RJ2594(33722)8/3/79
Computer Science

Research Report

USING PHOTO-COMPOSER EQUIPMENT FROM SAN JOSE RESEARCH VM SYSTEM (SJRLVM1)

James N. Gray
James C. King

IBM Research Laboratory
San Jose, California 95193

Copies may be requested from:
IBM Thomas J. Watson Research Center
Distribution Services
Post Office Box 218
Yorktown Heights, New York 10598

RJ2594(33722)8/3/79
Computer Science

USING PHOTO-COMPOSER EQUIPMENT FROM SAN JOSE RESEARCH VM SYSTEM (SJRLVM1)

James N. Gray
James C. King

IBM Research Laboratory
San Jose, California 95193

ABSTRACT: The Yorktown Formatting Language and PREP2 have been installed at San Jose Research. They allow script users to send documents to the photocomposition equipment in San Jose. This report serves as a users manual for these facilities.

Using Photo-Composer Equipment from San Jose Research VM System (SJRLVM1)

James N. Gray and James C. King

Introduction and Background

In the late 1960 ' s, IBM (Boulder) designed and built several computer controlled photographic "typesetting" machines. Their output is of very high quality on photographic print paper (about 800 spots/inch resolution). Two such machines (called **experimental printers**) reside in the San Jose GPD Information Systems Micrographics Laboratory (Building 26) although only one is operational at a time.

The Micrographics Laboratory also recently acquired an **APS5** photocomposer manufactured by AUTOLOGIC corporation. It provides many more fonts than the experimental printer and should be the machine of choice for new users. The IBM Yorktown Research Laboratory also has an APS5.

IBM developed a very sophisticated software package to format data for output on the experimental printer and on other devices. This software is called **TERMTEXT/Format** and provides a TERMTEXT/Format Language in which users may compose textual material for processing. Many San Jose PUBS people use this language as described in IBM IUP manual SH20-1372: *TERMTEXT/Format Language Guide.*

IBM Research in Yorktown Heights has provided the VM Script user with an alternative to TERMTEXT/Format by writing a Script-to-TERMTEXT conversion program which is now called the **Yorktown Formatting Language** (YFL). It accepts "roughly" Script/370 Version 3 input and does a "best effort" translation to TERMTEXT/Format Language. In addition, it has commands especially tailored to font selection, variable spacing, and equation processing. These commands are script-like extensions to the basic Script language. There are some Script features which are not supported (see Appendix D for a Script and YFL comparison.) YFL is well documented in: *YFL User's Guide* -- RC 6994 available from the library.

The default printing by YFL is as follows:

normal text uses the font:	W=PRESS(10)	(this font)
underlined text comes out in italic:	X=PRESS-ITALIC(10)	*(italic sample)*
overtyped text (bold):	Y=PRESS-BOLD(10)	**(bold sample)**
use .bi (like .us) for:	Z=PRESS-BOLD-ITAL(10)	***(bold italic sample)***
(see fonts samples in Appendix B.)		

Horizontal spacing is variable with the photo-composer so the lines in format mode will not match that of Script line-printer or typewriter output. Most horizontal spacing commands use 1/10 inch per unit, since there is no fixed character size. For example. .ll 60 means 6 inches. The same issue exists for vertical spacing where commands use 1/6 inch for one line. See YFL manual for details.

The special equation processing described in the YFL Users Guide and in the PREP2 manual is not generally available in San Jose. The reason is that Yorktown enhanced their TERMTEXT/Format System with an overlay command. Most "normal" systems (e.g., here in

San Jose) do not have this feature. Yorktown and San Jose are collaborating on a common TERMTEXT which will support equation processing on the APS5.

One more convenience program out of Yorktown is called PREP2. It is a preprocessor for Script (YFL) inputs which makes the use of subscripts, superscripts, bold, italic, and special characters much simpler. For example, to obtain a greek alpha one would normally type "(bgab/)bg" where b represents a backspace. Using PREP2 one simply types ".AL".

Use from SJRLVM1

A general purpose command (EXEC) is on mini-disk RESEARCH 200 and is called **PHOTO**. (See Appendix C for a listing of the QMARK files). If one types:

```
PHOTO DOG
```

the file "DOG SCRIPT" will be processed by YFL producing a file "DOG$ TERMTEXT" and a file "DOG$ YFLDIAG". The first file is the YFL conversion (a packed TERMTEXT/Format Language input) the second file lists errors found by YFL, if any. The file "DOG$ TERMTEXT" is surrounded by JCL and shipped to the MVS machine SJNMAS2 (alias SNJMVS3) in Building 26. There the TERMTEXT/Format processor is run producing a magnetic tape which is automatically routed to the APS5 for processing. The final output is put in an envelope and sent to our Computing Center Output Area by courier. This complete process typically takes about 24 hours. Confidential output is not couriered and must be picked up by the author at the Micrographics Laboratory in Building 26. (use the "CONF" option on the PHOTO command).

