

NETWORK WORKING GROUP
Request for Comment: 177
Obsoletes: none
Updates: 125
NIC: 7102

J. McConnell
Ames Research Center Moffet Field, CA
15 June 1971

A DEVICE INDEPENDENT GRAPHICAL DISPLAY DESCRIPTION

As more nodes are connected to the ARPA network, the types of graphical display processors available to users is quite varied. To attempt to facilitate the transmission of graphical information over the network, a device independent description of a display is described. The using host may make any conversions necessary to realize the picture on a specific device. It is also possible to interpose a form machine as proposed by Heafner and Harslem in RFC #94. Some of the items and entities described herein were first discussed by Steve Crocker in RFC #86. It is intended that this description is to be oriented to a refresh display with point, vector, and character drawing capability, these types of devices include the IBM 2250, IMLAC PDS-1, the DEC 338, and DEC 340, as well as the Evans and Sutherland LDS-1. However, direct video storage tube devices represented by the ARDS, Tektronix and Computer devices can also serve as clumsy interactive devices. Hard copy devices, such as microfilm or plotters, can also be used at the using host's discretion.

There are several items and constructs which will be defined before discussing the specifics of the description:

1. The network standard graphics description stream (NGDS) contains the description, data, and operators necessary to effect a display.
2. The network standard stream interpreter (NGSI) parses the NGDS into its components.
3. The network standard display list (NGDL) is the basic entity which, when executed, causes graphical information to be displayed.
4. The network standard list interpreter (NGLI) is the entity which executes the NGDL and controls the beam movement to effect generation of graphical information.
5. The network standard screen (NGS) is the entity on which the information is to be displayed. It may be divided into image areas to be defined later.

6. The network standard graphics list pointer (NGLP) and the network standard graphics execution stack (NGES) are entities which describe the state of the NGLI at any given time and provide the facilities to allow the NGLI to be a recursive interpreter.

Figure 1. summarizes the preceding discussion graphically, which is appropriate.

The network standard graphics screen is rectangular and positions on it are specified by an ordered pair of fractions representing the horizontal distance from the left edge and the vertical distance from the bottom edges respectively. These shall be termed the (x,y) coordinates as is standard. There is no specification of resolution, so that the quality of the picture will degrade with the decreasing resolution of a poor display device, or be improved if it is displayed on a higher resolution device. Coordinate values are specified by a 16 bit unsigned fractions with the binary point to the left of the most significant bit - this provides a normalized screen with coordinates between 0.0 and .999...9.

The NGS may be subdivided into rectangular image areas with possibility unique displays in each space. An image area has three attributes: a 16 bit integer name, an x,y pair specifying the lower left corner and an x,y pair to specify the upper right corner of the area. Image spaces may overlap, but may not be completely contained within each other. The main image space has the name 0 and is coincident with the NGS. The use of image spaces allows for manipulation of part of the NGS's contents, as well as redefining the coordinate space. Within an image area, coordinate values are fractional displacements from the lower left hand corner of the image space. Thus an image area defined from (0.0, 0.0) to (0.5, 0.5) would contain an image one-fourth of the size of what it would be on the full NGS.

Character strings may be displayed at arbitrary points in an image area. Because many display devices have hardware character generators capable of producing one, or a few sizes, character scaling within an image area will not be expected. Characters shall be assumed to be .014 screen width wide, and .025 screen height high including spacing. This gives a screen capacity of 72 characters and 40 lines. When the beam is moved to be a screen position prior to drawing a character, it is assumed to be in the center of the rectangle defining the character space. The beam position after drawing a character, or a string of characters, is undefined.

The format of the NGDS can now be specified. The NGSI parses the NGDS into commands. The commands are in a prefix format with an eight bit command followed by the necessary parameters. Their

descriptions follow the numerical order of the command codes.

ERASE (commands = 0) consists of a command byte and no parameters. Its effect is to erase the full NGS.

