CHAPTER 13 Text Attributes

An important feature of the TEXT, TITLE, LABEL, and LEGEND commands is the ability to use within-text subcommands to specify the following:

- to change the case (upper versus lower) in mid-text. For example, leading characters of words can be upper case and trailing characters can be lower case (this can now also be done by literally entering the desired case);
- shift to subscripts and superscripts in mid-text;
- generate Greek letters;
- generate mathematical symbols (for example, integral sign, partial derivative sign, etc.);
- generate other special symbols (for example, brackets, arrows, carats, daggers, etc.).

The above may be done whenever the Hershey fonts (simplex, duplex, triplex, triplex italic, complex, simplex script, and complex script) have been specified (see the FONT command in the Diagrammatic Graphics chapter). The only special symbols recognized with hardware fonts are the in-line case shifts (i.e., upper and lower case) and the space character.

Within-text subcommands (indicators) are used to specify the desired text operations. For example:

UC()	to shift to capital letters;
SUB()	to shift to subscript mode;
ALPH()	to draw a Greek alpha;
INTE()	to draw an integral;
RBRA()	to draw a right bracket.

The within-text subcommands are all distinguished by an open and closed parenthesis () at the end of the subcommand. The () is a flag to DATAPLOT that the previous character sub-string is not to be printed literally but rather should be converted and acted upon in a special fashion.

If an unsupported special character is encountered when a hardware font is in effect, the sequence is replaced with a single space in the current version. However, older versions of DATAPLOT will print the string literally. For example, if the sequence ALPH() is encountered in the current version, it is replaced with a single space character, but older versions will print ALPH().

CAPITALIZATION

DATAPLOT by default prints all text in upper case. The CASE command can be used to specify the case (upper, lower, or as typed on the command) of text.

There are CASE commands for individual plot elements (e.g., LEGEND CASE, LABEL CASE, TITLE CASE). These commands are documented in the Plot Control chapter. In the current version, the CASE commands have an ASIS argument that preserves the case as entered on the corresponding text commands (TEXT, TITLE, LEGEND, LABEL, CHARACTER). In addition, explicit () sequences within the text can be used for case shifting (this method may be required in older versions). Case shifts are recognized for both hardware and software generated characters. If you want the default to be that the character case is preserved as typed on the command, enter the following commands in one of the DATAPLOT login files (DPSYSF.TEX or DPLOGF.TEX):

```
CASE ASIS; TITLE CASE ASIS; LEGEND CASE ASIS
LABEL CASE ASIS; CHARACTER CASE ASIS
```

Shifts between capitalized letters and non-capitalized letters can be carried out within text strings generated by any TEXT, TITLE, LABEL, or LEGEND command. To shift to upper case, enter CAP(), CAPS(), or UC() (standing for upper case) followed by the desired text sub-string. To shift to lower case, enter LC() (standing for lower case) followed by the desired text sub-string.

The within-text case shifting overrides the setting from the CASE command. If the within-text case shift takes place mid-line, then the first part of the text string follows whatever the current setting is as given by the CASE command. At the end of a text line, the current CASE command setting takes effect again.

The capitalization indicators are:

UC(), CAP(), CAPS() for upper case;
LC() for lower case.

If all characters on a text line are to have the same case, either all upper or all lower, then it is easier to set the case globally with the CASE command rather than using within-text case shifts. For example:

```
CASE UPPER
CASE LOWER
```

For example, to go to the middle of screen and write the string “DATAPLOT is from NIST” with all symbols in simplex font, enter the following DATAPLOT commands:

```
CASE ASIS
FONT SIMPLEX
MOVE 50 50
TEXT DATAPLOT is from NIST
```

or

```
FONT SIMPLEX
MOVE 50 50
TEXT DATAPLOT LC()IS FROM UC()NIST
```

As another example, to go to the middle of screen and write ”Future Goals” in a triplex font, enter the following commands:

```
FONT TRIPLEX
MOVE 50 50
TEXT UC()FLC()UTURE UC()GLC()OALS
```

or

```
CASE ASIS
FONT TRIPLEX
MOVE 50 50
TEXT Future Goals
```

SUBSCRIPTS

Subscripts and superscripts can be generated within any TEXT, TITLE, LABEL, or LEGEND command whenever one of the Hershey fonts (simplex, duplex, triplex, triplex italic, complex, simplex script, and complex script) have been specified (see the FONT command in the Diagrammatic Graphics chapter). Subscripts and superscripts are not currently supported for hardware generated text (the tektronix font). This specifically includes Postscript fonts.