The PHOTO command will type YFL diagonstics at the terminal and if any serious errors are reported will query the user of his desire to stop or go on. Unless directed otherwise, PHOTO will erase all temporary files which it creates. If the original file is "DOG SCRIPT" PHOTO may create (and erase) "$DOG SCRIPT", "DOG$ SCRIPT", "DOG$ YFLDIAG", "DOG$ TERMTEXT", and "$DOG PREPDIAG". *No check is made for useful files a user may have by these names and no warning is given when they are overwritten or erased.*

A log of the TERMTEXT/Format processing done on the SNJMAS2 machine in Building 26 will be returned to your virtual machine's card reader and will be named "DOG OUTPUT", (if the original file was called DOG).

In order to compose the proper MVS JCL decks with which to surround your Job a file named "your-userid PHOTO" is required. A file "PHOTO PHOTO" is on Research 200 and should be copied, edited, and renamed for your use. Note that the fields must begin at fixed columns.

Confidential output may be obtained by:

```
PHOTO fn (CONF
```

Unfortunately, confidential output is not couriered and must be picked up by the author at the Micrographics Laboratory in Building 26.

The older experimental printer processor may be used by:

```
PHOTO fn (XPRT
```

Note that the fonts available on the APS5 are not the same (in name or content) as on the experimental printer and one should not expect to switch back and forth between the machines if

special fonts are used. The casual Script user (using YFL) can expect to use either machine with nearly equivalent results.

Preprocessing by PREP2 may be specified by:

```
PHOTO fn (PREP2
```

In this case PREP2 is invoked *before* YFL to expand font and special character abbreviations. Since PREP2 may be new to most users a brief summary of its advantages and use is included in the next section.

PREP2

PREP2 is a program written by George Leeman at IBM Research/Yorktown and documented in RC6948 - *PREP2: A Program to Simplify the Typing of Manuscripts*. The user wishing to do subscripts, superscripts, and special mathematical symbols will find PREP2 especially useful. Appendix A of this document shows abbreviations currently supported by PREP2 (in San Jose) for the APS5. For example in order to produce A_1 or $\alpha+\beta$ one simply codes A.SUB/1/ and .AL+.BE, respectively.

The abbreviations used by PREP2 are always five characters or less and denoted by a period followed 2 capital letters (e.g. .AL, .FR12). The PREP2 user must therefore develop the habit of using lower case letters for script commands (e.g., .sp instead of .SP).

In addition to the built-in abbreviations of PREP2, the user may also supply his own set of abbreviations and/or override those built into PREP2. These are supplied in a file called "`TABLE PREP2`" or "`fn PREP2`", where "`fn`" is the name of the script file being processed. See the QMARK file listing for PHOTOPEP in Appendix C for details.

In general, PREP2 abbreviations are string replacement functions of zero or more arguments. For example:

.AL	is the *ZERO* argument function producing α,
.BD/the dog/	is a one argument function producing the single argument "the dog" in font PRESS-BOLD(10) (**the dog**).

PREP2 automatically prefaces your file with secondary font definitions (YFL ' s .sf command) for many extra fonts. The exact list is given in Appendix A. Any additional secondary fonts needed can be included in the TABLE PREP2 file.

Notes and Hints

• YFL does not support **EZ-Script** commands (e.g., &P, &B). To get **EZ-Script** support, replace the ".ez on" by ".im ez" which causes a set of YFL macros to be included in the file. (See the file "EZ SCRIPT".)

• YFL does not support the Script command ".tr".

• As a temporary problem the APS5 makes errors in processing single and double quotes as entered as the keyboard characters. The result is that double quotes (") appear as a single quote (') and single quote appear as a (`). Also the underscore character (—) is treated as a null character, use E□F to get the underscore character. These should only be temporary problems!

• BEDFORD and COURIER are monospace fonts.

• YFL has a new command ".bl" which, by default, translates % to monospace blank. We have reversed this default by placing a ".bl off" in the header file PHOTO uses to supply the courier information (file -- PHOTO JCLC).

• The TERMTEXT/Format Processor for the APS5 will accept codes for hexadecimal characters as follows: the two hex digits (as characters themselves) separated by a backspace. Suppose that the user wants to enter the symbol ‡ from a PRESS font. Examining Appendix B one finds that ‡ is hexadecimal "DA". TERMTEXT for APS5 will accept this as D␈A (where ␈ represents a backspace).

• TERMTEXT/Format is run on a Building 26 MVS system. The JCL is set up to route two files back to your card reader, one Class A and the other Class Z. The Class A file has TERMTEXT/Format error messages and should be examined carefully. The Class Z file is the run-time log from MVS and is less interesting unless some abnormal condition arises.