DEFINE LIST (commands = 1) has three parameters. A 16 bit integer for the list name is first. If the name has already been used, then this command redefines the list. The count of the items in the list is also a 16 bit integer defining the length of the list in items. The list items follow the count terminating with an end list item.

The items themselves are used to control the NGLI and the movement of the beam when drawing pictures. They are also in prefix format, with an eight bit item code followed by the necessary parameters. All names are 16 bit integers, coordinate data are 16 bit unsigned fractions as previously described. Any other parameters will have their type and size specified in individual descriptions.

- a. OPEN IMAGE AREA (name) (item type = 0) instructs the NGLI that list commands which follow are to be adjusted to be contained within the named area. A nonexistent area name is treated as a NOP. If no image area is specified for the NGLI the default area is area 0, the NGS. This directive remains in effect until another type 0 item is encountered by the NGLI.
- b. RESET IMAGE AREA (name) (item type = 1) causes the NGLI to move the beam to the lower left corner of the image area. A nonexistent name is again treated as a NOP.
- c. CLEAR IMAGE AREA (name) (item type = 2) causes the NGLI to erase all graphical information in the specified area, and then to perform a reset command.

The following items are display items which actually cause beam movement by the NGLI. Coordinate data are relative to the origin of the current image area and are taken as fractional displacements within this area.

- d. MOVE BEAM (x,y) (item type = 3) causes the NGLI to move the beam to (x,y) relative to the current origin.
- e. DISPLAY CURRENT POSITION (item type = 4) causes the NGLI to display the current point on the NGS.

- f. DISPLAY VECTOR (x,y) (item type = 5) causes a vector to be drawn from the current point to (x,y). After the operation, (x,y) becomes the current point.
- g. DISPLAY CHARACTERS (n, 'C1, C2...Cn') (item type = 6) causes the n characters to be displayed starting from the current point. N is the character count and is an eight bit byte.

The character set recognized by the NGLI is composed of eight bit bytes interpreted in the following ways. Any character with the most significant bit a zero is interpreted as an ASCII character. Some non-graphic ASCII characters may have special functions within a string:

1. Line feed (LF) move the beam one line spacing downward.
2. Carriage return (CR) move the beam to the left most character position in the current line.
3. Backspace (BS) move the beam one character position to the left in the current line.
4. Tab move the beam a predetermined number of character positions to the right in the current line. The number will follow the tab character in the string, but may not exceed the capacity of the current line.
5. Vertical tab (VT) move the beam down a predetermined number of line spacings. The number of spacings follow the VT character and may not exceed the line capacity of the screen. This is expanded as multiple line feeds.
6. Any other non-graphic character - could cause a space character to appear in its place on the display, or it may be used for any purpose two processes choose.

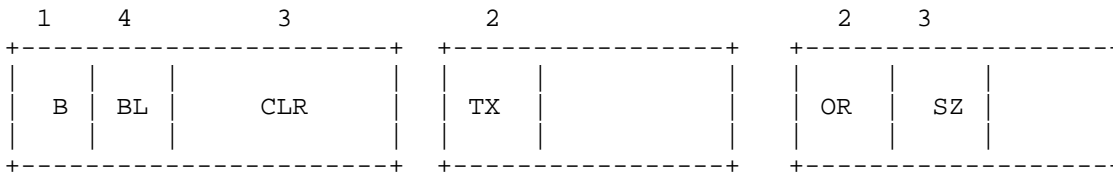
Characters with the most significant bit a one are used for an extended character set where a device can support this. These may also be used as an implicit stroke table, calls being the names of special graphic lists that may be defined in the NGDS. If no eight bit list name exists, then the character is a space.

