18973

RFC #570
NIC 18973
UPDATES: RFC 216
OBSOLETES: None
REFERENCES: RFC 216, RFC 398

Postel

1973

EXPERIMENTAL INPUT MAPPING

BETWEEN

NVT ASCII AND UCSB ONLINE SYSTEM

30 October 1973

John R. Pickens
Computer Systems Lab
University of California
Santa Barbara

RECEIVED AT NIC NOVEMBER 28, 1973.

Contents

		page
I.	Motivation	1
II.	Overview	1
III.	Effects on Previous Specification	3
IV.	Details	3
	A. Details of One-to-One Mapping	3
	B. Details of Two-to-One Mapping	6
V.	New or Revised SERVER TELNET Options	8
	A. OPERATOR/ALPHABET Option	8
	B. SHIFT/UNSHIFT Option	8
	C. INTERPRETLF/IGNORELF Option	9
VI.	Conclusions and Suggestions for Easy Usage	9

Figures

Figure 1.	OLS Keyboard	2
Figure 2.	One-to-One Mappings	4
Figure 3.	One-to-One Operator Mode Mappings, Expanded	5
Figure 4.	Visual Representation of One-to-One Operator Mode Mappings.	5
Figure 5.	Two-to-One Mappings	7
Figure 6.	Summary of SERVER TELNET commands	10

I. MOTIVATION

Numerous requests have been received in the last six months to improve the human interface to the UCSB On-Line System (OLS) for network users. The problem stems from having to enter as many as six keys to describe a single OLS key. For example, ;PRED [SP] must be entered to specify the [PRED] key (See RFC 216, Telnet Access to UCSB's On-Line System). One solution to the problem has been to obtain and to use the custom OLS keyboard. However, this implies a capital outlay which is not always justifiable for users with tight budgets. Thus the following solution is offered as a compromise between the expense of a hardware keyboard and the awkwardness of spelled out key mappings.

II. OVERVIEW

The OLS keyboard, pictured in Figure 1, contains 168 meaningful keys. ASCII, the network standard for keyboards, allows only 128. Thus the mapping described in this report relies on one-to-one mappings for the most common keys and two-to-one mappings using [ESC] for less commonly used keys. Key assignments in the former case are picked so that location is approximately the same as pictured in Figure 1. Key assignments in the latter case are picked, wherever possible, to be suggestive of function. In addition, options exist to reenable upper and lower case alphabetics and to ignore line feeds after carriage returns.