• Certain error messages in the "OUTPUT" file sent back to your card reader (Class A) by TERMTEXT/Format are normal and should be ignored. Two such are:

DHZ1085U FONT PARAMETER IN /CN/ OR /PN/ INCORRECT; ...
DHZ1054W /LS/ OF 0 MAY CAUSE STRIKEOVERS; ...

These occur roughly once per page and are caused by YFL forcing TERMTEXT/Format to do things it doesn't like!

• Users of PREP2 should beware of the use of ! (continue line) and of # (new line). Use the PREP2 verbs (.EX and .PD to avoid such problems).

• PREP2 does not automatically process imbedded files. (i.e., it ignores ".im" commands.)

• When using PREP2 a temporary file is made to be passed to YFL for processing. That file has eleven (11) lines of secondary font definitions placed at the beginning of it. Consequently, when using PREP2, the line numbers reported in YFL error messages will by off by 11 lines. Subtract 11 from the YFL error messages to find the line in error in the original file.

In preparing files be cautious of the your editor's treatment of backspace characters (i.e., canonicalization). For example, D␈A will be canonicalized to A␈D by most editors, resulting in the character [instead of ‡. This is particularly troublesome in processing PREP2 abbreviation tables.

References

APS-5 FONT GUIDE, Obtained from Ron Pierce, R64-H35, STL, tel: 115-3-4943.

TERMTEXT/Format Language Guide, IBM IUP manual SH20-1372.

YFL User's Guide -- RC 6994, IBM T. J. Watson Research Center, Yorktown Heights, NY.

PREP2: A Program to Simplify the Typing of Manuscripts -- RC 6948, IBM T. J. Watson Research Center, Yorktown Heights, NY.

Default font declarations (YFL and PREP2)

The fonts defined by YFL are:

w=press(10)	The normal default font
x=press-italic(10)	The SCRIPT underline font
y=press-bold(10)	The SCRIPT bold font
z=press-bold-ital(10)	A bold italic font

The fonts added to this list by PREP2 are:

```
.sf l=apl(10);
.sf m=greek(10);
.sf n=greek(8,-3);
.sf o=greek(8,3);
.sf p=utility-5-math(10);
.sf q=press(10)
.sf r=press(8,-3);
.sf s=press(8,3);
.sf t=press-bold(10);
.sf u=press-italic(10);
.sf v=press-undr(10);
```

Users can then freely use font letters **a** to **k**.

Built-in Functions (may appear anywhere in the text)

Name	Form	Sample Input	Makes Output	Font Letter
boldface	.BD/x/	abc.BD/defGH12/ijk	abc**defGH12**ijk	t
	.BBx.EB	abc.BBdefGH12.EBijk	abc**defGH12**ijk	t
italics	.IT/x/	abc.IT/defGH12/ijk	abc*defGH12*ijk	u
	.BIx.EI	abc.BIdefGH12.EIijk	abc*defGH12*ijk	u
underlining	.UL/x/	abc.UL/edfGH12/ijk	abcedfGH12ijk	v
	.BUx.EU	abc.BUdefGH12.EUijk	abcdefGH12ijk	v
superscripts	.SUP/x/	abc.SUP/edfGH12/ijk	abc^{edfGH12}ijk	s
subscripts	.SUB/x/	abc.SUB/defGH12/ijk	abc$_{defGH12}$ijk	r

Table of abbreviations for APS5 using PREP2 (at SJRLVM1)

Some APL Operators (secondary font 1=APL(10))

Name	Mnemonic	Symbol
APL alpha	.AAL	α
APL asterisk	.AAS	*
APL ceiling	.ACE	⌈
APL comma	.ACM	,
APL downward arrow	.ADA	↓
APL decode	.ADE	⊥
APL difference	.ADF	-
APL epsilon	.AEP	∈
APL equal	.AEQ	=
APL floor	.AFL	⌊
APL greater than	.AGT	>
APL greater than or equal	.AGE	≥
APL implication	.AIM	⊃
APL intersection	.AIN	∩
APL iota	.AIO	⍳
APL left arrow	.ALA	←
APL left bracket	.ALB	[
APL left parenthesis	.ALP	(
APL less than	.ALT	<
APL less than or equal	.ALE	≤
APL minus	.AMI	−
APL multiplication	.AML	×
APL not equal	.ANE	≠
APL omega	.AME	ω
APL or	.AOR	∨
APL plus	.APL	+
APL question mark	.AQM	?
APL rho	.ARH	ρ
APL right arrow	.ARA	→
APL right bracket	.ARB	]
APL right parenthesis	.ARP	)
APL semicolon	.ASC	;
APL slash	.ASL	/
APL stroke	.AST	|
APL subset	.ASB	⊂
APL tilde	.ATI	~
APL upward arrow	.AUA	↑
APL union	.AUN	∪

Some miscellaneous symbols (secondary font q=PRESS(10))