- h. EXECUTE LIST (name, x, y) (item type =7) causes the NGLI to suspend interpretation of the list and interpret the named list. It is equivalent to a subroutine call. The (x,y) pair specifies the origin of the new list relative to the current origin. The actions taken by the NGLI are specified in the next section

- i. COPY LIST (name, x,y) (item type = 8) cause the items contained in the named list to be copied into the place of the item with the origin of the list at (x,y) relative to the current origin. This is analogous to item (h) as a subroutine is to a macro.
- j. EXECUTE TABLE (name, mode, length) (item = 9) causes the NGLI to treat the named list as a special entity. It is constrained to contain only coordinate pairs. It may be executed in either point or vector mode as specified by the mode parameter 1= vector, 0= point. The length of the list is specified by a 16 bit integer parameter.
- k. END LIST (item type = 10) has no parameters and informs the NGLI that the end of a list has been reached.
- l. TABLE (n) (item type = 11) specifies that there are (x,y) pairs in this list. N is a 16 bit integer.
- m. Set vector mode (mode) (item type = 12)
- n. Set character mode (mode) (item type 13)

In addition to the above display items, some items exist which control the display mode of the above where this is applicable. There are two mode items, the one governing vector modes, and one for character modes. The quantities specified include brightness, blinking, color selection, and mode dependent descriptions.

For vector modes, the texture of vectors such as dashed, solid or, dotted can be specified. For characters, size, and orientation can be additionally specified. The scope of a mode item extends to the next encountered mode item of the same type. Mode specifications could have been included in the display items at the expense of more parameters and in possibly more data to be transmitted. The mode specification consists of two eight bit bytes. The bytes have the following organization:



BYTE0 - both modes

BYTE 1- vector mode

BYTE 1- Character mode

- Where
- B Blink if set
 - BL Brightness level - 0 = invisible, IIII brightness
 - CLR Color One bit for each primary
 - TX Texture
 - 00 = solid
 - 01 = dashed
 - 10 = dotted
 - 11 = dot - dash
 - OR Orientation binary value x 90deg
 - SZ Size

If the device will not support them, or provide subroutines to perform these enhanced functions, they are treated as NOPs.

Thus the NGDL is a set of named lists.

DEFINE IMAGE AREA (command = 2) has three parameters a 16 bit area name; an (x,y) pair defining the lower left corner of the area and an (x,y) pair defining the upper right corner. If the name has already been used to define an area then this command serves a re-definition of that area. If the two (x,y) information displayed in it. If Xl is greater than Xr or Yl is greater than Yr then the definition is treated as NOP.

The following commands are included to avoid retransmission of a list when some minor changes to it are necessary. All instances of a list, where an instance is a command to execute the list will be effected. Where a list has been copied into another list, no change will take place in the copied list.

TRANSLATE LIST (command = 3) provides for translation of all instances of a list within all of the image spaces it occurs in. Parameters are the 16 bit list name, and two 16 fractions for the coordinate values of the translation.

SCALE LIST (command = 4) can change the size of the picture created by a list in all image spaces. The parameters are the 16 bit list name followed by two 16 bit (X,Y) unsigned fractions with eight bits of exponent followed by eight bits of mantissa. These alter the scale factors in the X and Y directions.

ROTATE LIST (command = 5) will rotate all instances of a list. The parameters are for the 16 bit list name, and a 16 bit value corresponding to rotational angle in degrees with the sense being counter-clockwise for increasing values and the possible axis being zero degrees.

The NGLI executes in a loop, taking items from the main list. When the end of the main list is reached, the NGLI returns to the top of it. At any point, the NGLP gives the NGLI the positional displacement of the next display item it is processing. When a display item of the type h. is encountered, the current lists's execution is suspended, the name of the list, the current origin, and the NGLP are saved in the NGES. The NGLP is re-set to zero, the next current origin is calculated, and execution of the new list commences. When the end of the list is encountered, the old values are restored from the NGES and execution of the suspended list continues. The NGLI is then a recursive interpreter. Whenever a new image area is opened, the name of that image area is stored in the current image area name.

