MagnaType®

OPERATOR'S MANUAL

The typesetting for this manual was produced completely with MagnaType software using PostScript* to drive the Apple LaserWriter.**

*PostScript is a trademark of Adobe Systems, Inc.
**Apple and Apple LaserWriter are trademarks of Apple Computer, Inc.

INTRODUCTION

Thank you for purchasing the MagnaType Software Package from Magna Computer Systems, Inc. It is one of the first, fully functional pre-print typesetting systems available for use on the IBM® family of personal computers (or compatibles). We are sure you will agree after using the program that its speed and performance combined with its powerful typographic features makes MagnaType a wise investment for the growing demands of the industry.

In the interest of maximizing your use of the program, this manual has been carefully designed so you may become familiar with the many features of the program as quickly as possible. The instruction contained in this manual assumes the user to have a fundamental knowledge of typographic terms and a basic understanding of the operation procedures for the personal computer. Please take the time to examine the contents and note the sections which may require further study based on your own personal experience with other typesetting or word processor systems.

Before proceeding with Chapter 1, you should first study the *INSTALLATION SECTION* and *TYPESETTER SPECIFIC* information provided with this manual. This material will allow you to become familiar with the installation and operation of the hardware and the handling of the software program diskettes.

Other *required* reading prior to the use of MagnaType are the *GUIDE TO OPERATIONS* and the *DISK OPERATING SYSTEM (DOS) USER'S GUIDE* provided with your personal computer. For more detailed information on typography, you should consult one or more of the typesetting trade publications.

Although Magna Computer Systems, Inc. is confident you will find everything you need to get the maximum production from this software package, our representatives are willing to assist you and welcome any suggestions for the improvement of this instruction material.

ORGANIZATION OF THIS MANUAL

This manual contains 6 chapters and 3 appendices.

Chapter 1 is a detailed introduction to the preliminary software topics to consider before using MagnaType in a production environment.

Chapter 2 gives a detailed description of all the keyboard functions.

Chapter 3 describes the composition command codes used to create and edit a MagnaType text job.

Chapter 4 provides a guide through the main menu entries of the MagnaType software program with emphasis on creating and working with text jobs on the system.

Chapter 5 details procedures for production input and editing.

Chapter 6 provides detailed explanations and examples for practical application of MagnaType.

Appendix A lists the composition error messages that may appear in a job/take during H&J.

Appendix B is a quick reference guide to the features of the program.

Appendix C is a glossary listing descriptions of the typesetting and computer terminology used throughout this manual.

TYPESETTER SPECIFIC

Linotype L/202

As with any complex device, features and options vary from one manufacturer to another. Such is the case with output typesetters. Following is a breakdown of the specific features of the L/202 that will have impact on various MagnaType command codes and system standards. MagnaType requires the loading of the slave program on the 202 that accepts pre-justified CORA code input (see your typesetter manual).

FONT ACCESS

MagnaType instructs the L/202 to use specified fonts by referencing the CFD number. This is the identification given to each font at the time it is loaded onto the L/202 hard disk or L/202 font floppy disk by the user with the TM instruction (see the manual supplied with the typesetter).

The MagnaType program knows this font CFD number by the entry on the fourth line of the *FONT INFORMATION TABLE* for each font. This is the *FONT OUTPUT IDENTIFICATION NO:* and it is the operator's responsibility to see that these numbers are correct.

MagnaType also supports the *electronic italic* feature of the L/202. A *FONT INFORMATION TABLE* may be created to indicate that a specific font ID in the text refers to an electronic italic version of another standard font ID already established *(see FONT INFORMATION in chapter 1).*

MagnaType allows the operator to create FSTs and TMs on screen for direct transmission to the L/202 as if it were a paper tape reader. This is accomplished by opening a job/take with the <FT> command and keying in the contents of an FST or TM instruction just as if it were being punched on a paper tape. The job/take is then *prepared for output* and sent to the L/202 as a composed job. This feature is provided as a convenience for the user, however, refer to the *FONT HANDLING* section of the L/202 documentation for instructions regarding the use and creation of FSTs and Font Transfer information.

POINTSIZE

MagnaType currently supports pointsize requests in half point increments from 4.5 to 120. Refer to your typesetter documentation for the limits of your particular machine. To assist your operators in detecting oversized font parameter requests, the maximum allowed pointsize may be entered in the *SYSTEM STANDARDS-VALUES* screen (see chapter 1), and an ERROR will be flagged during H&J if it is exceeded in a job.

LEADING

The L/202 is currently capable of leading in quarter point increments and MagnaType supports this feature. To specify leading in quarter points, enter the amount in hundredths, for example, eleven and three quarter point lead is keyed as 11.75 in the appropriate command. Eleven and one half may be keyed as 11.5. Be sure to verify that your L/202 is equipped to receive quarter point lead instructions.

All horizontal escapement values are expressed in relative units. For the L/202, a relative unit is one fifty-fourth of an em, where em is equal in points to the current pointsize requested. For example, in 18 point type, a relative unit is equal to one third of a point or three relative units to a point. In 54 point type, a relative unit is equal to one point. The relative unit value applies to *character widths, spaceband limits, whitespace adjustment, tracking, kerning,* and *letterspacing.* The MAXIMUM value for tracking and whitespace adjustment is plus or minus 9 relative units. Accent characters are given zero widths in the *FONT INFORMATION* tables (see chapter 1).

RULE WEIGHTS

On the L/202, rule weights are established in the typesetter slave program and are requested by external front-end drivers in cora code by a number from 0 to 9. Refer to your typesetter manual for the exact weight achieved with each number or experiment with a job/take from MagnaType by requesting <WR0> through <WR9>.

TYPESETTER REQUIREMENTS

Selection 2 of *OTHER SYSTEM OPTIONS (see chapter 1)* requires the operator to indicate with a 'Y' that the fonts loaded on the L/202 typesetter are using the standard X124 FST provided by Allied Linotype. 'N' indicates your L/202 already has fonts loaded with the FST provided by CCI* and eliminates the need to reload the fonts. This is to assure the proper character access using the PC keyboard.

If you have superfonts loaded on your L/202, indicate yes with a 'Y' in the second screen menu of *OTHER SYSTEM OPTIONS* of the *STANDARDS AND DEFAULTS* menu (see chapter 1). This will alert the composition process to send a default superfont pointsize of 20 (>1P20< in cora code) to the L/202. If the user wishes to alter this default in a job/take, key in <SPx> where x is a pointsize from 20 to 72.

Indicate if your L/202 is equipped with a hard disk by keying a zero for number of floppy disks. If you will be using the <PD command, key in the maximum number of <PD commands allowed in a single job/take *(see <PDn,m> in chapter 3).*

If you have an old style (pre N) L/202, indicate with a 'Y' on the fourth line of the screen to allow the proper transmission of data to the paper tape reader port.

Any changes made to this screen menu requires that the program be rebooted in order to take effect.

---

*CCI is a trademark of Computer Composition International

KEYSTROKE/OUTPUT TABLE
L/202

This table indicates which keystroke (A) is associated with which typesetter sequence number (B). The sequence number (B) is also the width value position number in the width screen of the *FONT INFORMATION* table. The pi character (shown on screen as π) is accessed by typing a backslash on the keyboard.

The table also shows which Reverse TTS number is used to access a character for each keystroke. Column (C) shows the TTS code used when the *TYPESETTER REQUIREMENTS* menu indicates STANDARD FST, and column (D) shows the TTS code used when NON-Standard FST is indicated *(see page 202-3)*.

(A)	(B)	(C)	(D)	(A)	(B)	(C)	(D)	(A)	(B)	(C)	(D)	(A)	(B)	(C)	(D)
a	1	30	30	J	36	132	132	)	71	61	61	π H	106	226	205
b	2	23	23	K	37	136	136	-	72	62	62	π q	107	255	235
c	3	16	16	L	38	111	111	^	73	42	42	π e	108	243	220
d	4	22	22	M	39	107	107	`	74	142	142	π x	109	256	227
e	5	20	20	N	40	106	106	—	75	154	154	π 8	110	241	254
f	6	26	26	O	41	103	103	π-	76	262	262	π 9	111	265	243
g	7	13	13	P	42	115	115	$	77	70	70	π 7	112	240	274
h	8	5	5	Q	43	135	135	π $	78	246	270	π s	113	232	224
i	9	14	14	R	44	112	112	%	79	162	171	π v	114	230	152
j	10	32	32	S	45	124	124	/	80	261	162	π V	115	224	217
k	11	36	36	T	46	101	101	π/	81	242	174	π z	116	213	141
l	12	11	11	U	47	134	134	π j	82	223	232	π Z	117	205	221
m	13	7	7	V	48	117	117	π o	83	247	203	π y	118	274	165
n	14	6	6	W	49	131	131	π l	84	206	211	π Y	119	254	225
o	15	3	3	X	50	127	127	π r	85	207	212	π 5	120	270	241
p	16	15	15	Y	51	125	125	*	86	221	167	π 6	121	253	265
q	17	35	35	Z	52	121	121	π p	87	227	215	π &	122	167	240
r	18	12	12	&	53	143	143	π d	88	216	222	π]	123	171	250
s	19	24	24	1	54	67	67	π w	89	217	231	π [	124	160	242
t	20	1	1	2	55	71	71	π i	90	214	214	π,	125	152	246
u	21	34	34	3	56	60	60	π a	91	235	230	π.	126	141	247
v	22	17	17	4	57	52	52	π g	92	231	213	π;	127	165	253
w	23	31	31	5	58	41	41	π k	93	220	236	π !	128	174	204
x	24	27	27	6	59	65	65	π u	94	212	234	π *	129	236	206
y	25	25	25	7	60	74	74	π t	95	201	201	π %	130	211	266
z	26	21	21	8	61	54	54	π f	96	234	226	π :	131	210	261
A	27	130	130	9	62	43	43	π b	97	215	223	π)	132	233	256
B	28	123	123	0	63	55	55	π c	98	225	216	π (	133	237	276
C	29	116	116	.	64	47	47	π m	99	203	207	π^	134	244	251
D	30	122	122	,	65	46	46	π 1	100	267	267	π S	135	250	245
E	31	120	120	:	66	153	153	π 2	101	271	271	π n	136	257	275
F	32	126	126	;	67	53	53	π 3	102	260	260	π N	137	266	257
G	33	113	113	?	68	155	155	π 4	103	252	252	π X	138	272	244
H	34	105	105	!	69	170	170	π Ø	104	264	255	π W	139	276	150
I	35	114	114	(	70	161	161	π h	105	222	160	π U	140	277	104

HARDWARE CONNECTION
L/202

§1.0 WHICH 202?

Before attempting to cable connect the MagnaType PC to the typesetter, you must first determine which kind of 202 is present. Is it an "OLD" style 202 or "NEW" style 202. An 'old' style 202 is the version with the red/orange push-buttons. No LCD display is present. A 'new' style 202 has membrane-type switches and has an LCD display. A 202N or 202W is considered a 'new' 202.

Another 'pointer' is that on the older 202 the floppy disk drive is mounted vertically, whereas on the newer style 202 the floppy disk drive is mounted horizontally.

If you have an OLD style 202, *GO TO §2.0.* If you have a NEW style 202, *GO TO §3.0.*

§2.0 OLD 202

Determine which kind of paper tape reader interface board is installed. PART NO. 8402207. This is the original Paper Tape Reader interface board from Merganthaler (Allied Linotype). If you do NOT have an external floppy disk reader attached to your 202, this board is probably the one installed in your system (see §2.3 for some more info). *GO TO §2.1.*

PART NO. 8402201. This interface board is used when an external floppy disk reader is installed on the 202 (see also §2.3 for more info). *GO TO §2.2.*

§2.1 CABLE INSTALLATION

How to Install the Magna Cable to the 202

1. Turn off the power to the 202 (for safety reasons).
2. Unscrew the two small philips screws at the back of the 202 that retains the back cover of the operator's console.
3. Locate the connector/cable that is attached to the back of the paper tape reader board. The connector is a dual 10 pin edge connector, and is probably green in colour. Using either a marker or a small piece of tape, mark the top edge of the connector, and then unplug the connector from the reader board.
4. Take the cable supplied by Magna and locate the switch on the connector. Make sure the switch is in position '2'. Now plug the 202 cable/connector into the Magna connector, ensuring that the polarity of the two cables match (i.e. 'Top' to 'Top').
5. The other end of the Magna cable plugs into the parallel port of the IBM XT or AT (or compatible). If a MagnaLock is on parallel port 1, you may still connect the cable to this parallel port through the MagnaLock.
6. Replace the back cover to the operator's panel that was removed in (2).
7. Apply power to the 202. Boot the 202 slave program.
8. Installation of the cable is complete. To run the 202 from the Magna system, put the Local/Online switch to 'Local', initiate the typesetting transmission from the Magna system (see your Magna manual, Chapter 4, the section called 'Output to Typesetter'), and then hit the RUN button on the 202.

IMPORTANT NOTE:

If you have an IBM 'clone', or the IBM machine that you have is equipped with a non-IBM parallel port controller, the Magna system may not work correctly. In order for the Magna software to operate correctly, the parallel port controller must be an IBM board or an EXACT replica.

Also, the parallel card used when driving the "Old Style 202" must be set up to allow interrupts according to the IBM standard, IRQ7. A true IBM parallel card normally has this set up, however, a 'clone' or compatible may or may not have this option available. Refer to your manufacturer's manual or computer dealer for more information.

§2.2 CABLE INSTALLATION

How to Install the Magna Cable to the 202

Follow the instructions in §2.1 with the following change to step (4).

4. Take the cable supplied by Magna and locate the switch on the connector. Make sure the switch is in position "1". Now plug the 202 cable/connector into the Magna connector, ensuring that the polarity of the two cables match (i.e. "Top" to "Top").

§2.3 P.T.R. INTERFACE

Following is a description of the two types of Paper Tape Reader Interface Board.

The original Paper Tape Reader controller board in the 202 is Part No. 8402207, this number should be printed on the controller board in the lower left-hand corner (obviously, you will need to remove the board from the 202 first in order to see this). This board has only a few chips installed, and there is no edge connector on the board other than the main one that connects to the backplane of the 202.

The other Paper Tape Reader controller board is one that also controls the optional external floppy disk reader and also the optional line printer. The Part Number for this board is 8402201, which should also be printed somewhere on the board itself. There is an edge connector on this board (which is used to connect to a printer cable) located on the side of the board which faces towards the front of the 202 when the board is plugged in.

§3.0 NEW 202

Following is a step by step procedure for connecting the Magna cable to the "NEW" 202 (202N, 202W)

1. Turn off power to the 202 (for safety purposes).
2. Open the front door of the 202, and locate the Paper Tape Reader interface board at either location 6 or 7. (Some 202 units have retainer bars to hold the boards in place—you need to remove these bars).
3. Pull the Paper Tape Reader interface board out of the 202 by about 5 inches. Locate the 20 pin IDC connector (2 rows of 10 pins—also known as a "BERG" connector) on the interface board, situated just above the 25 pin male connector.
4. Plug the 20 pin connector of the Magna cable into the IDC connector. The connector on most of the Magna ribbon cables is "keyed" so you can only plug it in one way. The black wire will be on the top edge and the ribbon cable will naturally lead out of the machine toward you. Re-seat the interface board in the 202. Replace any retainer bars removed in step (2). Close the 202's front door.
5. Plug the other end of the Magna cable into the parallel port of your PC. You may attach the cable to the MagnaLock if you wish to use parallel port 1.
6. Re-apply power to the 202. Load the slave program, and put the Local/Online switch to "Local".
7. Cable installation is complete. To run the 202 from your Magna system, initiate the typesetting transmission (see your Magna manual, Chapter 4, section "Output to Typesetter"), and then hit the START button on the 202.

TYPESETTER SPECIFIC

CG-8400

As with any complex device, features and options vary from one manufacturer to another. Such is the case with output typesetters. Following is a breakdown of the specific features of the CG8400 that will have impact on various MagnaType command codes and system standards.

FONT ACCESS

MagnaType instructs the 8400 to use specified fonts by referencing the 12 digit font number. This is the identification given to each font at the time it is loaded onto the 8400 hard disk or 8400 font floppy disk by the user (see the manual supplied with the typesetter).

The MagnaType program knows this font ID number by the entry on the fourth line of the *FONT INFORMATION TABLE* for each font. This is the *FONT OUTPUT IDENTIFICATION NO:* and it is the operator's responsibility to see that these numbers are correct.

MagnaType also supports the *electronic italic* feature of the 8400. A *FONT INFORMATION TABLE* may be created to indicate that a specific font ID in the text refers to an electronic italic version of another standard font ID already established *(see FONT INFORMATION in chapter 1).*

Utilizing the semi-slave program requires that when you load your fonts onto your floppy or hard disk, that the widths for those fonts are also loaded. If you are currently driving your 8400 with an MCS or Quadex system, this will probably mean that you will need to re-structure your floppy or hard disks. If you are not currently driving your 8400 with either of those systems, there is no need to re-structure.

If you have any difficulties loading font widths into the 8400, then please call your Compugraphic support person, describing in detail what it is that you are attempting to accomplish.

POINTSIZE

MagnaType currently supports pointsize requests in ⅛th point increments from 4.5 to 96. Refer to your typesetter documentation for the limits of your particular machine. To assist your operators in detecting oversized font parameter requests, the maximum allowed pointsize may be entered in the *SYSTEM STANDARDS-VALUES* screen (see chapter 1), and an ERROR will be flagged during H&J if it is exceeded in a job.

LEADING

The 8400 is currently capable of leading in ⅛th point increments and MagnaType supports this feature. To specify leading in ⅛ths of a point, enter the amount in hundredths, for example, eleven and ⅛th point lead is keyed as 11.1 or 11.12 in the appropriate command. Eleven and one half may be keyed as 11.5. Eleven and ⅜ths is 11.3 or 11.35 (MagnaType will round to nearest ⅛th).

ESCAPEMENT

Horizontal escapement values are expressed in relative units. For the 8400, a relative unit is one fifty-fourth of an em, where em is equal in points to the current pointsize requested. For example, in 18 point type, a relative unit is equal to one third of a point or three relative units to a point. In 54 point type, a relative unit is equal to one point. The relative unit value applies to *character widths, spaceband limits, and kerning. Tracking, letterspacing, and whitespace adjustment* values are indicated as 1/9th of a point increments. Accent characters are given zero widths in the *FONT INFORMATION* tables (see chapter 1).

RULE WEIGHTS

On the 8400, rule weights are established as multiples of quarter points. For example, a one point rule would be coded as <WR4>.

TYPESETTER REQUIREMENTS

Selection 1 of *OTHER SYSTEM OPTIONS (see chapter 1)* allows the operator to indicate which serial port will output to the 8400. Type in a number with an "S" following.

Selection 2 of *OTHER SYSTEM OPTIONS (see chapter 1)* allows the operator to indicate if your 8400 is equipped with a hard disk by keying a zero for number of floppy disks. If you will be using the <PD command, key in the maximum number of <PD commands allowed in a single job/take *(see <PDn,m> located in chapter 3).*

Selection 3 of *OTHER SYSTEM OPTIONS* allows the operator to establish the transmission protocol for each serial port assigned in selection 1. The proper setup for the 8400 semi-slave program is 9600 baud, 8 word, 1 stop, and even parity.