Name	Mnemonic	Symbol
bullet	.BUL	•
cent sign	.CE	¢
degree sign	.DEG	°
em dash	.EM	—
en dash	.EN	–
exclamation point	.EX	!
fractions:	.FR12	½
	.FR13	⅓
	.FR14	¼
	.FR18	⅛
	.FR23	⅔
	.FR38	⅜
	.FR58	⅝
	.FR78	⅞
greater than or equal	.GE	≥
less than or equal	.LE	≤
left arrow	.LAR	←
left brace	.LBC	{
left bracket	.LBK	[
logical not	.LN	¬
minus	.MI	−
not equal	.NE	≠
multiplication	.ML	×
o umlaut	.OUM	ö
plus or minus	.PM	±
pound sign	.PD	#
right arrow	.RAR	→
right brace	.RBC	}
right bracket	.RBK	]
square	.SQ	□
u umlaut	.UUM	ü
vertical rule	.VR	|

Some mathematical symbols (secondary font p=UTILITY-5-MATH(10))

Name	Mnemonic	Symbol
infinity	.INF	∞
integral	.IN	∫
logical and	.LAN	∧
logical or	.LOR	∨
partial derivative	.PA	∂
prime	.PR	′
square root	.SQR	√
union	.UN	∪

Lower case Greek Letters (secondary font m=GREEK(10))

Name	Mnemonic	Symbol
alpha	.AL	α
beta	.BE	β
gamma	.GA	γ
delta	.DE	δ
epsilon	.EP	ϵ
zeta	.ZE	ζ
eta	.ET	η
theta	.TH	θ
iota	.IO	ι
kappa	.KA	κ
lambda	.LA	λ
mu	.MU	μ
nu	.NU	ν
xi	.XI	ξ
omicron	.OM	o
pi	.PI	π
rho	.RH	ρ
sigma	.SI	σ
tau	.TA	τ
upsilon	.UP	υ
phi	.PH	ϕ
chi	.CH	χ
psi	.PS	ψ
omega	.ME	ω

Upper case Greek Letters (secondary font m=GREEK(10))

Name	Mnemonic	Symbol
capital alpha	.CAL	A
capital beta	.CBE	B
capital gamma	.CGA	Γ
capital delta	.CDE	Δ
capital epsilon	.CEP	E
capital zeta	.CZE	Z
capital eta	.CET	H
capital theta	.CTH	Θ
capital iota	.CIO	I
capital kappa	.CKA	K
capital lambda	.CLA	Λ
capital mu	.CMU	M
capital nu	.CNU	N
capital xi	.CXI	Ξ
capital omicron	.COM	O
capital pi	.CPI	Π
capital rho	.CRH	P
capital sigma	.CSI	Σ
capital tau	.CTA	T
capital upsilon	.CUP	Υ
capital phi	.CPH	Φ
capital chi	.CCH	X
capital psi	.CPS	Ψ
capital omega	.CME	Ω

Lower case Greek Subscripts (secondary font n=GREEK(10))
(Letter X used to show relative position of subscript.)

Name	Mnemonic	X followed by Symbol
subscript alpha	.DAL	X_{α}
subscript beta	.DBE	X_{β}
subscript gamma	.DGA	X_{γ}
subscript delta	.DDE	X_{δ}
subscript epsilon	.DEP	X_{ϵ}
subscript zeta	.DZE	X_{ζ}
subscript eta	.DET	X_{η}
subscript theta	.DTH	X_{θ}
subscript iota	.DIO	X_{ι}
subscript kappa	.DKA	X_{κ}
subscript lambda	.DLA	X_{λ}
subscript mu	.DMU	X_{μ}
subscript nu	.DNU	X_{ν}
subscript xi	.DXI	X_{ξ}
subscript omicron	.DOM	X_{o}
subscript pi	.DPI	X_{π}
subscript rho	.DRH	X_{ρ}
subscript sigma	.DSI	X_{σ}
subscript tau	.DTA	X_{τ}
subscript upsilon	.DUP	X_{υ}
subscript phi	.DPH	X_{ϕ}
subscript chi	.DCH	X_{χ}
subscript psi	.DPS	X_{ψ}
subscript omega	.DME	X_{ω}

Lower case Greek Superscripts (secondary font o=GREEK(10,3))
(Letter X used to show relative position of superscript.)