The problem of interacting with the displayed picture has yet to be addressed since this is a more complicated area. Interaction may occur in two fashions: the first, and the easiest to handle, are those kinds of events which are separate from the picture itself. This encompasses key boards and function key types of devices. These can be handled as standard messages from the graphics device to the serving host. The second class of interactions is with the picture itself. This is more difficult because of the problem of associating the point selected with some meaningful entity such as a list. This association can probably only be made by the serving host since the using host, or a Form Machine, may have transformed the NGDS in unknown ways, and the NGDS may no longer exist at the using host's site.

There are essentially two classes of devices that can interact with the picture. The first class is synchronous devices, ones who only cause some attention to occur while a given display item is being

executed. Light pens are synchronous devices. Asynchronous devices are those which can interact regardless of which, or even when no, display item is being executed. Asynchronous devices make associations more difficult since no relationship can be easily inferred as they can for synchronous devices.

The NGES is created for the interaction with the picture. For synchronous devices the current values for the list name, origin, and NGLP, as well as the contents of the NGES provides a hierarchical structure where associations can be made. For asynchronous devices, the structure is probably not as necessary as the name and coordinates within the image space in which the interaction occurred. It will be necessary for the using host to understand which type of devices are available and to supply proper type of interaction information to the serving host.

The form of the interaction information for synchronous devices follows:

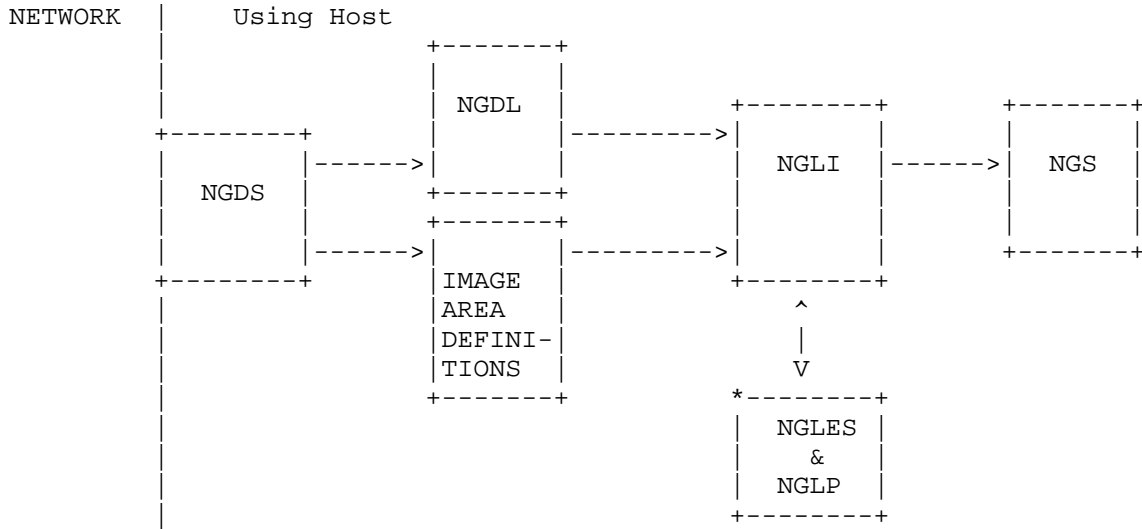
INTERACTION (type, cia, cln, cor, clp, n, l1, or1, lp1, ...ln, orn, lpn)

Where type describes the interaction type 0=synchronous
 cia current image area name
 cln current list name
 cor current origin value
 clp current NGLP value
 n number of entries in NGES, higher value of n is older entry
 li list name of the i th entry
 ori origin of the i th entry
 lpi NGLP of the i th entry

The form for asynchronous interaction is:

INTERACTION (type, cia, iax, iay, sx, sy)

Where type = 1 for asynchronous interactions
 cia current image name
 iax image area x coordinate
 iay image area y coordinate
 sx,sy screen x,y coordinates



[This RFC was put into machine readable form for entry]
 [into the online RFC archives by Kelly Tardif, Viagénie 12/1999]