Any changes made to this screen menu requires that the program be rebooted in order to take effect.

KEYSTROKE/OUTPUT TABLE
CG8400 LAYOUT 549

This table indicates which typesetter flash position is associated with which keystroke(s) to produce which output character. The position number is also the width value position number in the width screen of the *FONT INFORMATION* table. If the font requested is a 549 layout, the output flash position shown in this table will be used. The pi precedent character (shown on screen as π) is accessed by typing a backslash on the keyboard.

key(s)	flash posn	char	key(s)	flash posn	char	key(s)	flash posn	char
a	21	a	O	28	O	π D	81	‡
b	16	b	P	36	P	π hyphen	82	–
c	11	c	Q	51	Q	@	83	@
d	15	d	R	33	R	π ?	84	¿
e	13	e	S	43	S	)	85	)
f	19	f	T	27	T	(	86	(
g	8	g	U	50	U	π i	87	ı
h	3	h	V	38	V	π B	88	ß
i	9	i	W	48	W	hyphen	89	-
j	23	j	X	46	X	shft hyph	90	—
k	26	k	Y	44	Y	π b	91	•
l	6	l	Z	40	Z	π C	92	Ç
m	5	m	π shft hyph	53	_	π e	93	é
n	4	n	period	54	.	&	94	&
o	2	o	comma	55	,	π k	95	^
p	10	p	0	56	0	π K	96	^
q	25	q	1	57	1	shift 6	97	’
r	7	r	2	58	2	open quote	98	‘
s	17	s	3	59	3	π u	99	¨
t	1	t	4	60	4	π U	100	¨
u	24	u	5	61	5	π g	101	`
v	12	v	6	62	6	π G	102	`
w	22	w	7	63	7	!	103	!
x	20	x	8	64	8	π M	104	º
y	18	y	9	65	9	π F	105	ª
z	14	z	semicolon	66	;	#	106	#
A	47	A	colon	67	:	π r	107	®
B	42	B	$	68	$	π t	108	¯
C	37	C	plus	69	+	π n	109	.
D	41	D	%	70	%	π c	110	ç
E	39	E	?	71	?	π (	111	«
F	45	F	π A	72	´	π)	112	»
G	34	G	π a	73	´	π !	113	¡
H	29	H	π o	74	°	slash	114	/
I	35	I	π m	75	–	*	115	*
J	49	J	π plus	76	±	π $	116	¢
K	52	K	π slash	77	÷	π R	117	©
L	32	L	π x	78	×	π T	118	¯
M	31	M	=	79	=			
N	30	N	π d	80	†			

KEYSTROKE/OUTPUT TABLE
CG8400 DEFAULT (500)

This table indicates which typesetter flash position is associated with which keystroke(s) to produce which output character. The position number is also the width value position number in the width screen of the *FONT INFORMATION* table. If the font requested is NOT a 549 layout, the output flash position shown in this table will be used (the output characters shown are from a 500 layout). The pi precedent character (shown on screen as π) is accessed by typing a backslash on the keyboard.

key(s)	flash posn	char	key(s)	flash posn	char	key(s)	flash posn	char
a	21	a	O	28	O	π D	81	7
b	16	b	P	36	P	π hyphen	82	8
c	11	c	Q	51	Q	@	83	9
d	15	d	R	33	R	&	84	&
e	13	e	S	43	S	)	85	)
f	19	f	T	27	T	(	86	(
g	8	g	U	50	U	!	87	!
h	3	h	V	38	V	π B	88	.
i	9	i	W	48	W	hyphen	89	-
j	23	j	X	46	X	shft hyph	90	—
k	26	k	Y	44	Y	%	91	%
l	6	l	Z	40	Z	π slash	92	/
m	5	m	π shft hyph	53	_	π e	93	]
n	4	n	period	54	.	π ?	94	[
o	2	o	comma	55	,	π k	95	⅓
p	10	p	0	56	0	π K	96	⅔
q	25	q	1	57	1	shift 6	97	’
r	7	r	2	58	2	open quote	98	‘
s	17	s	3	59	3	π u	99	¼
t	1	t	4	60	4	π U	100	¾
u	24	u	5	61	5	π g	101	½
v	12	v	6	62	6	slash	102	/
w	22	w	7	63	7	π i	103	1
x	20	x	8	64	8	π M	104	2
y	18	y	9	65	9	π F	105	3
z	14	z	semicolon	66	;	#	106	4
A	47	A	colon	67	:	π r	107	5
B	42	B	$	68	$	π t	108	6
C	37	C	plus	69	$	π n	109	7
D	41	D	π b	70	¢	π c	110	8
E	39	E	?	71	?	π (	111	9
F	45	F	π A	72	¢	π)	112	0
G	34	G	*	73	´	π !	113	*
H	29	H	π o	74	0	π G	114	√
I	35	I	π m	75	1	π a	115	–
J	49	J	π plus	76	2	π $	116	®
K	52	K	π C	77	3	π R	117	•
L	32	L	π x	78	4	π T	118	□
M	31	M	=	79	5			
N	30	N	π d	80	6			

HARDWARE CONNECTION
CG 8400

CABLE INSTALLATION

How to Install the Magna Cable to the 8400

1. Remove the back panel of the 8400. (Depending on your expertise with your 8400 you may or may not wish to switch off power first.) Locate the two 37 pin connectors for the serial interfaces on the CPU board. (You may already be using one or both of these serial interfaces.)
2. Plug the 37 pin connector of the Magna cable into the serial port of your choice on the CPU board.
3. Plug the other end of the Magna cable into the serial port of your PC.

Note 1. The IBM AT and some of its "clones" use a 9 pin connector for the serial port. You must purchase an adapter (in this case a 9-pin female to 25-pin male adapter) from your computer store in order to connect properly to the Magna cable.

Note 2. Some IBM PC "clones" use a 25-pin female connector for the serial port. You must purchase an adapter (in this case a 25-pin male to 25-pin male "gender changer") from your computer store in order to connect properly to the Magna cable.

4. Cable installation is complete. Replace the back panel of the 8400.

TYPESETTER SPECIFIC

Apple LaserWriter

As with any complex device, features and options vary from one manufacturer to another. Such is the case with output typesetters. Following is a breakdown of the specific features of the Apple LaserWriter that will have impact on various MagnaType command codes and system standards.

FONT ACCESS

MagnaType commands the use of specified fonts on the LaserWriter by listing their names in a preamble to the PostScript code stream. The font names are listed literally based on the keyed font name entered on line two of each *FONT INFORMATION TABLE*. This is called the *FONT DESCRIPTION*. Nine fonts are initially supported by MagnaType with *FONT INFORMATION TABLES* already established along with an assortment of autokern values. As the user acquires additional digitized fonts for the LaserWriter, *FONT INFORMATION TABLES* must be established with the *FONT DESCRIPTION* keyed exactly as required by PostScript.

POINTSIZE

MagnaType currently supports pointsize requests in tenths of a point increments from 5 to 400. You may not, however, wish to allow setting pointsizes this large, so to assist your operators in detecting oversized font parameter requests, your maximum allowed pointsize may be entered in the *SYSTEM STANDARDS-VALUES* screen (see chapter 1), and an ERROR will be flagged during H&J if it is exceeded in a job.

LEADING

The LaserWriter is currently capable of leading in tenths of a point increments and MagnaType supports this feature.

HORIZONTAL ESCAPEMENT

All horizontal escapement values are expressed in relative units. For the LaserWriter, a relative unit is 1/108th of an em, where em is equal in points to the current pointsize requested. For example, in 18 point type, a relative unit is equal to one sixth of a point or six relative units to a point. In 54 point type, a relative unit is equal to one half point. The relative unit value applies to *character widths, spaceband limits, whitespace adjustment, tracking, kerning,* and *letterspacing. Accent characters should have actual width values indicated in the FONT INFORMATION* tables (see chapter 1).

RULE WEIGHTS

On the LaserWriter, rule weights are established by a number which is expressed in multiples of a quarter point. For example, <WR1> would produce a quarter point rule, <WR4> would produce a one point rule, etc.

TYPESETTER REQUIREMENTS

Selection 1 of *OTHER SYSTEM OPTIONS (see chapter 1)* allows the operator to indicate which serial port will output to the LaserWriter. Type in a number with an "S" following.

Selection 2 of allows the operator to indicate a default choice of *portrait* or *landscape* mode (see <PSa,b> command in chapter 3), and a default choice for the paper size.

Selection 3 of *OTHER SYSTEM OPTIONS* allows the operator to establish the transmission protocol for each serial port assigned in selection 1. The proper setup for the LaserWriter is 9600 baud, 8 word, 1 stop, and no parity.

Any changes made to this screen menu requires that the program be rebooted in order to take effect.

KEYSTROKE/OUTPUT TABLE
Apple LaserWriter

This table indicates which typesetter character sequence is associated with which keystroke(s) to produce which output character. The sequence number is also the width value position number in the width screen of the *FONT INFORMATION* table. The pi precedent character (shown on screen as π) is accessed by typing a backslash on the keyboard.

key(s)	output number	char	key(s)	output number	char	key(s)	output number	char
a	57	a	X	24	X	π m	192	¯
b	58	b	Y	25	Y	π 1	179	‹
c	59	c	Z	26	Z	π 2	180	›
d	60	d	&	150	&	π 3	181	«
e	61	e	1	132	1	π B	204	|
f	62	f	2	133	2	π 4	182	»
g	63	g	3	134	3	π zero	131	‰
h	64	h	4	135	4	π h	119	œ
i	65	i	5	136	5	π H	117	Œ
j	66	j	6	137	6	π q	160	¿
k	67	k	7	138	7	π e	159	¡
l	68	l	8	139	8	π x	148	ƒ
m	69	m	9	140	9	π 8	177	“
n	70	n	0	141	0	π 9	178	”
o	71	o	period	169	.	π 7	184	¸
p	72	p	comma	170	,	π s	122	ß
q	73	q	colon	172	:	π v	118	æ
r	74	r	semi-colon	171	;	π V	116	Æ
s	75	s	?	158	?	π z	88	
t	76	t	!	157	!	π Z	32	
u	77	u	(	168	(	π y	106	ø
v	78	v	)	185	)	π Y	49	Ø
w	79	w	hyphen	165	-	π 5	120	fi
x	80	x	shift 6	176	’	π 6	121	fl
y	81	y	open quote	175	‘	π E	147	˜
z	82	z	shift hyph	163	—	π C	128	¤
A	1	A	π hyphen	164	–	π P	151	¶
B	2	B	$	123	$	@	149	@
C	3	C	π $	124	¢	π I	174	˝
D	4	D	%	130	%	π shft hyph	166	
E	5	E	slash	202	/	π comma	143	¯
F	6	F	π slash	129	/	π period	144	˘
G	7	G	π j	125	£	#	156	#
H	8	H	π o	126	·	plus	142	+
I	9	I	π l	198	‚	{	186	{
J	10	J	π r	194	˙	}	200	}
K	11	K	*	155	*	[	201	[
L	12	L	π p	152	§	]	205	]
M	13	M	π d	153	†	=	145	=
N	14	N	π w	154	‡	π shift 6	146	^
O	15	O	π i	98	ı	π S	203	\
P	16	P	π a	187	´	π n	195	˚
Q	17	Q	π g	188	`	π M	115	º
R	18	R	π k	189	ˆ	π J	127	¥
S	19	S	π u	190	¨	π W	114	ª
T	20	T	π t	191	˜	π U	197	˝
U	21	U	π f	199	ˇ	π L	167	…
V	22	V	π b	193	˘	π O	162	•
W	23	W	π c	196	¸	π F	173	˙

KEYSTROKE/OUTPUT TABLE
Apple LaserWriter $PI Access

This table indicates which typesetter character sequence is associated with which $ mnemonic keystrokes. The sequence number is also the width value position number in the width screen of the *FONT INFORMATION* table.

$	char	seq	$	char	seq	$	char	seq	$	char	seq
$GA	A	1	$HU	υ	77	$JO	⟩	84	$LI	)	185
$GB	B	2	$HV	ϕ	62	$JP	⊗	191	$LJ	[	201
$GC	Γ	7	$HW	φ	66	$JQ	⊕	192	$LK	⌈	36
$GD	Δ	4	$HX	χ	59	$JR	∅	193	$LL	⌊	38
$GE	E	5	$HY	ψ	81	$JS	∩	194	$LM	⌊	115
$GF	Z	26	$HZ	ω	79	$JT	∪	190	$LN	⌉	92
$GG	H	8	$IA	ϖ	78	$JU	⊃	30	$LO	⌉	94
$GH	Θ	17	$IB	1	132	$JV	⊇	195	$LP	⌋	122
$GI	I	9	$IC	2	133	$JW	⊄	196	$LQ	]	205
$GJ	K	11	$ID	3	134	$JX	⊆	197	$LR	{	186
$GK	Λ	12	$IE	4	135	$JY	∈	198	$LS	⌈	39
$GL	M	13	$IF	5	136	$JZ	∉	199	$LT	{	40
$GM	N	14	$IG	6	137	$KA	϶	176	$LU	⌊	41
$GN	Ξ	24	$IH	7	138	$KB	⌟	160	$LV	⌋	108
$GO	O	15	$II	8	139	$KC	/	129	$LW	⌉	95
$GP	Π	16	$IJ	9	140	$KD	◊	27	$LX	}	96
$GQ	P	18	$IK	0	141	$KE	∠	163	$LY	⌋	97
$GR	Σ	19	$IL	#	156	$KF	⊥	146	$LZ	}	200
$GS	T	20	$IM	%	130	$KG	∃	123	$MA	⌋	85
$GT	Y	21	$IN	*	155	$KH	∀	174	$MB	⌈	103
$GU	ϒ	159	$IO	′	124	$KI	/	202	$MC	⌋	89
$GV	Φ	6	$IP	″	153	$KJ	∇	101	$MD	⌋	98
$GW	X	3	$IQ	=	145	$KK	ℵ	44	$ME	√	54
$GX	Ψ	25	$IR	≠	184	$KL	ℑ	188	$MF		175
$GY	Ω	23	$IS	≡	178	$KM	℘	189	$MG	&	150
$GZ	α	57	$IT	≅	149	$KN	ℜ	187	$MH	·	111
$HA	β	58	$IU	<	143	$KO	∏	52	$MI	!	157
$HB	γ	63	$IV	>	144	$KP	∑	109	$MJ	?	158
$HC	δ	60	$IW	≤	125	$KQ	↔	181	$MK	:	172
$HD	ε	61	$IX	≥	154	$KR	↑	180	$ML	;	171
$HE	ζ	82	$IY	∧	86	$KS	→	120	$MM	.	169
$HF	η	64	$IZ	∨	93	$KT	↓	121	$MN	,	170
$HG	θ	73	$JA	−	165	$KU	←	179	$MO	...	167
$HH	ϑ	10	$JB	+	142	$KV	|	131	$MP		166
$HI	ι	65	$JC	±	164	$KW	⇔	47	$MQ	®	45
$HJ	κ	67	$JD	×	126	$KX	⇑	53	$MR	©	46
$HK	λ	68	$JE	÷	183	$KY	⇒	110	$MS	™	102
$HL	μ	69	$JF	≈	182	$KZ	⇓	51	$MT	®	29
$HM	ν	70	$JG	~	147	$LA	⇐	104	$MU	©	114
$HN	ξ	80	$JH	¬	43	$LB	(	168	$MV	™	33
$HO	o	71	$JI	∞	127	$LC	⎛	31	$MW	•	162
$HP	π	72	$JJ	∴	203	$LD	⎜	34	$MX	♣	152
$HQ	ρ	74	$JK	|	204	$LE	⎝	35	$MY	♦	128
$HR	σ	75	$JL	ƒ	148	$LF	⎞	87	$MZ	♥	173
$HS	ς	22	$JM	∂	151	$LG	⎟	90	$NA	♠	177
$HT	τ	76	$JN	⟨	28	$LH	⎠	91			

HARDWARE CONNECTION
APPLE LaserWriter

CABLE INSTALLATION

Following is a step by step procedure for connecting the Magna to the Apple LaserWriter.

1. Turn off the LaserWriter.
2. Set the switch at the back of the LaserWriter to "9600".
3. Install the null-modem cable supplied by Magna to the 25 pin connector at the back of the LaserWriter.
4. Install the other end of the Magna cable to the serial port of your PC.

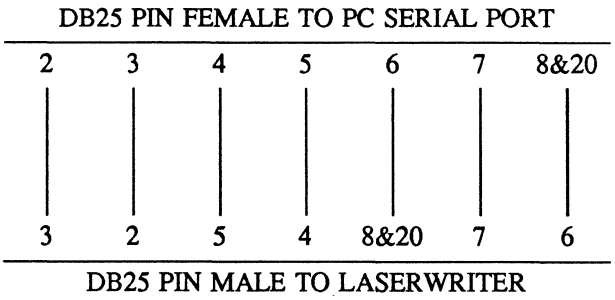
Note 1: The IBM AT and some of its "clones" use a 9-pin connector for the serial port. You need to purchase an adapter (in this case a 9-pin female to 25-pin male adapter) from your computer store in order to connect correctly to the Magna cable.

Note 2: Some PC "clones" use a 25-pin female connector for the serial port. You need to purchase an adapter (in this case a 25-pin male to 25-pin male "gender-changer") from your computer store in order to connect correctly to the Magna cable.

5. Cable installation is complete. Turn on power to the LaserWriter, ensuring that the check-sheet ejected after the self-test shows that you are in "9600" mode.
6. You may now initiate transmission to the Laserwriter, see your Magna manual (Chapter 4, section on "Output").

FOR YOUR INFORMATION

Cable pinouts for Magna to Laserwriter.



INSTALLATION SECTION

Included in this MagnaType Software package is a hardware device called the MagnaLock® and two floppy diskettes, one labeled "Program" and the other "Utilities." Locate these items and carefully proceed with the following instructions.

The MagnaLock must be connected to parallel port one (LPT1) of the PC. It is essential, if your PC is equipped with more than one parallel port, that only one is physically directed to be LPT1. If you are not sure, now would be a good time to become familiar with your PC, its Guide to Operations, and obtain a basic understanding of the DOS functions used to maintain your PC's hard disk drive.

The *PROGRAM* diskette contains four DOS files, two are executable programs, one is a system configuration file, and one is an ANSI.SYS file.

The *UTILITIES* diskette contains eleven DOS files. All of these files have an FIL extension and are essential to the operation of the executable MagnaType program. It is recommended that you make copies of the MagnaType floppy diskettes for backup *(see your DOS manual for the proper procedure).*

LOADING THE SOFTWARE

The following procedure gives instructions on how to create a subdirectory on the hard disk for loading and operating the MagnaType software. In the interest of avoiding any present or future confusion, we strongly suggest that you do NOT load the Magna programs or utilities into the ROOT directory.

1. Bring up the PC and get the DOS prompt C> in the ROOT directory of the hard disk on the screen. It is essential that you begin in the ROOT directory, and necessary that you understand the concept of the ROOT directory.

2. Insert the MagnaType "Program" diskette into the floppy disk drive and copy CONFIG.SYS and ANSI.SYS from the diskette into the ROOT directory using the DOS command:

   copy a:*.sys **RETURN**

   (assuming the use of floppy drive A).