CURRENT OLS KEYBOARD

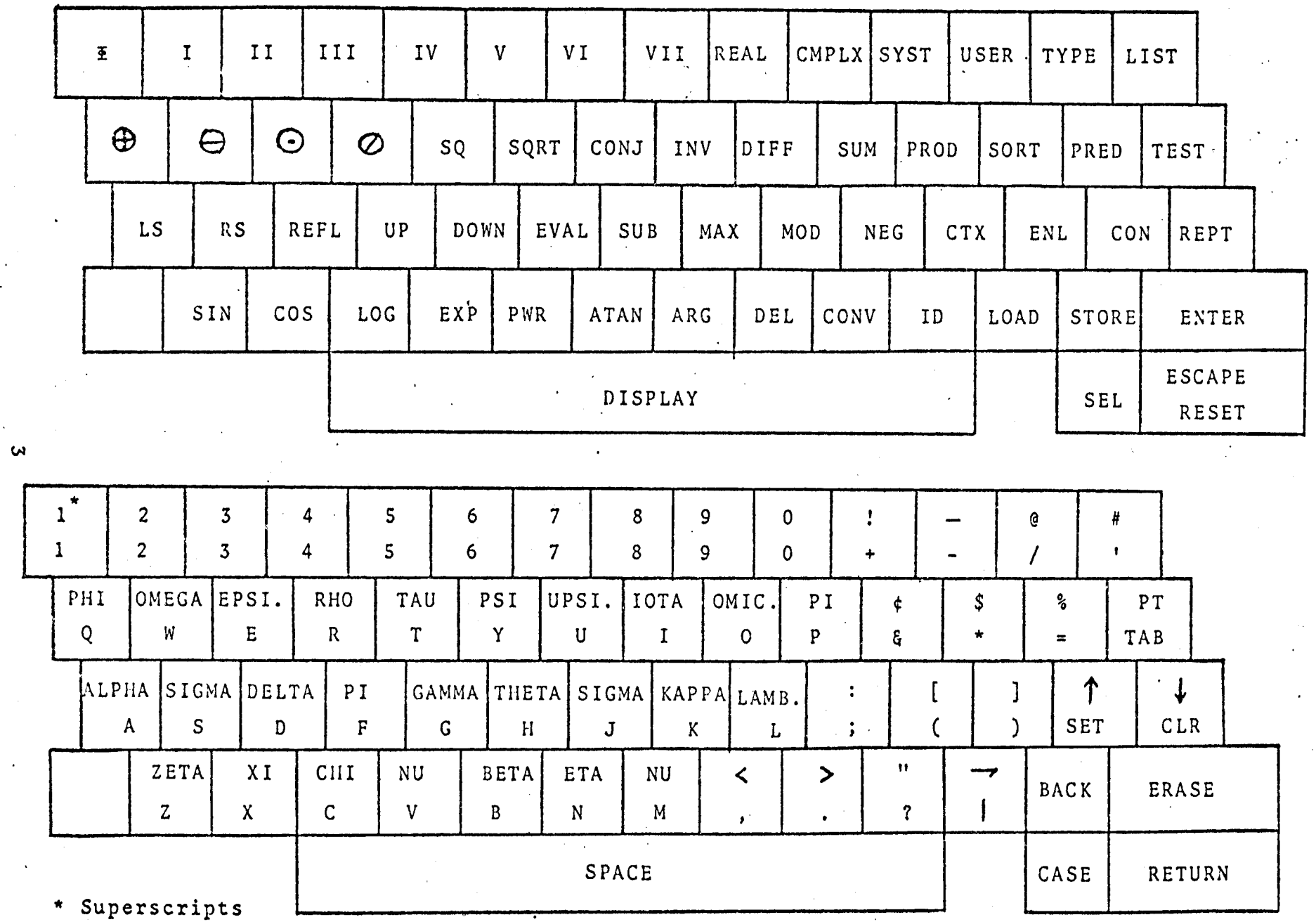

Figure 1. OLS Keyboard

These options (see Figure 6) are selectable via SERVER TELNET commands following the conventions of RFC 216.

III. EFFECTS ON THE PREVIOUS SPECIFICATION

The one-to-one and two-to-one mappings described herein fall completely within the previous specification of UCSB's teletype-compatible interface. At any time the user may specify an operator using the old style prefix and spelling. The old SERVER TELNET commands still apply. It is suggested, however, that the user change his prefix to be other than ; so that ; may be sent as a single key.

All UCSB sockets which allow TELNET input may use this new mapping. The socket which supports graphics display on TEKTRONIX series 4000 terminals and one socket which issues NGP0 output are two such cases. (See RFC 398 for a listing of UCSB's graphic sockets.)

Control characters, which were ignored in the previous specification, now map always into operators. Figures 3 and 4 show this mapping.

IV. DETAILS

A. Details of One-to-One Mapping

One-to-one mappings exist for A-Z, 0-9, most punctuation, ERASE, and most upper keyboard operators. The mapping for operators is accomplished with upper case alphabetics and control characters

MODE	TO PUSH OLS	OLS EXPLANATION	SEND ASCII
BOTH MODES	0-9	Numerics	0-9
	[SPACE] [RETURN] [BACK] [TAB] ,<.>?;: ([)]&*$ =%+!- |,¬,_ LINEFEED DOWN (↓)	Punctuation	[SP] [CR] [BACK] [TAB] ,<.>?;: ([)]&*$ =%+!- |,~,← [LF]
	[ERASE] [USER] - [NEG] *	Erase Screen Operators	[DEL] ↑a-↑z
OPERATOR	A-Z	UC Alphabetics	a-z
	[LS] - [SIN] *	Operators	A-Z
ALPHABET	A-Z a-z**	UC Alphabetics LC Alphabetics	A-Z a-z

Figure 2. One-to-One Mappings

* See Figures 3 and 4.