Subscript and superscript strings may be of any length. Nested subscripts and superscripts are permitted 7 deep. The size of a sub/super-script is always half the size of the previous level.

The sub/super-script indicators are:

SUB()	to enter subscript mode;
UNSB()	to exit subscript mode;
SUP()	to enter superscript mode;
UNSP()	to exit superscript mode.

As an example, to go the middle of screen, and write “e = mc squared” (Einstein’s classic equation) with all symbols in lower case simplex font, enter:

```
CASE LOWER
FONT SIMPLEX
MOVE 50 50
TEXT E = MCSUP()2
```

As another example, to go to the middle of screen, and write “T (with superscript *) = e (with superscript integral f)” in a lower case triplex font, enter:

```
CASE LOWER
FONT TRIPLEX
MOVE 50 50
TEXT TSUP()*UNSP() = ESUP()INTE()F
```

Greek Characters

Greek symbols can be generated within any TEXT, TITLE, LABEL, or LEGEND command whenever one of the Hershey fonts (simplex, duplex, triplex, triplex italic, complex, simplex script, and complex script) have been specified (see the FONT command in the Diagrammatic Graphics chapter). Both lower and upper case Greek symbols are available (see the CASE command). If hardware generated characters are in effect (the tektronix font), a single blank space is drawn in place of the Greek character (older versions will simply print the literal text string).

To indicate that a Greek symbol should appear in some text string, simply enter the English name of the desired Greek letter and append an open and closed parenthesis after the name, as in PI(), RHO(), and TAU(). Greek names longer than 4 letters should be truncated to 4 letters, as in ALPH(), GAMM(), and OMEG().

The CHARACTER command also accepts Greek characters, The only difference is that the open and closed parenthesis is omitted. For example, enter CHARACTER ALPH to use the Greek alpha as a plot character.

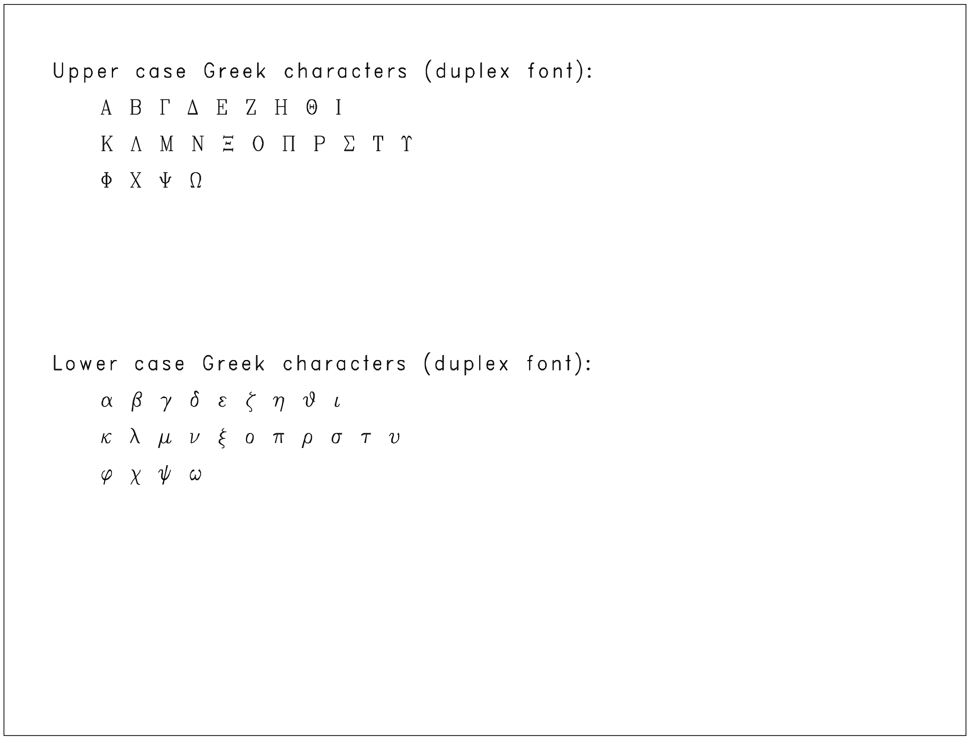
The Greek symbols are:

alpha	ALPH()
beta	BETA()
gamma	GAMM()
delta	DELT()
epsilon	EPSI()
zeta	ZETA()
eta	ETA()
theta	THET()
iota	IOTA()
kappa	KAPP()
lambda	LAMB()
mu	MU()
nu	NU()
xi	XI()
omicon	OMIC()
pi	PI()
rho	RHO()
sigma	SIGM()
tau	TAU()
upsilon	UPSI()
phi	PHI()
chi	CHI()
psi	PSI()
omega	OMEG()

Postscript printers also support Greek characters via the symbol font. However, DATAPLOT does not use the symbol font to draw Greek characters when one of the Greek () strings is encountered. It works the same as other devices where a space is drawn if hardware characters are in effect and it draws the character if one of the software characters is in effect. However, the analyst can specify the symbol font by entering the command SET POSTSCRIPT FONT SYMBOL. Be aware that ALL hardware generated text will be drawn in this font since only one Postscript font can be active at a time. Also, the Greek characters are specifed by entering the corresponding alphabetic character, not the () strings (e.g., enter a lower case a to get α).

The following program generates the Greek characters in both upper an lower case using the duplex Hershey font.

```
FONT DUPLEX
CASE UPPER
MOVE 5 90
TEXT ULC()PPER CASE UC()GLC()REEK CHARACTERS (DUPLEX FONT):
MOVE 10 85
TEXT ALPH() BETA() GAMM() DELT() EPSI() ZETA() ETA() THET() IOTA()
MOVE 10 80
TEXT KAPP() LAMB() MU() NU() XI() OMIC() PI() RHO() SIGM() TAU() UPSI()
MOVE 10 75
TEXT PHI() CHI() PSI() OMEG()
MOVE 5 50
TEXT LLC()OWER CASE UC()GLC()REEK CHARACTERS (DUPLEX FONT):
CASE LOWER
MOVE 10 45
TEXT ALPH() BETA() GAMM() DELT() EPSI() ZETA() ETA() THET() IOTA()
MOVE 10 40
TEXT KAPP() LAMB() MU() NU() XI() OMIC() PI() RHO() SIGM() TAU() UPSI()
MOVE 10 35
TEXT PHI() CHI() PSI() OMEG()
```



Upper case Greek characters (duplex font):

Α Β Γ Δ Ε Ζ Η Θ Ι

Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ ϒ

Φ Χ Ψ Ω

Lower case Greek characters (duplex font):

α β γ δ ε ζ η ϑ ι

κ λ μ ν ξ ο π ρ σ τ υ

φ χ ψ ω

Mathematical Symbols

Mathematics symbols can be generated within any TEXT, TITLE, LABEL, or LEGEND command. This may be done whenever the Hershey fonts (simplex, duplex, triplex, triplex italic, complex, simplex script, and complex script) have been specified (see the FONT command in the Diagrammatic Graphics chapter).

To generate a mathematical symbol, enter the abbreviated (never more than 4 characters) name from the list below adding an open and closed parenthesis after the name, such as INTE(), SUMM(), and DOTP(). The () is a flag to DATAPLOT that the previous character sub-string is not to be printed literally but rather should be converted and drawn as a special symbol.

The CHARACTER command also accepts mathematical symbols, The only difference is that the open and closed parenthesis is omitted. For example, enter CHARACTER CARA to use the math symbol carat as a plot character.