3. Create a subdirectory on the hard disk using the DOS command for "make directory" (MKDIR or MD):
   md *name* **RETURN**
   where *name* is the name you wish to give the subdirectory, something significant such as "LASER" or "L202".

4. GO TO the subdirectory that you created in step 3 by using the DOS command:
   cd *name* **RETURN**

5. Copy the COMP program and the INITL program from the "Program" diskette into this hard disk subdirectory using the DOS command:
   copy a:*.exe **RETURN.**

6. Insert the MagnaType "Utilities" diskette into the floppy disk drive and copy all the files from the diskette into this hard disk subdirectory using the DOS command:
   copy a:*.* **RETURN.**

7. Connect the MagnaLock device to LPT1.

8. Reset the PC by holding down CTRL, ALT, and DELETE all at the same time so the new configuration file may be properly recognized.

9. After getting the C> prompt in the ROOT directory, go to the Magna subdirectory using the DOS command:
   cd *name.*

10. You may now bring up either of the two executable Magna programs.

The composition program may then be called up by typing in xCOMP **RETURN** at the C> prompt. The 'x' is dependent on the type of output device you have, and is indicated by the label on the "Program" diskette. Wait for the program to load into memory which should only take several seconds. When the first screen appears, type in your two initials and the *PRIMARY MENU* will appear. Chapter 1 begins a detailed discussion of the options available.

The INITL (initialize) program is explained in the following section.

INITL PROGRAM

The program disk also contains an INITL.EXE program. This program is booted just as the xCOMP program by typing:

INITL **RETURN**

You will find that it contains selections for the creation or initialization of certain DOS files used by the MagnaType program. Understand that if these selections are utilized, they will clear the contents of these files.

CREATE WORKAREA FILE will wipe out all the existing temporary *JOBS READY FOR OUTPUT* stored in WORK.FIL as well as the four stored blocks in F6. The Activity Log will also be cleared. *See pages 4-11, 4-12, 4-15, and 5-2 for more information.*

CREATE PARAMETER FILE will reset all the STANDARDS AND DEFAULTS stored in PARAM.FIL to the way they were when you first received your MagnaType package, and clear out any stored programmable keys and the mnemonic/format reference table. *See pages 1-2 thru 1-7 for more information.*

CREATE GLOBAL FORMAT FILE will clear the FORMAT.FIL of all global formats and allow the operator to create a new set of globals. *See page 1-8 for more information.*

CREATE JOB FILE will clear all the job/takes available in the MagnaType program stored in JOB.FIL and reset all the directory entries in DIRECTRY.FIL which will then open up all the hard disk sectors previously occupied by MagnaType job/takes.

It is important to understand that all the MagnaType job/takes are stored in the one DOS file called JOB.FIL and must be accompanied by the DIRECTRY.FIL that exists at the same time when a backup is executed. *See pages 4-2 and 4-10 for more information.*

CREATE DICTIONARY FILE will clear all exception word lists. *See pages 1-9 thru 1-11 for more information.*

CREATE FONT FILE will clear out the font directory. *See pages 1-12 thru 1-23 for more information.*

Each of these *.FIL files may be saved prior to reinitializing by copying them to floppy diskettes or to other directories of the hard disk allowing the user to have different sets of STANDARDS AND DEFAULTS, GLOBAL FORMATS, and JOBS ON DISK FOR EDITING. *SEE YOUR DOS MANUAL FOR INSTRUCTIONS ON PERFORMING THESE OPERATIONS.*

Chapter 1
INITIAL PREPARATIONS

System Management Tasks 1-1

System Standards and Defaults 1-2
- System Standards-Values 1-2
- System Standards-Yes or No 1-3
- System Defaults-Values 1-4
- System Defaults-Yes or No 1-5
- Other System Options 1-6

Mnemonic/Format Reference Table 1-7

Global Format Options 1-8

Hyphenation Dictionaries 1-9

Font Information 1-12
- Font Information Table 1-12
  - Character Widths 1-14
  - AutoKern Values 1-15
  - Fixed Spaces/Ligatures 1-16
  - $PI Access Overrides 1-17
  - Tracking 1-18
- Master AutoKern Table 1-19
- $PI Access Table 1-20
- Font Family Assignment Table 1-21
- Display Font Directory 1-22
- Font Management Tasks 1-23

INITIAL PREPARATIONS

Now that you have carefully studied the *INSTALLATION SECTION* and *TYPESETTER SPECIFIC* information provided with this manual, you should consider the information contained in this chapter. Here we will explore the preliminary topics contained in menu selections concerning the effective preparation of the MagnaType Software program.

SYSTEM MANAGEMENT TASKS

Selection 4 of the *PRIMARY SELECTION MENU (Perform System Management Tasks)* allows a *NEW OPERATOR* to sign onto the system, allows the operator to work with the *SYSTEM STANDARDS AND DEFAULTS*, the *MNEMONIC/FORMAT REFERENCE TABLE*, the *GLOBAL FORMAT OPTIONS*, the *HYPHENATION DICTIONARIES*, and the *FONT INFORMATION*.

Selection 1 of the *SYSTEM MANAGEMENT TASKS* allows the current operator to close out and returns the system to the *OPERATOR SIGN ON* screen for a new operator to sign onto the system. The following selections will each be discussed in detail.

The *SYSTEM STANDARDS AND DEFAULTS* contain information to reduce the necessity of redundant coding in system jobs. Before deciding on how to alter any of these selections, you may wish to explore the various features as they are defined throughout this manual.

The *MNEMONIC/FORMAT REFERENCE TABLE* allows the operator to set up meaningful mnemonic code equivalents to format calls.

The *GLOBAL FORMAT OPTIONS* allows the user to enter up to 999 command and/or text strings which may be accessed from any job on the system.

The *HYPHENATION DICTIONARIES* contain the information essential to proper hyphenation.

The *FONT INFORMATION* tables contain all the information essential to proper justification.

SYSTEM STANDARDS AND DEFAULTS

From the *SYSTEM MANAGEMENT TASKS* menu, selection 2 will allow the operator access to several standards and default options. Careful planning of these *STANDARDS AND DEFAULTS* based on the specific needs of your operation will allow the system to make many of your choices automatically, thereby reducing the number of code keystrokes in a job.

TYPE: 2 RETURN

Select one of the five screen options by number followed by RETURN.

1. SYSTEM STANDARDS — VALUE RESPONSES
2. SYSTEM STANDARDS — YES/NO RESPONSES
3. SYSTEM DEFAULTS — VALUE RESPONSES
4. SYSTEM DEFAULTS — YES/NO RESPONSES
5. MAGNAPAGE DEFAULTS — VALUE RESPONSES
6. OTHER SYSTEM OPTIONS

1. SYSTEM STANDARDS - VALUES

Move the cursor to the appropriate specification, type a value followed by RETURN.

FRACTION WIDTH AND HEIGHT: expressed as a percent value of the current pointsize for setting elements in an autofraction (see <BF> command).

SMALL CAPS WIDTH AND HEIGHT: expressed as a percent value of the current pointsize for setting system generated small caps (see <SC> command).

SUPERIOR/INFERIOR WIDTH AND HEIGHT: expressed as a percent of the current pointsize for generating superiors and inferiors (see <IN> and <SU> commands).

STANDARD RULE: expressed in typesetter units (refer to the *TYPESETTER SPECIFIC* supplement)

MAXIMUM POINTSIZE: expressed in points and fraction points, this is the maximum pointsize allowed by the output device.

MAXIMUM COLUMN MEASURE: expressed in picas and points, this is the maximum limit allowed by the output device.

SLUG TO COPY LEADING: expressed in picas and points, this is the space between slugline and start of actual job copy on output.

INTERGALLEY LEADING: expressed in picas and points, this is the space between output jobs on the output device.

HANGING PUNCTUATION INDENT: expressed in picas and points, this is the automatic output indent for hanging punctuation (see <HP> command).

FORCED TO JUSTIFY RANGE: expressed in picas and points, if the last line of a paragraph comes within this range of the margin, the system will force justify the line of type by expanding the interword spacing.

FRACTION BAR $PI CODE: expressed as two characters from the *PI TABLE SETUP PROGRAM* which will assign the true fraction bar character to be used when the system generates autofractions (see <BF> command). If no mnemonic is specified, the system will use the slash character to generate autofractions.

TOTAL KERN FOR (/) IN FRACTION: expressed in relative units to specify spacing around the slash in system generated autofractions. This value may vary depending on the character and output device being used. It is advised that the user experiment until a proper value is determined. *The kern value is ignored if a* $PI *code is used to indicate the existance of a true fraction bar.*

STANDARD FONT FAMILY NAME: expressed in font family names as established in the *FONT FAMILY SETUP PROGRAM*. These are the two font families that will be brought into memory automatically every time the program is booted up. Whenever a change is made to this standard value, the program must be rebooted to take effect. *One of these eight fonts is referenced during compose for production of the job slugline (see SYSTEM DEFAULTS - VALUES below).*

Type ESCape to return to the previous menu.

2. SYSTEM STANDARDS - YES OR NO

Move the cursor to the appropriate specification, type a Y or N followed by RETURN.

NUMERIC HYPHENATION: Yes will allow numeric strings in text to be hyphenated during justification.

AUTO QUOTES: Yes will allow the system to automatically set open or close quotes when the autoquote key is used in a job/take (see *KEYBOARD FUNCTIONS*).

AUTO LEADING: Yes will allow the system to calculate leading when a <CPx> is used without <CLx>. The lead value is calculated by using ⅓ the pointsize of the text line above plus ⅔ the pointsize of the current text line. The <IAx> command is use to specify an automatic addition to this calculated value.

HYPHENATION IN RAGGED: Yes will allow the automatic hyphenation when a ragged mode is requested (see <RC>, <RL>, and <RR> command).

DOUBLE SPACING ON PRINTER: Yes will allow the line printer output to double space. No is single space.

LINE LOOSENESS SHOWN IN PICAS: (as opposed to EN's).

COMBINATION COMMAND <CP/CLxCC,CF>: (as opposed to <CC,CF,CP,CL>). For example, <CP10> <CL12> <CC25> <CFTR> would be keyed as <CP10/12x25,TR>

HYPHENATION PREFIX MINIMUM OF 2 CHARACTERS: (as opposed to 3 characters).

HYPHENATION SUFFIX MINIMUM OF 2 CHARACTERS: (as opposed to 3 characters).

Type ESCape to return to the previous menu.

3. SYSTEM DEFAULTS - VALUES

Move the cursor to the appropriate specification, type a value followed by RETURN.

COLUMN MEASURE: expressed in picas and points, used if no <CCx> is specified in job/take.

FONT NUMBER: expressed as a number from 1 to 8 taken from one of the standard font families *(1-4 is first font family and 5-8 is second font family)*, used if no <CFx> is specified in job/take. *Used to produce job slugline.*

POINT SIZE: expressed in points and fractions, used if no <CPx> is specified in job/take. *Used to produce job slugline.*

LEADING: expressed in points and fractions, used if no <CLx> is specified in job/take. *Used to produce job slugline.*

SET WIDTH: expressed in points and fractions, used if no <CPx> is specified in job/take in combination with default point size. *Used to produce job slugline.*

TRACK NUMBER: expressed as value from 0 to 3, used if no <Tx> is specified in job/take.

PRIMARY LANGUAGE: expressed as an alpha character to determine which language dictionary to use if no <FHx> is specified in job/take.

SPACEBAND VALUES: expressed in relative units, used if no <CSx,y,z> is specified in job/take.

CONTROL RAG: expressed in picas and points, used if no <CRx,y> is specified in the job/take. May enter zeroes if no control rag is desired as default.

MAXIMUM LETTERSPACE UNITS: expressed as value from 0 to 255, used if no <LSx> is specified in job/take and if letterspacing is allowed *(see TYPESETTER SPECIFIC information).*

CONSECUTIVE HYPHEN COUNT: expressed as value from 1 to 255, used if no <HCx> is specified and hyphenation is allowed.

UNDERSCORE RULE THICKNESS: expressed in typesetter units, used if no y is specified in the <USx,y> command in the job/take.

Type ESCape to return to the previous menu.

4. SYSTEM DEFAULTS - YES OR NO

Move the cursor to the appropriate specification, type a Y or N followed by RETURN.

HYPHENATION: Yes is same as <AH> in job/take.

LETTERSPACING: Yes is same as <AL> in job/take.

AUTOKERN: Yes is same as <AK> in job/take.

LIGATURES: Yes is same as <AG> in job/take.

KERN NUMERIC ONE: Yes is same as <KO> in job/take.

HYPHENATE WIDOW: Yes is same as <AW> in job/take.

BUILD FRACTIONS: Yes is same as <BF> in job/take.

Whether mode is set to Yes or No, it can be reversed in the job/take with the appropriate command.

Type ESCape to return to the previous menu.

5. MAGNAPAGE DEFAULTS — VALUE RESPONSES

This is an optional feature of MagnaType. It is described in a supplemental chapter of this documentation.

6. OTHER SYSTEM OPTIONS

This selection will allow the user to set up variables specific to the type of output device and hardware being used.

1. SYSTEM CONFIGURATION

Indicate with 'Y' or 'N' if a color monitor is being used. If there is a line printer physically connected to this unit, and/or if you have a typesetting device physically connected to this unit, indicate which port is being used. *You may only assign one device per port.* If there is no typesetter or line printer, enter **OP** (zero P). Indicate which serial port is being used to connect to the File Server by typing the number 1 or 2. If this system is not connected to a File Server, type a zero.

USE 7 BIT ASCII FOR DOS IMPORT (Y/N).
'Y' indicates the ASCII file will be examined, any high order bits will be stripped and all characters within the range of HEX 20 through 7E will be imported. For example, if HEX A1 exists in the file, it would be imported as HEX 21.

'N' indicates that the ASCII file will be examined and only characters within HEX 20 and 7E will be imported. For example, if HEX A1 exists in the file, it would NOT be imported.

After making any changes to this screen, ESCape back to the *PRIMARY* menu, EXIT to DOS, and then reboot the xCOMP program.

Whenever a change is made to this menu screen, the program must be rebooted for the changes to take effect.

2. TYPESETTER REQUIREMENTS

These options are dependent on the type of output device you are using. *See TYPESETTER SPECIFIC supplement.*

3. PORT ASSIGNMENTS

These options apply only if a serial port is established in selection 1 *SYSTEM CONFIGURATION.* The operator may then establish the baud rate, data bits (word size), stop bits, and parity for each of the serial communication ports *(designated by DOS as COM1, COM2, etc.).*

All of the parameters stored in the selections of *SYSTEM STANDARDS AND DEFAULTS* are stored in the Magna DOS file called PARAM.FIL and may be copied, stored, and reinitialized with the INITL.EXE program. *See the INSTALLATION SECTION.*

MNEMONIC/FORMAT REFERENCE TABLE

The third option under the *SYSTEM MANAGEMENT TASKS* menu is the setup table for associating a ;XX equivalent to a format call. This allows the user to equate meaningful, easily keyed mnemonic codes to arbitrary format numbers using a semi-colon as a precedent key.

For example, assume the standard procedure for your operation is to always put standard text parameters in format 100 and standard extract style text in format 110. It would be possible then to assign ;TX to format 100 and ;EX to format 110. While entering text data, the operator would type ;TX or ;EX rather than <UF100> or <UF110> to call on the proper format parameters.

From the *SYSTEM MANAGEMENT* menu,

TYPE: 3 RETURN

Move cursor to any available position on the screen, enter the two character mnemonic followed by the format number, type RETURN. If the look up is to a *GLOBAL FORMAT*, type a "G" following the format number. After entering all the new combinations required, type ESCape and the combinations will be stored in alpha/numeric order for easy reference. The maximum number of mnemonic combinations allowed is 256, there are 128 locations per screen. Use PGDN and PGUP to get from screen to screen.

> NOTE: You may have two or more different mnemonic codes equal to the same format number, but not more than one format number equal to each mnemonic.

TYPE ESCape to return to the *SYSTEM MANAGEMENT* menu.

GLOBAL FORMAT OPTIONS

The fourth selection of the *SYSTEM MANAGEMENT TASKS* menu allows the operator to load command and/or text strings as global (system wide) formats which may then be accessed from any job/take in the system.

From the *SYSTEM MANAGEMENT TASKS* menu, type 4 and RETURN. Three selections will appear.

Typing 1 and RETURN will display ranges of global format numbers not yet occupied with text and/or command codes. This will assist the operator in selecting an available global for entering new format data. Type ESCape to display the previous menu.

Typing 2 and RETURN will allow the operator to select a global format number for editing an existing format or create a new one. Type a number followed by RETURN and a screen will appear similar to a job/take data entry screen. *Move the cursor right and begin typing format data after the* <SGx> *code and prior to the* <EF> *code using the INSERT key.* After editing or keying in new data, type ESCape for the *EDITING COMPLETE* menu and select a disposition option with the cursor key and type RETURN. The previous menu will be displayed.

Typing 3 and RETURN allows the operator to display the entire format file on the screen. Each format will begin with a <SGx> where x is the global number and end with <EF>. The operator may also select to print the file to the line printer. A window menu will appear and allow the options of printing all global formats to paper in a stacked fashion as they appeared on screen or print a separate format on each sheet of printer paper.

If the operator were to EXIT to *DOS* and request a DIRectory of the hard disk, a file named *FORMAT.FIL* would appear. This *DOS* file contains all the global formats. This file could be replaced by another or copied for archival purposes. The Magna directory to the data contained in the global format file is created when the program is booted from the C> prompt.

All of the global formats stored in the program are stored in the Magna DOS file called FORMAT.FIL and may be copied, stored, and reinitialized with the INITL.EXE program. *See the INSTALLATION SECTION.*

HYPHENATION DICTIONARIES

The Hyphenation Dictionaries are the essential information required by the program to perform proper hyphenation in the H&J process. A hyphenation logic has been designed into the program as well as a comprehensive list of roots and exception words ready for immediate use. However, the user may create additional roots or exception words to the master dictionary as well as up to 99 special or *SECONDARY DICTIONARY* lists for special access in any single job created on the system. These lists would contain exception roots with alternate hyphen points. The user may also examine and/or change any of the existing master dictionary entries, but care must be taken not to undermine the proper hyphenation of all words containing an altered root. Any exception word list may be viewed on the screen or sent to a line printer for examination.

From the *SYSTEM MANAGEMENT TASKS* menu, TYPE 5 and RETURN. Another screen menu will display allowing the operator to enter either the *ENGLISH* (Master) Dictionary or the *SUPPLEMENTARY* Dictionaries. In either case, additional menu screens will appear to allow the examination (or testing) of word hyphenation as the dictionary currently exists, allow a change to be entered, or allow the display (to screen or line printer) of any existing dictionary.

For example, if a word is found in a job to be hyphenating incorrectly, the operator has several options. A discretionary hyphen may be keyed in (ALT H) at the proper hyphenation point for that one specific occurrence; the word could be entered as an <ED*word*> with corrected hyphenation points and take effect for that one job only; or the operator could go into the *HYPHENATION DICTIONARIES* menu and enter the word with the proper break points into the exception table. *ENTER* the English Dictionary, select *EXAMINE OR CHANGE THE HYPHENATION OF A WORD*, and key in the word with the proper hyphenation points indicated by *hyphens*. This new exception may then be stored to the list.

