provision for code extension in the design of general purpose data handling systems.

The standard procedures presented herein for the use of one of these special characters (Escape) were developed after extensive study of various potential applications and of trends expected in system design.

This standard was approved as a USA Standard by the United States of America Standards Institute on

Suggestions for improvement gained in the standard will be welcomed. They should be sent to the United States of America Standards Institute, 10 East 40th Street, New York 10016.

USASI Subcommittee X3.2, Codes and Input-Output, through which the standard was developed and processed, had the following membership at the time of approval:

(List of X3.2 member names)

The following members of USASI Task Group X3.2.4, Code Development, participated in the development of this standard:

It should be recognized that although X3.2 and X3.2.4 members are variously affiliated, work on a USASI Sub-committee or Working Group is achieved primarily on an individual and experience basis.

Contents

USA STANDARD

ESCAPE SEQUENCE PROCEDURES

FOR USE WITH

THE USA STANDARD CODE

FOR INFORMATION INTERCHANGE

1. Scope

This standard specifies a set of procedures for the representation, by sequences of ASCII* characters, of graphic symbols or control functions not directly represented in ASCII which may be required for a specific application or system.

* USA Standard Code for Information Interchange

2. General

2.1 The Characters provided in ASCII for code extension purposes are:

SO Shift Out
SI Shift In
ESC Escape
DLE Data Link Escape

2.2 This standard deals only with the use of Escape.

2.3 A brief discussion of the use of the remaining code extension characters will be found in the appendix.

2.4 The promulgation of these standard procedures is in no way meant to deprecate the use of other code extension procedures, so long as the implications of such usage upon system compatibility are recognized. (See also Appendix section A2.4)

2.5 The ASCII code table is shown for reference in the Appendix as Figure A1.

3. Standard Escape Sequence Procedures

3.1 The character ESC (Escape) is used as a prefix to a sequence of one or more additional ASCII characters used to represent a control function or graphic symbol not directly represented within the code.

3.2 An escape sequence is considered to include its associate ESC character, and its length is defined accordingly.

3.3 Such sequences - known as "escape sequences" - should not be used to represent additional communication control functions. (See Appendix section A5)

3.4 This standard provides a uniform method for the definition of code extension sequences of any length (two characters or greater).

3.5 The means of marking the end of a sequence depends upon division of the characters of the code into two classes, known as "intermediate" and "final" characters respectively.

3.6 A standard variable length sequence begins with ESC, continues, if necessary, with any number of "intermediate" characters, and invariably ends with one "final" character. Two-character sequences, therefore, contain no "intermediate" characters, but consist of ESC followed by one "final" character.

3.7 The intermediate characters are those in column 2 of the ASCII code table, that is, space and certain special graphics.

3.8 The final characters are those in columns 0, 1, and 3 through 7 of the code table, that is, the control characters, the alphabetics, the numerics and several of the special graphics, except as noted in section 3.9.

3.9 The following characters should be excluded from assignment in escape sequences: (See Appendix section A3.3)

Designation	Name	Code Table Column/Row
NUL	Null	0/0
SOH	Start of Heading	0/1
STX	Start of Text	0/2
ETX	End of Text	0/3
EOT	End of Transmission	0/4
ENQ	Enquiry	0/5
ACK	Acknowledge	0/6
DLE	Data Link Escape	1/0
NAK	Negative Acknowledge	1/5
SYN	Synchronous Idle	1/6
ETB	End of Transmission Block	1/7
CAN	Cancel	1/8
SUB	Substitute	1/10
ESC	*Escape	1/11
DEL	Delete	7/15

* Except as first character.

APPENDIX

(This appendix is not a part of the USA Standard Escape Sequence Procedures for use with the USA Standard Code for Information Interchange, but is included to facilitate its use.)

Contents

A1. Introduction

This appendix to the USA Standard Escape Sequence Procedures for Use with the USA Standard Code for Information Interchange contains a discussion of the objectives, criteria, and other considerations that were used in the development of the standard, as well as supplementary information to facilitate the effective application of these procedures.

A2. General

A2.1 Background

In the establishment of a general purpose code such as the USA Standard Code for Information Interchange (ASCII), or its international counterpart, the ISO 7-bit code, a fundamental decision must be made as to the size of the code. In making such a decision there is usually a conscious effort to avoid the most obvious problems with a code which is either too large or too small. Should the number of characters included be too small, many individual users will find their needs not accommodated, and will be forced to adopt "parochial" codes for their applications. Should the number of characters be too large, many potential users will find the standard code disproportionately costly to implement, or untenably inefficient in transmission or storage, and will again be driven to the use of some other code. Thus, either extreme in code sizing will reduce the generality of application of the code, defeating the very purpose of standardization in this field.

