Network Working Group
Request for Comments #86
NIC 5631

S. Croker
January 5, 1971
UCLA

# Proposal for a Network Standard Format for a Data Stream to Control Graphics Display

A typical arrangement of facilities is to have a console attached to a computer at the user's site, and to be using the computational facilities of a remote site. Information entered by the user is transmitted to the remote Host, and output from the remote Host is transmitted back to the local user. In this proposal I am concerned with specifying the form of the output stream for the case that the output portion of the console is a typical refresh display with point, vector and character drawing capability. Devices in this class include the DEC 338, DEC 340, IBM 2250, and IMLAC PDS-1.

It must be understood that this proposal is illustrative only, and knowingly avoids important issues. Its main purpose is to provide a basis for discussion and development.

In order to specify the semantices of the network standard graphics data stream (NGDS), I will postulate

1. a network standard display list (NGDL)
2. a network standard stream interpreter (NGSI)
3. a network standard list interpreter (NGLI), also called the display controller, and
4. a network standard screen (NGS)

The NGDS is accepted into the local Host and interpreted by the NGSI. The NGSI is a process which modifies the NGDL according to inputs in the NGDS. The NGDL is the display list for the NGLI; the NGLI executes the NGDL and controls the beam which writes on the NGS.

The NGS is square, has horizontal and vertical sides, and positions on it are specified by an ordered pair of unsigned 16 bit fractions. The first fraction specifies the horizontal distance from the left hand edge, and the second specifies the vertical distance from the bottom edge. The resolution of the screen is unspecified.

The lack of specification of the resolution of the NGS is intentional; programs designers should not interpret this to mean that they may impose a particular requirement on the using system. Thus the quality of the displayed picture should degrade gradually with decreasing resolution.

The NGLI has primitives for moving the beam to a particular point, intensifying a point, drawing a vector, or drawing a character. Characters are assumed to be not more than .015 screen width wide, and not more than .025 screen height high. When the beam is moved to a screen position before drawing characters, that position should be at the lower left hand corner of the first character drawn. The beam position after drawing a character is immediately to the right of the character, properly positioned for another character. However, after drawing one or more characters, the exact horizontal

position of the beam is unspecified.

The imprecise specification of character size is intended to take cognizance of the existance of character drawing hardware which is capable of producing only one or a few character sizes. The particular proportions of .015 wide by .025 high provides for 67 characters to a line and 40 lines, and seems well within the capability of most displays in the class considered. As in the case of screen resolution, variations in character drawing hardware should cause only minor degradation in the usefulness of the picture.

The character set intrepreted by the NGLI is ASCII, excepting all form control characters, but including the space character. The tab, return and line feed functions should be simulated by explicit positioning of the beam.

The NGDL consists of a set of named and possibly null lists. The names are 16 bit integers, and the name zero is the name of the chief list. Each list is composed of zero of more display items. An item is any of the following:

a) "Move beam to xxxx,yyyy, relative to current origin"

b) "Display vector from current position to xxxx,yyyy, relative to current origin"

c) "Display current point"

d) "Display the following n characters:

c c ...c "
1 2 n

e) "Execute list gggg with origin xxxx,yyyy relative to current origin"

The NGLI is constantly in a loop executing the chief list, the origin of the chief list is always <0,0>. When the NGLI comes to the end of the chief list, it returns to the top of it. When the NGLI encounters a type e item, it suspends execution of the current list, set the new origin to <xxxx,yyyy> + <the old origin>, and executes the list named gggg. When finished with the list, the old origin is restored and execution of the old list resumed. The NGLI is, therefore, a recursive interpreter, and type e items are subroutine calls.

There remains only the matter of the NGDS and the NGSI. The NGDS is parsed into commands by the NGSI and each command is executed as soon as it is recognized. The commands in the NGDS are in prefix form, consisting of an eight bit operation code followed by any arguments. Only two commands are defined: Erase and Replace.

The Erase command consists of a single zero-valued opcode and no arguments. The NGSI executes the Erase command by making all lists into null lists.

This erases the screen.

The Replace command has an opcode of 1, followed by a 16-bit list name, followed by an 8 bit subargument count, followed by the indicated number of subarguments.  The NGSI executes the Replcae command by deleteing all items from the indicated list, and rebuilding the list from the subarguments. There are five kinds of subarguments, corresponding to the five item types.

<subargument>::=<atype> | <btype> | <ctype> |

<dtype> | <etype>

<atype>::= <a> <x> <y>

<btype>::= <b> <x> <y>

<ctype>::= <c>

<dtype>::= <d> <n> <c> ... <c>
n

<etype>::= <e> <g> <x> <y>

<a>, <b>, <c>, <d>, and <e> are 8 bit bytes valued at 0, 1 ... 4,

respectively and corresponging to the item types listed on page 4.

<x> and <y> are 16 bit numbers

<g> is a 16 bit list name

<n> is an 8 bit count

<c> is an ASCII character code

[ This RFC was put into machine readable form for entry ]
[ into the online RFC archives by Anand Kumria 6/97 ]