All of the exception word lists of the *HYPHENATION DICTIONARIES* are stored in the Magna DOS file called DICT.FIL and may be copied, stored, and reinitialized with the INITL.EXE program. *See the INSTALLATION SECTION.*

HOW MAGNATYPE HYPHENATION WORKS

When a line ending is reached and it is determined that a "straddle" word must be hyphenated in order to properly justify the line, the first step is to establish where this "straddle" word may legally be broken. The first two characters of the "straddle" word are used to find the starting location in the exception word/root list to begin a comparative search.

If an exact word match is found, the hyphenation points indicated are processed for a word break to complete the line.

If a match is found to be only the first portion of the word, the hyphenation points indicated are processed and this portion of the word is protected from logical processing. The balance of the word is checked against the suffix logic for possible break points or protection from hyphenation.

If no match is found or part of the word is unprotected by exceptions or suffix logic, this word or partial word is then processed by the logic routines to determine possible break points.

Now that the possible break points have been assigned to the "straddle" word, the second step of the hyphenation process begins. The first attempt to justify the line is made by dropping the last portion of the "straddle" word after the last hyphenation point. If the line will justify within the specified parameters, the process is complete. If the line is still too long, the next attempt is made by dropping the last portion of the "straddle" word after the second to the last hyphenation point. This testing will continue until a break point is determined for the "straddle" word that will allow the line to justify with the greatest amount of text.

HOW TO BUILD AN EXCEPTION LIST

The most efficient method for building the exception word/root list is to note incorrectly hyphenated words encountered during production and then on a regular basis enter exception words or roots as necessary. *ENTER* the English Dictionary, select *EXAMINE OR CHANGE THE HYPHENATION OF A WORD*, and key in the word with the proper hyphenation points indicated by *hyphens*. This new exception may then be stored to the list.

It is neither recommended nor practical to enter every word in the Webster's dictionary as an exception word. The probability of a line ending with a word that will not hyphenate correctly is low, it is a time consuming process to enter every word, and a list of this size would take up valuable disk space which might otherwise be used for job storage.

When it is necessary to enter an exception, try to use a root entry that will cover as many words as possible. This will take up less disk space, keep the list of exceptions down to a manageable size, and subsequently allow the hyphenation process to work more efficiently.

For example, assume the word *somersault* is incorrectly hyphenating between the *e* and the *r (some-rsault).* It is not necessary to enter the entire word when entering a root would be more efficient. A *root* consists of the fewest number of characters to force the word to properly break without adversely affecting any other words with the same root. In this case, the root *som-e* would not be sufficient because although it would make *somersault* and *somerset* break correctly, it would adversely affect the words *somebody, someday, somedeal, somehow, someone, someplace, sometime, something, someway, somewhat,* and *somewhere.* To correctly force the hyphenation of all these words, two short roots could be entered. *SOM-ER-.* would cover *somersault* and *somerset. SOME-.* would cover all the other words listed above. With this procedure, it is possible to allow for proper hyphenation of a number of words many times the amount of entries used to construct and build on to an exception list.

All of the exception word lists of the *HYPHENATION DICTIONARIES* are stored in the Magna DOS file called DICT.FIL and may be copied, stored, and reinitialized with the INITL.EXE program. *See the INSTALLATION SECTION.*

FONT INFORMATION

From the *SYSTEM MANAGEMENT TASKS* menu, selection 6 will bring up the *FONT INFORMATION* options. This portion of the MagnaType Program allows the user to establish the essential font information necessary for proper character spacing in justification.

The *FONT INFORMATION* menu contains six selections which will each be explained in detail. Remember, the RETURN key (or ENTER key) is used to move forward through the menus, and ESCape will return the screen to the prior menu.

1. FONT INFORMATION TABLE
2. MASTER AUTOKERN TABLE
3. $PI ACCESS TABLE
4. FONT FAMILY ASSIGNMENT TABLE
5. FONT DIRECTORY
6. FONT MANAGEMENT TASKS

1. FONT INFORMATION TABLE

This is where the specific information for each individual font (or typeface) is located. After typing 1 RETURN from the *FONT INFORMATION* menu, the screen responds with

TYPE IN YOUR FONT ID:

Here you may select a font that already exists for examination or further updates. A list of fonts that reside on the system can be listed with selection 5 of the *FONT INFORMATION* menu, which will be discussed in a section below.

You may also create a new font specification table by typing in a new font ID of your choice consisting of 1 to 6 alpha/numeric characters which is referenced in a job/take with the <CF command.

TYPE: *font id* RETURN

The screen will respond with

FONT DESCRIPTION:

This is the font name given to the font by the user which is listed in the font directory.

For a new font, you may enter up to 24 alpha/numeric characters. How this name is keyed in should be selected carefully, the most descriptive name will allow the operator to easily know exactly which font this is. For example, rather than typing GARAMOND, type in ITC GARAMOND BOOK to distinguish this font from Garamond No. 3, Stempel Garamond, or Simoncini Garamond.

Or the name will appear here if the font selected already exists, in which case a RETURN may be typed to proceed.

TYPE: *font name* RETURN

The screen will respond with

IF ELECTRONIC ITALIC, TYPE IN ORIGINAL FONT ID:

Here you may enter the font ID of a true roman face that will be electronically slanted to create a pseudo-italic when a job/take requests *YOUR FONT ID* entered above. This opportunity only exists when creating a new font table. The width values of the true roman face will automatically be displayed in this font specification table.

TYPE: *existing roman font id* and/or RETURN.

The screen will respond with

FONT OUTPUT IDENTIFICATION NO:

This is the reference number which the composition portion of the program uses to access the font on the output device. You should consult the font loading and storage information for your output device and the *TYPESETTER SPECIFIC* supplement included with this manual to determine what numbers are used to label the fonts.

TYPE: *font number* RETURN

Five more menu selections now appear that are related specifically to the font ID entered. Type the number followed by RETURN to examine, edit, or enter new data.

1. CHARACTER WIDTH VALUES

Here are stored the values for each font character in relative width units appropriate for the output device (see *TYPESETTER SPECIFIC* supplement). The number of character width positions provided will depend on which output device is being addressed.

To enter or change a character width, move the cursor to the position and key in the *width value* followed by RETURN. The cursor will automatically advance to the next position. If this font has been designated as an electronic italic font, the width values cannot be altered.

Accent characters are indicated in this width table by typing the letter "A" prior to the width value. Whether the width values keyed in are zero or actual values depends on the output device (see *TYPESETTER SPECIFIC* information at the beginning of this manual). When a character is requested in a job/take that has the letter "A" entered for its width value, an automatic centering routine is triggered to position this character (an accent) centered over the following character in the text.

As you must realize, it is absolutely essential that this width information be identical to the actual widths of the typesetter font characters in order for proper justification by the MagnaType program. When all new values are entered or you are finished viewing the data, type ESCape for selection of another menu option for this font.

2. AUTOKERN VALUES

This will display all the kern character pairs where the amount of kerning may be entered for each pair of characters. A hyphen can be used to indicate space added between two characters. The *PGDN* and *PGUP* keys will allow access to additional screens. Each screen contains 160 pairs and values. To add or change a value, move the cursor to the desired pair, type the *value* followed by RETURN. The character pairs appearing in this table are copied from the *MASTER AUTOKERN TABLE* generated by entering selection 2 of the *FONT INFORMATION* which will be explained on page 1-19.

Type ESCape to return to the previous menu.

3. FIXED SPACES/LIGATURES

The value of the fixed spaces can be established in relative width units for use when this font is requested in a job/take. When the fixed space codes are used (see em, en, thin, figure, and unit space in chapter three) in a job, the actual amount of horizontal space occupied by each code will depend on the current point size and the values entered here for the current font. *See TYPESETTER SPECIFIC section at the beginning of this manual.*

Ligature character positions are also established for automatic look-up if ligatures are allowed. The *EN LEADER* is the default value for the <WL> and <WX> command.

To add or change a value, move the cursor to the appropriate position, key in the *value* followed by a RETURN.

Type ESCape to return to the previous menu.

4. $ PI ACCESS OVERRIDES

As many as 30 exceptions to the system $PI table may be established for each font. For example, assume $CG is equivalent to a serif copyright symbol on a pi font (*see PI TABLE SETUP PROGRAM* later in this section of the manual). Further assume for this particular font, which is a sans serif typeface, the user wishes $CG to be equivalent to a sans serif copyright symbol. A $PI call, such as $CG, in the text will look to this table of the current font before going to the system *PI TABLE SETUP*.

You will see the headings of three columns with the cursor sitting at the first position. If some values already exist, move cursor to next available location, type the *two character mnemonic* (without the $), type RETURN, type the position of the character on the font, type RETURN, type the *ID* of the font (or asterisk if the character exists on this font).

TYPE ESCape to bring up the previous menu.

5. TRACKING

Tracking is very much like kerning in that the space between characters is adjusted for a tight or loose appearance in the typeset copy. But unlike kerning, which effects only specific pairs of characters, tracking is a specific amount that is subtracted or added equally to every character. As typeset characters become larger, the space between them becomes more obvious, therefore, a tighter track may be required.

MagnaType allows the establishment of three separate tracks for each range of pointsizes which may be requested as necessary in the job/take with the <Tx> command code.

In the track table, a positive number is a reduction of track spacing in relative units. A hyphen may be used for indicating an increase in track spacing.

To enter or change existing values, move the cursor to the appropriate pointsize range, type the value required in each track followed by a RETURN. Use the *SPACE* key to move across entries you do not wish to change. If a pointsize requested in a job/take is between two ranges defined in this table, it will use the values of the lower range.

Type ESCape to return to the *FONT INFORMATION* menu.

2. MASTER AUTOKERN TABLE

This is where the user establishes the list of character kern pairs to be consulted during the justification process if kerning is allowed. This list appears in the *FONT SPECIFICATION TABLE* for each font loaded in the system.

When this table is stored (by typing ESCape) the character pairs are put in alphabetical and numerical order for easy reference. When a new pair is added, it will also be inserted in its proper order in this master table as well as the individual tables for each font where a value may be entered.

When a screen is full with 300 combinations, the *PGDN* key will bring up additional screens for entering up to 1500 total pairs.

To add a character pair, move the cursor to an *OPEN* location, type the appropriate characters followed by RETURN. There is no provision in the program for removing an autokern pair, so care should be taken when establishing this master list. However, if the user decides that a kern pair should not remain in the table, it may be overstriked with a dummy pair that does not already exist. When a new pair is to be added, the dummy pair may then be overstriked.

For example, assume 'ab' is a pair that already exists in the master table and it carries a kern value of 4 in font HR. If 'qq' does not already exist, it may be keyed in over 'ab' in the master table. When the table is stored, 'ab' will be removed and 'qq' will be put in proper alphabetical order. In font HR, the 'ab' will no longer exist and 'qq' will show the value of 4. Now assume that at some later date, the user wishes to add the pair 'av' to the master table. The procedure would be to overstrike 'qq' with 'av' and then go into font HR and replace the old value of 4 (now showing for 'av') with a proper value for this new pair.

Type ESCape to return to the previous menu.

3. $ PI ACCESS TABLE

Here you will find 13 screens with every $PI combination code from AA to ZZ, which may be accessed by using the *PGUP* and *PGDN* keys. After each pair, a font and character position on that font may be loaded with keystrokes and RETURN. Any job/take in the system may contain $PI combinations to request a certain special character. In this table, to specify access to a certain font character regardless of which font is currently being used, type * (asterisk) in place of *FONT ID*. It is recommended that one of these $PI combinations be assigned the character access of the fraction bar character to be used by the system for automatic generation of fractions (see *SYSTEM STANDARDS-VALUES* screen in the *SYSTEM STANDARDS AND DEFAULTS* section).

Type ESCape to return to the previous menu.

4. FONT FAMILY ASSIGNMENT TABLE

This is where the user establishes the layout for *FONT FAMILIES*. Each family may have a name of up to 6 characters. The fonts assigned to a font family may then be called for use in a job/take by using the <FS>, <FI>, <FB>, or <FO> command codes or ALT/S, ALT/I, ALT/B, or ALT/O keystrokes after <FFa>, where *a* is the font family name, establishes the font family.

Any font family setup may be altered in a job/take with the <FFa,b,c,d,e> command code.

Fonts in the font family setup are established by using their *FONT ID* reference (same as in the <CFx> code).

To add or change a font family setup, move cursor to the appropriate line, use the *SPACE* key to move across entries you do not wish to change, and type new entries followed by RETURN. Use *PGUP* and *PGDN* keys to move from screen to screen. Up to 540 families may be established.

For example, assume HR is helvetica roman, HI is helvetica italic, HB is helvetica bold, and HBI is helvetica bold italic. The command <FF3,HR,HI,HB,HBI> establishes Font Family 3 and after appearing in a job/take, alt/S indicates helvetica roman, alt/I indicates helvetica italic, alt/B indicates helvetica bold, and alt/O indicates helvetica bold italic. Now, if you later decide to change from helvetica to times roman, all that is necessary is to change the Font Family code or setup rather that changing every font call in the entire job.

Type ESCape to return to the previous menu.

5. DISPLAY FONT DIRECTORY

This program allows access to the complete list of all fonts loaded in the system. They are listed in alpha and numeric order by ID and the 24 character description assigned by the user is displayed for easy reference. Move the cursor to a specific font on the screen (which may be scrolled) and RETURN will allow direct access to the *FONT INFORMATION TABLE* for that specific font.

Typing *CONTROL/END* will bring the end of the list to the screen and *CONTROL/HOME* will return screen to the beginning of the list. This list may also be sent to a line printer.

6. FONT MANAGEMENT TASKS

Selection 6 of the *FONT MANAGEMENT* menu will bring up several options that may be applied to the font information.

1. CHANGE FONT ID allows the operator to change the mnemonic *FONT ID* of any existing font in the system.

2. DUPLICATE FONT allows the font table to be copied along with all related specification tables to another *FONT ID*.

3. DELETE FONT allows the operator to erase from the system disk a font table and all related information tables.

4. LIST FONT allows the operator to print a font table with all related specification tables to a line printer.

5. ARCHIVE FONT allows the copying of a font with all related information tables to a floppy disk or restore a font from an archive floppy disk.

6. DOWNLOAD FONT WIDTHS FROM TYPESETTER allows the downloading of font width information if the typesetting device supports such a feature.

7. DUPLICATE AUTOKERN WIDTHS allows the user to copy an autokern table from one font to another.

Chapter 2
KEYBOARD FUNCTIONS

Function Keys 2-1
Typewriter Keys 2-2
Numeric Keys 2-6

KEYBOARD FUNCTIONS

Before attempting to perform the functions of the MagnaType Software Package, it is essential that the operator take some time to become familiar with the computer keyboard.

For the purpose of explanation, we will divide the keyboard into three areas. The *FUNCTION KEYS* on the left side of the keyboard, the *TYPEWRITER KEY AREA* in the center, and the *NUMERIC KEYPAD* on the right. Be sure to note the minor differences in some key locations on the computer being used. The IBM-AT® is used for illustration.

THE FUNCTION KEYS

The **F1** key will put a job on screen through the H&J process and allow the display of the text lines as they would break when typeset.

SHIFT plus the **F1** keys will store the job/take on the screen and then allow direct entry into another take of the same job number.

The **F2** key toggles the screen display between line ending width/depth numbers and line ending arrows only.

SHIFT plus the **F2** keys toggles the screen display between full command code display and compressed code display ♦.

SHIFT plus the **F3** keys defines cursor at START of a text block to be removed or copied.

The **F3** key defines cursor at END of text block to be removed or copied and will generate a menu window.

The **F4** key will INSERT a text block defined by SHIFT/F3 and F3.

SHIFT plus the **F4** keys indicates re-insert COPY of last text block inserted with F4.

The **F5** key defines cursor to be on a WORD that will DELETE. Type F5 again to delete.

SHIFT plus the **F5** keys defines cursor to be on a LINE that will DELETE. Type SHIFT/F5 again to delete.

The **F6** key allows the storage keystrokes to be DISPLAYED, ALT/0 through ALT/9 keys plus the first line of each of the four stored BLOCKS.

SHIFT plus the **F6** keys allow the contents of the programmable keys ALT/0 through ALT/9 to be defined for storage.

The **F7** key instructs the character indicated by the cursor to change CASE (shift to unshift or unshift to shift).

SHIFT plus the **F7** keys moves the cursor to the next paragraph.

The **F8** key will display the compressed code in the ♦ indicated by the cursor position.

SHIFT plus the **F8** keys allows the window display of a stored format.

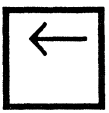
SHIFT plus the **F9** keys allows a search string to be defined.

The **F9** key will continue a search.

The **F10** key allows display of the HELP screens.

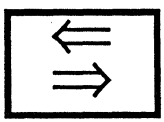
THE TYPEWRITER KEYS

Most of the keys in this center section of the keyboard are standard alpha/numerics and are used the same as on a typewriter for data entry.



The **BACK SPACE** key will delete the character to the left of the cursor and pull the balance of the line to the left.

The **OPEN QUOTE** key is used to override the auto-quotes feature and force an open quote. *Used in the unshift position.*



The **TAB KEY** is used in shift or unshift to precede a superior or inferior character during text entry. It displays as an UP or DOWN ARROW on the screen.

The **CONTROL** key is always used with another key to perform a function. For example, CRTL/END will bring the end of a job/take to the screen.

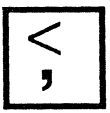
The **SHIFT** keys are always used with another key to produce a character or perform a function. For example, SHIFT ALPHA keys will produce capital letters, or SHIFT/FUNCTION 2 keys will toggle between code supression and code display.

The **ALT** key is always used with another key to perform a function or produce a code. The ALT key is used as a precedent key followed by an alpha key to produce a one-character version of the more frequently used commands.

ALT U	<UN>	Unit Space
ALT T	<TH>	Thin Space
ALT F	<FG>	Figure Space
ALT N	<EN>	En Space
ALT M	<EM>	Em Space
ALT L	<QL>	Quad Left
ALT R	<QR>	Quad Right
ALT C	<QC>	Quad Center
ALT J	<JU>	Force Justify
ALT P	<EP>	End Paragraph
ALT S	<FS>	Standard Font
ALT I	<FI>	Italic Font
ALT B	<FB>	Bold Font
ALT O	<FO>	Other Font
ALT H	<HH>	Discretionary Hyphen
ALT X	<MC>	Merge Copy

The **BACK SLASH** key is used while a job/take is displayed on the screen as a precedent key with many other keys to produce additional characters from the typesetter font (see specific font layout). The **BACK SLASH** shows on the screen as a PI (π) symbol.

The **HYPHEN** key is used to separate a JOB number from its TAKE number when requesting a job. It will produce a hyphen during text entry. *Used in the unshift position.*



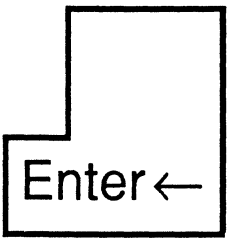
The **SHIFT COMMA** key is used during text entry to produce a MagnaType OPEN command character (see the *PLUS* key on the *NUMERIC KEYPAD*).

