

*Posted*

NETWORK WORKING GROUP  
REQUEST FOR COMMENTS: 74

Jim White  
UCSB  
16 Oct 70

SPECIFICATIONS FOR NETWORK USE OF  
THE UCSB ON-LINE SYSTEM

## INTRODUCTION

UCSB's On-Line System (OLS) is available to Network users as socket number x'101' at site 3. Network users should log in with the following OLS accounting parameters:

USER NUMBER= 196  
ID NUMBER= 57372  
USER NAME= site name--UCLA,SRI,UTAH,BBN,MIT,  
SDC,RAND--whichever is appropriate.

Users communicate with OLS through an intermediary process, hereafter called the Interface, which is addressed as socket number x'101' (which is termed OLS's "primary socket"), and can be invoked through the Logger. This document is intended to provide programmers with the information necessary to communicate with the Interface and to

reader is assumed familiar with the Culler-Fried system at UCSB from a user's standpoint. Specifically, this document

## LOGGING INTO THE INTERFACE

To use the On-Line system, the Network user must establish a full-duplex connection with the Interface. The Interface is core resident only while at least one such duplex connection is established (i.e. while at least one Network user is connected). At all other times, the Interface resides on direct-access storage and must be invoked through the Logger. A login sequence can always

accept any call it receives; at all other times, the Logger listens on that socket and will reject the first call it receives, read the Interface into core, and dispatch it. The Interface will then listen on the primary socket as before. Thus to initiate a login sequence, the user requests

## THE INPUT CONNECTION

With the exception of the first two bytes, data received by the Interface over the Input Connection is treated as a continuous stream of one-byte key codes, potentially endless in extent. The Interface passes each key code--unexamined--to the On-Line System, which in turn processes it exactly as it would input from a key-board connected directly to the System. The set of valid key codes and its relation to the standard OLS key-board are depicted in Figure 1. The Interface makes no validity check of the incoming data, but OLS will detect and discard invalid key codes.

Normally, the first keys sent over the Input Connection (i.e. the first keys that the Network user pushes) should be those necessary to log in to OLS. The user may log in and out many times during the life of the Network connection, and these operations are transparent to the Interface. The last keys sent over the Input Connection should log the user off of OLS (SYST DOWN). Failing to log off before terminating the Network connection allows the possibility of a later Network user's finding himself already logged in.

The first byte of data received over the Input Connection is discarded unexamined by the Interface, which assumes it to be zeros indicating message type zero in compliance with Host-Host protocol. No significance is attached to Imp-message boundaries. The second byte of data received is not passed to OLS but is examined by the Interface. By appropriately selecting that second byte, the user can cause to be suppressed by the Interface, any or all of the three classes of output generated by OLS and ~~potentially available to the user over the Output Connection~~

The byte is interpreted as follows:

Bit 0	=	1:	suppress all alphameric output.
Bit 1	=	1:	suppress all curvilinear output.
Bit 2	=	1:	suppress all special character output.
Bits 3-7:			not examined, should be zeros.

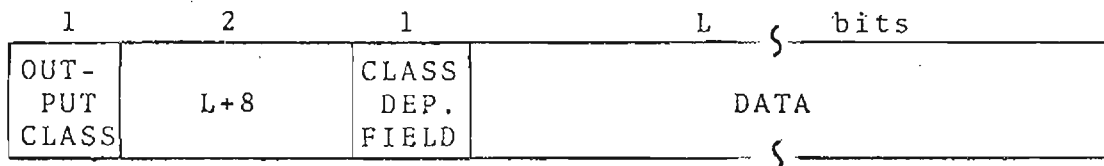
Once made, this declaration prevails for the life of the Network connections. A user can avoid transmission of output classes he is unable to process and would therefore ~~have to discard anyway, thus avoiding needless Network traffic~~

⊖ 30	I 31	II 32	III 33	IV 34	V 35	VI 36	VII 37	REAL 38	CMPLX 39	SYST 3A	USER 3B	TYPE 3C	LIST 3D
⊕ 11	⊖ 17	⊙ 05	∅ 12	S <sup>Q</sup> 14	SQRT 19	CONJ 15	INV 09	DIFF 0F	SUM 10	PROD 1C	SORT 1D	PRED AC	TEST BA
L <sup>S</sup> 01	R <sup>S</sup> 13	REFL 04	UP 06	DOWN 07	EVAL 08	SUB 0A	MAX 0B	MOD 0C	NEG 1B	CTX 2A	ENL 27	CON 28	REPT AA

