

```
*****
*****
***                                     ***
*** Name:                             ***
***                                     ***
*** Project:   1      Programmer:  MWK  ***
***                                     ***
*** File Name: XGP.DOC [XGP,TVR]       ***
***                                     ***
*** File Last Written:  4:58 19 Apr 1973 ***
***                                     ***
*** Time: 18:34          Date: 15 Jul 1973 ***
***                                     ***
***           Stanford University       ***
*** Artificial Intelligence Project     ***
*** Computer Science Department       ***
*** Stanford, California              ***
***                                     ***
*****
*****
```

SAILON NUMBER XX.

XGP SOFTWARE

STANFORD ARTIFICIAL INTELLIGENCE LABORATORY  
OPERATING NOTE NUMBER XX.

APRIL 1973

draft - draft - draft - draft - draft - draft - draft - draft - draft  
XEROX GRAPHICS PRINTER SOFTWARE.

Bruce Baumgart  
Ralph Gorin

Robert Mass

Tovar Mock  
Fred Wright

ABSTRACT:

This operating note is an anthology of documentation  
for the several programs that are relevant to XGP output.

CONTENTS:

SYSTEM MONITOR UUD - Gorin.  
COPY COMMAND MODE - Gorin, Helliwell.  
CAL COMP MODE - Wright, Helliwell.  
REM's Things.  
TYFONT - Baumgart, Tovar.  
EDFONT - Tovar.  
CRE - Baumgart.  
XAP - Baumgart, Tovar.

## FONT FORMAT:

## WORDS 0-177:

XWD CHARACTER\_WIDTH, CHARACTER\_ADDRESS  
 ;Character address is the word number  
 ;in the file where the character  
 ;definition begins.  
 ;(unused by FC)

## WORDS 200-237:

CHARACTER\_SET\_NUMBER ;(unused by FC)  
 HEIGHT (in scan lines)  
 MAX\_WIDTH (in bits) ;(unused by FC)  
 BASE LINE (# of scan lines from top of character)

## WORDS 240-377:

ASCIZ/FONT DESCRIPTION/

## REMAINDER OF FILE:

;(consists of "character definitions, packed sequentially into  
 ;the file).

## EACH CHARACTER DEFINITION:

CHARACTER\_CODE,,WORD\_COUNT+2 ;(wc+2 is total space devoted  
 ;to this definition).  
 ROWS\_FROM\_TOP,,DATA\_ROW\_COUNT ;Count of blank rows from the  
 ;top. Count of non-blank data  
 ;rows. (pad with blank rows  
 ;at the bottom)  
 BLOCK WORD\_COUNT ;Data packed into words such that  
 ;an ILDB gets the next scan line  
 ;unless character width > 36  
 ;where next scan line starts  
 ;at first available word boundary.

## DATA MODES FOR XGP.

This document reflects the state of the software as of version 6.09W.

The information contained herein is subject to change without notice and should not be construed as commitment by anyone.

## Buffered modes:

1.	0 and 13	Stanford ASCII
2.	100 and 113	Extended ASCII
3.	10	Video
5.	14	Stanford Ascii & Vectors
6.	114	Extended Ascii & Vectors

## Dump modes:

7.	17	Video
9.	16	Stanford Ascii
10.	116	Extended Ascii
11.	15	Stanford Ascii & Vectors
12.	115	Extended Ascii & Vectors

## Implemented modes:

## Mode 17

In this mode data is obtained from the user under direction of his dump mode command list. The data is logically one or more groups, each group corresponding to (a portion of) one scan line. A group consists of a group command word followed by some number of video data words.

The group command word is interpreted as follows:  
 BYTE (1) MARK (11) LNSKIP (12) COLSKP (6) UNUSED, DWCNT

The paper will be advanced by LNSKIP (blank) lines before printing. (LNSKIP=1 gives normal single spacing; LNSKIP=0 allows this group to be printed on the same line as the previous group.) Then, starting at column Colskp in the scan line buffer DWCNT words will be taken from user data and deposited as video data. Finally, if the Mark bit is set then the paper will be marked for cutting. (Paper cutting is not exact so a mark should be preceded and followed by several blank lines.)

If the iowd of this command has not been exhausted, the next word will be fetched and interpreted as a group command word. The dump mode command list is logically processed as if it were exactly one dump mode command.

!!!! THE IMPLEMENTATION OF MODE 117 HAS BEEN DELETED !!!!