**a-z is actually Alpha-Zeta for local users. However, network users will see and use these keys as a-z.

ASCII	LOWER SHIFT	UPPER SHIFT	CONTROL
a	A	LS	USER
b	B	ATAN	DISPLAY
c	C	LOG	RESET
d	D	REFL	LIST
e	E	⊙	II
f	F	UP	ENTER
g	G	DOWN	REPT
h	H	EVAL	BACK
i	I	INV	(TAB)*
j	J	SUB	(↓) *
k	K	MAX	LOAD
l	L	MOD	STORE
m	M	DEL	(CR) *
n	N	ARG	ID
o	O	DIFF	CMPLX
p	P	SUM	SYST
q	Q	⊕	‡
r	R	⊘	III
s	S	RS	TYPE
t	T	SQ	IV
u	U	CONJ	REAL
v	V	EXP	CTX
w	W	⊖	I
x	X	COS	PWR
y	Y	SQRT	V
z	Z	SIN	NEG

Figure 3. One-to-One Operator Mode Mappings, Expanded

* [TAB], [↓], and [CR] are not really operator keys, but are included to be compatible with ASCII.

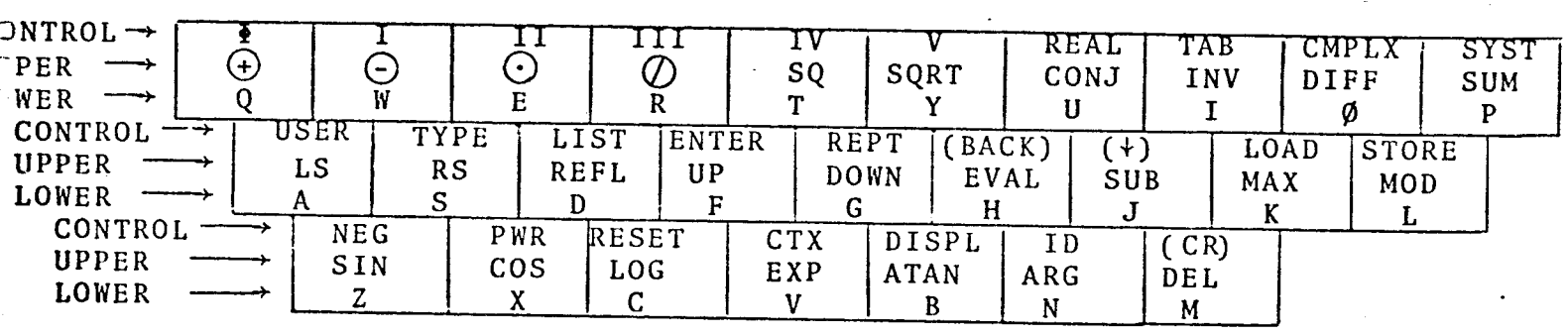

Figure 4. Visual Representation of One-to-One Operator Mappings

in the OPERATOR mode. ALPHABET mode sacrifices the upper case portion of this mapping, but don't panic. [ESC] may be used to return to OPERATOR mode for a single key (see details of Two-to-One Mapping). Figures 2, 3, and 4 contain the details of the one-to-one mapping. The ASCII keys FS, GS, RS, US, `, \, {, }, ^ are currently unassigned and may be used as convenient command prefixes.

B. Details of Two-to-One Mapping

Two-to-one mappings exist for the remainder of the OLS keys. In all cases the first key of the two key sequence is [ESC]. Either of two actions may be performed by [ESC] depending upon the successor key. If the successor is not an alphabetic the appropriate OLS key will be selected independent of mode, as outlined in Figure 5. For example, [ESC]? will always give the [TEST] key. If the successor is an alphabetic then it is interpreted as though the opposite mode were in effect (i.e. GENERATOR is the opposite of ALPHABET). This may seem confusing, but some useful effects are generated and will be illustrated by the following two examples.

1. If while in OPERATOR mode the keys [ESC]a are entered the interpretation will be "a" (as in ALPHABET mode) rather than "A". Thus users may enter lower case alphabetics while in OPERATOR mode.
2. If while in ALPHABET mode the keys [ESC]A are entered, the interpretation will be [LS] (as in OPERATOR mode) rather than "A". Thus users may enter the operators [LS] - [SIN] while in ALPHABET mode. The entire mapping is outlined in Figure 5.

