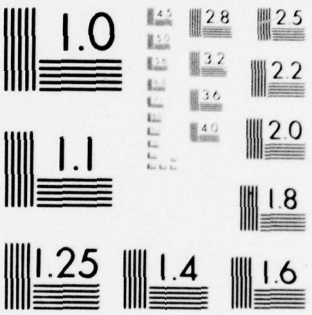
AD-A059 902 ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG MISS F/G 9/2
DEVICE IMPLEMENTATION GUIDELINES FOR 3-D GCS.(U)
JUL 78

UNCLASSIFIED NL

| OF |
AD
A059902

END
DATE
FILMED
12-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

WATERWAYS EXPERIMENT STATION

Vicksburg, Miss.

U.S. ARMY · CORPS OF ENGINEERS · 1775 · BICENTENNIAL · 1975

200th

DEVICE IMPLEMENTATION GUIDELINES FOR 3-D GCS.

This program is furnished by the Government and is accepted and used by the recipient with the express understanding that the United States Government makes no warranties, expressed or implied, concerning the accuracy, completeness, reliability, useability, or suitability for any particular purpose of the information and data contained in this program or furnished in connection therewith, and the United States shall be under no liability whatsoever to any person by reason of any use made thereof. The program belongs to the Government. Therefore, the recipient further agrees not to assert any proprietary rights therein or to represent this program to anyone as other than a Government program.

for the WES Automatic Data
Processing Center

3-D VERSION OF GCS

*** BLOCKED 10 CARDS PER 84 CHARACTER PHYSICAL RECORD

The first series of files describe the computers presently available. In front is a series of statements that must be inserted in two places. The first is in the block data routines and the second is in subroutine "URESET" in the device dependent routines for each graphics device supported.

FILE NUMBER	TYPE	
1	CDC	COMPUTER DEPENDENT
2	CDC	"
3	CDC	"
4	HWL	"
5	HWL	"
6	IBM	"
7	DEC	"
8	BUR	"
9	DEVICE INDEPENDENT	
10	ALP	DEVICE DEPENDENT
11	ARD	"
12	CAL	"
13	C56	"
14	COM	"
15	DAT	"
16	FR8	"
17	GLD	"
18	GOU	"
19	IML	"
20	PTR	"
21	TEK	"
22	TK4	"
23	TSP	"
24	TTY	"

DEVICES SELECTED BY CODE AND DESCRIPTION

ALP --	Interactive Alphanumeric Terminals
ARD --	ARDS Storage Tube Terminals
C56 --	Calcomp 563 Drum Plotter
C93 --	Calcomp 936 Drum Plotter
COM --	Computek 400/15 Storage Tube Terminal W/6-Bit Interface
DAT --	Datapoint 3300 Refresh Alphanumeric CRT Terminal
GLD --	Gould 4800 Dot-Matrix Printer -- Bit-Matrix Interface
GOU --	Gould 4800 Dot-Matrix Printer -- Calcomp Compatible Interface
IML --	Imlac PDS-1D and PDS-4 Terminals -- Tis Package
PTR --	Alphanumeric Line Printers
TEK --	Tektronix 4010/4012/4013 Storage Tube Terminals
TER --	Terminet 300
TK4 --	Tektronix 4014/4015 Storage Tube Terminals
TSP --	Timesharing Peripherals TSP-212 Analog Pen Plotter Terminal
TTY --	Remote Batch Alphanumeric Terminals

COMPUTERS SELECTED BY CODE AND DESCRIPTION

CDC --	CDC 6000 Series Computers
DEC --	DEC PDP-10 Computers
HWL --	HONEYWELL 6000 Series Computers
IBM --	IBM 360/370 Series Computers
SUN --	Sperry UNIVAC 1100 Series Computers
BUR --	Burroughs 5700/6700/7700 Series Computers

APPENDIX G

STANDARD CHARACTER SET

Standard Character Set	Internal Machine Code	↓ Octal	Hollerith Card Code	ASCII Code	UASCI Code	EBCDIC Code
0	000000	00	0	060	060	360
1	000001	01	1	061	061	361
2	000010	02	2	062	062	362
3	000011	03	3	063	063	363
4	000100	04	4	064	064	364
5	000101	05	5	065	065	365
6	000110	06	6	066	066	366
7	000111	07	7	067	067	367
8	001000	10	8	070	070	370
9	001001	11	9	071	071	371
[	001010	12	2-8	133	133	
#	001011	13	3-8	043	043	173
@	001100	14	4-8	100	100	174
:	001101	15	5-8	072	072	172
>	001110	16	6-8	076	076	156
?	001111	17	7-8	077	077	157
b̸	010000	20	(blank)	040	040	100
A	010001	21	12-1	141	101	301
B	010010	22	12-2	142	102	302
C	010011	23	12-3	143	103	303
D	010100	24	12-4	144	104	304
E	010101	25	12-5	145	105	305
F	010110	26	12-6	146	106	306
G	010111	27	12-7	147	107	307
H	011000	30	12-8	150	110	310
I	011001	31	12-9	151	111	311
&	011010	32	12	046	046	
.	011011	33	12-3-8	056	056	113
]	011100	34	12-4-8	135	135	
(	011101	35	12-5-8	050	050	115
<	011110	36	12-6-8	074	074	114
\	011111	37	12-7-8	134	134	
↑	100000	40	11-0	136	136	
J	100001	41	11-1	152	112	321
K	100010	42	11-2	153	113	322
L	100011	43	11-3	154	114	323
M	100100	44	11-4	155	115	324
N	100101	45	11-5	156	116	325
O	100110	46	11-6	157	117	326
P	100111	47	11-7	160	120	327
Q	101000	50	11-8	161	121	330
R	101001	51	11-9	162	122	331