The **SHIFT PERIOD** key is used during text entry to produce a MagnaType CLOSE command character to end a variable command code (see the *MINUS KEY* on the *NUMERIC KEYPAD).*
For example, assume the operator wanted an extra lead amount prior to a series of typeset numbers. If <EL123456789 were typed without a close command, the program would not know what amount of extra lead was really required.
So, if 12 points were required before 3456789, the correct keystrokes would be:

<EL12>3456789

The **QUOTE** and **SHIFT QUOTE** key is used during text entry to produce auto-quotes in regular quote applications. Based on adjacent characters, the program automatically generates the open or close quotes if the feature is allowed (see *SYSTEM STANDARDS-YES/NO*).

The **SHIFT 6** key is used to override the auto-quotes feature and force a close quote (or apostrophe).

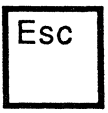


The **RETURN** or **ENTER** key is used to move from one screen menu to another. It is also used to boot up the program and in the *UTILITY* programs to enter variables. During job display, it will bring the cursor to the beginning of the next text line.

The **CAPS LOCK** key is a toggle between shift/alpha characters and unshift/alpha characters. While the lock (shift) is in effect, a C in reverse video will display in the lower right corner of the screen.

Esc	Num Lock	Scroll Lock	Sys Req
7 Home	8 ↑	9 PgUp	PtrSc *
4 ←	5	6 →	-
1 End	2 ↓	3 PgDn	+
0 Ins		. Del	

THE NUMERIC KEYPAD

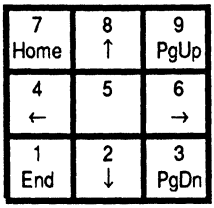


The **ESCape** key is used when moving from one screen menu to another (see section on screens) and to exit after creating or editing a job/take.

The **NUMbers LOCK** key is a toggle to activate the shift versions of the number keys, 0 through 9 and decimal point. While the lock is in effect, an N in reverse video will display in the lower right corner of the screen *and the cursor control keys will be inactive.*

The **SCROLL LOCK** key (not used within the MagnaType program)

The **SYStem** key (not used within the MagnaType program)



The **HOME** key will put the cursor in the upper left corner of the text screen.

The **CONTROL** key plus the **HOME** key will return job/take display to first screen of text.

The **UP ARROW** key moves the cursor up one line for every keystroke.

The **PAGE UP** key used with the **CTRL** key will instantly scroll up one screen of a job/take being displayed.

The **LEFT ARROW** key moves the cursor left one character position for every keystroke.

The **RIGHT ARROW** key moves the cursor one character position for every keystroke.

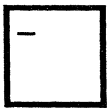
The **END** key will put the cursor in the lower right corner of the text screen.

The **CONTROL** key plus the **END** key will move job/take display to last screen of text.

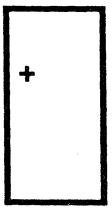
The **DOWN ARROW** key will move the cursor down one screen line for every keystroke.

The **PAGE DOWN** key used with the **CTRL** key will instantly scroll down one screen of a job/take being displayed.

The PRINT SCREEN† key may be used to print the contents of the screen to the line printer. *Used in the SHIFT position.*



The MINUS key will generate a MagnaType CLOSE command during text entry or edit. Same as a shift period.



The PLUS key will generate a MagnaType OPEN command during text entry or edit. Same as a shift comma.

The INSert key will open a line of text at the position indicated by the cursor during text entry or edit to allow additional keystrokes to be entered without overstriking the following characters.

The DELete key will eliminate the character at the current cursor position during text entry or edit.

† Because of the intervention of IBM's BIOS operation with MagnaType, the Print Screen function may not always operate consistently.

Chapter 3
COMMAND CODES

<AG>	Allow Ligatures	3-1
<AH>	Allow Hyphenation	3-2
<AK>	Allow Kerning	3-3
<AL>	Allow Letterspacing	3-4
<AW>	Allow Widow Hyphenation	3-5
<BF>	Build Fractions	3-6
<BN-	Begin Number Counter	3-7
<BP>	Break Page	3-8
<BX-	Set Ruled Box	3-9
<CC-	Change Column Measure	3-10
<CF-	Change Font	3-11
<CL-	Change Leading	3-12
<CP-	Change Pointsize	3-13
<CR-	Change Ragged Zone	3-14
<CS-	Change Spaceband Values	3-15
<CW-	Change Set Width	3-16
<DL-	Delay Lines Depth	3-17
<DM-	Delay Measure Depth	3-18
<DV-	Define Vertical Rule	3-19
<ED-	Enter Dictionary Word	3-20
<EF>	End Format	3-21
<EI-	Escape Immediate	3-22
<EL-	Extra Leading	3-23
<EM>	Em Space	3-24
<EN>	En Space	3-25
<EP>	End Paragraph	3-26
<ES>	Enhanced Scaling	3-27
<ET>	End of Take	3-28
<FB>	Font Bold	3-29
<FF-	Font Family	3-30
<FG>	Figure Space	3-31
<FH-	Foreign Hyphenation	3-32
<FI>	Font Italic	3-33
<FO>	Font Other	3-34
<FS>	Font Standard	3-35
<FT>	FST Transfer	3-36
<HC-	Consecutive Hyphen Maximum	3-37
<HH>	Discretionary Hyphen	3-38
<HP>	Hanging Punctuation	3-39
<IA-	Increment Auto Leading	3-40

<IF-	Indent First Line	3-41
<IH-	Indent Hang	3-42
<IL-	Indent Left	3-43
<IN>	Set Inferior Character	3-44
<IP-	Indent Paragraph	3-45
<IR-	Indent Right	3-46
<IT-	Indent Take	3-47
<IX-	Set Text Indent	3-48
<JT-	Jump Tab Columns	3-49
<JU>	Justify Line (elevate)	3-50
<KA-	Kern Additive	3-51
<KC-	Kern Character	3-52
<KO>	Kern On Number One	3-53
<LS-	Letterspace Maximum	3-54
<MA>	Mark Tab Beginning	3-55
<MC>	Merge Copy	3-56
<MS-	Operator Message	3-57
<NE>	No Escape	3-58
<NF>	No Flash	3-59
<NM-	Job Name Command	3-60
<PD-	Position Font Disk	3-61
<PL-	Paragraph Extra Leading	3-62
<PM-	Point Mark	3-63
<PN>	Print Number Counter	3-64
<PS-	Page Setup	3-65
<PT-	Pass Through Command	3-66
<PX-	Point Maximum	3-67
<QC>	Quad Center	
<QL>	Quad Left	
<QR>	Quad Right	3-68
<QM>	Quad Middle	3-69
<QT>	Quit Tab	3-70
<RC>	Ragged Center	
<RL>	Ragged Left	
<RR	Ragged Right	3-71
<RF>	Reuse Format	3-72
<RI-	Return to Indent	3-73
<RM-	Return to Mark	3-74
<RV-	Reverse Leading	3-75
<RX-	Return to Maximum	3-76
<SC>	Start Small Caps	3-77
<SD-	Supplemental Dictionary	3-78

<SF-	Store Format	3-79
<SL-	Skew Left	3-80
<SP-	Superfont Default	3-81
<SR-	Skew Right	3-82
<SU>	Set Superior Character	3-83
<SV-	Set Vertical Rule	3-84
<SW-	Set White Space	3-85
<T-	Track Selection	3-86
<TB-	Tab Stub	3-87
<TH>	Thin Space	3-88
<TN-	Tab Number	3-89
<TP-	Tab Proportional	3-90
<TR-	Set Tabular Rule	3-91
<TS-	Tab Set	3-92
<TT-	Tab Text	3-93
<UF-	Use Format	3-94
<UG-	Use Global Format	3-95
<UN>	Unit Space	3-96
<US-	Set Underscore	3-97
<VR-	Set Reverse Video	3-98
<WL-	Set Leader Dots	3-99
<WR-	Set Horizontal Rule	3-100
<WX-	Set Leader Character	3-101
<XB>	Cancel Fractions	3-102
<XC>	Cancel Small Caps	3-103
<XD>	Cancel Supplemental Dictionary	3-104
<XF>	Cancel No Flash	3-105
<XG>	Cancel Ligatures	3-106
<XH>	Cancel Hyphenation	3-107
<XI>	Cancel Indent Hang	3-108
<XK>	Cancel Kerning	3-109
<XL>	Cancel Letterspacing	3-110
<XM>	Cancel Point Marks	3-111
<XO<	Cancel Kern Numeric One	3-112
<XP>	Cancel Hanging Punctuation	3-113
<XR>	Cancel Ragged	3-114
<XS>	Cancel Enhanced Scaling	3-115
<XT>	Cancel Tab	3-116
<XW>	Cancel Widow Hyphenation	3-117
<XX>	Select Output	3-118
<XY-	Move to X-Y Coordinate	3-119
<ZL>	Zero Leading	3-120

ALLOW LIGATURES

Code	<AG>
Variables	n/a
Alt Key	n/a
Cancel Code	<XG>
Explanation	The code will allow the automatic look up of ligature combinations for the current font as specified in the Font Information Table. May be established as a system default.
Special Conditions	If established as default, <XG> will act as a temporary cancel until an <AG> or the end of job/take. If NOT established as default, <AG> will enable ligatures until <XG> or end of job/take. Ligature combinations will be ignored in a text line that is letterspaced.

ALLOW HYPHENATION

Code	<AH>
Variables	n/a
Alt Key	n/a
Cancel Code	<XH>
Explanation	The code will enable the program to hyphenate line endings as necessary to facilitate H&J. May be established as a system default
Special Conditions	If established as default, <XH> will act as a temporary cancel until an <AH> or the end of job/take. If NOT established as default, <AH> will enable hyphenation until <XH> or end of job/take.

ALLOW KERNING

Code	<AK>
Variables	n/a
Alt Key	n/a
Cancel Code	<XK>
Explanation	The code will enable the automatic look up of kern pair combinations as specified for the current font in the Font Information Table. May be established as a system default.
Special Conditions	If established as default, <XK> will act as a temporary cancel until an <AK> or the end of job/take.

ALLOW LETTERSPACING

Code	<AL>
Variables	n/a
Alt Key	n/a
Cancel Code	<XL>
Explanation	The code will enable the program to select the letterspacing option as necessary to facilitate H&J. May be established as a system default
Special Conditions	If established as default, <XL> will act as a temporary cancel until an <AL> or the end of job/take. If NOT established as default, <AL> will enable letterspacing option until <XL> or end of job/take.

ALLOW WIDOW HYPHEN

Code	<AW>
Variables	n/a
Alt Key	n/a
Cancel Code	<XW>
Explanation	The code will allow the hyphenation of the last word in a paragraph if it is necessary to avoid the next to last line being set loose. May be established as a system default.
Special Conditions	If established as default, <XW> will act as a temporary cancel until <AW> or the end of job/take. If NOT established as default, <AW> will allow widow hyphenation until an <XW> or the end of a job/take. If hyphenation is NOT allowed, the <AW> command is not sufficient to allow a widow to hyphenate.

BUILD FRACTIONS

Code	<BF>
Variables	n/a
Alt Key	n/a
Cancel Code	<XB>
Explanation	The code will allow the program to create a fraction with numerals from the current font and the specified fraction bar character. The fraction is identified as a numeric string followed by a regular slash followed by a numeric string. If a whole number preceeds the fraction, it must be seperated by a hyphen (which will not be output).
Example	To create: 25½ Keyboard as: 25-1/2
Special Conditions	Build fractions may be established as a system default. Percentage of reduction for numerator and denominator is established in the system standards-values. The fraction bar $PI code is established in the system standards-values. The amount of kern or white space around the fraction bar is established in the system standards-values.

BEGIN NUMBER COUNTER

Code	<BNx,y,z>
Variables	x = a number from 1 to 4000 y = a number from 1 to 9 z = a number from 1 to 4
Alt Key	n/a
Cancel Code	n/a
Explanation	The command is used to initiate the automatic number counter which is printed and incremented by 1 for every <PN> command. The first variable indicates with which number to start printing the sequence. The second variable indicates which of the 9 counters to initiate (same as the x variable in the <PNx> command). The third variable indicates the number of digits for which to hold a place. A positive number instructs the setting of leading zeros, a hyphen before the number instructs the setting of leading figure spaces *(see chapter 6).*
Special Conditions	The first <PNx> command in a job/take will begin print with 1 if no <BN command is indicated. A missing variable will default to 1.

BREAK PAGE

Code	<BP>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to indicate a page end to the sheet feed output device and to begin output of a new page.
Special Conditions	this code is only valid with PostScript output modules such as LCOMP for the LaserWriter.

SET RULED BOX

Code	<BXa,b,c>
Variables	a = width of box in picas and points b = depth of box in picas and points c = rule weight
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to indicate the setting of a ruled box of width and depth indicated. The command code must be followed by a quad code that will determine the horizontal position of the box on the measure. After the setting of the box, the current vertical position will be the rule at the base of the box.
Special Conditions	If c is specified, it will override the default rule weight. To set material within the box, use a <PMx> prior to the <BX code and a <RMx> following the quad code.

CHANGE COLUMN MEASURE

Code	<CCx>
Variables	x = a number of picas and points
Alt Key	n/a
Cancel Code	another <CC command
Explanation	One of the four basic parameters for typeset material, it indicates the full width of justified copy and the maximum width for ragged copy.
Example	To set measure of 21 picas, Keyboard as: <CC21> or, to set measure of 21 picas and 8 points, Keyboard as: <CC21.8>
Special Conditions	The maximum column measure is limited by the output device as indicated in the System Standards and Defaults, If no <CC is indicated, the program will use the system default-value.

CHANGE FONT

Code	<CFx>
Variables	x = 1 to 6 alpha/numeric characters
Alt Key	see <FF Font Family command
Cancel Code	another <CF command
Explanation	Used to indicate the typeface to be set.
Special Conditions	The font requested must be set up in the *FONT INFORMATION TABLE.*

CHANGE LEADING

Code	<CLx>
Variables	x = a number of points and fraction/points
Alt Key	n/a
Cancel Code	another <CL command
Explanation	One of the four basic parameters for typeset material, it indicates the amount of vertical space between the baseline of the current line to the baseline of the type above it.
Example	To vertically advance 12 points between baselines, Keyboard as: <CL12> or, to vertically advance 121/2 points between baselines, Keyboard as: <CL12.5>
Special Conditions	The fraction increments are determined by the output device. If auto-leading is established in the system standards, and no <CL is indicated following a <CP command, the program will calculate the leading based on the formula of ⅓ the point size of the type above plus ⅔ the point size of the type below. If auto-leading is *not* established in the system standards, and no <CL is indicated at the beginning of the job/take, the program will use the system default-value. If auto-leading is established in the system standards-mode, a <CL immediately following a <CP will override auto-leading. <CL takes effect on the line it appears.

CHANGE POINTSIZE

Code	<CPx>
Variables	x = a number of points and fraction/points
Alt Key	n/a
Cancel Code	another <CP command
Explanation	One of the four basic parameters for typeset material, it indicates the set size of the characters.
Example	To set 24 point type, Keyboard as: <CP24> or, to set 241/2 point type, Keyboard as: <CP24.5>
Special Conditions	The maximum and minimum point size allowed is determined by the output device. The fraction increments are determined by the output device. If no <CP is indicated, the program will use the system default-value.

CHANGE RAGGED ZONE

Code	<CRx,y>
Variables	x = the distance in picas and points from the end of the measure where the line may terminate, the *RAGGED ZONE.* y = the space within the ragged zone that straddles the prior line ending which must be avoided when the current line is terminated, the *HOT ZONE.*
Alt Key	n/a
Cancel Code	another <CR command or end of job/take
Explanation	The Ragged Zone command is used to control the aesthetics of ragged line endings by establishing the amount of change from one line to another.
Special Conditions	There is NO hot zone for the first line ending to avoid. Spaceband expansion and letterspacing (if allowed) will be used to keep a line from ending within the hot zone of prior line. The *HOT ZONE* specifier could be 0 if only a line ending range is desired. If no control over the rag is required, use 0 for both specifiers.

CHANGE SPACEBAND VALUES

Code	<CSx,y,z>
Variables	x = minumum spaceband value y = optimum spaceband value z = maximum spaceband value where values are specified in relative units
Alt Key	n/a
Cancel Code	another <CS command
Explanation	The code is used to establish the range of spaceband variation for justified copy.
Special Conditions	*Minimum* must be less than or equal to *optimum* which is less than or equal to *maximum* which is less than or equal to 255. Optimum spaceband value is used for ragged copy UNLESS a ragged zone is in effect. Zero is not a legal value A missing value uses system default-value, for example, <CS3,,9> will use default for optimum value

CHANGE SET WIDTH

Code	<CWx>
Variables	x = a number of points and fraction/points
Alt Key	n/a
Cancel Code	another <CW or <CP command
Explanation	Used to indicate a character typeset width that is different than the current point size. Can be used to create condensed or expanded typefaces as allowed by the output device.
Example	To set 24 point with 18 point widths, Keyboard as: <CP24> <CW18>
Special Conditions	The maximum variation between current point size and set width is limited by the output device. The fraction increments same as <CP command. If no <CW is indicated, the program will use the system default-value until a <CP is indicated.

DELAY LINES DEPTH

Code	<DLx>
Variables	x = a number of text lines
Alt Key	n/a
Cancel Code	upon completion or a <MC> in the text
Explanation	The code is used within a format to act as a conditional merge code. The program will consider all commands in a format up to the <DL, exit the format until the number of text lines is completed, and then return to the format for further command instructions.
Example	Assume the operator wants a 2 pica indent for 10 lines of text followed by an indent of 5 picas for the balance of the copy to be set, Keyboard as: <SF1> <IT2> <DL10> <IT5> <EF> <UF1> that is: store format 1 indent take 2 picas delay lines 10 indent take 5 picas end format use format
Special Conditions	If another delay line format is requested prior to completion of the current delay line in effect, the program will attempt to complete the second level delay line format before returning to the first.

DELAY MEASURE DEPTH

Code	<DMx>
Variables	x = a vertical depth in picas and points
Alt Key	n/a
Cancel Code	upon completion or a <MC> in the text
Explanation	The code is used within a format to act as a conditional merge code. The program will consider all commands in a format up to the <DM, exit the format until the amount of depth is completed, and then return to the format for further command instructions.
Example	Assume the operatior wants a 2 pica indent for a depth of 10 picas followed by an indent of 5 picas for the balance of the copy to be set, Keyboard as: <SF1> <IT2> <DM10> <IT5> <EF> <UF1> that is: store format 1 indent take 2 picas delay measure depth 10 picas indent take 5 picas end format use format or, if depth of 10 picas and 8 points, Keyboard as: <DM10.8>
Special Conditions	If another delay measure format is requested prior to completion of the current delay in effect, the program will attempt to complete the second level delay before returning to the first. The delay measure is considered complete on the first text line that reaches or exceeds the depth indicated.

DEFINE VERTICAL RULE

Code	<DVx>
Variables	x = a number from 1 to 40
Alt Key	n/a
Cancel Code	<SV set vertical command
Explanation	The <DV command establishes a start point for the construction of a vertical rule at the horizontal point in the text on the baseline.
Example	see <SV set vertical command
Special Conditions	Keying <DV> will be considered as <DV1>.