The mathematical symbols are:

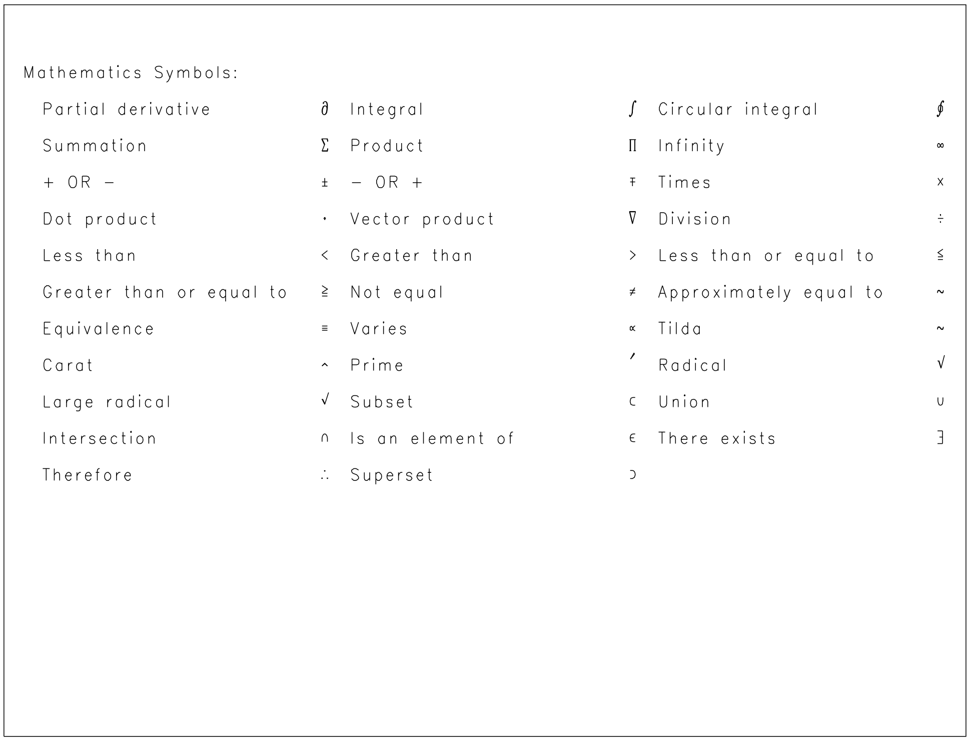
partial derivative	PART()
integral	INTE()
circular integral	CINT()
summation	SUMM()
product	PROD()
infinity	INFI()
+ or -	+-()
- or +	-+()
times	TIME()
dot product	DOTP()
vector product	DEL()
division	DIVI()
less than	LT()
greater than	GT()
less than or equal to	LTEQ()
greater than or equal to	GTEQ()
not equal	NOT=()
approximately equal to	APPR()
equivalence	EQUI()
varies	VARI()
tilda	TILD()
carot	CARA()
prime	PRIM()
radical	RADI()
large radical	LRAD()
subset	SUBS()
superset	SUPE()
union	UNIO()
intersection	INTR()
is an element of	ELEM()
there exists	THEX()
therefore	THFO()

See the comments on the previous page regarding Greek characters with Postscript. Likewise, some of the characters above can be generated using the Postscript symbol font. See the Postscript Language Reference Manual (published by Addison-Wesley) for a table of the available characters with the symbol font.