## Mode 117

In this mode each data word is interpreted as two commands to the XGP interface. Each halfword is interpreted as either a DATA0 or a CONO command to the interface as follows:

600000+DATA	→DATA0	XGP, [BYTE (20)1(16)DATA]	DATA IS:
400000+DATA	→DATA0	XGP, [BYTE (20)0(16)DATA]	12 VIDEO + 4 WIDTH
140000+DATA	→CONO	XGP, 140000+DATA	16 VIDEO
150000+DATA	→CONO	XGP, 150000+DATA	12 COLUMN ADDRESS
170000+DATA	→CONO	XGP, 170000+DATA	12 COUNT OF 1 BITS TO WRITE
			3 COMMAND BITS

Each scan line must be terminated by a halfword with 170004; the switch buffers cono. This instruction will not be executed as part of your XGP program. CAUTION: beware of using more than about 50. words (100. commands) in a single scan line. A longer line may not be processed fast enough.

All combinations not specified above are reserved for future expansion, except that zero in a halfword is always a no-op.

## XGP MODE 0 - DEFINITION

Scope of mode 0: Mode 0 is limited to one line of active text at any time. More general modes will be implemented as extensions of this mode.

XGP mode 0 shall interpret 7 bit bytes taken from the user's buffer as follows:

Byte	Usual meaning	Escape significance
0	Null - byte is ignored	Normal
1	Normal	XGP ESCAPE 1
2	Normal	XGP ESCAPE 2
3-10	Normal	Reserved
11	TAB	Normal
12	LF	Normal
13	Normal	Reserved
14	FF	Normal
15	CR	Normal
16-37	Normal	Reserved
40-176	Normal	Normal
177	ESCAPE	Normal

Normal means the definition of this byte in the current font will be printed. If this byte is undefined in the current font, it will be ignored.

ESCAPE cause the next byte to have an alternate meaning selected from the column "Escape significance".

TAB produces a column select to the column which is at least the width of a blank to the right of the current column position, and some multiple of 8 blank widths to the right of the left margin.

LF signifies the end of a line of text. The entire current line will be printed before any further characters are processed.

FF is like LF except that the paper will be spaced to the bottom of the currently active text area, spaced past the page bottom margin, marked for cut, and spaced past the page top margin before any further characters are processed.

CR causes a column select to the current left margin to be generated.

XGP ESCAPE 1 ('177&'001) causes the next 7 bits to be read as a special operation code. The following codes are proposed:

- 0-17 Font select. The code, 0 to 17 is taken as the font identification number of the font to use.
- 20-37 Reserved for future use.
- 40 XGP Column Selector  
The next 14 bits are taken modulo 4096 as the x position to print at next. (The intention is to allow arbitrary width spaces for text justification.)
- 41 XGP Underscore

The next 7 bits are taken as the scan line number on which to underscore. (Scan line 0 is the first scan-line in the character). The next 14 bits are taken modulo 4096 as the length of the underscore.

42 Line space.

This does a line feed and then takes the next 7 bits as the number of blank lines to insert before the next line.

43 Base-line adjust.

The next 7 bits are taken in two's complement as the base-line adjustment to the current font. The adjustment sticks until reset by another adjust command or a font select. The intention is to allow a font to be used for subscripts and superscripts. (Increment baseline for superscript, decrement for subscript).

XGP ESCAPE 2 ('177&'002) causes the next 7 bits to be taken as the column increment. This quantity is signed: 0-77 are positive increments 100 to 177 are negative increments (100 → -100, 177 → -1).

The escape significance of codes 3 through 10, 13, and 16 through 37 is not defined at the present time but reserved for future use.

MTAPE UUD for the XGP.

MTAPE CHAN,ADR  
where CHAN is the channel number on which the XGP has been opened is interpreted as follows:

ADR contains the operation selector. The data at ADR+1 and following depends on the operation selected.

#### Operation

- 0 Return error status.  
ADR+1/ major error code  
ADR+2,3,4/ error data
- 1 Font selection.  
ADR+1/ Font file name in sixbit  
ADR+2/ Font extension  
ADR+3/ PPN of font file  
ADR+4/ font identification number. (0 to 15.)  
(This UUD will skip if there is no error).

The font named will be read by the font compiler. It will be assigned the font identification number that you supply. The identification number is used only by the Font selection operator.

- 2 Read Margins  
ADR+1/ Top of page margin  
ADR+2/ Page body size  
ADR+3/ Bottom of page margin.  
ADR+4/ Left margin  
ADR+5/ Right side margin  
ADR+6/ Minimum interline space
- 3 Set Margins  
ADR+1/ Top of page margin  
ADR+2/ Page body size  
ADR+3/ Bottom of page margin.  
ADR+4/ Left Margin  
ADR+5/ Right side margin.  
ADR+6/ Minimum interline space



New copy features:

- 1) XGPLIST, new command, equivalent to COPY XGP:←.
- 2) New switch for font selects, format:

/FONT {#n} =<filename>{.ext} {[<p>, <pn>]}

If no ext is given, FNT is assumed. If no p,pn is given XGP,SYS is assumed. The #n is the font id number, if it is left out, it is assumed to be 0.