ENTER DICTIONARY WORD

Code	<EDx>
Variables	x = a word of up to 24 characters
Alt Key	n/a
Cancel Code	end of job/take
Explanation	Used to enter a word to the dictionary with hyphens to indicate possible word breaks. Words entered could override or be added to current dictionary and are in effect only for current job/take.
Example	Assume the operator does not wish for the word *computer* to be hyphenated within the current job/take, Keyboard as: <EDCOMPUTER> or, assume the operator wishes for the word *computerization* to only be hyphenated between the *z* and *a*, keyboard as: <EDCOMPUTERIZ-ATION>
Special Conditions	Number of entries is limited to a total of 100 characters.

END FORMAT

Code	<EF>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The <EF> is used to indicate the end of a format. The program will consider all data after a <SF command as part of the format until it encounters an <EF command.
Special Conditions	As with any terminate code, care must be taken not to forget the <EF command after the <SF command or an entire job/take could mistakenly become a format. An <EF> command could be used in the text to terminate an open format rather than a <MC>.

ESCAPE IMMEDIATE

Code	<EIx,y>
Variables	x = a number of picas and points y = rule thickness value (as required by the output device)
Alt Key	n/a
Cancel Code	n/a
Explanation	Used in text to indicate an immediate horizontal escapement with or without a rule.
Example	Assume the operator requires a space of 38 points between two words that will not be expanded or contracted during justification, Keyboard as: word<EI3.2>word or, assume a 42 point rule of thickness value 3 between two words, keyboard as: word<EI3.6,3>word
Special Conditions	An <EI command will not expand or contract to facilitate justification.

EXTRA LEADING

Code	<ELx>
Variables	x = a number of points and fractions
Alt Key	n/a
Cancel Code	n/a
Explanation	Used in the text to indicate an immediate vertical escapement.
Special Conditions	The <EL command will add to the <CL that is currently in effect if no typeset characters have yet appeared on the current line. An <ELx> may be used to return to a baseline adjusted by a prior <RVx> of same x value if both appear before the next quad code.

EM SPACE

Code	<EM>
Variables	n/a
Alt Key	M
Cancel Code	n/a
Explanation	A fixed space in relative units as specified in the *FONT INFORMATION TABLE* of the current font.
Special Conditions	Unlike a spaceband in justified copy, the <EM> is NOT variable to facilitate H&J. The width value of the <EM> will key off the current <CW in effect.

EN SPACE

Code	<EN>
Variables	n/a
Alt Key	N
Cancel Code	n/a
Explanation	A fixed space in relative units as specified in the *FONT INFORMATION TABLE* of the current font.
Special Conditions	Unlike a spaceband in justified copy, the <EN> is NOT variable to facilitate H&J. The width value of the <EN> will key off the current <CW in effect.

END PARAGRAPH

Code	<EP>
Variables	n/a
Alt Key	P
Cancel Code	n/a
Explanation	The end paragraph command is used to indicate the end of a paragraph instead of a specific quad command. The quad effect of an <EP> will depend on the type of rag indicated. The <EP> is also used as a trigger for some indent commands and as a cancel to other indent commands.
Special Conditions	An <EP> in itself is not sufficient to generate a line of leading. It must be preceded by a typeset character. Multiple <EP> commands will be considered as one <EP>. To add a line of lead following an end of paragraph, Keyboard as: ...text<EP> <EM> <EP>

ENHANCED SCALING

Code	<ES>
Variables	n/a
Alt Key	n/a
Cancel Code	<XS>
Explanation	The code is used to turn on the enhanced scaling feature of the APS output device if it is so equipped.

END OF TAKE

Code	<ET>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The <ET> indicates the end of the job/take. It can be used to eliminate the balance of a job/take without having to manually delete every line of type following.
Special Conditions	When a newly created job/take is stored to disk, the program automatically generates an <ET> at the end.

FONT BOLD

Code	<FB>
Variables	n/a
Alt Key	B
Cancel Code	another change font command
Explanation	The <FB> or ALT/B command may be used in a job/take as an alternate way of requesting a font change. It is the third font in the current font family in effect.
Special Conditions	Use of the ALT/B keys produces high intensity characters on the monochrome screen to represent BOLD type.

FONT FAMILY

Code	<FFa,b,c,d,e> or <FFa>
Variables	a = font family name b = standard font c = italic font d = bold font e = other font where family name and font may be 1 to 6 alpha/numeric characters
Alt Key	see <FS>, <FI>, <FB>, and <FO> commands
Cancel Code	n/a
Explanation	<FFa> is used to establish a font family other than the default font family for the current job/take. <FFa,b,c,d,e> is used to establish a font family for the current job/take as well as font overrides within the family.
Example	Assume HELV (helvetica) and TR (times roman) are the two default font families of the program, but the operator wishes to use fonts from the GARA (garamond) family in the current job/take, Keyboard as: <FFGARA> at the beginning of the job/take or, assume the font family HELV (helvetica) is H (helvetica), HI (helv italic), HB (helv bold), and NP (news pi) in the *FONT FAMILY ASSIGNMENT TABLE*, but the operator wishes to use HBI (helv bold italic) as the ALT/O in the current job/take, Keyboard as : <FFHELV,,,,HBI> in the job/take If an invalid font ID is assigned to a *FONT FAMILY*, the request in a job/take for that family will show a FONT error message.

FIGURE SPACE

Code	<FG>
Variables	n/a
Alt Key	F
Cancel Code	n/a
Explanation	A fixed space in relative units as specified for the current font. The value of the <FG> is equal to the width value of the figure 0.
Special Conditions	Unlike a spaceband in justified copy, the <FG> is NOT variable to facilitate H&J. The <FG> width value will be affected by tracking and <SW just as the width value of the number 0 would be affected.

FOREIGN HYPHENATION

Code	<FHx
Variables	x = single alpha character indicating the language to be used E = English F = French S = Spanish I = Italian G = German D = Danish P = Portugese
Alt Key	n/a
Cancel Code	another <FH command or end of job/take
Explanation	The code is used to indicate a foreign hyphenation dictionary and logic program be used other than the system default.

FONT ITALIC

Code	<FI>
Variables	n/a
Alt Key	I
Cancel Code	another change font command
Explanation	The <FI> or ALT/I command may be used in a job/take as an alternate way of requesting a font change. It is the second font in the current font family in effect. Use of the ALT/I keys produces underscored characters on the monochrome screen to represent ITALIC type.

FONT OTHER

Code	<FO>
Variables	n/a
Alt Key	O
Cancel Code	another change font command
Explanation	The <FO> or ALT/O command may be used in a job/take as an alternate way of requesting a font change. It is the fourth font in the current font family in effect. Use of the ALT/O keys produce high intensity, underscored characters on the monochrome screen to represent BOLD ITALIC type.

FONT STANDARD

Code	<FS>
Variables	n/a
Alt Key	S
Cancel Code	another change font command
Explanation	The <FS> or ALT/S command may be used in a job/take as an alternate way of requesting a font change. It is the first font in the current font family in effect.

FST TRANSFER

Code	<FT>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to begin a job/take containing FST information or font transfer instructions for the L/202. The contents must be keyed exactly as if it were a paper tape.
Special Conditions	To transfer the job to the L/202, it must first be *PREPARED FOR OUTPUT* and then sent to the L/202 as a regular MagnaType job when the typesetter instructions require a paper tape to be input.

CONSECUTIVE HYPHEN MAXIMUM

Code	<HCx>
Variables	x = a number from 1 through 255
Alt Key	n/a
Cancel Code	another <HC, <XH or end of take
Explanation	The code is used to control the number of consecutive lines hyphenated.
Special Conditions	For ragged copy, the number of consecutive hyphens allowed is one fewer than specified in the code or system default-value if hyphenation is allowed in ragged mode.

DISCRETIONARY HYPHEN

Code	<HH>
Variables	n/a
Alt Key	H
Cancel Code	n/a
Explanation	This code embedded in a word indicates line breaking possibilities should it be necessary to justify a line properly.
Special Conditions	This code at the end of a word indicates the word should not break over a line ending.

HANGING PUNCTUATION

Code <HP>

Variables n/a

Alt Key n/a

Cancel Code <XP>

Explanation The code indicates the following copy will be indented a number of points as specified in the system standards-values from the left where start of line punctuation will be set adjacent to the first alpha/numeric on the line. End of line punctuation will be set beyond the end of the measure.

START OF LINE
- Single quote
- Double quote
- Asterisk

END OF LINE
- Period
- Hyphen
- Comma
- Single quote
- Double quote
- Asterisk

INCREMENT AUTO LEADING

Code	<IAx>
Variables	x = a number of points and fractions
Alt Key	n/a
Cancel Code	another <IAx> or <IA0>
Explanation	If AUTO-LEADING is enabled in system standards-mode, the leading is calculated by adding ⅓ the point size of the prior line to ⅔ the point size of the current line (if no <CLx> is specified). The <IAx> code specifies amount to be added to this calculation.
Special Conditions	<IA0> does NOT cancel auto-leading.

INDENT FIRST LINE

Code	<IF[R]x,y>
Variables	x = left indent in picas and points y = right indent in picas and points R = values are relative ems and 1/18ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	<IFx,y> where x and/or y is zero, or end of take
Explanation	Establishes an amount of indent for the next text line following an <EP>.
Special Conditions	May be used in a format or at the beginning of a job/take to establish an automatic indent following every <EP> code. To cancel the indent, an <IF0> must appear prior to the <EP> before the line that is NOT to be indented. Indents First add to indent take <ITx,y>

INDENT HANG

Code	<IH[R]x,y>
Variables	x = left indent in picas and points y = right indent in picas and points R = values are relative ems and 1/18ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	<XI> or another <IHx,y> or end of take
Explanation	Establishes an amount of indent for all text lines following the first line of a paragraph (following an <EP> code). The first line sets full measure.
Special Conditions	The <EP> that cancels an <IXn> could be used to trigger an <IHx,y> code. Hang indents add to indent take <ITx,y> The indents take effect immediately on the line following the code.

INDENT LEFT

Code	<IL[R]$x_1/y_1,...,x_{20}/y_{20}$>
Variables	x = a number of lines y = amount of indent in picas and points R = y values are relative ems and ⅛ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	completion of lines, another <IL string or end of take
Explanation	Up to 20 pairs of x/y values may be included in one code string, used to set up a shape of line indents. Produces same effect as combining <IT and <DL codes without need of a format. Left indents add to <ITx> indent take

SET INFERIOR CHARACTER

Code	<IN>
Variables	n/a
Alt Key	n/a
Cancel Code	after one character
Explanation	The code indicates that the following character is to be set reduced in size and width as specified in the system standards-values and sightly below the baseline.
Special Conditions	The program will convert the <IN> code to display a *DOWN ARROW* on the screen. The *UNSHIFT/TAB* key will also generate a *DOWN ARROW* inferior code.

INDENT PARAGRAPH

Code	<IP[R]x,y>
Variables	x = left indent in picas and points y = right indent in picas and points R = values are relative ems and 1/18ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	another <IPx,y>, <EP> or end of take
Explanation	Establishes amounts of indent for all text lines following until an <EP> is encountered. Same as <ITx,y>, but for a paragraph only.
Special Conditions	Multiple <IPx,y> codes could be used in a format with <DLx> or <DMx> codes to set up various levels of indents. Paragraph indents add to <ITx,y> (indent take).

INDENT RIGHT

Code	<IR[R]$x_1/y_1,...,x_{20}/y_{20}$>
Variables	x = a number of lines y = amount of indent in picas and points R = y values are relative ems and ⅛ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	completion of lines, another <IR string or end of take
Explanation	Up to 20 pairs of x/y values may be included in one code string, used to set up a shape of line indents from the right. Produces same effect as combining <IT,y> and <DL codes without need of a format Right indents add to <IT,y> (indent take).

INDENT TAKE

Code	<IT[R]x,y>
Variables	x = left indent in picas and points y = right indent in picas and points R = values are relative ems and ⅛ths units optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	another <ITx,y> or end of take
Explanation	Specifies an amount of left and/or right indent for all copy starting with the next complete line
Special Conditions	The amount of indent is subtracted from the total <CCx> specified and also creates a new point for which other indents are counted: INDENT FIRST INDENT PARAGRAPH INDENT HANG INDENT LEFT INDENT RIGHT

SET TEXT INDENT

Code	<IXn>
Variables	n = number from 1 to 20
Alt Key	n/a
Cancel Code	<EP> or end of take
Explanation	The horizontal position of the <IX code is remembered and each line of type following is indented to that position until an <EP> is encountered. The <RIn> code with n = the number in the <IXn> code will indent again all the following copy lines to the same position remembered until an <EP> is encountered.
Special Conditions	If n is not specified, the program assumes it to be 1, <IX> is same as <IX1>. If another <IXn> replaces a prior <IXn> of same n, the new indent is retained. <XY commands will adjust the horizontal positions of <IXn>, but indent take <IT will not.

JUMP TAB COLUMNS

Code	<JTx>
Variables	x = number of columns
Alt Key	n/a
Cancel Code	A quad code
Explanation	Used in a tabular line to instruct the following copy to consider the current column plus the number x of following columns as one column. Used to construct straddle heads or a temporary combination of columns.
Special Conditions	A <JTx> in effect will suspend the automatic setting of vertical gutter rules if they have been requested.

JUSTIFY LINE (ELEVATE)

Code	<JU>
Variables	n/a
Alt Key	J
Cancel Code	n/a
Explanation	Code is used to quad justify a line of text that would normally set short of measure.
Special Conditions	Code is used in place of a quad code.

KERN ADDITIVE

Code	<KAx>
Variables	x = a number of relative width units
Alt Key	n/a
Cancel Code	after one character
Explanation	The code is used to reduce the width space occupied by the previous character. Generally used to selectively tighten a pair of characters.
Special Conditions	A negative value (using a hyphen) can specify additional width space to a character. A <KAx> placed between two characters in the text will add or subtract to the value specified for that pair in the autokern table if there is one.

KERN CHARACTER

Code	<KCx>
Variables	x = a number of relative width units
Alt Key	n/a
Cancel Code	after one character
Explanation	The code is used to reduce the width space occupied by the previous character. Generally used to selectively tighten a pair of characters.
Special Conditions	A negative value (using a hyphen) can specify additional width space to a character. A <KCx> placed between two characters in the text will override the value specified for that pair in the autokern table if there is one.

KERN ON NUMBER ONE

Code	<KO>
Variables	n/a
Alt Key	n/a
Cancel Code	<XO>
Explanation	The code instructs the program to kern the number 1 in combination with any other character.
Special Conditions	May be established as a system default-mode. Automatically cancelled in the tabular mode unless coded following the <MA>.

LETTERSPACE MAXIMUM

Code	<LSx>
Variables	x = relative units 0 through 99
Alt Key	n/a
Cancel Code	<LS0>,<XL> or end of take
Explanation	Established in the system default-values, the code is used to override this default value for the current job/ take.

MARK TAB BEGINNING

Code	<MA>
Variables	n/a
Alt Key	n/a
Cancel Code	<QT> or another <MA>
Explanation	The code is used to indicate the beginning of a tabular line controlled by a prior tab set-up instruction. *SEE* tab text, tab number, tab set, tab proportional, and tab stub
Special Conditions	A tab set-up command must already be established. Automatically sets a <QT> to end prior <MA> line and begins a new tabular line at the left margin.

MERGE COPY

Code	<MC>
Variables	n/a
Alt Key	X
Cancel Code	n/a
Explanation	The merge copy code (or RETURN code) is used in formats to break out and return to the text stream, or in text to indicate a return to the format.

OPERATOR MESSAGE

Code	<MSn>
Variables	n = a string of alpha/numeric characters
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to key a message anywhere in a job/take to alert another operator of any special circumstances. The code is ignored during H&J and preparation for output. It will display as underlined text on the monochrome monitor and in the color of ALT/I text on the color monitor.
Special Conditions	The message string may be completed with a close command (>) or another MagnaType code.

NO ESCAPE

Code	<NE>
Variables	n/a
Alt Key	n/a
Cancel Code	after one character
Explanation	The code is used to indicate the following character is to be set as if it had no width value. The character will be printed but no escapement will take place.
Special Conditions	If the last character on the full measure text line is <NE>, its width will not be considered in calculating the line. The effect is that character will set outside the right margin. A <NE> character in the middle of the text line will overprint the following character.

NO FLASH

Code	<NF>
Variables	n/a
Alt Key	n/a
Cancel Code	<XF> or end of line
Explanation	The code is used to instruct the output device not to print the following material, but to escape across the measure as if the copy was being printed.

JOB NAME COMMAND

Code	<NMx>
Variables	x = up to 16 alpha/numerics
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used at the beginning of an ascii file. When the job is imported to MagnaType, the code is translated into a job name that will appear in the job directory.

POSITION FONT DISK

Code	<PDn,...n>
Variables	n = number from zero through 999
Alt Key	n/a
Cancel Code	n/a
Explanation	This code is used by the operator to communicate the necessary font disks that must be loaded on the output device to accomodate the font requests within the job/take.
Special Conditions	A <PD command may be entered anywhere within a job/take, and more than one disk number may be specified in one <PD command as specified in the *TYPESETTER REQUIREMENTS* menu of *OTHER SYSTEM OPTIONS.* The disk numbers indicated with the <PD commands will display in the *JOBS READY FOR OUTPUT* list and set as part of the job/take slug on output. If more than the specified number of disks are requested, a FONT DISK error will display in the job/take on H&J. This command is applicable only to floppy font disk output devices.

PARAGRAPH EXTRA LEADING

Code	<PLx>
Variables	x = a number of points and fractions
Alt Key	n/a
Cancel Code	another <PLx> where x could be zero
Explanation	Used in the text to indicate an amount of extra leading between paragraphs. The amount of leading specified will be added to the current lead of a text line following an <EP> command.

POINT MARK

Code	<PMx>
Variables	x = a number from 1 to 6
Alt Key	n/a
Cancel Code	<XM>, another <PMx> same x or end of take
Explanation	The code is used to mark and remember a vertical depth in a job/take. The point actually remembered is the current baseline minus the current leading. *SEE* return to mark, return to maximum, and clear maximum.

PRINT NUMBER COUNTER

Code	<PNx>
Variables	x = a number from 1 to 9 to indicate which number counter is to be printed
Alt Key	n/a
Cancel Code	n/a
Explanation	The current number in the designated automatic number counter will typeset in place of the <PNx> in the current typeface and pointsize. The counter will also be incremented by one for the following <PNx> code of same x.
Special Conditions	The code may be used in combination with the <SU> and <IN> codes to produce superior or inferior numbers that sequence automatically. *SEE* <BNx> code.

PAGE SETUP

Code	<PSa,b>
Variables	a = 0 (zero) for portrait set 1 for landscape b = 0 (zero) for 11 inch length paper 1 for 14 inch length paper
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used at the beginning of a job/take to indicate how the typeset material is to be positioned on the output paper. The first variable indicates which mode is to be used; *portrait* is when the typeset material is positioned with the column measure running across the 8½ inch width, and *landscape* is when the column measure is running across the 11 or 14 inch width. The second variable indicates the maximum size of the output paper being used. In *portrait* mode, this would be the maximum depth of the page, in *landscape* mode, this would be the maximum width of the page.
Special Conditions	This code is only valid with PostScript output modules such as LCOMP for the LaserWriter. The code must appear as the first code in the job/take. If the code does not appear, default values will be recognized. If landscape is indicated, the second variable will determine the maximum column width of the copy.

