DEC STANDARD 165 REV. A

STD FOR DOC. SYM-BOLOGY

TITLE: Standard for Documentation Symbology

ABSTRACT: This standard defines character names, special key names, and notation conventions that are to be used in user documentation.

DATE	ECO #	AUTHOR	APPROVED	REV	SEC	PAGES
21 Sep 78	----	S. Porada	C. Noelcke	A	-	ALL

Size	Code	Number	Rev.
A	DS	EL00165-00	A

INDEX

PAGE REVISION CONTROL

PAGE NO.	PAGE REVISIONS	PAGE NO.	PAGE REVISIONS
1	A		
2	A		
3	A		
4	A		
5	A		
6	A		
7	A		
8	A		
9	A		
10	A		
11	A		
12	A		
13	A		
14	A		
SEC. REV	-	SEC. REV	
STD REV	A	STD REV	

DEC STANDARD 165

Standard For Documentation Symbology

December 1977

Publications Standards Subcommittee

ABSTRACT

This standard defines character names, special key names, and notation conventions that are to be used in user documentation.

1.0 INTRODUCTION

1.1 Goals

- To promote documentation compatibility in the following areas:

    1. Character Naming - standard names for all printing characters, to be used when the character itself must be singled out for attention.

    2. Special Key Representation - standard graphic representation of special keys and multiple key combinations; standard names for these keys and representations.

    3. Notation Conventions - standard graphic representation of notation in command strings and other examples of computer input/output.

- To define a standard usage of symbols and names that is meaningful to customers.

- To ensure consistency in our documentation

1.2 Scope

This standard applies to all departments that are involved in the creation or production of manuals. These departments include, but are not limited to:

- Software Publications
- Software Development
- Production Groups
- Software Services
- Educational Services
- Software Quality Management
- Software Product Management
- Technical Publications

1.3 History of Previous Standardization Efforts

Prior to its approval as a DEC Standard, this standard was a policy of Software Publications.

1.4 Related Current Standards

- Standard for Indexes, Appendixes, Running Heads, and Section Numbering for Software Documentation Manuals: DEC STD 118.
- Format Standard for Manuals Produced on Typeset Media: DEC STD 124.
- Standard for Updating Software Manuals: DEC STD 143.
- Format Standard for Manuals Produced on Non-Typeset Media: 005-003-026-nn.
- Keyboard Standard: DEC STD 107
- ANSI X3.4-1977 ASCII and DEC STD 051
- ANSI X3.32 Graphic Representation of the Control Character of ASCII
- ISO 646 7-bit coded Character Set

1.5 Future Standards Activities

This standard will be updated as needed.

2.0 CONFORMANCE

Upon approval of this standard, writing and production groups will conform when creating new manuals. No manual already existing should be redone simply to conform to this standard.

Documents being partially updated should remain consistent with their present symbology.

If no symbology exists in the manual being modified and the changes introduce symbology, that symbology should be in conformance with this standard.

3.0 DEFINITION OF THE STANDARD

The definition of the standard is divided into three parts:

1. Character naming
2. Key naming
3. Notation conventions

The definition of the standard follows.

PART I

CHARACTER NAMING STANDARDS

This part states the appropriate term to refer to a character within the text. For all non-printing characters, the names used are the character names given in the international code standard ISO 646. When referring to a key or its action, follow Part II. Thus Carriage Return is the character name; RETURN is the name of the key.

The names of the characters identify the characters; the names do not identify the functions of the characters.

Character	Name
	Space
.	Period - when used in any sense but arithmetic.
	Example: In any command string, filename and file type are separated by a period, as in FILE01.MAC.
	Radix point - when used with non-decimal numbers in an arithmetic sense.
	Example: The octal number is expressed as nnnn, with an implied radix point; e.g., the number 1234 expresses the value 1.234.
	Decimal point - when used with decimal numbers in an arithmetic sense.
	Example: The number is expressed as nnnn, with an implied decimal point; e.g., the number 3141 expresses the value 3141.00.
,	Comma
/	Slash
?	Question mark
:	Colon
;	Semicolon
`...' or '...'	Single quotation marks
	Example: A quotation within a quotation is enclosed in single quotation marks.
`	Left single quotation mark - when used to reference the mark itself.

Example: To illustrate a quotation within a quotation, enter a left single quotation mark, the text, and then a right single quotation mark.