Name	Mnemonic	X followed by Symbol
superscript alpha	.UAL	X^{α}
superscript beta	.UBE	X^{β}
superscript gamma	.UGA	X^{γ}
superscript delta	.UDE	X^{δ}
superscript epsilon	.UEP	X^{ϵ}
superscript zeta	.UZE	X^{ζ}
superscript eta	.UET	X^{η}
superscript theta	.UTH	X^{θ}
superscript iota	.UIO	X^{ι}
superscript kappa	.UKA	X^{κ}
superscript lambda	.ULA	X^{λ}
superscript mu	.UMU	X^{μ}
superscript nu	.UNU	X^{ν}
superscript xi	.UXI	X^{ξ}
superscript omicron	.UOM	X^{o}
superscript pi	.UPI	X^{π}
superscript rho	.URH	X^{ρ}
superscript sigma	.USI	X^{σ}
superscript tau	.UTA	X^{τ}
superscript upsilon	.UUP	X^{υ}
superscript phi	.UPH	X^{ϕ}
superscript chi	.UCH	X^{χ}
superscript psi	.UPS	X^{ψ}
superscript omega	.UME	X^{ω}

Upper case Greek Subscripts (secondary font n=GREEK(10))
(Letter X used to show relative position of subscript.)

Name	Mnemonic	X followed by Symbol
subscript alpha	.DCAL	X_{A}
subscript beta	.DCBE	X_{B}
subscript gamma	.DCGA	X_{Γ}
subscript delta	.DCDE	X_{Δ}
subscript epsilon	.DCEP	X_{E}
subscript zeta	.DCZE	X_{Z}
subscript eta	.DCET	X_{H}
subscript theta	.DCTH	X_{Θ}
subscript iota	.DCIO	X_{I}
subscript kappa	.DCKA	X_{K}
subscript lambda	.DCLA	X_{Λ}
subscript mu	.DCMU	X_{M}
subscript nu	.DCNU	X_{N}
subscript xi	.DCXI	X_{Ξ}
subscript omicron	.DCOM	X_{O}
subscript pi	.DCPI	X_{Π}
subscript rho	.DCRH	X_{P}
subscript sigma	.DCSI	X_{Σ}
subscript tau	.DCTA	X_{T}
subscript upsilon	.DCUP	X_{Υ}
subscript phi	.DCPH	X_{Φ}
subscript chi	.DCCH	X_{X}
subscript psi	.DCPS	X_{Ψ}
subscript omega	.DCME	X_{Ω}

Upper case Greek Superscripts (secondary font o=GREEK(10))
(Letter X used to show relative position of superscript.)

Name	Mnemonic	X followed by Symbol
superscript alpha	.UCAL	X^{A}
superscript beta	.UCBE	X^{B}
superscript gamma	.UCGA	X^{Γ}
superscript delta	.UCDE	X^{Δ}
superscript epsilon	.UCEP	X^{E}
superscript zeta	.UCZE	X^{Z}
superscript eta	.UCET	X^{H}
superscript theta	.UCTH	X^{Θ}
superscript iota	.UCIO	X^{I}
superscript kappa	.UCKA	X^{K}
superscript lambda	.UCLA	X^{Λ}
superscript mu	.UCMU	X^{M}
superscript nu	.UCNU	X^{N}
superscript xi	.UCXI	X^{Ξ}
superscript omicron	.UCOM	X^{O}
superscript pi	.UCPI	X^{Π}
superscript rho	.UCRH	X^{P}
superscript sigma	.UCSI	X^{Σ}
superscript tau	.UCTA	X^{T}
superscript upsilon	.UCUP	X^{Υ}
superscript phi	.UCPH	X^{Φ}
superscript chi	.UCCH	X^{X}
superscript psi	.UCPS	X^{Ψ}
superscript omega	.UCME	X^{Ω}

Below are listed a few of the fonts currently avaiable on the APS5. For a full list of all of the fonts available get an APS5 fonts manual.

This is a table of the **PRESS** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10			½	⅓	¼				⅛							
20				⅔										•	'	'
30							ı		⅜	‘		—	®		“	”
40		b									¢	.	<	(	+	|
50	&								⅝		!	$	*	)	;	¬
60	-	/									|	,	%		>	?
70									⅞		:	#	@	'	=	'
80		a	b	c	d	e	f	g	h	i	↑	. .	≤			
90		j	k	l	m	n	o	p	q	r	†		¤	×	±	–
A0		°	s	t	u	v	w	x	y	z	□	—	→	[	≥	•
B0	⁰	¹	²	³	⁴	⁵	⁶	⁷	⁸	⁹		–		]	≠	÷
C0	{	A	B	C	D	E	F	G	H	I	↓			‘		©
D0	}	J	K	L	M	N	O	P	Q	R	‡		§			®
E0	\		S	T	U	V	W	X	Y	Z				’		—
F0	0	1	2	3	4	5	6	7	8	9		ffl	ffi	fl	fi	ff

This is a table of the **PRESS-ITALIC** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10			½	⅓	¼				⅛							
20				⅔										•	'	'
30							ı		⅜	‘		—	®		“	”
40		*b*									*¢*	.	<	(	+	|
50	&								⅝		*!*	$	*	)	;	–
60	-	/									|	,	%		>	?
70									⅞		:	#	@	'	=	'
80		*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	↑	. .	≤			
90		*j*	*k*	*l*	*m*	*n*	*o*	*p*	*q*	*r*	†		¤	×	±	–
A0		°	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*	□	—	→	(	≥	•
B0	⁰	¹	²	³	⁴	⁵	⁶	⁷	⁸	⁹		–		)	≠	÷
C0	{	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	.			‘		©
D0	}	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*	‡		§			®
E0	\		*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*				’		—
F0	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*		*ffl*	*ffi*	*fl*	*fi*	*ff*