## THE OUTPUT CONNECTION

With the exception of the first byte, data transmitted over the Output Connection by the Interface consists of a continuous string of variable-length records. The first byte sent consists of zeros, indicating message type zero, ~~to comply with Host Host protocol and should be discarded~~

by the user. At present there are three classes of records defined, one corresponding to each class of OLS output--alphameric, curvilinear, and special characters. Only records of those classes which have been enabled by the user will be transmitted; all other output will be suppressed locally by the Interface. Each record consists of a one-byte field specifying the output class, a one-byte output-class-dependent field, a variable-length data field, and a two-byte field containing the combined length in bits (unsigned) of the data and output-class-dependent fields. Each record has the following form:



The integer above each field is the length of that field in bytes (except where stated to the contrary). The length of a record, then, is given in bits by the contents of the length field plus twenty-four. The significance of the data and class-dependent fields, and the output class assignments are given in the following sections for each output class.

A. ALPHAMERIC OUTPUT (CLASS 1)

For alphameric output, the output class field contains the following:

Bits 0-3:	unpredictable
Bits 4-7:	0001

The contents of the class-dependent field are unpredictable. The data field contains the alphameric display in the form of a contiguous string of one-byte characters. Any character listed in Figure 2 may be present. The list includes the Greek and Latin alphabets, a variety of special symbols, as well as carriage control characters such as carriage return, line feed, backspace, and erase.

Alphameric output records embody system-generated messages, LIST mode displays, lower key-board activity on the TYPE level, TYPE level operators such as UP and DOWN, etc. The appearance of the character pair 'BACK ERASE' (x'59BC') in a record represents a command to erase the display scope. When not immediately followed by ERASE, BACK indicates a backspace operation. 'BREAK' (x'79') is used to facilitate formatting of long messages that may be either printer- or display-scope-destined. In generating scope display, where there are twenty-five characters per line, 'BREAK' should be interpreted as a carriage return; in generating printer output, where longer lines are possible, it should be interpreted as a space or blank.

FIGURE 2. ALPHAMERIC OUTPUT CHARACTER SET

<u>NAME</u>	<u>Lower Case</u>	<u>CODE</u>	<u>NAME</u>	<u>Upper Case</u>	<u>CODE</u>
A		C1	ALPHA		81
B		C2	BETA		82
C		C3	CHI		83
D		C4	DELTA		84
E		C5	EPSILON		85
F		C6	PI		86
G		C7	GAMMA		87
H		C8	THETA		88
I		C9	IOTA		89
J		D1	SIGMA		91
K		D2	KAPPA		92
L		D3	LAMBDA		93
M		D4	MU		94
N		D5	ETA		95
O		D6	OMICRON		96
P		D7	PI		97
Q		D8	PHI		98
R		D9	ΒΗΩ		99
S		E2	SIGMA		A2
T		E3	TAU		A3
U		E4	UPSILON		A4
V		E5	NU		A5
W		E6	OMEGA		A6
X		E7	XI		A7
Y		E8	PSI		A8
Z		E9	ZETA		A9
0		F0	ss 0		B0
1		F1	ss 1		B1
2		F2	ss 2		B2
3		F3	ss 3		B3
4		F4	ss 4		B4
5		F5	ss 5		B5
6		F6	ss 6		B6
7		F7	ss 7		B7
8		F8	ss 8		B8
9		F9	ss 9		B9

(cont'd)



<u>NAME</u>	<u>CODE</u>	<u>NAME</u>	<u>CODE</u>
PLUS +	4E	UNDERSCORE _	6D
MINUS -	60	AT SIGN @	7C
SLASH /	61	POUND SIGN #	7B
APOSTROPHE '	7D	CENT SIGN ¢	4A
LOGICAL AND &	50	DOLLAR SIGN \$	5B
ASTERISK *	5C	PERCENT SIGN %	6C
EQUALS =	7E	COLON :	7A
SEMI-COLON ;	5E	LEFT BRACKET [	73
LEFT PAREN (	4D	RIGHT BRACKET ]	74
RIGHT PAREN )	5D	LESS THAN <	4C
GREATER THAN >	6E	GREATER THAN OR EQUAL TO >=	6F