Standard Character Set	Internal Machine Code	Octal	Hollerith Card Code	ASCII Code	UASCI Code	EBCDIC Code
-	101010	52	11	055	055	140
$	101011	53	11-3-8	044	044	133
*	101100	54	11-4-8	052	052	134
)	101101	55	11-5-8	051	051	135
;	101110	56	11-6-8	073	073	136
'	101111	57	11-7-8	047	047	175
+	110000	60	12-0	053	053	116
/	110001	61	0-1	057	057	141
S	110010	62	0-2	163	123	342
T	110011	63	0-3	164	124	343
U	110100	64	0-4	165	125	344
V	110101	65	0-5	166	126	345
W	110110	66	0-6	167	127	346
X	110111	67	0-7	170	130	347
Y	111000	70	0-8	171	131	350
Z	111001	71	0-9	172	132	351
←	111010	72	0-2-8	137	137	
,	111011	73	0-3-8	054	054	153
%	111100	74	0-4-8	045	045	154
=	111101	75	0-5-8	075	075	176
"	111110	76	0-6-8	042	042	177
!	111111	77	0-7-8	041	041	132

THREE-DIMENSIONAL

GRAPHICS COMPATIBILITY SYSTEM

Version 2.0

DEVICE IMPLEMENTATION GUIDELINES

July 5, 1978

3-D GCS Version 2.0 Implementation Guidelines for Devices

I. Introduction.

This report

The following paragraphs develop guidelines for implementing new devices for 3-D GCS. Included are ~~will be~~ the definition of device-dependent Graphics Status Area initialization values, functional specifications for each of the 3-D GCS device-dependent routines, a discussion of factors involved in supporting intelligent refresh devices, and a description of a phased sequence for implementing new devices.

II. Graphics Status Area Initialization.

In this section, each of the device-dependent elements of the Graphics Status Area (GSA) will be listed along with the procedures required to calculate the value, if appropriate.

KBEAMX Logical pen x-position in raster units. Default position is lower left corner of the largest square on plotting surface. If screen is rectangular, the square is right- or bottom-justified.

KBEAMY Logical pen y-position in raster units. Default position is lower left corner of the largest square on the plotting surface. If the screen is rectangular, the square is right- or bottom-justified.

KBEAMZ Logical pen z-position. Default value is zero.

TRSVX Ratio of screen to virtual units in x-direction. Default is width of UDAREA in raster units divided by width of UWINDO (100.).

TRSVY Ratio of screen to virtual units in y-direction. Default is height of UDAREA in raster units divided by height of UWINDO (100.).

KMINSX Minimum x screen window (UDAREA) boundary in raster units. Default is the same as KBEAMX.

KMAXSX Maximum x screen window (UDAREA) boundary in raster units. Default is maximum address in x-direction.

KMINSY Minimum y screen window (UDAREA) boundary in raster units. Default is same as KBEAMY.

KMAXSY Maximum y screen window (UDAREA) boundary in raster units. Default is the top boundary of largest square in plotting surface.

KMINSZ Minimum z screen window (UDAREA) boundary in raster units. Default is same as KBEAMZ.

KMAXSZ Maximum z screen window (UDAREA) boundary in raster units. Default is the back plane of the cubical display volume for 3D devices. For 2D devices, the default is the same as KMAXSX.

KCOLOR Color or greyscale switch. Default is the actual color produced by the plot head when device is initialized.

TMRKSZ Marker horizontal size in raster units. Default is one per cent of the maximum physical x address but no smaller than one.

KHORSZ Horizontal hardware character position size in raster units. Default is width of smallest size character position.

KVERSZ Vertical hardware character position size in raster units. Default value is height of smallest size character position.

KLMRGN Left alphanumeric margin boundary in raster units. Default is normally zero (minimum plotting surface address in x-direction).

KRMRGN Right alphanumeric margin boundary in raster units. Default is maximum plotting surface address in x-direction.

KTMRGN Top alphanumeric margin boundary in raster units. Default is maximum plotting surface address in y-direction.

KBMRGN Bottom alphanumeric margin boundary in raster units. Default is zero (minimum plotting surface address in y-direction).

KGIN Graphics input device selector. Default is primary graphics input device on display.

KMINHX Minimum device x-boundary of last NEWSCALE. Default is KMINSX.

KMAXHX Maximum device x-boundary of last NEWSCALE. Default is KMAXSX.

KMINHY Minimum device y-boundary of last NEWSCALE. Default is KMINSY.

KMAXHY Maximum device y-boundary of last NEWSCALE. Default is KMAXSY.

KMINDX Minimum x-address of display surface in raster units. Default is zero.

KMAXDX Maximum x-address of display surface in raster units. Default for devices with variable size plotting surface is largest square using normal plotting medium.

KMINDY Minimum y-address of display surface in raster units. Default is zero.

KMAXDY Maximum y-address of display surface in raster units. Default for devices with variable size plotting surface is largest square using normal plotting medium.

KMINDZ Minimum z-address of display surface in raster units. Default is zero.

KMAXDZ Maximum z-address of display surface in raster units. Default is same as KMAXDX.

TRASTX X-resolution in raster units per inch. Default is KMAXDX divided by width of plotting surface in inches.

TRASTY Y-resolution in raster units per inch. Default is KMAXDY divided by height of plotting surface in inches.

TRASTZ Z-resolution in raster units per inch. Default is KMAXDZ divided by depth of plotting surface in inches (or width if 2D device).