MODE	TO PUSH OLS	OLS EXPLANATION	SEND ASCII
BOTH MODES	Shift 0 -Shift 9	Shift Numerics	[ESC] 0- [ESC] 9
	¢	Cent Sign	[ESC] $
	↑	Line Feed Up	[ESC] [LF]
	TEST	Operators	[ESC] ?
	PRED		[ESC] #
	SORT		[ESC] %
	CONV		[ESC])
	CASE		[ESC] (
	ENL,CON		[ESC] >, [ESC] <
	VI,VII		[ESC] &, [ESC] '
	PT		[ESC] .
	SET,CLR		[ESC] +, [ESC] -
OPERATOR	A-Z*	UC Alphabetics	[ESC] A- [ESC] Z
	a-z	LC Alphabetics	[ESC] a- [ESC] z
ALPHABET	A-Z*	UC Alphabetics	[ESC] a- [ESC] z
	[LS] - [SIN]	Operators	[ESC] A- [ESC] Z

** (applies to the TEST through SET,CLR rows)

Figure 5. Two-to-One Mappings

*These cases are degenerate and minimally useful, but are included for completeness. A-Z is best sent by a-z in OPERATOR Mode and A-Z in ALPHABET Mode.

**Note: in actual usage these keys might be best handled by spelling them out rather than looking at a chart.

V. New or Revised SERVER TELNET Options*

A. OPERATOR/ALPHABET Option

A new option set by the OPERATOR command enables the one-to-one mapping described above. The ALPHABET command restores the interpretation of lower/upper case alphabetics to that of RFC #216. Users of the OPERATOR mode will seldom return to ALPHABET mode. When ALPHABET mode is invoked, however, it will probably be to use message generation on the TYPE level or upper/lower case file creation using COL.

B. SHIFT/UNSHIFT Option

Another option, set by the SHIFT command, reverses the interpretation of upper/lower case ASCII alphabetics. For example, if SHIFT is invoked while in ALPHABET mode, a lower case "a" would receive an upper case "A" interpretation (i.e. send an OLS "A"). Or while in OPERATOR and SHIFT modes a lower case "a" would receive the upper case "A" interpretation, LS. The SHIFT option was useful in the previous specification to gain convenient access to upper case. For this new specification, however, its usefulness is minimal, but might be used for a user who would rather map upper case ASCII to upper case OLS and lower case ASCII to operators.

*See Figure 6 for a summary of all available commands.

C. INTERPRETLF/IGNORELF Option

The options INTERPRETLF/IGNORELF exist due to the desire to map [LF] directly into the OLS key [↓] (located over [CLR]). Some terminals always send [LF] following [CR]. For these terminals IGNORELF ignores the trailing [LF], thus avoiding spurious sends of [↓] to OLS. Most terminals, however, allow the separation of [CR] and [LF]. For these, INTERPRETLF can be used to cause all [LF] keys to be interpreted as [↓].

VI. CONCLUSIONS AND SUGGESTIONS FOR EASY USAGE

For these mappings to be most useful, the user should have tape-on labels for the fronts of keys A-Z and ↑A-↑Z (the same keys for most keyboards). A reference card summarizing double keypushes and special modes completes the necessary equipment. The Computer Systems Lab will work with individuals to create customized label sets.

It is hoped that the current experimental mapping will provide more convenient input to the UCSB OLS. As network graphics protocols become better defined, UCSB hopes also to enable more output features not now available to network users. Any criticisms, comments, or suggestions with this and future improvements should be sent via network mail (e.g. SNDMSG) to GRIPE at UCSB-MOD75.

COMMAND		MEANING
*HELP		Lists all NETOLS commands.
PREFIX	*(default=;)*	Change NETOLS command prefix.
SHIFT		Lower and upper case ASCII is interpreted as upper and lower case ASCII, respectively.
UNSHIFT	*(default)*	Lower and upper case ASCII is interpreted as lower and upper case ASCII, respectively.
*FULLDUPLEX		Causes NETOLS to echo all characters.
*HALF DUPLEX	*(default)*	Disables NETOLS character echoing.
*STATE		Displays status of all options.
LOGOUT		Sign off and break connection.
IGNORELF		Ignore any [LF] following [CR].
INTERPRETLF	*(default)*	Interpret [LF] following [CR] as in Figure 2.
OPERATOR		Lower case and upper case ASCII are interpreted as upper case alphabetics and operators respectively.
ALPHABET	*(default)*	Lower case and upper case ASCII are interpreted as lower case (greek) and upper case alphabetics, respectively.

Figure 6. Summary of SERVER TELNET commands

*These commands are ignored on all sockets which output graphics.