' Right single quotation mark - when used to reference the mark itself.

Example: To illustrate a quotation within a quotation, enter a left single quotation mark, the text, and then a right single quotation mark.

' Single quotation mark - when used to reference the mark itself in a non-grammatical sense.

Example: Strings preceded and followed by a single quotation mark are stored right justified.

Apostrophe - when used in a grammatical sense.

Example: The possessive case of the noun 'person', expressed as person's, is formed by adding an apostrophe and an s to the noun.

` Grave accent

"..." or "..." Quotation marks

Example: A quotation is enclosed in quotation marks.

`` Left quotation mark - when used to reference the mark itself.

Example: To illustrate a quotation, enter the left quotation mark, the text, and then the right quotation mark.

'' Right quotation mark - when used to reference the mark itself.

Example: To illustrate a quotation, enter the left quotation mark, the text, and then the right quotation mark.

" Quotation mark - when used to reference the mark itself.

Example: To enter literal text, type a quotation mark followed by the text.

! Exclamation point

@ At sign

Example: An indirect file is indicated by typing an at sign (@) before the name.

#	Number sign
$	Dollar sign
%	Percent sign
¢	Cent sign
&	Ampersand
*	Asterisk
(	Left parenthesis
)	Right parenthesis
[	Left square bracket
]	Right square bracket
_	Underline
-	Hyphen - when used in any sense but arithmetic.

Example: Parameters in the statement are separated by hyphens; e.g., CALL arg1-arg2-arg3.

Minus sign - when used in an arithmetic sense.

Example: Use of a minus sign in a statement implies that subtraction is to be performed; e.g., 'arg1-arg2' implies that the value of arg2 is to be subtracted from the value of arg1.

—	Em dash
–	En dash
+	Plus sign
=	Equal sign
≠	Not-equal sign
{	Left brace
}	Right brace
←	Back-arrow
^	Circumflex
↑	Up-arrow

↓ Down-arrow

→ Front-arrow

< Left angle bracket - when used in any sense but arithmetic.

Example: In RUNOFF, a word preceded by a left angle bracket is printed as all capital letters; e.g., <dec is printed as DEC.

Less-than sign - when used in an arithmetic sense.

Example: Inequality can be expressed using the less-than sign; e.g., the term arg1<arg2 means that the value of arg1 is less than the value of arg2.

> Right angle bracket - when used in any sense but arithmetic.

Example: In RSX-11D, the right angle bracket is typed as a prompt to indicate that the system is ready to receive typed input; e.g., the typeout MCR> indicates that MCR input can be typed.

Greater-than sign - when used in an arithmetic sense.

Example: Inequality can be expressed using the greater-than sign; e.g., the term arg2>arg1 means that the value of arg2 is greater than the value of arg1.

≥ Greater-than-or-equal sign

≤ Less-than-or-equal sign

\ Backslash

| Vertical line

~ Tilde

£ Pound sign (English currency)

¤ Lozenge

¤ Currency symbol (International)

PART II

KEY (FUNCTION) NAMING STANDARDS

This part provides the appropriate symbol to indicate that a particular terminal key is to be pressed. These symbols are mainly used in command line descriptions and examples. The symbols in this part have been chosen to correspond to the characters found on the keys. Therefore, these symbols conform to the DEC Keyboard Standard (DEC STD 107) in preference to ISO 646-1973 or ANSI X3.4-1977.

These symbols are to be used in printed manuals. In documents where these symbols are not feasible (e.g., drafts, functional specs), these symbols are to be enclosed in angle brackets instead of being contained in an oval.

KEY	SYMBOL FOR USE IN PRINTED MANUALS	SYMBOL FOR USE IN DRAFTS	MEANING AND EXAMPLE
RETURN	(RET)	<RET>	Wherever possible, use of the RETURN key should be implied rather than expressed. For example, it is advisable (when true) to state that "all user input must be terminated by pressing the RETURN key, except where otherwise indicated." In cases where it is not obvious that the RETURN key must be pressed, the symbol should be used. Example: .R PIP (RET)
LINE FEED	(LF)	<LF>	When the LINE FEED key must be pressed, use the symbol to denote the fact. Example: INSERT ARG1 (LF) ARG2
CTRL	(CTRL/x)	<CTRL/x>	Whenever the CTRL key and another key must be pressed at the same time, denote the combination as CTRL/x. Example: At the end of the input text, enter a CTRL/Z to indicate the end-of-file. For example: ...At THE END (CTRL/Z)