The following program draws all the available mathematical symbols using a simplex font.

```
HW 1.6 0.8;FONT SIMPLEX; JUSTIFICATION LEFT
LET X1 = 4; LET X2 = 36; LET X3 = 68
LET X4 = X1+30; LET X5 = X2+30; LET X6 = X3+30; LET YINC = 5
.
MOVE 2 90; TEXT MLC()ATHEMATICS UC()SLC()YMBOLS:
LET YPOS = 85; MOVE X1 YPOS; TEXT PLC()ARTIAL DERIVATIVE
MOVE X2 YPOS; TEXT ILC()NTEGRAL
MOVE X3 YPOS; TEXT CLC()IRCULAR INTEGRAL
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT SLC()UMMATION
MOVE X2 YPOS; TEXT PLC()RODUCT
MOVE X3 YPOS; TEXT ILC()NFINITY
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT + OR -
MOVE X2 YPOS; TEXT - OR +
MOVE X3 YPOS; TEXT TLC()IMES
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT DLC()OT PRODUCT
MOVE X2 YPOS; TEXT VLC()ECTOR PRODUCT
MOVE X3 YPOS; TEXT DLC()IVISION
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()ESS THAN
MOVE X2 YPOS; TEXT GLC()REATER THAN
MOVE X3 YPOS; TEXT LLC()ESS THAN OR EQUAL TO
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT GLC()REATER THAN OR EQUAL TO
MOVE X2 YPOS; TEXT NLC()OT EQUAL
MOVE X3 YPOS; TEXT ALC()PPROXIMATELY EQUAL TO
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT ELC()QUIVALENCE
MOVE X2 YPOS; TEXT VLC()ARIES
MOVE X3 YPOS; TEXT TLC()ILDA
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT CLC()ARAT
MOVE X2 YPOS; TEXT PLC()RIME
MOVE X3 YPOS; TEXT RLC()ADICAL
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()ARGE RADICAL
MOVE X2 YPOS; TEXT SLC()UBSET
MOVE X3 YPOS; TEXT ULC()NION
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT ILC()NTERSECTION
MOVE X2 YPOS; TEXT ILC()S AN ELEMENT OF
MOVE X3 YPOS; TEXT TLC()HERE EXISTS
LET YPOS = YPOS - YINC; MOVE X1 35; TEXT TLC()HEREFORE
MOVE X2 YPOS; TEXT SLC()UPERSET
.
JUSTIFICATION RIGHT; LET YPOS = 85; MOVE X4 YPOS; TEXT PART()
MOVE X5 YPOS; TEXT INTE(); MOVE X6 YPOS; TEXT CINT()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT SUMM()
MOVE X5 YPOS; TEXT PROD(); MOVE X6 YPOS; TEXT INFI()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT +-()
MOVE X5 YPOS; TEXT -+(); MOVE X6 YPOS; TEXT TIME()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT DOTP()
MOVE X5 YPOS; TEXT DEL(); MOVE X6 YPOS; TEXT DIVI()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT LT()
MOVE X5 YPOS; TEXT GT(); MOVE X6 YPOS; TEXT LTEQ()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT GTEQ()
```

```
MOVE X5 YPOS; TEXT NOT=(); MOVE X6 YPOS; TEXT APPR()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT EQUI()
MOVE X5 YPOS; TEXT VARI(); MOVE X6 YPOS; TEXT TILD()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT CARA()
MOVE X5 YPOS; TEXT PRIM(); MOVE X6 YPOS; TEXT RADI()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT LRAD()
MOVE X5 YPOS; TEXT SUBS(); MOVE X6 YPOS; TEXT UNIO()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT INTR()
MOVE X5 YPOS; TEXT ELEM(); MOVE X6 YPOS; TEXT THEX()
LET YPOS = YPOS - YINC; MOVE X4 35; TEXT THFO()
MOVE X5 YPOS; TEXT SUPE()
```



Mathematics Symbols:

Partial derivative	∂	Integral	∫	Circular integral	∮
Summation	Σ	Product	Π	Infinity	∞
+ OR −	±	− OR +	∓	Times	×
Dot product	·	Vector product	∇	Division	÷
Less than	<	Greater than	>	Less than or equal to	≤
Greater than or equal to	≥	Not equal	≠	Approximately equal to	~
Equivalence	≡	Varies	∝	Tilda	~
Carat	^	Prime	′	Radical	√
Large radical	√	Subset	⊂	Union	∪
Intersection	∩	Is an element of	∈	There exists	∃
Therefore	∴	Superset	⊃		

Miscellaneous Symbols

Miscellaneous symbols can be generated within any TEXT, TITLE, LABEL, or LEGEND command. This may be done whenever the Hershey fonts (simplex, duplex, triplex, triplex italic, complex, simplex script, and complex script) have been specified (see the FONT command in the Diagrammatic Graphics chapter).