This is a table of the **PRESS-BOLD** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10			½	⅓	¼				⅛							
20				⅔										•	'	'
30							ı		⅜	˚	.	_	◌	.	"	"
40		b									¢	.	<	(	+	|
50	&								⅝		!	S	*	)	;	¬
60	-	/									|	,	%		>	?
70									⅞		:	#	@	'	=	'
80		a	b	c	d	e	f	g	h	i	↑	..	≤			
90		j	k	l	m	n	o	p	q	r	†		¤	×	±	–
A0		°	s	t	u	v	w	x	y	z	□	—	→	[	≥	•
B0	⁰	ⁱ	²	³	⁴	⁵	⁶	⁷	⁸	⁹		–		]	≠	÷
C0	{	A	B	C	D	E	F	G	H	I	↓			'		©
D0	}	J	K	L	M	N	O	P	Q	R	‡		§			®
E0	\		S	T	U	V	W	X	Y	Z				,		—
F0	0	1	2	3	4	5	6	7	8	9		ffl	ffi	fl	fi	ff

This is a table of the **PRESS-BOLD-ITAL** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10			½	⅓	¼				⅛							
20				⅔										•	'	'
30							ı		⅜	˚	.	_	◌	.	"	"
40		b									¢	.	<	(	+	|
50	&								⅝		!	S	*	)	;	¬
60	-	/									|	,	%		>	?
70									⅞		:	#	@	'	=	'
80		a	b	c	d	e	f	g	h	i	↑	..	≤			
90		j	k	l	m	n	o	p	q	r	†		¤	×	±	–
A0		°	s	t	u	v	w	x	y	z	□	—	→	[	≥	•
B0	⁰	ⁱ	²	³	⁴	⁵	⁶	⁷	⁸	⁹		–		]	≠	÷
C0	{	A	B	C	D	E	F	G	H	I	↓			'		©
D0	}	J	K	L	M	N	O	P	Q	R	‡		§			®
E0	\		S	T	U	V	W	X	Y	Z				,		—
F0	0	1	2	3	4	5	6	7	8	9		ffl	ffi	fl	fi	ff

APPENDIX B

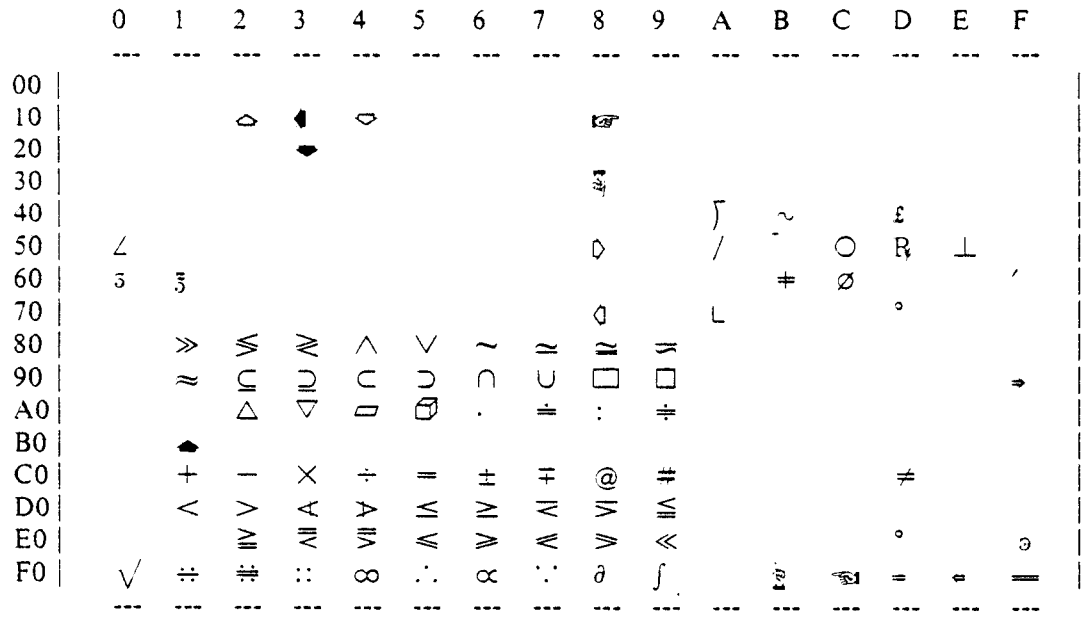
This is a table of the **GREEK** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20																
30																
40												.		(		
50														)		
60	-	/										,				
70																
80		α	β	ψ	δ	ϵ	ϕ	γ	η	ι						
90		θ	κ	λ	μ	ν	o	π	σ	ρ						-
A0			σ	τ	υ	ξ	ω	χ	ϑ	ζ						
B0				ε					$\varkappa$					$\eth$		
C0		A	B	Ψ	Δ	E	Φ	Γ	H	I						
D0		Θ	K	Λ	M	N	O	Π	ϕ	P						
E0			Σ	T	Υ	Ξ	Ω	X		Z						
F0		∇	f	∂				$\cdot$								