KFMXMX Formatted display surface x-dimension in raster units. Default is maximum normal x-dimension.

KFMYMX Formatted display surface y-dimension in raster units. Default is maximum normal y-dimension.

KFMZMX Formatted display surface z-dimension in raster units. Default is maximum normal z-dimension (same as KFMXMX for 2D devices).

THDCPY Hard copy delay time in seconds. Default is the number of seconds of wait time required while a hard copy is being generated. Default is zero if no hard copy capability is available.

KPHXMX Maximum possible physical x-boundary in raster units. Default is largest possible address. This could include a full roll of plotter paper.

KPHYMX Maximum possible physical y-boundary in raster units. Default is largest possible y-address.

KPHZMX Maximum possible physical z-boundary in raster units. Default is largest possible z-address (same as KPHXMX for 2D devices).

TRESOX Actual physical x-resolution in raster units per inch. Default is KPHXMX divided by actual width of display surface in inches.

TRESOY Actual physical y-resolution in raster units per inch. Default is KPHYMX divided by actual height of display surface in inches.

TRESOZ Actual physical z-resolution in raster units per inch. Default is KPHZMX divided by actual depth of display surface in inches (actual width is used for 2D devices).

KTYMNU Device type for menu generation. Possible values are: Graphics device = 1, Alphanumeric device = 2, Batch device =3.

KNRCLR Number of modifiable color table entries supported. Default is zero for devices which have no modifiable color table.

KBAUDR Speed of communication line in characters per second. Default is usual speed for device.

KPCHAR X,Y component buffer. Used by GCSPEN as a work area. Default should reflect impossible or unusable value.

KPREVC Color which terminal is currently generating. Default is same as KCOLOR.

KPREVI Current device intensity setting. Default is set to sixty percent of maximum screen intensity.

III. 3-D GCS Device-Dependent Routines.

This section describes the functions embodied in each of the 3-D GCS device-dependent routines. This discussion will include interface specifications, functional specifications, and indication of whether the function can be supported at the display-device either in hardware or software, relationships to other functions, and special instructions which may be required for 3-D display-devices. The functions will be listed alphabetically within the two groups, user-interface and GCS-internal.

UHDCPY

Comments: This routine causes a hard copy of the current display surface to be produced if an on-line immediate hard copy facility is available for the display device. A delay can be programmed by checking the GSA variable THDCPY which indicates the number of seconds of delay required. The delay can be generated by sending an appropriate number of 'NUL' or 'SYN' characters to the display device based on THDCPY seconds times the contents of KBAUDR (communication line speed in characters per second).

This function is null for display devices with no on-line immediate hard copy capability.

URESET

Comments: This routine returns the Graphics Status Area to its initial condition. Aside from setting the device-dependent GSA elements to agree with those in the Block Data subroutine for this device before compilation, no further modification is required.

This routine can not be supported at the display-device.

UWAIT(TIME)

TIME indicates the amount of time which GCS will wait before sending further commands to the display.

Comments: This routine is usually implemented by sending a sufficient number of null-operation control characters across the communication line to cause the required delay.

The KBAUDR GSA element specifies the line speed in characters per second. For batch devices, this routine is a null-operation.

This routine cannot be supported at the display.

GCSALF(ICHAR)

ICHAR is one ASCII character, right-justified, zero-fill.

Comments: This routine causes one hardware character to be displayed after the terminal is placed in alpha mode and KKMODE is set to reflect alpha mode. If a hardware character generator is not available and if the device is not intelligent, a device-indpendent hardware character simulator version of GCSALF is available. If a hardware character generator or simulator is available at the display, the printable character is simply sent to the display. Note that on three-dimensional devices, the characters are always produced on the screen plane. Only printable ASCII characters are output from this routine. The receipt of control characters implies that the terminal is is to be placed in alpha mode and that the character size is to be updated, if necessary.

Two control characters have special extra meaning for intelligent displays. A value of 001_8 indicates start of alpha string and 002_8 indicates end of alpha string. This allows strings of characters to be processed as one primitive operation at the display.

To update the character size, on input the value of KSIZEF is stored in KPREVS and the new sizes are stored in KHORSZ and KVERSZ.

This function may be supported at an intelligent display-device either by sending single characters or be using the 001_8 and 002_8 brackets to send entire character strings.

GCSARC(X,Y,ANGLE,RADIUS,ICHECK)

X,Y	are the coordinates of the center of the circle defined by the arc.
ANGLE	indicates the angle the arc will subtend and the direction of travel of the arc. A positive angle is counterclockwise, a negative angle is clockwise.
RADIUS	specifies the radius of the circle.
ICHECK	is set to 1 if hardware arc generation is not available or the arc was not drawn for this request. It is set to zero if the requested arc was drawn.

Comments: If hardware arc generation is available and if the arc can be drawn and does not intersect the window, the flag ICHECK is set to zero and the appropriate instruction is sent to the display. If, for some reason, the arc can not be drawn at the display, the flag is set to 1. Reasons which might apply are lack of an arc generation facility at the display device, intersection of the arc with the window, or the viewing and positional transformations are such that the arc cannot be drawn in the current X,Y plane by the hardware.

This function may be supported at the device if the arc to be drawn does not intersect the window. If it does intersect the window, GCSARC may calculate the visible arc segments and issue several instructions to draw each segment. However, the instructions to perform these calculations are somewhat complex and it is usually left to UARC to draw the desired arc.

GCSBEG

Comments: This routine performs any initialization functions required by the display. For intelligent devices, it may pass the initialization request to the device. For 3-D display-devices, it may have to initially load the viewing transform parameters to their default values.