PASS THROUGH COMMAND

Code	<PTn>
Variables	n = contents of a variable instruction
Alt Key	n/a
Cancel Code	n/a
Explanation	The command is used to allow a variable instruction in typesetter code to pass through the *PREPARATION FOR OUTPUT* process directly to the output device.
Special Conditions	Only the contents of the variable instruction is necessary, the *open command* and the *close command* of the appropriate device surrounding the instruction is automatic. For cora code, *bell coding* cannot be used in the <PT command

POINT MAXIMUM

Code	<PXn>
Variables	n = a number from 1 to 6
Alt Key	n/a
Cancel Code	<XM>, another <PXn> same n, or end of take
Explanation	The code is used to mark a maximum depth for copy under a <PMx> where x = n which may later be returned to with a <RXn>.
Special Conditions	Code may be used to manually mark or update an existing <PXn> without changing the current <PM. Point maximum is automatically evaluated and updated for the current point mark whenever another point mark is established, or an <RM or <RX is requested.

QUAD LEFT
QUAD RIGHT
QUAD CENTER

Code	<QL>, <QR>, and <QC>
Variables	n/a
Alt Key	L, R, and C
Cancel Code	n/a
Explanation	The quad codes are used to determine the horizontal position of a short line or last line of a text paragraph using optimum spaceband values. <QL> quad left indicated the copy is to set flush against the left margin. <QR> quad right indicates the copy is to set flush against the right margin. <QC> quad center indicates the copy is to set centered with the space equally divided to the left and right margins.
Special Conditions	When copy is set in ragged mode, the quad copy is normally required to be the same. Using an <EP> in place of a quad code will assume the appropriate quad type. In tabular material, quad codes advance to the next column.

QUAD MIDDLE

Code	<QM>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The quad middle code is used to expand a short line of copy to full measure by pushing the copy before the <QM> to the left margin and the copy after to the right margin. Spacebands assume optimum value.
Example	Keyboard: A<QM><QM>B<QM>C<QL> will expand ABC to full measure with twice the space between A and B than B and C.
Special Conditions	All <QM> codes in a single line will be of equal width value (escapement). <QM> codes may be used in multiples to create multiple amounts of space between amounts of set copy.

QUIT TAB

Code	<QT>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The <QT> quit tabular code is used to end a tab line and return text to normal full measure starting at the left margin.
Special Conditions	The <MA> code automatically creates the effect of a <QT> and begins a new tab line.

RAGGED LEFT
RAGGED RIGHT
RAGGED CENTER

Code	<RL>, <RR>, and <RC>
Variables	n/a
Alt Key	n/a
Cancel Code	<XR> or another rag command
Explanation	The ragged codes are used to override the default justify mode and instruct the set of ragged copy. <RL> ragged left indicates the text lines will all set flush left and terminate at different horizontal points. <RR> ragged right indicates the text lines will all set flush right and begin at different horizontal points. <RC> ragged center indicates the text lines will begin and terminate at different horizontal points from line to line with each line being indented from the left and right margins equally.
Special Conditions	The different horizontal points of begin and terminate will be determined by the <CRx,y> code in effect or default.

REUSE FORMAT

Code	<RF>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The <RF> code is used in the text stream if the operator wishes to reuse the last called format.
Special Conditions	The <RF> code may be used within a format to loop a repeating set of commands or text string, but care must be taken that there are exits from the format prior to the <RF> code such as <MC> merge copy, <DLx> delay lines, or <DMx> delay measure. A <SFx>*commands*<RF> <EF> string requested during H&J will yield a FORMAT error.

RETURN TO INDENT

Code	<RIx>
Variables	x = a number from 1 to 20
Alt Key	n/a
Cancel Code	<EP> or end of take
Explanation	Used after establishing a <IXn> where n = x of <RIx>. All lines following will indent to the horizontal point established until an <EP> is encountered.

RETURN TO MARK

Code	<RMx>
Variables	x = a number from 1 to 6
Alt Key	n/a
Cancel Code	<XM>
Explanation	The code is used to immediately return to the vertical depth of the <PMx> where the number x is the same. The current depth is compared to any existing maximum depth for the current point mark x and the greater is retained. That maximum depth can be returned to with an <RXn> code where n = x. The depth is also compared to an overall maximum depth and the greater is retained. That maximum depth can be returned to with an <XM> command.
Special Conditions	If an <RMx> code is keyed after a <XM>, an error message will indicate that <PMx> doesn't exist and must be reinstated.

REVERSE LEADING

Code	<RVx>
Variables	x = points and fractions
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to indicate an immediate shift up of the baseline.
Special Conditions	The <RVx> will be subtracted from the current <CLx> in effect if no typeset characters have yet appeared on the line. An <RVx> may be used to return to a baseline adjusted by a prior <ELx> of same x value if both appear before the next quad code.

RETURN TO MAXIMUM

Code	<RXn>
Variables	n = a number from 1 to 6
Alt Key	n/a
Cancel Code	<XM>
Explanation	The code is used to return to the maximum depth for n that has been retained from all <RMn> codes, provided that depth has not yet been exceeded.

START SMALL CAPS

Code	<SC>
Variables	n/a
Alt Key	n/a
Cancel Code	<XC>
Explanation	Instructs the program to set the following lowercase alpha characters as CAPS reduced in size and width as specified in the system standards-values.

SUPPLEMENTAL DICTIONARY

Code	<SDx>
Variables	x = a number from 1 to 99
Alt Key	n/a
Cancel Code	<XD> or another <SDx> or end of take
Explanation	The code is used to call in a supplemental dictionary that has been established in the Hyphenation Dictionaries.
Special Conditions	When a supplemental dictionary is in effect, it is checked first for hyphenation breaks before the program goes to the regular dictionary.

STORE FORMAT

Code	<SFx>
Variables	x = a number from 1 to 500
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to begin a string of command codes and/or text that will be stored and used when a <UFx> code is entered in a job/take.
Special Conditions	Formats may be stored at the beginning of a job/take and used anywhere within the job/take. Formats may be stored in take F of a job number and used in any take of the same job number. Requests to store a format number greater than 500 will yield a FORMAT error.

SKEW LEFT

Code	<SL$n_1/m_1,...,n_{20}/m_{20}$>
Variables	n = number of lines to skew m = indent in picas and points
Alt Key	n/a
Cancel Code	completion of line or end of take
Explanation	The code is used to produce a graduated left indent over a number of text lines.
Special Conditions	Up to 20 pairs of values n/m may be included in one code string. A hyphen before the number of lines indicates a skew toward the left margin from the indent specified. The skew indent takes effect immediately, so if a full line of text is required, use a pair consisting of 1/0, or if a full indent line is required before beginning a negative skew, use a pair consisting of 1/x where x = full indent amount.

SUPERFONT DEFAULT

Code	<SPx>
Variables	x = a pointsize between 20 and 72
Alt Key	n/a
Cancel Code	another <SPx> command
Explanation	The code is used to change the superfont default pointsize specified in the *TYPESETTER REQUIREMENTS* menu.
Special Conditions	The command code is used to instruct the L/202 to set superfonts at or above the specified default.

SKEW RIGHT

Code	<SR$n_1/m_1,...,n_{20}/m_{20}$>
Variables	n = number of lines to skew m = indent in picas and points
Alt Key	n/a
Cancel Code	completion of lines or end of take
Explanation	The code is used to produce a graduated right indent over a number of text lines.
Special Conditions	Up to 20 pairs of values n/m may be included in one code string. A hyphen before the number of lines indicates a skew toward the right margin from the indent specified. The skew indent takes effect immediately, so if a full line of text is required, use a pair consisting of 1/0, or if a full indent line is required before beginning a negative skew, use a pair consisting of 1/x where x = full indent amount.

SET SUPERIOR CHARACTER

Code	<SU>
Variables	n/a
Alt Key	n/a
Cancel Code	after one character
Explanation	The code indicates that the following character is to be set reduced in size and width as specified in the system standards-values and top aligned with the cap height of the current point size.
Special Conditions	The program will convert the <SU> code to display an *UP ARROW* on the screen. The *SHIFT/TAB* key will also generate an *UP ARROW* superior code.

SET VERTICAL RULE

Code	<SVx,y,z>
Variables	x = number 0 through 40 y = thickness of rule z = offset to left
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to instruct the setting of a vertical rule from the start point defined by <DVx> to the baseline of the <SV code.
Special Conditions	The thickness of the rule and the amount of left offset are dependent on the output device. Request to set vertical rule that has no <DVx> code will yield VRULE error.

SET WHITE SPACE

Code	<SWx>
Variables	x = a number of relative units
Alt Key	n/a
Cancel Code	<SW0>
Explanation	The code is used to adjust the overall tracking of the typeset material. The values expressed are reductions to each character width. A hyphen is used to indicate a value added to each character width.
Special Conditions	The <SWx> value will add or subtract from the tracking value in effect.

TRACK SELECTION

Code	<Tx>
Variables	x = a number 0 through 3
Alt Key	n/a
Cancel Code	another <Tx> code
Explanation	The track selection code is used to override the track specified in the system default-values.
Special Conditions	Tracking is an overall addition or subtraction of letterspace value in addition to any kerning, letterspacing, or set white space in effect. <T0> indicates NO tracking.

TAB STUB

Code	<TBp[,Gm/r],n[,Gm/r]>
Variables	m = gutter width in picas and points r = vertical rule weight p = column width in picas and points n = number of columns Gm/r is optional, do not key brackets[ ]
Alt Key	n/a
Cancel Code	<XT>
Explanation	The tab stub command is used to set up a specified width (stub) column with the balance of the measure divided equally among a number of additional columns. Gutters and vertical rules may be set using the Gm/r variables.
Special Conditions	Vertical rules specified by weight will be centered within the gutter width. The Gm/r variables indicate the width of the gutter that will be set between the columns. For vertical rules on the outside margins, use the <DV and <SV codes. Rule weight is dependent on the output device.

THIN SPACE

Code	<TH>
Variables	n/a
Alt Key	T
Cancel Code	n/a
Explanation	A fixed space in relative units as specified in the *FONT INFORMATION TABLE* of the current font.
Special Conditions	Unlike a spaceband in justified copy, the <TH> is *not* variable to facilitate H&J. The width value of the <TH> will key off the current <CWx> in effect.

TAB NUMBER

Code	<TNn[,Gm/r]>
Variables	m = gutter width in picas and points r = vertical rule weight n = number of columns 1 to 40 Gm/r is optional, do not key brackets [ ]
Alt Key	n/a
Cancel Code	<XT>
Explanation	The tab number command is used to set up a number of equal width columns across the current measure. Gutters and vertical rules may be set using the Gm/r variables.
Special Conditions	Vertical rules specified by weight will be centered within the gutter width. The Gm/r variable indicates the width of current and all subsequent gutters between each pair of tabular columns. For vertical rules on the outside margins, use the <DV and <SV codes. Rule weight is dependent of the output device.

TAB PROPORTIONAL

Code	<TPx_1,[Gm/r,]x_2,[Gm/r,]...x_{40}>
Variables	m = gutter width in picas and points r = vertical rule weight x = proportional units Gm/r is optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	<XT>
Explanation	The tab proportional command is used to set up a number of columns where the widths will be multiples of a proportional unit. Gutters and vertical rules may be set using the Gm/r variables.
Example	The most narrow column would be x = 1. Column x = 2 would be twice the width, x = 3 would be three times the width, etc. To set up a table where the second column is twice the width of the first, and the third column is twice the width of the second, keyboard as: <TP1,2,4>
Special Conditions	Vertical rules specified by weight will be centered within the gutter width. The Gm/r variables indicate the width of the current gutter and all subsequent gutters, unless otherwise specified, that will fall between the columns as indicated. For vertical rules on the outside margins, use the <DV and <SV codes. Rule weight is dependent on the output device.

SET TABULAR RULE

Code	<TRx,y>
Variables	x = rule weight y = override leading value in points and fractions
Alt Key	n/a
Cancel Code	any typeset character or end of line
Explanation	The code is used to indicate the setting of a horizontal rule within a tab column (or columns if <JTx> is used) that will extend halfway into the surrounding gutters and meet the vertical rules if they exist.
Special Conditions	If x is specified, it will override the default rule weight. If y is specified, it will override the current leading in effect. Within tabs, it must be the only command within the column (or columns) and immediately followed by a quad code. Outside of tabs, it operates as a <WR and it will draw all existing vertical rules within tab gutters down to meet the horizontal rule.

TAB SET

Code	<TS[Gm/r,]p_1,...p_{40}>
Variables	m = gutter width in picas and points r = vertical rule weight p = tab indent from left in picas and points Gm/r is optional, do not set brackets [ ]
Alt Key	n/a
Cancel Code	<XT>
Explanation	The tab set command is used to set up columns that start at various points across the measure with or without gutter specifications. Gutters and vertical rules may be set using the Gm/r variables.
Special Conditions	Vertical rules specified by weight will be centered within the gutter width. The Gm/r variables indicate the width of the gutter that will fall prior to the following tab set points. For vertical rules on the outside margins, use the <DV and <SV codes. Rule weight is dependent on the output device.

TAB TEXT

Code	<TT[Gm/r]>
Variables	m = gutter width in picas and points r = vertical rule weight Gm/r is optional, do not key brackets []
Alt Key	n/a
Cancel Code	<XT>
Explanation	The tab text code is used to mark a horizontal point on the measure based on the space occupied by typeset characters preceeding the <TT. If the gutter specifier is used, the gutter will precede the horizontal point marked by the <TT, and if a rule weight is specified, a vertical rule will be centered for every tab line of text starting with an <MA> code.
Special Conditions	Rule weight is dependent on the output device.

USE FORMAT

Code	<UFx>
Variables	x = a number from 1 to 500
Alt Key	n/a
Cancel Code	n/a
Explanation	Used in the text stream to request the commands and/or text contained in format of number x.
Special Conditions	Request for format x that has not been established will yield a FORMAT error.

USE GLOBAL FORMAT

Code	<UGx>
Variables	x = global format number 1 to 1000
Alt Key	n/a
Cancel Code	n/a
Explanation	Used in the text stream to request the commands and/ or text contained in global format of number x.
Special Conditions	Request for format x that has not been established will yield a FORMAT error.

UNIT SPACE

Code	<UN>
Variables	n/a
Alt Key	U
Cancel Code	n/a
Explanation	A fixed space in relative units as specified in the *FONT INFORMATION TABLE* of the current font.
Special Conditions	Unlike a spaceband in justified copy, the <UN> is NOT variable to facilitate H&J. The width value of the <UN> will key off the current <CWx> in effect.

SET UNDERSCORE

Code	<USx,y>
Variables	x = points below baseline (or above baseline indicated by a hyphen) y = weight of underscore rule
Alt Key	n/a
Cancel Code	<US>
Explanation	The code is used to turn of the underscore condition until a <US> is encountered. Rule weight is dependent on the output device. If points below the baseline areommitted, the program will position the underscore a number of points below the baseline equal to one third the current pointsize.

SET REVERSE VIDEO

Code	<VRx,y>
Variables	x = number of points and fractions above the baseline y = number of points and fractions below the baseline
Alt Key	n/a
Cancel Code	<VR0>
Explanation	The code is used to instruct the setting of reversed type. The feature is subject to the limitations of the output device.

SET LEADER DOTS

Code	<WLx>
Variables	x = relative units between dots
Alt Key	n/a
Cancel Code	any typeset character or end of line
Explanation	The code is used to indicate a string of periods (full stops) to be set in the horizontal space left over after justification between the typeset characters where the code appears. In effect, the code is much like a <QM> and may be used in multiples to create one row of leader dots proportionally longer than another row on the same line. If no value x is specified, the program will default to the value of the en leader for the current font.

SET HORIZONTAL RULE

Code	<WRx,y>
Variables	x = rule weight y = override leading value in points and fractions
Alt Key	n/a
Cancel Code	any typeset character or end of line
Explanation	The code is used to indicate the setting of a horizontal rule between margins, between characters, or between characters and the margin. The effect on a line is like a <QM> and may be used to create more than one rule on a line.
Special Conditions	If x is specified, it will override the default rule weight. If y is specified, it will override the current leading in effect. If more than one <WR command appears on any line with different override leading values, the last override leading value will take effect. If the text line contains a <ZL> command, the y value and the current leading will be ignored.

SET LEADER CHARACTER

Code	<WXn,m>
Variables	n = typeset character m = relative units between characters
Alt Key	n/a
Cancel Code	any typeset character or end of line
Explanation	The code is used to produce a row of the same character repeated with a specific amount of space between each.
Special Conditions	The effect on the line is like a <QM> and may be used to create more than one set of leadered characters on a line. The typeset character n may be an alpha/numeric, a pi character, or a $XX pi character.

CANCEL FRACTIONS

Code	<XB>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <BF> code and allow the text entry of numbers-slash-numbers without converting them to a fraction.

CANCEL SMALL CAPS

Code	<XC>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <SC> code and allow the lower case letters without converting them to small caps.

CANCEL SUPPLEMENTAL DICTIONARY

Code	<XD>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to suspend the access to a secondary dictionary requested previously by an <SDx> command. The job/take justification will be returned to the control of the standard dictionary.

CANCEL NO FLASH

Code	<XF>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <NF> code, allowing the continued printing of typeset material.

CANCEL LIGATURES

Code	<XG>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <AG> command or the system defaults-mode of allowing ligatures.

CANCEL HYPHENATION

Code	<XH>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <AH> command of the system defaults-mode of allowing hyphenation.

CANCEL INDENT HANG

Code	<XI>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the effect of the <IHx,y> on the following line.

CANCEL KERNING

Code	<XK>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is use to cancel the <AK> command or the system defaults-mode that allows kerning.

CANCEL LETTERSPACING

Code	<XL>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <AL> command or the system defaults-mode of allowing letterspacing.

CANCEL POINT MARKS

Code	<XM>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to return to the overall maximum point remembered by all <PMx> commands are cleared.
Special Conditions	The code takes effect on the current line. An <RMx> code used after a <XM> will generate a POINTMARK error to indicate <PMx> must be reinstated.

CANCEL KERN NUMERIC ONE

Code	<XO>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <KO> command or the system defaults-mode that allows the kerning of numeric one.

CANCEL HANGING PUNCTUATION

Code	<XP>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the <HP> command.

CANCEL RAGGED

Code	<XR>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel the effect of the <RC>, <RL>, or<RR> commands and restores the H&J program back to setting justified lines.

CANCEL ENHANCED SCALING

Code	<XS>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel a request for enhanced scaling <ES> on the APS output device.

CANCEL TAB

Code	<XT>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to cancel a tab setup. An <MA> command issued after the <XT> would have no effect.

CANCEL WIDOW HYPHENATION

Code	<XW>
Variables	n/a
Alt Key	n/a
Cancel Code	n/a
Explanation	The code is used to inhibit the hyphenation of the last word in a paragraph. It cancels the <AW> code or the system default-mode that would normally allow widow hyphenation.

SELECT OUTPUT

Code <XX>

Variables n/a

Alt Key n/a

Cancel Code <EP>

Explanation The code is used to indicate that a current paragraph be prepared for output when *SELECTIVE COMPOSE* is requested. During the *preparation for output* process, the take is scanned for paragraphs containing <XX>s and flagged for composition. Only one <XX> is required within each paragraph to be output selectively. Multiple select output paragraphs will have space between them equal to the inter galley leading indicated in the *SYSTEM STANDARDS*.