The 7-bit size (128 characters) adopted for ASCII is thought to be near optimum at present with respect to the above considerations. Nevertheless, there will doubtlessly be numerous applications with requirements that are not accommodated by a code of this size, or at least not by the specific characters assigned within it. Still, it is hoped that many of these applications can be served by the use of the standard code augmented in some appropriate manner. Through such an approach, the user may be able to implement much of his system with standard hardware or software. More significantly, perhaps, he will thereby be able to retain compatibility with other systems for the interchange of that information which can adequately be directly represented by the standard code.

The concept of augmenting the standard code for such purposes may be spoken of in a generic way as "code extension".

A2.2 Standardization of Procedures

The codes with which we are concerned contain four characters whose definitions indicate their relationship to code extension. They are:

SO (Shift Out)
SI (Shift In)
DLE (Data Link Escape)
ESC (Escape)

The use of these characters is not treated in detail in the code standards. Actually, the very nature of code extension inherently limits the degree to which standards for it may be constructed: it is a means of operating "beyond the standard". Nevertheless, there are several advantages to establishing a standard general procedure. This standard does so with respect to the use of Escape.

First, such standardization can prevent undesirable conflict between independently contrived applications of code extension. For example, a code extension procedure used by a data communication terminal device should be inherently free from any hazard of conflict with a code extension procedure used in a communications system which may be called upon to serve the terminal.

Second, the availability of such standards can provide guidance to system designers to facilitate the advance inclusion of general provisions for code extension operations in information handling equipment.

A2.3 Application of Standard Procedures

The standard procedures are directed at the application of code extension to those portions of a system where the use of the standard code itself would ordinarily be appropriate; that is, in what is spoken of as "information interchange".

Naturally, there are other functions within many information interchange systems for which an extremely unusual usage of the standard code, or some entirely different representation of information (e.g., the points of a character matrix), may be entirely appropriate. Such functions are often thought of as being internal to some autonomous sytem component. Just as the code standard is not presumed to be appropriate for such functions, it is not presumed that the procedures of this standard are appropriate for them.

A2.4 Related Approaches

The suggested procedures presented here for code extension should in no way be considered to deprecate the practice of using sequences of graphic characters to represent machine instructions, graphic characters not otherwise available, and so forth. Programming languages used in data processing, for example, are based upon such an approach.

A2.5 ASCII

Figure 1 shows the USA Standard Code for Information Interchange and is provided for reference. The code consists of two general categories of characters, graphics and controls. There are 32 controls, 95 graphics, and the character DEL (Delete) which in reality is neither. The 95 graphics include both upper and lower cases of the Latin (often called "Roman") alphabet, the Arabic numerals 0 to 9, a number of punctuation marks and special symbols, and SP (space), the "nonprinting graphic".

A3. USE OF ESC (ESCAPE) SEQUENCES

A3.1 Sequence Length

A3.1.1 In order that an Escape sequence may invariably be identifiable as such, each such sequence begins with the prefix character ESC (Escape), which has no other use.

A3.1.2 It was at one time proposed that code extension sequences should be standardized as always consisting of ESC and a single-following character. While this would be adequate for many applications, there are a number of considerations which may make longer sequences desirable in many cases. One such consideration is just that of having an adequate number of sequences available for the functions required in one system, or in a number of systems requiring nonconflicting function representations. Another consideration is that it is sometimes desirable to represent a critical function by a long sequence to gain security against accidental or malicious operation. A third consideration is the desire, in some systems, to have a mnemonic relationship between the character sequence and the designation of the function to be controlled.

A3.1.3 In many systems it is very useful to have a doctrine which allows sequences of various lengths to coexist in the same system.

Paramount among the requirements for a variable-length doctrine is the need to have a simple means for a device to determine the end of each sequence which it receives: that is, how many of the characters following ESC are associated with it. This is necessary so that the device may avoid giving the normal interpretation to individual characters of a code extension sequence, even when the specific sequence is not to be recognized and acted upon.

A3.1.4 The procedures of section 3 provide this flexibility without requiring the use of an "ending" character in each sequence, which carries no other information.

X3.2/622
X3.2.4/235
December 12, 1967

A3.2 Partition of the Code

A3.2.1 There are a number of criteria which affected the way in which the characters of the code was divided into "intermediate" and "final" groups. Among the significant ones were:

1. "Intermediates" should be distinguishable from "finals" by a simple logical test, preferable by the sense of 1 bit in the coded representation.

2. A given class of character, such as alphabetic, numeric, etc., should be entirely within one group.

3. Upper- and lower-cases of any specific alphabetic character should be in the same group. This allows a system designer to assign sequences so that no distinction is made on the basis of case, if desired.

4. A number of 2-character (i.e., ESC-plus-one-"final") sequences should be available which use only letters or numerals, because such sequences are convenient for use by humans.

5. The "final" group should contain some characters which are likely to occur with reasonable frequency in a stream of data. This ensures that, should the legitimate final character of a sequence be lost or mutilated, the system will soon be restored to its normal mode of character interpretation.