GCSBPK(IX,IY,ICHAR)

IX,IY is the location to be marked.

ICHAR is the fill character for the designated raster position. For some raster devices, this argument may be used to indicate the number of raster positions along the raster to be marked starting at the indicated raster position.

Comments: This routine maintains the internal raster buffer for raster devices such as line printers, alphanumeric terminals without cursor addressing, and direct support of electrostatic printers. The buffer is a Fortran Common Area GCSBUF which contains storage for the entire frame or a page of the frame. The IX,IY values are mapped into buffer locations and the character (after translation into internal computer character set) or raster positions are stored there.

This routine may be supported at the display-device if an intelligent raster-scan terminal or an unintelligent raster-scan terminal with local raster storage. In this case, this routine should be eliminated with GCSPEN sending the vector request directly to the terminal. This will require implementation of the scan conversion at the terminal either in hardware or software.

GCSCLR(INDEX,CLRVEC)

INDEX is the color table entry being defined.

CLRVEC is an array of seven elements. Each element indicates a percentage of maximum participation for a particular hue. Basic hues are in the order: RED, GREEN, YELLOW, BLUE, MAGENTA, CYAN, and WHITE.

Comments: This routine allows redefinition of color table entries for display devices supporting a modifiable color table. For all other devices, this routine is null.

If INDEX exceeds the maximum color table index, the value of INDEX modulo the number of color table entries available should be used.

Since the values in CLRVEC represent percentages of the maximum participation for a particular basic hue, they can also be thought of as percentages of maximum intensity. The actual intensity value for the hue can be obtained by multiplying the maximum intensity setting by the CLRVEC entry divided by 100. Note that if only a three entry color table is available (red, blue, green; or magenta, yellow, cyan), the complementary hues may be obtained from the appropriate components each of which has the intensity setting desired for the unavailable hue. For example, 75% yellow may be obtained on additive color systems from 75% red and 75% green.

GCSCLS

Comments: This routine closes the currently open segment at the display device. Note that if the device draws vectors or characters by some kind of vector string or character string, this string should be terminated prior to closing the segment. Failure to do this when required can cause misinterpreted information to be displayed when the segments are being posted.

GCSDEL(NAMSEG)

NAMSEG is the number of the segment to be deleted.

Comments: This routine sends a command to the display device causing the requested segment to be deleted. For display devices which have no inherent segmentation capability or requirement, the deletion is handled entirely by UDELET and this routine is null. If NAMSEG has value zero, all segments are to be deleted.

Note htat when the delete is requested at the display device, the deleteion should occur immediately (i. e., the segment should be removed from the display surface.

GCSDSP

Comments: This routine transmits the user display area location to the display-device. This will only be required for 3-D devices which perform the viewing transform and clipping at the display-device.

GCSEND

Comments: This routine performs any termination functions required by the display-device. For intelligent devices, it may pass the termination request to the device.

GCSERA

Comments: This routine provides a clean plotting surface by either erasing the plotting surface or by advancing the plotting medium to an unused frame.

This function may be supported at the display-device. In this case, only an indication that the erase must occur need be transmitted to the device.

Note: in earlier versions of GCS, this function was incorporated within a device-dependent UERASE. UERASE has been made device-independent. It now calls GCSERA to actually perform the erasure.

GCSFMT(INDEX,ASPECT,WIDTH)

INDEX is the GCS index number for the supported format type. A listing of the formats supported may be found in UFORMT.

ASPECT is the aspect ratio of the requested format.

WIDTH is the width of the format in inches.

Comments: This routine corrects the display surface format boundaries established by UFORMT. Since UFORMT centers the format on the physical display surface boundaries, no other action is required if that position is appropriate for the requested format. This routine will only be different from the null function for devices with variable size plotting surfaces such as computer-output-microfilm (COM) recorders or pen plotters.

For COM recorders, the appropriate formats should be mapped onto the address space in such a way that when the COM device is configured with the camera/lens combination for the requested format, the format boundaries exactly match the edges of the frame on the film.

For pen plotters, paper (or other medium) of the appropriate size is requested to be made ready on the plotter surface.

If INDEX has value zero, GCSFMT will have been invoked by UDIMEN. In this case, adjustments as necessary to support the requested aspect ratio and width should be made.

GCSFRA(INDEX,OPCODE)

INDEX is the numeric identifier for the frame.

OPCODE is the action to be performed related to the designated frame.

Comments: This routine processes the frame manipulation function requests of "open", "close", "show", and "unshow". Since these operations are currently supported only in an intelligent display, this function becomes a null routine for unintelligent devices. The function of this routine is primarily to reformat the requested operation as necessary and transmit it to the display.

This is an interface routine which communicates with a program executing in the display-device general processor if the display is intelligent.

GCSIN(ICOUNT,IBUF,IPRMPT)

ICOUNT indicates the upper limit of the number of ASCII characters to be received as input.

INBUF is a buffer of length ICOUNT in which the ASCII characters will be placed right-justified, zero-fill.

IPRMPT is an array containing the number of prompt characters to be produced in the first word and the prompt characters in the succeeding words, one ASCII character/word, right-justified, zero-fill.

Comments: This is primarily in interface routine to the operating system function which actually sends the prompt and obtains the characters. If less than ICOUNT characters are received, the remaining buffer positions should be zeroed.

This function can not be supported at the display device since it is really not device-dependent but operating system dependent. However, to insure correct linkage editing on all operating systems, it is grouped with the device-dependent routines. Note that on many operating systems, a routine has been written to perform this function (see 3D GCS Host-Computer Implementation Guidelines).

For batch devices, this routine issues an error message and returns null characters.