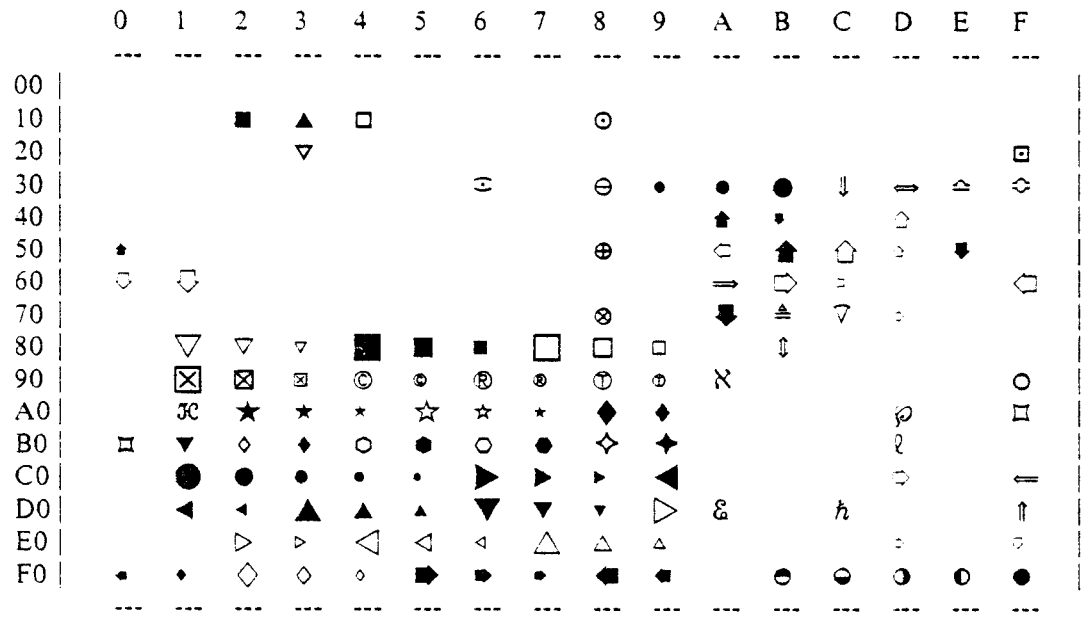
This is a table of the **APL** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00																
10																
20															'	
30						$\perp$										
40		*B*	★		$\cap$		$\subset$		$\check{}$			.	<	(	+	|
50														)	;	¯
60	–	/	∘	⊃		$\perp$			$\ominus$	Δ		,		-	>	?
70			¨								:			'	=	`
80	~	α	*B*	*C*	*D*	ϵ	*F*	*G*	*H*	ι	⁚		≤			
90		*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	ρ			⩽	×	≐	
A0			*S*	*T*	*U*	*V*	ω	*X*	*Y*	*Z*	⊡	—	—	[	≥	
B0											⋃	–		]	≠	÷
C0	{	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	.	°				
D0	}	*J*	*K*	*L*	*M*	*N*	*O*	*P*	*Q*	*R*					⍟	
E0	\		*S*	*T*	*U*	*V*	*W*	*X*	*Y*	*Z*	⍞	⌜			⊃	
F0	0	1	2	3	4	5	6	7	8	9			⌊			

This is a table of the **UTILITY-5-MATH** 10 point font.



This is a table of the **UTILITY-4-SHAPE** 10 point font.



This is a table of the **UTILITY-3-SYMBOL** 10 point font.

This is a table of the **PASS-SPEC-CHARS** 10 point font.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00		0	1	2	3		5	6	7	8						
10	9	¬	¬	'	'		ϵ	{	}	{						
20	}	□	□			¤			<	≤						
30	≤	=	=	≥		ω	>		≠	−						
40		÷	÷	+	+	×	×	¦	↑	↓						
50	↓	—	—	→	→	'	"	|	|	\						
60	\	¨				ω	ϵ	ρ	¯	ι						
70	∘	★	α	⌜	⌞	−	©	Δ	°	⌝						
80	(	)	[	]	⊂	⊃	⋂	⋃	⊥	;						
90	:	,	.	/		⊖	ˇ	∘								
A0																
B0																
C0																
D0																
E0																
F0	4	±	±	¤	<	=	−									