To generate a special symbol, enter the abbreviated (never more than 4 characters) name from the list below adding an open and closed parenthesis after the name, such as LAPO(), LBRA(), and RBRA(). The () is a flag to DATAPLOT that the previous character sub-string is not to be printed literally but rather should be converted and drawn as a special symbol.

The CHARACTER command also accepts miscellaneous symbols, The only difference is that the open and closed parenthesis is omitted. For example, enter CHARACTER CARA to use the math symbol carat as a plot character.

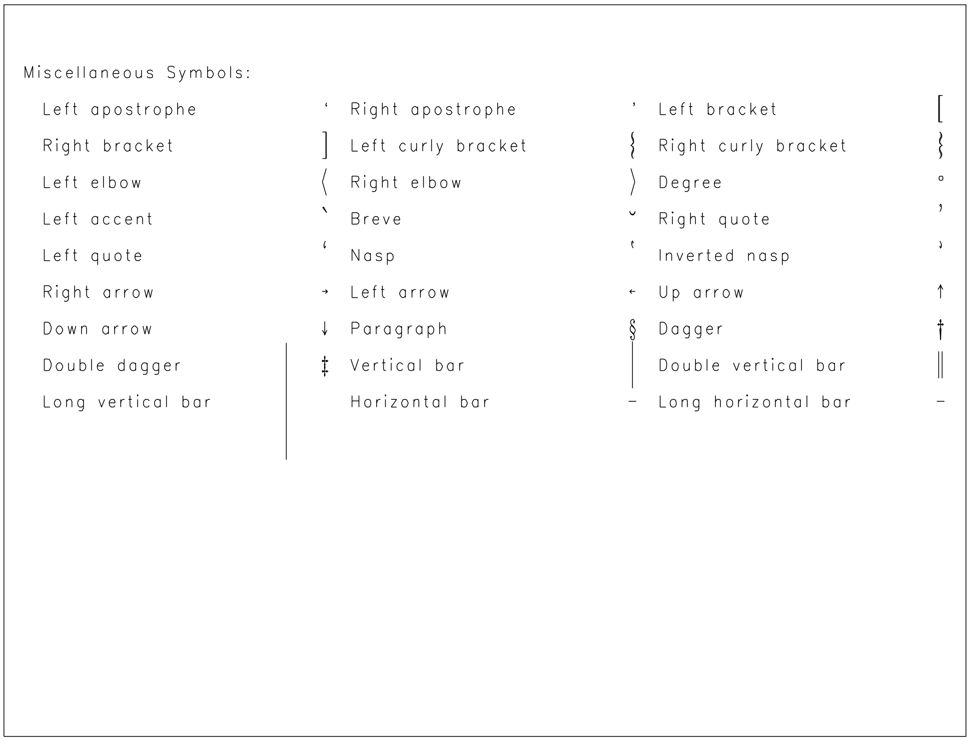
The miscellaneous symbols are:

space	SPAC() or SP()
left apostrophe	LAPO()
right apostrophe	RAPO()
left bracket	LBRA()
right bracket	RBRA()
left curly bracket	LCBR()
right curly bracket	RCBR()
left elbow	LELB()
right elbow	RELB()
left accent	LACC()
breve	BREV()
right quote	RQUO()
left quote	LQUO()
normal aspirate	NASP()
inverted aspirate	IASP()
right arrow	RARR()
left arrow	LARR()
up arrow	UARR()
down arrow	DARR()
paragraph	PARA()
dagger	DAGG()
double dagger	DDAG()
vertical bar	VBAR()
double vertical bar	DVBA()
long vertical bar	LVBA()
horizontal bar	HBAR()
long horizontal bar	LHBA()
degree	DEGR()