TAB	(TAB)	<TAB>	When the TAB key must be pressed, use the symbol to denote the fact. Do not delimit the symbol with spaces unless they are meant to be entered as spaces. Example: NAME (TAB) ADDRESS
DELETE (Rubout)	(DEL)	<DEL>	When the RUBOUT or DELETE key must be pressed, use the symbol to denote the fact. However, do not use the symbol when showing actual echoed output at the terminal. Example: RNU (DEL) (DEL) UN echoes as RNU\UN\UN.
ESC (Altmode)	(ESC)	<ESC>	When the ESC or ALT key must be pressed, use the symbol to denote the fact. Example: EX (ESC) (ESC)
FF	(FF)	<FF>	When the FF key must be pressed, use the symbol to denote the fact. Example: END OF PAGE (FF)
SHIFT	(SHFT/x)	<SHFT/x>	Whenever the SHIFT key and another key must be pressed at the same time, denote the combination as SHFT/x. Example: To type the $ character, press (SHFT/4)

SPACE	SP	<SP>	Wherever possible, use of the SPACE key should be implied rather than expressed. If a need arises to indicate that the space bar must be pressed, or that one or more spaces must be entered, use the symbol to denote each space. Example: .R SP PIP SP SP NOW
VT	VT	<VT>	Whenever the VT key must be pressed, use the symbol to denote the fact.

Note that documentation describes keys as being pressed, not depressed or hit.

PART III

NOTATION CONVENTIONS STANDARDS

This part illustrates the symbols to be used in a command line when certain items are required, variable, optional or a choice of the user. In addition, this section discusses how to differentiate between user input and machine output.

Symbols	Meaning
[ ]	Characters enclosed in these special square brackets indicate optional information that can be omitted from a command string without affecting the basic meaning. These special brackets are chosen to avoid confusion with the standard ASCII characters [and]. When these symbols are used to denote options, the corresponding default options must be described in accompanying text.

Example: .R SOS-file.typ [/NOBAK]

If the /NOBAK switch is omitted, a backup file is created for the file specified by file.typ.

This symbol is to be used in printed manuals. In documents where this symbol is not feasible (e.g., drafts, functional specs), the equivalent symbol can be created by typing an upper-case I over a bracket.

Example: .R SOS-file.typ [/NOBAK]

{ } Characters enclosed in braces indicate a required choice; i.e., any parameter enclosed in braces can be used with a particular command, with different effects (which must be explained in accompanying text).

Example: .LOAD prog/AT: {TOP}
{BTM}

where TOP = load program from top of memory downward.

BTM = load program from bottom of memory upward.

This symbol is to be used in printed manuals. In documents where this symbol is not feasible (e.g., drafts, functional specs), a set of braces one line in height enclose the entire list of required choices.

Example: .LOAD prog/AT: {TOP
BTM}

lower-case letters — Lower-case characters presented in a command string indicate variable information to be supplied by the user. The possible range of values must be supplied in accompanying text.

Example: #LPn:*.*=DKn:*.*

where n is a unit number (0 or 1 for LP, 0 through 7 for DK).

UPPER CASE LETTERS — Upper-case characters presented in a command string indicate fixed (literal) information that must be entered exactly as shown. Note that fixed information (e.g., commands, switches) should not be abbreviated in a command string, especially in a batch command string.

Example: .R PIP

Underline (_) — Characters underlined in an example denote those characters that the user entered; the user must assume that characters not underlined constitute computer output. [Note: In some cases, for example, editing programs, the ratio of user input to machine output may be very high, causing readability problems when underlining. In such cases, it is acceptable to underline machine output, but that fact must be called out in accompanying text.]

Example: The command used to kill a page of a file using TECO is:

*HK

Contrasting Colors — When a manual is printed in two colors, the color that is not used for the textual material is considered the contrasting color. In examples, the contrasting color denotes those characters that the user entered; the user must assume that in an example the information not in the contrasting color constitutes computer output. [NOTE: When using contrasting colors, do not use the underline convention.]

Example: The command used to kill a page of a file using TECO is:

*HK In case you can't tell, "HK" is printed in the contrasting color.