This Appendix shows the "QMARK" files for the EXEC's which are used to do the text processing. PHOTO EXEC is the only one the user generally needs to use since it invokes the others.

```
PHOTO ?
```

one gets the output:

Form is: PHOTO fn <(options)>
Where ' fn ' is the name of a script file to be processed.
The options come in three sets:

Final output:	nothing	--	APS5
	XPRT	--	old experimental printer
Processing:	nothing	--	NONE
	PREP2	--	PREP2 (as modified for San Jose)
	PREP2=	--	same execpt allows explicit specification of a translation table. (The name of the table must follow and have filetype PREP2)
Confidential:	nothing	--	NON-CONFIDENTIAL
	CONF	--	confidential output.
Debugging:	SAVE	--	Does not erase any intermediate files (so you can look at them later).
	STEP	--	Forces query pause at each step (one can stop at any phase)

PHOTO wraps the file in a descriptor with the user ' s name, etc.
The result ($FN SCRIPT) is fed to YFL.
The YFL Output is shipped to SNJMAS3 for further processing.
TTF is run there and the output is sent to the APS5 or experimental printer.

The output will eventually appear in your envelope in Bldg. 28

If you have special instructions, call the photocomposer operator at 6-7289.

For help, call Jim Gray 6-6408 or Jim King 6-7443.

This exec writes to files '$fn SCRIPT' and 'fn$ SCRIPT', and erases them. It is a good practice not to name script files beginning or ending with a '$'.

The exec reads several files which usually will reside with it on the disk. It will not work by itself.

The user must have a file called 'userid PHOTO' specifying the information required on the JCL, and on the output to identify it. The format of this file is shown by example in file 'PHOTO PHOTO'. Copy it to your disk, rename it to 'userid PHOTO', and edit it to have your name, phone number, etc. where indicated.

The EXEC used to do the YFL processing (used inside PHOTO) is called PHOTOYFL. If one types:

```
PHOTOYFL ?
```

one gets the output:

Form is: PHOTOYFL fn <(<dev> p1 p2 ... p15

Where ' fn ' is the name of the file to be processed by YFL,
' dev ' is the particular device which will process the YFL output:
APS5 -- the APS5 -- the default
XPRT -- the experimental printer

This program requires the PLI library 'PLILIB TXTLIB'.

p1 p2 ... p5 are the parameters passed to YFL.

The PREP2 precessing is done (inside the PHOTO EXEC) by the EXEC PHOTOPEP. If one types:

```
PHOTOPEP ?
```

one gets the output:

Form is: PHOTOPEP fn <tn> <(vers)>
The file 'fn SCRIPT *' will be processed by PREP2. An output file named '$fn SCRIPT a' and a diagnostics file 'fn PREPDIAG a' will be made. The PREPDIAG file will also be typed at the terminal at the end of processing.
This version of PREP2 has the ability to use a translation table supplied by the user. Three possibilities are explored in order:
1) if the user supplies a tn, the file 'tn PREP2 *' is used.
2) if a file named 'fn PREP2 *' exists, it is used.
3) if a file 'TABLE PREP2 A' exists, it is used.
4) none of the above, no table is used.
The particular version of PREP2 to be used can be given as the option 'vers'.
APS5 -- a version compatible with the APS5 fonts,
XPRT -- a version compatible with the experimental printer,
APS5 is the default, if none is specified.

Differences between YFL and SCRIPT/VS

The following is a list of the differences between SCRIPT/VS and YFL commands. In summary the following commands are identical in the two systems:

.bc	.bm	.br	.bt	.cb
.ce	.cm	.*	.cp	.cw
.ds	.eb	.et	.fm	.fo
.go	.hm	.h0-6	.hw	.im
.im	.ir	.li	.ll	.ls
.mc	.nb	.nc	.nf	.nj
.ot	.pa	.pl	.pp	.ps
.qu	.rc	.re	.ri	.rv
.sc	.se	.sk	.sp	.ss
.tb	.tc	.te	.tt	.ty
.uc	.un	.up	.us	..l

The following commands are quite similar in YFL and SCRIPT3. The most common differences being that YFL tends not to support ON | OFFF (marked by an *). Use '.nc' for '.co OFF', '.nj' for '.ju OFF' and '.nf' for '.fo OFF'.

.ap	.co *	.hy	.kp *	.bc *
.ju *	.fo *			

The following commands are supported by SCRIPT/VS but not by YFL:

.bf	.cl	.cs	.dc	.dd
.dm	.du	.ec	.ef	.em
.fn	.hn	.il	.it	.lb
.ly	.mg	.nl	.oc	.rd
.rh	.rt	.sl	.sv	.sy
.tr	.ud	.wf		

The following commands are supported by YFL but not by SCRIPT/VS:

.bd	.bi	.bl	.eq	.fl

The following commands are substantially different in the two systems

.bx	.cd	.du	.ms	.ep
.pf				