The following program draws all the available miscellaneous symbols using a simplex font.

```
HW 1.6 0.8; FONT SIMPLEX; JUSTIFICATION LEFT
LET X1 = 4; LET X2 = 36; LET X3 = 68
LET X4 = X1+30; LET X5 = X2+30; LET X6 = X3+30; LET YINC = 5
.
MOVE 2 90; TEXT MLC()ISCELLANEOUS UC()SLC()YMBOLS:
LET YPOS = 85; MOVE X1 YPOS; TEXT LLC()EFT APOSTROPHE
MOVE X2 YPOS; TEXT RLC()IGHT APOSTROPHE
MOVE X3 YPOS; TEXT LLC()EFT BRACKET
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT RLC()IGHT BRACKET
MOVE X2 YPOS; TEXT LLC()EFT CURLY BRACKET
MOVE X3 YPOS; TEXT RLC()IGHT CURLY BRACKET
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()EFT ELBOW
MOVE X2 YPOS; TEXT RLC()IGHT ELBOW
MOVE X3 YPOS; TEXT DLC()EGREE
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()EFT ACCENT
MOVE X2 YPOS; TEXT BLC()REVE
MOVE X3 YPOS; TEXT RLC()IGHT QUOTE
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()EFT QUOTE
MOVE X2 YPOS; TEXT NLC()ASP
MOVE X3 YPOS; TEXT ILC()NVERTED NASP
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT RLC()IGHT ARROW
MOVE X2 YPOS; TEXT LLC()EFT ARROW
MOVE X3 YPOS; TEXT ULC()P ARROW
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT DLC()OWN ARROW
MOVE X2 YPOS; TEXT PLC()ARAGRAPH
MOVE X3 YPOS; TEXT DLC()AGGER
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT DLC()OUBLE DAGGER
MOVE X2 YPOS; TEXT VLC()ERTICAL BAR
MOVE X3 YPOS; TEXT DLC()OUBLE VERTICAL BAR
LET YPOS = YPOS - YINC; MOVE X1 YPOS; TEXT LLC()ONG VERTICAL BAR
MOVE X2 YPOS; TEXT HLC()ORIZONTAL BAR
MOVE X3 YPOS; TEXT LLC()ONG HORIZONTAL BAR
.
JUSTIFICATION RIGHT
LET YPOS = 85; MOVE X4 YPOS; TEXT LAPO()
MOVE X5 YPOS; TEXT RAPO(); MOVE X6 YPOS; TEXT LBRA()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT RBRA()
MOVE X5 YPOS; TEXT LCBR(); MOVE X6 YPOS; TEXT RCBR()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT LELB()
MOVE X5 YPOS; TEXT RELB(); MOVE X6 YPOS; TEXT DEGR()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT LACC()
MOVE X5 YPOS; TEXT BREV(); MOVE X6 YPOS; TEXT RQUO()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT LQUO()
MOVE X5 YPOS; TEXT NASP(); MOVE X6 YPOS; TEXT IASP()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT RARR()
MOVE X5 YPOS; TEXT LARR(); MOVE X6 YPOS; TEXT UARR()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT DARR()
MOVE X5 YPOS; TEXT PARA(); MOVE X6 YPOS; TEXT DAGG()
LET YPOS = YPOS - YINC; MOVE X4 YPOS; TEXT DDAG()
```

```
MOVE X5 YPOS; TEXT VBAR(); MOVE X6 YPOS; TEXT DVBA()
LET YPOS = YPOS - YINC; LET XTMP = X4 - 4;MOVE XTMP YPOS; TEXT LVBA()
MOVE X5 YPOS; TEXT HBAR(); MOVE X6 YPOS; TEXT LHBA()
```



Miscellaneous Symbols:

Left apostrophe	‘	Right apostrophe	’	Left bracket	[
Right bracket	]	Left curly bracket	{	Right curly bracket	}
Left elbow	⟨	Right elbow	⟩	Degree	°
Left accent	`	Breve	˘	Right quote	’
Left quote	‘	Nasp	‘	Inverted nasp	’
Right arrow	→	Left arrow	←	Up arrow	↑
Down arrow	↓	Paragraph	§	Dagger	†
Double dagger	‡	Vertical bar	|	Double vertical bar	‖
Long vertical bar	│	Horizontal bar	−	Long horizontal bar	−

(blank page)