- 3) New format for /EXTRA switch, if you use /EXTRA=n. (yes that's an equivalence sign), then n is the number of scan lines to skip between text lines, not the number of extra line feeds to insert.

A	ASSIGN ASCII CODE TO IMAGE.
B	EXPAND/CONTRACT BY CONSTANT
$\alpha$ B	EXPAND/CONTRACT IN Y DIRECTION
$\beta$ B	EXPAND/CONTRACT IN X DIRECTION
$\epsilon$ B	SLANT CHARACTER (1/2 SLANTS TO 45 DEGREE ANGLE)
C	MAKE THRESHOLD CUT.
$\epsilon$ C	MAKE POLYGON IMAGE OUT OF BIT REPRESENTATION OF FONT.
D	ENABLE/DISABLE DELETION OF BABY POLYGONS (DEFAULT IS OFF).
F	LOCATE NEAREST POINT, $\epsilon$ F USE LIGHT PEN
G	LEVEL OF CORRESPONDING CHARACTER CODE
H	HISTOGRAM, " $\alpha$ H" , " $\beta$ H" BI-MODAL CUT.
I	INPUT TV PICTURE FROM DISK.
$\alpha$ I	INPUT CRE FILE
$\beta$ I	INPUT POLYGON FILE (CRUNCHED CRE)
$\epsilon$ I	INPUT FONT FILE
K	KILL POLYGON OR VERTEX
L	SHOW LAST BIT IMAGE
$\alpha$ L	SHOW CHARACTER FROM FONT IN FNTSEG
M	MOVE POLYGON TO NEXT IMAGE.
$\alpha$ M	MOVE TO NEW IMAGE
$\beta$ M	MIDPOINT LINE
$\epsilon$ M	MUNG ONTO GRID POINT (AS SEEN IN $\epsilon$ Y)
N	NEXT IMAGE
$\alpha$ N	PREVIOUS IMAGE
$\beta$ N	REPEAT NEXT IMAGE UNTIL A CHARACTER IS TYPED
$\epsilon$ N	REPEAT PREVIOUS IMAGE UNTIL A CHARACTER IS TYPED
O	OUTPUT CAREYE FILE
$\alpha$ O	OUTPUT CRE FILE
$\beta$ O	OUTPUT POLYGON FILE
$\epsilon$ O	OUTPUT FONT FILE
P	PLOT OUTPUT FILE
Q	MAKE FONT
$\alpha$ Q	MAKE 1/2 SIZE FONT
R	DISPLAY BIT MATRIX FOR THIS CHARACTER.
$\alpha$ R	ROTATE IMAGE, LEVEL OR POLYGON (ANGLE IN RADIANS)
S	SMOOTH
$\alpha$ S	SMOOTH AND KILL VIDEO INTENSITY CONTOUR
$\beta$ S	REPEAT 'S' FOR EACH IMAGE
$\epsilon$ S	REPEAT ' $\alpha$ S' FOR EACH IMAGE
T	TAKE A TV PICTURE
V	CREATE VERTEX AT CENTER
$\alpha$ V	CREATE NEW VERTEX AT CURRENT VERTEX
$\beta$ V	CREATE NEW VERTEX IN NEW IMAGE
W	CENTER IN THE WINDOW.
$\alpha$ W	CENTER Y-POSITION ONLY.
$\beta$ W	CENTER X-POSITION ONLY.
$\epsilon$ W	MOVE POINT SPECIFIED BY LIGHT PEN TO CENTER.
X	XTEND MODE COMMANDS
Y	DISPLAY SMOOTHED FORM
$\beta$ Y	DISPLAY VIDEO INTENSITY CONTOUR
$\alpha$ Y	DISPLAY BOTH OF ABOVE
$\epsilon$ Y	DISPLAY VIDEO INTENSITY CONTOUR MUNGED ONTO PIXELS
$\alpha$ Z	RESET LOGICAL CAMERA POSITION
$\beta$ Z	RESET DISPLAY

(MORE ON NEXT PAGE)

+        FETCH FILM NODE  
 α+        FETCH FIRST IMAGE FROM FILM  
 β+        FETCH FIRST LEVEL FROM FILM  
 c+        FETCH FIRST POLYGON FROM FILM

IF A NODE IS CURRENTLY BEING DISPLAYED, THESE COMMANDS AFFECT THAT NODE, OTHERWISE THEY AFFECT THE CAMERA (VIEWERS) POSITION. <CONTROL> MULTIPLIES BY 2, <META> MULTIPLIES BY 4.

