Network Working Group
Request for Comments: 74

J. White
UCSB

October 16, 1970

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 accounts 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 define the input expected and the output returned. The readers is assumed familiar with the Culler-Fried system at UCSB from a user's standpoint. Specifically, this document is not a user's manual for OLS.

The interface conducts all Network transactions through the NCP, which operates under the Host-Host protocol of 3 August 70. The first message sent by the Interface is of Type 0: the first eight bits are zeros and thereafter, for the life of the connection Imp-message boundaries are not significant. Similarly, the Interface expects the first message it receives to be Type 0, discards the first eight bits assuming them to be zeros, and thereafter for the life of the connection takes no notice of Imp-message boundaries.

A word about terminology. The 360/75 is a 32-bit machine, but its instruction set is byte-oriented. A byte is eight bits, and those eight bits are numbered 0-7 from left to right. Terms such as

"listen", "request connection", "accept a connection", and "reject a connection" are used freely herein to describe those primitive Network functions, which are user at a foreign site presumably has available to him through his NCP. They are used here in the same senses in which they have frequently been used in the NWG literature.

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 be initiated by requesting connection to OLS's primary socket. While in core, the interface listens on that socket and will 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 connection to the primary socket. If accepted, he is in contact with the Interface. If rejected, he should reissue the connection request; when accepted, he will be connected to the Interface. A second rejection would indicate that the On-Line System was inactive, or that either the Interface or the NCP had exhausted its resources.

Over this initial connection, the Interface will send eight bits of zeros, indicating message type zero, followed by a 32-bit socket number, which it will select from a pool of socket numbers allocated to it. It will then promptly close the connection and reissue the listen, to allow other users to begin login. It will then request connection of the local socket whose number was sent to the user, with the foreign socket whose number is one greater than that of the user's socket. Similarly, it will request connection of the local socket whose number is one greater than that sent to other user, with the user's socket. Once the two connections have been established, the Interface will consider the user logged in.

The two connections thus established are maintained indefinitely by the Interface. Over its receive connection (hereafter termed the "Input Connection"), the Interface accepts input fro OLS. Over its send connection (the "Output Connection"), the Interface relays displays from OLS generated in response to the input. The Interface will terminate the connections only should the On-Line System terminate. The user is expected to close the two connections when finished, making the local sockets available for reallocation, at which time the Interface will consider the user logged off.

The Input Connection

With the exception of the first tow 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 keyboard connected directly to the System. The set of valid key codes and its relation to the standard OLS keyboard 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 key s 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 relayable to the user over the Output Connection. The byte is interpreted as follows:

```
        Bit     0  =    1: suppress all alphanumeric 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. A user operating from a teletype and capable of displaying only alphameric output, for example, might specify x'60' and thereby suppress all else.

Figure 1. Input Key Code Set [Please view PDF version.]

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:

```
   1            2            1                 L        bits
---------------------------------------------------S-----------
|OUT-   |               |  CLASS  |                           |
|PUT    |       L+8     |   DEP.  |             DATA          |
|CLASS  |               |  FIELD  |                           |
---------------------------------------------------S-----------
```

The integer above each field is the length of that field in bytes (except where stated to the contrary). The lengthy of a cord, 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 keyboard 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

NAME	Lower Case	CODE	NAME	Upper Case	CODE
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	RHO		99
S		E2	SIGMA		A2
T		E3	TAU		A3
U		E4	UPSLION		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

NAME	CODE	NAME	CODE
PLUS +	4E	UNDERSCORE _	6D
MINUS -	60	AT SIGN @	7C
SLASH /	61	POUND SIGN #	7B
APOSTROPHE ’	7D	CENT SIGN [cent sign]	4A
LOGICAL AND &	50	DOLLAR SIGN $	5B
ASTERISK *	5C	PERCENT SIGN %	6C
EQUALS =	7E	COLON :	7A
SEIM-COLON ;	5E	LEFT BRACKET [	73
LEFT PAREN (	4D	RIGHT BRACKET]	74
RIGHT PAREN)	5D	LESS THAN <	4C
COMMA ,	6B	GREATER THAN >	6E
PERIOD .	4B	QUOTE "	7F
QUESTION MARK ?	6F	LOGICAL NOT [half arrow]	5F
LOGICAL OR |	4F	EXCLAMATION !	5A

Carriage Control		Special List Mode Characters	
BACK (backspace)	59	SPACE _	62
RETURN (carriage return)	49	POST LIST :	63
		DIVIDE [0with /]	64
TAB (advance to next line)	77	MULTIPLY [0 with .]	65
		SUBTRACT [0 with -]	66
UP (line feed up)	06	ADD [0 with +)	67
ENL (line feed up)	27	CARRIAGE RETURN	
DOWN (line feed down)	07	[diagonal left down arrow]	68
		DELETE [box with ///]	69
CON (line feed down)	28	Pointer _	6A
RS (position to upper left of display area)	13	**Miscellaneous**	
ERASE	BC		
BREAK (for display scope: RETURN for line printer: SPACE)	79	DOT (curvilinear display dot-dot mode)	78
SPACE (blank)	40		

Note:
Codes are specified in hexadecimal and are eight bits. ’ss’ means ’superscript’

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:

```
  2     2       2       2                               2      2
--------------------------------------S----------------------------
| X1   | Y1  |  X2  |  Y2  |         ...             | Xn  |  Yn  |
--------------------------------------S----------------------------
```

Xi and Yi are the X and Y display coordinates -- after scaling -- of the ith 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 original 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

Direction	Code
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 0o at positive X in an X-Y coordinate system.

* Text enclosed in brackets describe non-ascii characters that were present in the original document. Please see the PDF file for the actual representations.