If the operating system routine does not handle prompts, the prompt may be generated by the following statements prior to calling the operating system input routine:

```
IFRAME=KFRAME
KFRAME=0
CALL GCSOUT(IPRMPT(1),IPRMPT(2),0)
KFRAME=IFRAME
```

GCSINN(ICOUNT,ISTRNG)

ICOUNT indicates the number of ASCII characters to be read from the terminal.

ISTRNG is a buffer into which ICOUNT characters will be placed.

Comments: This routine accepts strings of ASCII characters from the terminal. Priop to issuing an input operation using GCSIN, the terminal is placed in the appropriate mode to activate the logical text device, usually realized as a keyboard. If ICOUNT is less than or equal to zero, no input operation takes place.

This function can not be supported at the display.

GCSMAP(XIN,YIN,ZIN,XOUT,YOUT,ZOUT)

XIN,YIN,ZIN	are the coordinates before projection in viewing units.
XOUT,YOUT,ZOUT	are the coordinates after projection is applied in viewing units.

Comments: This routine performs orthogonal or perspective transformations for devices which do not perform viewing transformations at the display-device. In this case, a version of this routine is provided. For devices which do perform the viewing transformation at the display-device, this routine consists simply of the input coordinates mapping into the output coordinates.

GCSMOD(NAMSEG,NAMAT,NEWVAL,ILOC)

NAMSEG is the number of the segment one of whose attributes is to be modified.

NAMAT is the number of the attribute to be modified:

0 = Visibility
1 = Highlighting
2 = Detectibility
3 = 2D Translate Image Transformation
4 = 2D General Image Transformation
5 = 3D Translate Image Transformation
6 = 3D General Image Transformation

NEWVAL contains the new setting for the attribute of the specified segment. If NAMAT has values 0 - 2, NEWVAL will have values 0 = OFF or 1 = ON. Otherwise, NEWVAL is an array containing the parameters for the image transformation requested.

ILOC is the record number within the GCS library work file where the segment is located if segmentation is occurring entirely in the host-computer.

Comments: This routine causes the indicated segment attribute to be modified as directed at the display device. Highlighting should be supported by blinking whenever possible.

For devices which support segmentation at the display device, this routine sends the requested segment attribute modifications to the display device. Note that if the device does not support image transformations at a particular level, requests for such transforms should not be sent but ignored or processed as well as possible.

For other devices, segmentation is supported within the host by saving the segments in the library work file. The segment attributes are contained in the segment table which is inaccessible to the device-dependent routines. However, the current image transformation is stored in the segment header as follows:

Header Word 5 indicates the number of words in the segment header extension in which are stored the transformation.

Header Word 6 and the following words, if required, contain the current image transformation.

GCSMOD (cont.)

If the current appearance of the segment is changed by the attribute modification, then the display surface must be updated to reflect the change unless posting modifications is being delayed (KPOST .EQ. 0).

GCSMRK(INX,INY,INZ,IMARK)

INX,INY,INZ is the location of the marker (symbol) in raster units.

IMARK is the number of the marker to be generated.

Comments: This routine produces a marker at the indicated location in raster units. At least ten markers are supported by GCS. Marker 0 is always a point and calling GCSMRK with IMARK .EQ. 0 will result in point plotting and should generate point plotting display commands if they are available. If markers are not supported at the display device, the markers are simulated by using GCSSIM. Descriptors of the markers have been defined for input to GCSSIM. This version of GCSMRK exists in current device libraries. Note that, even in this case, IMARK .EQ. 0 is produced by the special call to GCSPEN with IPEN .EQ. 2.

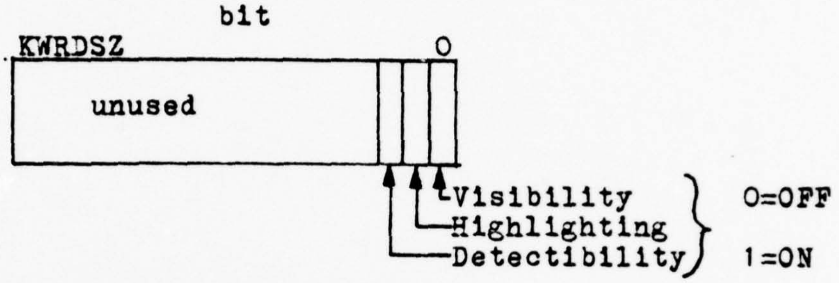
If markers are supported at the display device, then commands should be generated which will produce the requested markers at the indicated location.

If IMARK exceeds the highest marker index, an IMARK modulo the number of markers available is used.

GCSOPN(NAMSEG,NRTSEG,ILOC)

NAMSEG is the number of the segment to be opened.

NRTSEG are the initial segment attribute settings as follows:

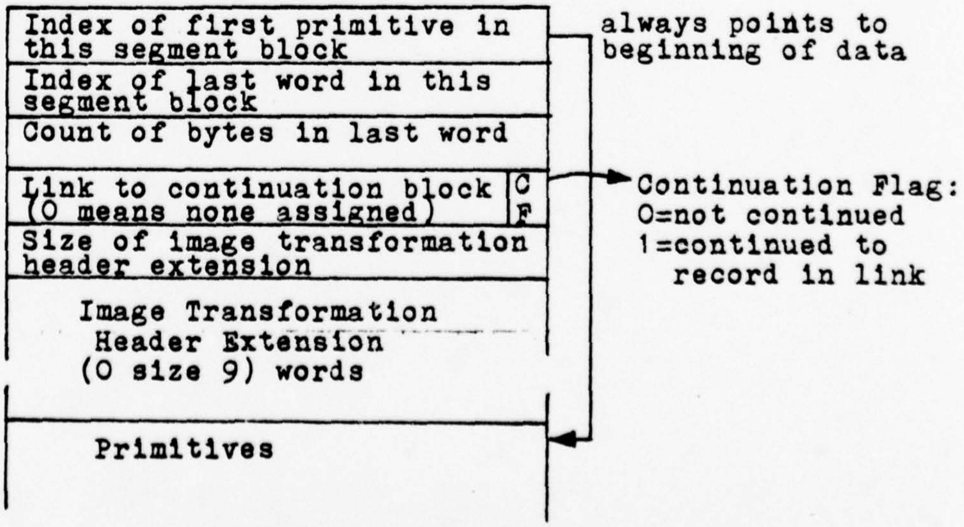


ILOC is the number of a record on the GCS library work file in which the segment can be stored if the display device does not have inherent segmentation capability.

Comments: This routine opens the requested segment at the display device. Before a segment is opened, any vector or character strings which are currently being processed must be terminated to avoid problems during posting.

If segmentation is maintained at the display device, an "open segment" command should be sent to the device along with the initial segment attributes.

If segmentation is being handled within the host-computer, the segment header must be initialized. The segment header is located in the first words of the first record of the segment and has the following format:



Index of first primitive in this segment block — always points to beginning of data

Index of last word in this segment block

Count of bytes in last word

Link to continuation block (0 means none assigned) | CF — Continuation Flag: 0=not continued 1=continued to record in link

Size of image transformation header extension

Image Transformation Header Extension (0 size 9) words

Primitives

GCSOPN (cont.)

Note that for devices which are byte-oriented and segmentation is maintained in the host-computer, the device-dependent segmentation routines for the Tektronix 4014 probably can be used without modification.

GCSOUT(ICOUNT,IBUF,IRESP)

ICOUNT indicates the number of ASCII characters to be transmitted to the display.

IBUF is the buffer containing ICOUNT ASCII characters right-justified, zero-fill.

IRESP is a count indicating the number of acknowledgement characters required from the terminal. This is only used for supporting intelligent devices.

Comments: This is the interface routine between GCS and the operating system routine which sends characters to the terminal. It is called by all GCS routines which need to transmit to the display-device. If ICOUNT=0, the operating system is requested to flush any internal buffers.

This routine cannot be supported at the display-device.

For batch device, this routine performs only one function: it intercepts GCS error messages and displays them on the system print file.

For supporting intelligent displays, the IRESP argument can be used to request an acknowledgement from the device before further characters are sent. It is usually used when sending functions which may require a large execution time at the display.

For devices without inherent segmentation capability, the initial action of this routine should be to save bytes being directed to the plot device if a segment is open. This is acccomplished by passing the input to device-dependent routine GCSPRM. When a segment is open, the bytes are only directed to the display surface if the segment is being constructed visibly (i. e., the open segment has visibility attribute of "on"). If no segment is open, the bytes are always directed to the display surface.

GCSPEN(IX,IY,IZ,IPEN)

IX,IY,IZ is the end point of the line to be drawn in raster units.

IPEN indicates the operation to be performed. Zero specifies a move; one specifies a point; three specifies a buffer flush request; and four specifies a dashed line.

Comments: This routine is the basic drawing routine for each device. Several variables have been set aside in the GSA for use by this routine. The array KPCHAR is for use as an internal work space and usually is where the command to be sent to the display-device is constructed. KPREVX, KPREVY, and KPREVZ are updated with the value of IX, IY, and IZ to always point to the current pen position when GCSPEN terminates. GCSPEN may also need to fetch additional information to insert intensity, blink, or other parameters into the command. The variable KKMODE is updated to reflect any mode changes initiated by GCSPEN.

If the display-device is such that a separate invisible line mode is used, GCSPEN will set KMOVEF to one whenever he will exit with the display-device primed for a blank line. Otherwise KMOVEF is set to zero or left at zero which is the default.

For two-dimensional devices, the IZ and KPREVZ variables may be ignored.

For terminal devices, output from GCSPEN is sent through GCSOUT. For batch devices, output is written on the appropriate files either directly or using a device specific support package.

Note: GCS assumes that the plotting surface is addressed from zero to some maximum positive number. If the actual address range contains negative numbers, it is the responsibility of GCSPEN to add in the appropriate displacement before building the display processor commands.

GCSPOS(ISEG,ILOCAT)

ISEG is the number of the segment to be posted.

ILOCAT is the record number of the segment location if segmentation is maintained in the host-computer.

Comments: This routine prepares the display surface for posting a segment and also posts (i. e., causes the segment to be displayed) the segment.

ISEG haveing value zero indicates a request that the display surface be prepared for posting, if required. For non-refresh display surfaces, this indicates a clearing of the display surface in preparation for redrawing the retained segments. For refreshed display surfaces, this indicates that all non-retained information (information not in retained segments) be deleted. In either case, when all segments have been posted, the display surface should not contain any information prepared outside of retained segments.

If ISEG has value other that zero, the segment should be displayed on the display surface it its visibility attribute is set to visible. For displays which maintain segmentation at the device, no action should be required since the display of a segment is controlled by commands from GCSMOD, GCSOPN, or GCSDEL. If, however, the host-computer maintains the segments, the segment primitives must be copied to the display surface (determination of visibility occurs in the routine which calls GCSPOS).

Note that for byte-oriented devices with no inherent segmentation capability, the Tektronix 4014 version should be usable unmodified.

GCSPRM(INCNT,INBUF)

INCNT is the number of bytes of information to be inserted in the segment.