;        MOVE LEFT (←) BY DELTA  
 :        MOVE RIGHT (→) BY DELTA  
 (        MOVE UP BY DELTA  
 )        MOVE DOWN BY DELTA

THESE COMMANDS AFFECT THE CAMERA (VIEWERS) POSITION.

\*        INCREASE MAGNIFICATION BY DELTA  
 -        DECREASE MAGNIFICATION BY DELTA

THESE COMMANDS CHANGE NODE BEING DISPLAYED.

.        FETCH COUNTER CLOCKWISE NODE IN RING.  
 ,        FETCH CLOCKWISE NODE IN RING.  
 ^        FETCH FATHER OF NODE  
 v        FETCH SON OF NODE  
 <        FETCH ARC [OF POLYGON OR VERTEX]  
 >        FETCH POLYGON [OF VERTEX]  
 ^        EQUIVALENT TO '<, >'  
 v        EQUIVALENT TO '<., >'  
 !        FLUSH NODE DISPLAY

THESE COMMANDS AFFECT THE PUSHDOWN LIST

U        PUSH NODE BEING DISPLAYED ONTO STACK  
 n        POP NODE OFF STACK AND DISPLAY IT

#### EXTENDED COMMANDS

XEROX    OUTPUT TV IMAGE TO XGP  
 HELP     DESCRIBE XTENDED FUNTIONS  
 DDT     INVOKE DDT IF PRESENT, RETURN WITH αP  
 EXIT     EXIT TO MONITER  
 ARCWID   SET CONSTANT FOR SMOOTHING  
 DISPLA   TURN ON DISPLAY  
 -DISPL   TURN OFF DISPLAY  
 KILVIC   FLUSH PRE-SMOOTHING LINES  
 GRID     TURN ON GRID  
 -GRID    TURN OFF GRID  
 CAMERA   SELECT CAMERA, "αS" BCLIP, "βS" TCLIP.  
 KILARC   KILL ARC VECTORS  
 CENTER   CENTER ALL IMAGES  
 POPJ     LEAVE TTY LOOP  
 BABYKI   TOGGLE FLAG WHICH BABY POLYGONS  
 SCALE    SCALE ALL IMAGES BY CONSTANT  
 xSCALE   SCALE IN X ALL IMAGES BY CONSTANT  
 ySCALE   SCALE IN Y ALL IMAGES BY CONSTANT  
 SLANT    SLANT ALL IMAGES BY CONSTANT  
 MUNG     FORCE VECTICES OF CURRENT POLYGON OR LEVEL ONTO PIXEL BOUNDARIES  
 POLYGO   MAKE HOLE INTO POLYGON  
 HOLE     MAKE POLYGON INTO HOLE

SORT SORT IMAGES ON FILM ACCORDING TO ASCII CODE  
READFO READ FONT FILE INTO POLYGON REPRESENTATION  
CNTFLG TOGGLE CENTERING WRT PIXEL GRID BEFORE CONVERTING TO BIT REPRESENTATION  
ORTHMU MUNG VERTICES WHICH FORM PERPENDICULAR LINES ONTO PIXEL GRID  
SETORTH SET MINIMUM LENGTH FOR ORTHMUNG  
ZERO RESET DATA STRUCTURE

VERTEX/ARC NODE.		POLYGON/REGION NODE.	
0	VERTEX-RING	0	POLYGON-RING.
1	ROW,,COL	1	DAD,,SON
2	TYPE,,RELOC	2	TYPE,,RELOC
3	-,,-	3	-,,-
4	ARC,,PDOPY	4	ARC,,NCNT
5	-,,PGON	5	-,,PGON
6	RT SEG,,LT SEG	6	-,,-
IMAGE NODE.		LEVEL NODE.	
0	IMAGE-RING	0	LEVEL-RING
1	-,,SON	1	-,,SON
2	TYPE,,RELOC	2	TYPE,,RELOC
3	-,,-	3	-,,-
4	-,,-	4	-,,NCNT
5	-,,-	5	-,,WIDTH
6	-,,-	6	NTIME,,PTIME
FILM NODE.		EMPTY NODE.	
0	CORESIZ	0	-,,AVAIL
1	-,,SON	1	-,,-
2	TYPE,,RELOC	2	TYPE,,RELOC
3	-,,AVAIL	3	-,,-
4	BLOCK COUNT	4	-,,-
5	-,,-	5	-,,-
6	NTIME,,PTIME	6	NTIME,,PTIME
SEGMENT NODE.			
0	SEGMENT RING		
1	-,,-		
2	TYPE,,300003		
3	LDEL,,RDEL		
4	LCOL,,RCOL		
5	LROW,,RROW		
6	LT,,RT		