PERIOD .	4B	QUOTE "	7F
QUESTION MARK ?	6F	LOGICAL NOT ~	5F
LOGICAL OR	4F	EXCLAMATION !	5A

Carriage  
Control

Special List  
Mode Characters

BACK (backspace)	59	SPACE	62
RETURN (carriage return)	49	POST LIST :	63
TAB (advance to next tab)	77	DIVIDE Ø	64
UP (line feed up)	06	MULTIPLY @	65
ENL (line feed up)	27	SUBTRACT @	66
DOWN (line feed down)	07	ADD @	67
CON (line feed down)	28	CARRIAGE RETURN ↵	68
RS (position to upper left of display area)	13	DELETE ☒	69
ERASE	BC	POINTER _	6A
BREAK (for display scope: RETURN for line printer: SPACE)	79		
SPACE (blank)	40		

Miscellaneous

DOT (curvilinear display, dot-dot mode)	78
-----------------------------------------	----

NOTE:

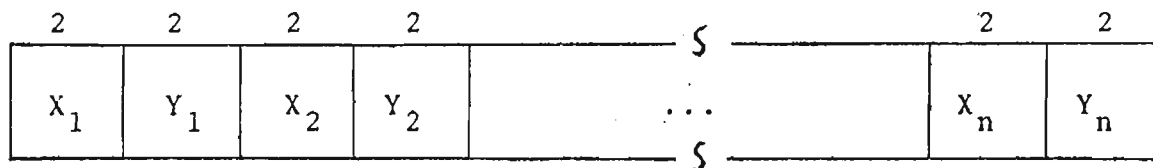
Codes are specified in hexadecimal and are eight bits.

B. CURVILINEAR OUTPUT (CLASS 2)

For curvilinear output, the output class field contains the following:

Bits 0-1:           00 indicates line segment mode  
                               (adjacent display points are to  
                                   be connected by straight lines).  
                               01 indicates dot mode  
                               10 indicates character mode (the  
                                   class-dependent field contains a  
                                   character from Figure 2 which is  
                                   to be displayed at each point  
                                   ('dot-dot' mode is character mode  
                                   with the display character 'DOT'  
                                   (x'78')).  
 Bits 2-3:           unpredictable  
 Bits 4-7:           0010

For character mode, the class-dependent field contains the display character; in other cases, the contents of that field are unpredictable. The data field contains a list of X-Y display coordinates as depicted below:



X<sub>i</sub> and Y<sub>i</sub> are the X and Y display coordinates--after scaling--of the i<sup>th</sup> component of the vector represented by this record. Each coordinate is contained in a two-byte field, therefore one component in four bytes, and hence the context of the vector being displayed is given by the contents of the length field minus eight divided by thirty-two. The assumed display area is square, with origin at lower left, and both X and Y ranging between 0 and 4095. There

is a one-to-one correspondence between vectors displayed and curvilinear output records transmitted.

C. SPECIAL CHARACTER OUTPUT (CLASS 3)

For special character output, the output class field contains the following:

Bits 0-3:           unpredictable  
Bits 4-7:           0011

The contents of the class-dependent field are unpredictable. The data field contains a contiguous string of variable-length characters, each representing either a move in one of sixteen directions or a change in position relative to the lower right corner of the last character frame (where for alphameric and special character display, the display area is square, 4096 units in extent vertically and horizontally, and a character frame is 160 units wide and 224 units high).

The sixteen characters which define move operations are listed in Figure 3, and each is one byte long. Such a character indicates a move from the current position, in the specified direction, a distance equal to that of a move in the same direction from the center of a 64-unit square to its perimeter. The length of the move is therefore functionally related to its direction.

A change in position relative to the lower right corner of the last character frame is represented by a four-byte character of the form:

1	12 bits	12 bits
x'70'	$\Delta X$	$\Delta Y$

where  $\Delta X$  and  $\Delta Y$  are signed quantities indicating the number of units change along each coordinate.

FIGURE 3. SPECIAL CHARACTER VECTOR CHARACTER SET

<u>DIRECTION</u>	<u>CODE</u>
000.0	47
022.5	48
045.0	51
067.5	52
090.0	53
112.5	54
135.0	55
157.5	56
180.0	57
202.5	58
225.0	41
247.5	42
270.0	43
292.5	44
315.0	45
337.5	46

NOTE:

Codes are specified in hexadecimal and are eight bits.

Directions are specified in degrees, increasing counter-clockwise from 0° at positive X in an X-Y coordinate system.