6. The structure should admit the use of reasonable mnemonic relationships.

A3.2.2 These criteria led to partition of the code into "intermediate" and "final" characters is as follows: (see section 3.6)

Columns 0, 1 and 3 through 7 of the code table contain final characters. $(b_7, b_8, b_5) \neq (0, 1, 0)$

Column 2 of the code table contains intermediate characters $(b_7, b_6, b_5) = (0, 1, 0)$

(See A 3.3 below, for restrictions)

This partition is felt to produce the most useful balance in the degree to which these criteria are satisfied.

A3.3 Restrictions

The restrictions of section 3.9 were imposed in order to avoid certain potentially serious problems.

A3.3.1 The ten communication control characters should never be used in Escape sequences. Such use could cause interference with the control logic of communication systems through which the data may be passed, unless the systems were arranged to detect the sequences and determine their lengths, an unnecessary burden. These ten characters are:

Designation	Name	Code Table Column/Row
SOH	Start of Heading	0/1
STX	Start of Text	0/2
ETX	End of Text	0/3
EOT	End of Transmission	0/4
ENQ	Enquiry	0/5
ACK	Acknowledge	0/6
DLE	Data Link Escape	1/0
NAK	Negative Acknowledge	1/5
SYN	Synchronous Idle	1/6
ETB	End of Transmission Block	1/7

Also, additional communication controls should not be represented by ESC sequences, but rather by DLE sequences, as described in section A5.

A3.3.2 The following characters also should not be assigned in Escape sequences:

Designation	Name	Code Table Column/Row
NUL	Null	0/0
CAN	Cancel	1/8
SUB	Substitute	1/10
ESC	Escape*	1/11
DEL	Delete	7/15

A3.3.2.1 NUL is excluded due to the hazards associated with the lack of clearly established conventions for its use and because some systems may be unable to process this character.

A3.3.2.2 CAN is excluded since its purpose is to "cancel" a portion of the data, and may thus appear abruptly in a stream of data and may even be used to "cancel" an Escape sequence.

A3.3.2.3 SUB is similarly excluded because it may be used to replace a character determined to be in error, and may thus unpredictably appear in an escape sequence as a result of this process.

A3.3.2.4 ESC is excluded to avoid confrontation with the paradox created by its definition as a "final" character: after the first ESC of a sequence another would mean at once that the sequence was starting and ending. It therefore seems better to avoid this problem than to become dependent upon specific resolutions of it in equipment.

A3.3.2.5 Finally, DEL is excluded because in some systems it may unpredictably appear as a result of correction of operator errors in perforated tape, and because some portions of a system may "delete" this character from the data stream.

---

* Except, of course, as the first character.

A3.3.3 The use of the remaining control characters in any ESC sequence should be avoided whenever possible, due to the effects which they may cause if the ESC is lost or mutilated or is not recognized for some other reason. (See also A3.4)

A3.4 Anomalies

A sequence should always be considered ended if a final character is recognized, whether or not the sequence is recognized and regardless of whether the final character is an "allowable" one in escape sequences. If that final character is ESC it should preferably be considered to start a second sequence.

A4. SO (Shift Out) and SI (Shift In)

A4.1 SO and SI are defined in USA Standard X3.4-1967 as:

SO (Shift Out) : A control character indicating that the code combinations which follow shall be interpreted as outside of the character set of the standard code table until a Shift In character is reached.

SI (Shift In) : A control character indicating that the code combinations which follow shall be interpreted according to the standard code table.

No further doctrine for their use has yet been standardized.

A5. Code Extension For Communication Controls: Use of DLE (Data Link Escape)

Standardization of specific procedures for the use of DLE falls within the jurisdiction of USASI Task Group X3.3.4, Communication Control Procedures. The subject is discussed here only to show

the relationships of this use to other aspects of code extension.

It is necessary for a communication system to be able to readily distinguish between the communication controls which are of concern to it and other controls with which it is not concerned. The assignment of specific communication control characters in the code provides this distinction under ordinary circumstances. It is necessary that this distinction be preserved when additional controls of one type or the other are represented by escape sequences. The character DLE (Data Link Escape) is provided for use in lieu of ESC as the first character of sequences used to represent additional communication controls. Thus, communication link control logic may ignore ESC entirely, passing it and the characters which follow as any other "text" characters. Code extension sequences of concern to the communication control logic can inveriably begin with DLE, to which the logic may be made sensitive.

The prohibition previously expressed against the use of communication control characters in ESC sequences is intended to prevent direct interference with the communication control logic.

USA STANDARD CODE FOR INFORMATION INTERCHANGE
USAS X3.4-1967

Bits b_7 b_6 b_5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4	b_3	b_2	b_1	Row ↓ Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	-	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	_	o	DEL