INBUF is the buffer containing INCNT bytes right-justified, zero-fill.

Comments: This routine is only required for display devices which do not have inherent segmentation capability. The function of this routine is to receive bytes from GCSOUT and to pack the bytes into the segment which is saved on records of the GCS library work file. Note that for byte-oriented devices, the version of this routine in the Tektronix 4014 library should be usable without modification. If the input to GCSOUT is different from 8-bits, this will require that during execution of GCSPRM the value of KBYTEL should be set to the appropriate value and the value of ICPW (in GCSPRM) should be recomputed.

GCSPST(OPTION)

OPTION is a standard form (four-character, left-justified, blank-fill) UPSET option name passed from UPSET.

Comments: This routine is responsible for any device-dependent parameter setting required by the parent UPSET call.

One option must always by processed in GCSPST. This is the "SETDASH" function. If the value of KDASHT is less than 10, this indicates that a hardware dash mode is requested. The value 9 is reserved for dotted lines and 1 through 8 for the first eight hardware dash modes. If the display-device does not support hardware dotted lines, the 9 in KDASHT should be replaced with 944. If the display-device supports less than eight hardware dash modes, unsupported dash mode numbers in KDASHT should be replaced by 56.

Other UPSET options may be checked for as required by the display-device and appropriate action taken.

This function cannot be supported at the display-device. However, intelligent devices may be passed the requested setting for local processing. Intelligent display-devices which support mode setting will require a parallel routine to receive the requests sent by GCSSET.

The setting of certain hardware attributes should also be performed in this routine. These are listed below:

SPACE (device or virtual)
PROJECTION
WIDTH of lines
BLINK rate
COLOR
BRIGHTNESS (intensity)
SETDASH
MARKER selection
LETTER (character type)
CSPACING (character spacing)
ANGLE of text string
SIZE of characters
XSIZE of characters
YSIZE of characters
FONT
ASPECT RATIO

GCSPYR(XIN,YIN,ZIN,XOUT,YOUT,ZOUT)

XIN,YIN,ZIN	are the coordinates of the head endpoint of the line in viewing coorinates.
XOUT,YOUT,ZOUT	are the coordinates of the head end of the visible portion of the line in viewing coordinates.

Comments: This routine provides general clipping to the viewing pyramid for those display-devices in which the 3-D clipping functions is not performed at the display-device. The algorithm provided is that of Sutherland and Cohen as published in Principles of Interactive Graphics by Newman and Sproull.

For three-dimensional devices which perform the 3-D clipping function at the display-device, this routine whould set KPYRCP = -1 indicating that no clipping took place and then copy the input arguments into the output arguments.

This routine can only be supported in the display-device if the following conditions hold:

1) a clipping/divider is provided in the hardware or the Sutherland-Cohen algorithm is implemented in the display-device mini-computer.

2) GCSMAP is also supported at the display-device.

Although it is theoretically unnecessary for viewing transformations to be supported at the display-device, most devices which support 3-D clipping also support the viewing transformation in rotational hardware.

GCSRNM(NAMOLD,NAMNEW)

NAMOLD is the number of a currently existing segment.

NAMNEW is the new number by which the segment referred to in NAMOLD will henceforth be called.

Comments: This routine renames a segment. It is a null routine if segmentation is maintained within the host-computer.

For display devices which maintain segmentation at the device, the old name and new name are sent to the display device where the name change takes place.

GCSVPT

Comments: This routine transmits the viewport dimensions and position relative to the viewpoint/viewsite to the display device. For all devices which do not support viewing transformations at the display, this operation is null.

GCSVTD(XIN,YIN,ZIN,IXOUT,IYOUT,IZOUT)

XIN,YIN,ZIN contains the input coordinates in viewing space units after 3-D clipping and projection.

IXOUT,IYOUT,IZOUT contains the output coordinates in device raster units.

Comments: For 2-D devices or 3-D devices without clipping at the device, this routine accepts input virtual viewing space units on the projection plane and converts them to device units after clipping at the UWINDO boundaries. For these devices, a routine is provided.

For 3-D devices with clipping at the device, this routine accepts input virtual viewing space units and converts them to 3-D device units without any clipping operation.

The clipping function of this routine can be supported at the display. This requires that the window definition be sent to the display by GCSDSP and GCSWDO.

GCSVUD

Comments: This routine transmits the viewing vector specifications to the display-device if viewing transforms are to be applied at the display-device. For 2-D devices and 3-D devices without a local view transformation, the location of the view point and the view site or the view point and the direction cosines are sent in display-device units.

GCSVUE(XOBJ,YOBJ,ZOBJ,XEYE,YEYE,ZEYE)

XOBJ,YOBJ,ZOBJ contain the input coordinates in object space units.

XEYE,YEYE,ZEYE will receive the output coordinates in viewing coordinate units.

Comments: This routine converts object space coordinates to view space coordinates, if this function is to be performed in the host-computer. For 2-D devices and 3-D devices without local viewing transformation capability, this routine applies the viewing transformation. For 3-D devices with local viewing transformation capability, this routine simply copies the input coordinates into the output coordinates. Note that if viewing transformation is to be applied at the display-device, the following functions must also be performed at the display-device: 3-D clipping, projection, and windowing.

GOSWDO

Comments: This routine transmits the window boundaries to the display-device, if required. For 2-D devices and 3-D devices without a local scissoring facility, this routine is null. For 3-D devices with a local scissor facility, this routine transmits the boundary specifications in view space units to the terminal.

GCSWHR(IX,IY,IC)

IX,IY will contain the raster unit coordinates of the input cursor position.

IC is a mode, status, or option character obtained during the input operation. If IC = 0 upon entry to GCSWHR, only the character is requested.

Comments: This routine performs "positioning"-type input operations from the input device specified by GSA variable KGIN or the closest available equivalent. Whatever the form of the input coordinates when received, they must be converted to standard raster unit format. The mode, status, or option character should be converted to a form which can be identified by comparison with a Hollerith string.

This routine cannot be supported at the display but may require a parallel function to execute in an intelligent device.

IV. Guidelines for Intelligent Devices.

Display-devices which have computing poser of their own may be termed intelligent. Appropriate use of this intelligence can greatly increase the capabilities of the entire system. Most such display-devices have a refresh capability. Thus, much of the capacity of the general processor of the device is involved with display list management.

In GCS, a display list management facility has been provided to support dynamic pictures in a device-independent manner to the extent that the program will execute correctly even while using dynamic instructions on a non-refresh device. This is the segmentation facility. A complete name for this feature is transformed, segmented, unstructured display list. Several functions are available in GCS to create, manipulate, and delete segments. Segments are "opened" with a UOPEN (UFRAME) function and are closed with a UCLOSE (UFREND) function. Associated with each segment are attributes which may be modified by the UMODFY routine and by the image transformation routines UIMAGE, U3IMAG, UPLACE, and U3PLAC. Segments may be deleted with the UDELET function. All segments can be deleted with the UDELAL function. Associated with these user-callable routines are several device-dependent system routines which either simulate the functions or send the requests to the display-device. These are GCSCLS, GCSDEL, GCSMOD, GCSOPN, GCSPOS, GCSPRM, and GCSRNM.

In the display, a UOPEN (UFRAME) operation starts up a new segment for the indicated segment name. If a segment for that name already exists, a double buffering takes place which continues showing the elder of the two segments with identical numbers until the newer segment definition is complete (indicated by receiving a UCLOSE (UFREND) command). At this time, the elder segment is deleted and the newer segment is displayed (if visible). Since the double buffering if performed entirely by the device-independent routines, it need not be supported at the display device. If a segment is constructed visibly, it will be shown when completed. It can then be made invisible by executing a UMODFY (UNSHOW) command. If a segment is constructed invisibly, it will not be displayed until a UMODFY (USHOW) command is received making it visible. Subsequent UMODFY commands can make any segment visible or invisible as required.

The logical organization of the display list is as above. The actual physical organization and technique of memory management will depend on the particular display-device.

In addition to display list management routines, the intelligent device general processor will also need to interface available hardware functions with the device-dependent routines of GCS and to provide simulators as desired for device-dependent functions not available in the particular hardware under consideration.

In designing the format in which the device-dependent GCS routines communicate with the display-device, it is wise to select an encoding scheme that does not violate ASCII coding conventions. This requires that only seven bits be used as data and the parity-bit reserved for parity or marking even if the particular operating system does not require it. Following this procedure will allow connection of the terminal with the host-computer over standard communication lines without speciall communications software.

In selecting which functions should be simulated in an intelligent 3-D display-device or whether a particular hardware function should be utilized, the following table may be of use. For correct operation of 3-D functions, the cut-off in the list of functions should be as close to the top as possible always remembering that all functions below the topmost function to be implemented in the display-device must also be implemented in the display-device.

Curve Generation
Viewing Transformation
3-D Clipping
Projection (Perspective or Orthogonal)
Windowing
Hardware Character Generation
Hardware Dashed Line Processing
Vector Drawing

V. Implementation Sequence.

GCS is a modular system. Since many of the devices supported by GCS are relatively unsophisticated, many of the GCS device-dependent libraries contain simulators for many of the hardware functions. By using these simulators, it is possible to shorten the initial implementation time for new devices allowing later replacement of the simulators as time permits and as required by the display-device capabilities.

The following scheme for phased implementation of new devices has been designed to maximize the benefits mentioned above.

Phase 1: Identify the required initialization values for each device-dependent GSA variable and insert them in the Block Data and URESET subroutines.

Phase 2: For intelligent devices, design the communications data structures to be used between the device-dependent GCS routines and the program executing in the display-device. This program should be developed in parallel to the device-dependent GCS routines in the remaining phases if not already available.

Phase 3: Implement the GCSBEG, GCSPST, and GCSEND functions, if necessary. For 3-D devices with viewing transform capability, this may require initializing the transform matrix to the GCS default.

Phase 4: Implement the GCSERA, GCSPEN, GCSVUE, GCSVUD, GCSVPT, GCSVTD, GCSDSP, GCSWDO, GCSPYR, and GCSMAP routines. Note that for 2-D devices and 3-D devices with no local viewing transform capability, the GCSVUE, GCSVUD, GCSVPT, GCSPYR, and GCSMAP routines already exist and may be used unmodified. The GCSVTD, GCSWDO, and GCSDSP routines should also be used unmodified for these devices even if windowing is available at the device during the first iteration. The GCSOUT routine can usually be used without modification from one of the other libraries. Simulator or null routines can be used for the remaining functions.

Phase 5: Implement the GCSALF routine if a hardware character generator is available. This would replace the hardware character simulator accessing version of GCSALF.

Phase 6: Implement the segmentation routines, if the device has an inherent segmentation capability. If not, the simulator version of these routines is appropriate.

Phase 7: Implement the GCSIN, GCSINN, and GCSWHR routines if the device is interactive. These would then replace the null routines which probably were used until this phase.

Phase 8: Implement all other functions which do not yet support the capabilities of the device.

By following the sequence above, experience has shown that implementation of new devices should proceed smoothly.