Example For example, to selectively output only the second paragraph in the following text, an <XX> must appear somewhere in the second paragraph between the prior <EP> and the following <EP>.

This is the first paragraph in the job take. This is the first paragraph in the job take. This is the first paragraph in the job take. <EP>
<XX>This is the second paragraph in the job take. And this is a line in the second paragraph in the job take. This is the second paragraph in the job take. <EP>
This is the third paragraph in the job take. This is the third paragraph in the job take. This is the third paragraph in the job take. <EP>

X-Y COORDINATES

Code	<XYx,y>
Variables	x = indent from left in picas and points y = extra lead amount in picas and points
Alt Key	n/a
Cancel Code	n/a
Explanation	The X-Y coordinates are used to move blocks of copy to specific start points within a job/take. The top left corner of a job/take is <XY0,0>. The x value in the code effectively moves the horizontal start point of all indents that normally measure from the left margin. The <CCx> code also measures from the x value.
Special Conditions	The X-Y coordinate code establishes a start point for text from which the measure and leading is started. Example: <CP10> <CL12> <CC20> <XY6,12> would position the first line of copy starting at 6 picas from the left margin, ending at 26 picas from the left margin, and on a baseline 13 picas from the top of the job/take. The maximum y coordinate is 132, which is equivalent to 22 inches. In order to alert the operator of conditions where the <XY command exceeds the maximum measure of the output device, the 'x' coordinate is added to the current column measure and the total compared to the maximum allowable measure in the *STANDARDS AND DEFAULTS*. A COLUMN ERROR is flagged if this total exceeds the maximum.

ZERO LEADING

Code	<ZL>
Variables	n/a
Alt Key	n/a
Cancel Code	end of line
Explanation	The code is used to suspend the current leading in effect for the current line only.
Special Conditions	The <ZL> command may only be used immediately following a quad code.

Chapter 4
PRIMARY MENU FEATURES

Space Available 4-2
Create a New Job 4-3
Exit a New Job 4-5
Select an Existing Job 4-7
Exit an Existing Job 4-9
Output to Typesetter 4-11
Transfer Jobs 4-13
Display & Clear Log 4-15

PRIMARY MENU FEATURES

The following is a detailed discussion of the other four options and the *SPACE AVAILABLE* information of the Primary Selection menu that follows the *OPERATOR SIGN ON*. Option 4 was discussed in chapter 1 and option 6 is used to close out the program and return to *DOS*.

DISPLAY THE JOB DIRECTORIES and *SELECT AN EXISTING JOB* are options leading to the performance of functions on jobs that already exist in the system. *CREATE A NEW JOB* allows the operator to open a new job/take for input. *TRANSFER JOBS* allows the operator to export and import selected job/takes to and from floppy disks for archive and transport as well as importing composed job/takes from other MagnaType stations.

As with every menu-driven software program, the tasks available to the user are displayed in a series of screens from which choices are made. After selecting a screen option, the computer will display the next logical collection of choices for the user to proceed.

The general guide to using the MagnaType Software program is that RETURN (or ENTER) will move the operator forward through the menu selection screens, and ESCape will move back to previous selection screens.

Now, lets go to the computer and examine several possibilities from the main menu screen that appears after booting the program and entering operator initials (see *INSTALLATION SECTION* and *DISKETTE* provided with this manual).

SPACE AVAILABLE

Before this information in the lower left corner of the Primary Selection Menu may be of any use, you must first understand how MagnaType maintains job storage. In order to realize economies in hard disk storage space, MagnaType holds all job data in one *DOS file (named JOB.FIL)* rather than creating a separate *DOS file* for each job. As new jobs are created or imported into the MagnaType system, this JOB.FIL is expanded to store this new data by taking blocks of available free sectors *(512 character bytes per sector)* from the unused portion of the hard disk using *DOS*. The amount of free sector space available from *DOS* is indicated by the byte count displayed in the lower left portion of the Primary Selection Menu.

For example, let's assume after loading the MagnaType programs and utility files on a ten megabyte hard disk, you are left with 9,216,000 bytes of storage space. The Primary Selection Menu would display *SPACE AVAILABLE* as 9,216,000 for DOS and zero for JOB. Then assume a job/take called TEST is created and stored into one sector. The space available count for DOS will drop (as a block of free sectors is assigned to the JOB.FIL) and the count for JOB will remain at zero.

Now assume it is some time later, many job/takes have been created or imported, and a display of *SPACE AVAILABLE* shows DOS with 6,144,000 and JOB with zero. The job/take called TEST is now killed from the system, the sector it occupies is released to the job storage area *(JOB.FIL)* and a display of *SPACE AVAILABLE* will show DOS still with 6,144,000 and JOB with 512. Now a new job/take is created that requires 1000 bytes of storage space. The 512 bytes freed by killing TEST will be used to store the first portion of this new job and another block of sectors will be pulled from DOS into JOB.FIL to store the rest of this new job. Now a display of *AVAILABLE SPACE* will show a drop for DOS equivalent to the size of the assigned block of sectors* and JOB again at zero.

One way to visualize this situation is to imagine the JOB.FIL as the boundaries to a *city* within a *county* that is the free space on the hard disk. As the city grows because new residents move in (creating new job/takes), it will expand by taking land from the county. But when residents move away (killing job/takes) and space is left unoccupied, the county doesn't reclaim the space, but rather allows the city to offer it up to any new residents that may move in (creating more job/takes).

*This block of sectors will vary depending on the size of the disk drive. A 10 megabyte disk will take 8 sector blocks (4096 characters) and a 20 megabyte disk will take 16 sector blocks (8192 characters). This is a function of DOS.

CREATING A JOB

The first two selections of the primary menu allows the user access to all the job/takes that exist in the system. However, before jobs can be called up for proofing and editing, they must first be created. Selection 3, *CREATE A NEW JOB*, is used for this purpose.

TYPE: 3 RETURN

TYPE IN THE JOB NUMBER:

Here the operator will enter a job number consisting of up to 8 alpha/numeric characters followed by a hyphen followed by a take number from 1 to 999. If no hyphen and take number is entered, the program will automatically assign a take number 0.

TYPE: (*job number, hyphen, take*) RETURN

TYPE IN THE JOB NAME:

The program allows the operator to give the job/take a descriptive name of up to 16 alpha/numeric keystrokes. This name appears next to the job/take number when the *JOB DIRECTORY* list is displayed for easy reference.

TYPE: (*job name*) RETURN

The program will now show a job header at the top of the screen. Note the information contained in the job header which remains on the screen during text entry and editing. The job number and take number along with the job name is displayed as a reminder to the operator of which job is being worked on. The day, month, and year that the job was first entered is also displayed. The day, month, year and time of last revision is displayed along with the initials of the operator. The character count is the size of the job in bytes (characters), which is updated whenever the H&J (F1) is requested. The cursor is positioned on the first text data line of the new job and the program is now ready to accept the entry of keystrokes (see following sections on *INPUT AND EDITING FEATURES* and *APPLICATIONS EXAMPLES* for details on text data entry).

Along with text entry keystrokes, the user has a full complement of typesetting command codes to produce any style of typeset copy required (see *COMMAND CODES* section for detailed information). However, the only codes necessary to produce type are the four basic parameters: Pointsize, Leading, Column Width (or Measure), and Font.

Now that you have a job/take opened, enter the four basic parameters. Open command, <, is the *PLUS KEY* on the number pad and close command, >, is the *MINUS KEY* on the number pad. TYPE: <CP10> <CL12> <CC25> <CFHR>

This indicates to the program a change pointsize of 10, a change leading of 12 points, a column measure of 25, and a change font of HR (for Helvetica, provided already setup in the *FONT INFORMATION TABLES*.

Now type a few paragraphs of text using the ALT/P to indicate the end of paragraph.

After completing a few paragraphs, type F1. This will instruct the program to perform the H&J (hyphenation and justification) process.

Several things are now happening very rapidly in the memory of the computer. The system is considering the width values of every character, adding them up until the total nears the end of the column measure. As this happens, spacebands must be expanded or contracting in order to end the line precisely at 25 picas. However, if by expanding or contracting spacebands to the limits specified in the *SYSTEM DEFAULT-VALUES*, the program discovers it must break a word, it looks to the dictionary logic program for possible hyphenation points. After correctly constructing the justified line, the system moves on to consider the text data of the next line. When the entire job has been put through this process, you will see the depth indicator in the top right of the job header display the total depth of the job. You will also note that each line end is detailed with some numbers. These numbers indicate the length of the line of type as it would be when sent to the output device, as well as the depth from the top of the job of each baseline. With 12 pt. lead (1 pica), you will see each line numbered 1,2,3,... etc. Each line length except the last one in the paragraph will show a number equivalent to the column measure, 25 picas.

Now change the <CC25> to <CC15> and type F1 to re-H&J. Note how the amount of text on each line has decreased and the line endings show the new measure 15 at the end of each line except the last in the paragraphs. You will also note that the depth indicator at the top right has increased because more lines are required to contain the same amount of type.

Now type INSERT located at the bottom right of the keypad under the cursor control keys (note the cursor is at the top left corner of the job). Type <IT2,2>, this is to indicate an indent of 2 picas from the left and right margin.

Type F1 and note how the copy has changed. Line endings show each line beginning 2 picas indented and ending at 13 (2 picas from 15).

Spend some time changing some of the other parameters like leading and pointsize and use F1 to see the calculated results change. When you feel comfortable with keyboarding and interpreting the line ending values, go to the *COMMAND CODE* section of this manual and experiment with some indent and leading commands on the H&J process.

TO EXIT A NEW JOB

TYPE: ESCape

The *EDITING COMPLETE* menu will appear on the screen with the cursor positioned next to the first option. This menu also appears when the operator exits (ESCape) an existing job/take that was called up for proofing and editing.

CHANGE THE JOB NO.:

Here, the job number and take number of the new job could be changed before storing it to disk.

TYPE: (*new number*) RETURN

Or just move the cursor to the next option line with the cursor down key.

CHANGE THE JOB NAME:

A new name could be entered for this job/take before storing it to disk.

TYPE: (*new name*) RETURN
or move the cursor to the next option line with the cursor down key.

SAVE THE JOB/SEND TO PRINTER:

The job/take could be sent directly to the line printer program for output to a line printer while automatically stored away in the *JOB DIRECTORY* for future reference.

TYPE: RETURN

A window will appear at the right of the screen to display options for line printing.

1. TEXT AND COMMANDS
2. TEXT AND COMMANDS ♦
3. TEXT ONLY

IF EXPEDITE, ENTER "Y"

If no line printer output is required, move the cursor to the next option line with the cursor down key.

SAVE THE JOB/PREPARE FOR OUTPUT:

The job/take could be sent directly to the compose program for typesetter output while automatically stored away in the *JOB DIRECTORY* for future reference.

TYPE: RETURN

A window will appear at the right of the screen to display options for output to the typesetter (see <XX> in chapter 3).

1. COMPOSE
2. COMPOSE/EXPEDITE
3. SELECT COMPOSE
4. SELECTIVE EXPEDITE

If no typesetter output is required, move the cursor to the next option line with the cursor down key.

SAVE THE JOB:

The job/take may now be stored away in the *JOB DIRECTORY* on the disk for future reference.

TYPE:RETURN

Now the primary menu will reappear on the screen. If the operator chooses not to store the job/take, the cursor may be moved to the next option line with the cursor down key.

RETURN TO JOB:

The job may be recalled to the screen with the cursor positioned in the job at the same place it occupied when ESCape was typed for the *EDITING COMPLETE* menu.

TYPE: RETURN

Or move the cursor to the last option line with the cursor down key.

IGNORE ALL CHANGES:

TYPE: RETURN
The main menu will reappear and the new job will no longer exist.

SELECT AN EXISTING JOB

To bring up an existing job/take to the screen for proofing and/or editing, one of the first two selections of the main menu may be used.

TYPE: 1 RETURN

JOBS ON DISK FOR EDITING
JOBS IN PREPARATION FOR OUTPUT
JOBS READY TO BE OUTPUT
JOBS THAT REQUIRE ATTENTION
DISPLAY LOG

The cursor may be positioned on line with one of the five selections, by typing RETURN, the contents of each list will be displayed. *JOBS ON DISK FOR EDITING* will display a complete list of all jobs in the system in alphabetic and numeric order for easy reference. To access the *JOB LIST*, type RETURN. You will see a list of job numbers followed by its name and the number of takes or extensions associated with it. Also available on the screen to the right are a number of options. *L* allows the operation of the line printer, *C* allows the jobs to be prepared for the output device with a slugline, *F* allows the jobs to be prepared for output without a slugline, *N* allows the job number to be renamed, and @ allows the directory listing of all the takes of the job to the line printer.

To access a list of every JOB/TAKE, type * RETURN

To access only jobs beginning with a specific letter or number, type that character followed by RETURN and only the list of those jobs will be displayed. Move the cursor to the line with the job to be displayed and type RETURN. All the takes of that job number will list to the screen. Now the operator may move the cursor next to a specific take of the job and select any option listed to the right by typing the appropriate letter followed by RETURN. *H* or *P* will bring the job data to the screen in H&J form or non-H&J form, *L* will list the take to the line printer, *C* will prepare the take for the output device with a slugline, *F* will prepare the take for output without a slugline, and *K* will allow the operator to kill a job.

The second method for selecting a job for proof or edit is also from the primary menu.

TYPE: 2 RETURN

TYPE IN JOB NO.:

Type in the job number followed by a hyphen followed by the take number, then type RETURN.

SELECT OPTION:

Type one of the letters for a function described in the list to the right of the screen.

TYPE: H RETURN
This will bring the job to the screen in H&J form showing line breaks as they would be in output.

TYPE: P RETURN
This will bring the job to the screen in non-H&J, or raw form for easy proofing or editing.

TYPE: L RETURN
This will send the job/take to the line printer in one of three forms selected from a menu window.

TYPE: C RETURN
This will send the job/take to prepare for output to the typesetter complete with a header slugline. This slugline contains much of the job header information including the job number and name for easy identification.

TYPE: F RETURN
This will send a job to be prepared for output without a header slugline for final output such as on the laser printer where a slugline is not required.

TYPE: K RETURN (and a security keystroke)
This will remove the job/take from the system.

TYPE: D RETURN
This will allow the job/take to be duplicated with the ability to key in a new job/take number.

TYPE: R RETURN
This will allow the operator to replace the current job/take number.

TYPE: J RETURN
This will allow the operator to join any number of other job/takes to the selected job/take. Type the job numbers to be added following each with a return to assemble a list. Typing an * (asterisk) RETURN will execute the join feature.

Obviously, there are may ways to find and bring a job/take to the screen for proofing and editing or performing other functions on it. The best method depends on the specific needs of the user.

TO EXIT AN EXISTING JOB

TYPE: ESCape

The *EDITING COMPLETE* menu will appear on the screen with the cursor positioned next to the first option. The same possibilities exist as when the operator exits a new job with the following exceptions:

CHANGE THE JOB NO.:

Type a new number for storage of edited version, while the original remains in its previous form under its previous job/take number. Or move cursor to next line.

CHANGE THE JOB NAME.:

Type a new job name. If a revised number was entered above, the new name goes with it leaving the old name on the old job. If no revised number was entered, a new name will replace the old name of the existing job. Or move cursor to next line.

SAVE THE JOB/SEND TO PRINTER:

The job/take could be sent directly to the line printer program for output to a line printer while automatically stored away in the *JOB DIRECTORY* for future reference.

TYPE: RETURN
A window will appear at the right of the screen to display options for line printing.

1. TEXT AND COMMANDS
2. TEXT AND COMMANDS ♦
3. TEXT ONLY

IF EXPEDITE, ENTER "Y"

If no line printer output is required, move the cursor to the next option line with the cursor down key.

SAVE THE JOB/PREPARE FOR OUTPUT:

The job/take could be sent directly to the compose program for typesetter output while automatically stored away in the job directory for future reference.

TYPE: RETURN
A window will appear at the right of the screen to display options for output to the typesetter (see <XX> in chapter 3).

1. COMPOSE
2. COMPOSE/EXPEDITE
3. SELECT COMPOSE
4. SELECTIVE EXPEDITE

If no typesetter output is required, move the cursor to the next option line with the cursor down key.

SAVE THE JOB:

If revised job number and/or revised job name is required, RETURN must be typed to STORE in the job directory. Or move cursor to next line.

RETURN TO JOB:

Type RETURN and the job will reappear on screen with the cursor at the position in the job before ESCape was typed to bring up *EDITING COMPLETE.* Or move cursor to last line.

IGNORE ALL CHANGES:

Type RETURN and the screen will display the primary system menu. All keystrokes entered will be ignored including any entered on the *EDITING COMPLETE* menu.

All the job/takes available for editing on the MagnaType system are stored in a single DOS file called JOB.FIL. The key to the location of each job/take within JOB.FIL is another DOS file called DIRECTRY.FIL and it is used to build the directory when the program is booted. These files may be copied for backup, however, it is important that they be kept together. One is worthless without the other. Everytime a job is pulled to the screen for editing and then stored, the JOB.FIL changes and likewise, the DIRECTRY.FIL keeps record of this change. *See the INITL PROGRAM in the INSTALLATION SECTION.*

OUTPUT

Before a job/take can be sent to the output typesetting device, it must first be *prepared for output.* This process directs the job/take through H&J and adds to this the appropriate coding for the output device. This becomes only a temporary file and the job/take itself remains in its most recent version in the job directory.

Job/takes may be sent to *PREPARATION FOR OUTPUT* from the first two selections of the *PRIMARY* menu or from the *EDITING COMPLETE* menu upon exiting a job/take. Since only one job/take can be prepared for output at any given time, they may be queued up in the order that they are to be prepared for output. This queue or *list* may be viewed by the operator from selection 1 of the *PRIMARY* menu. Move the cursor to the second line, *JOBS IN PREPARATION FOR OUTPUT,* and type RETURN. This will display all the job/takes that have been requested for preparation to output. The first job/take at the top of the list is actually the one currently being processed for the output device.

As these job/takes are completed, they will appear in order and may be viewed by moving the cursor to the third line of selection 1 from the *PRIMARY* menu. A window will appear with options to view only the job/takes not yet sent to the output device, all job/takes ready to go to the output device, or allow the operator to kill the temporary output job/takes from the list that have already been sent to the output device.

After selecting one of the first two options, a list of job/takes ready for the output device will appear and another menu of options will display in the upper right corner of the screen. Move the cursor to the job/take required and type the key initial of the option requested.

> *T* followed by RETURN indicates that the job/take will be directed to the output device as soon as the operator types ESCape. Several job/takes from the list may be requested at one time before initiating the request with ESCape.
>
> *L* followed by RETURN and ESCape will send a command code printout to the line printer displaying the job/take in the language of the output device for diagnostic purposes.

K will remove the *ready for output* version of the job/take from this list after a RETURN, but as with the *KILL JOBS TYPESET* option above, the most recent version of this job/take will remain in the job directory.

X will cancel any of the above requested options if typed before RETURN.

V will allow the operator to view the slug line information of the job to get information prior to output such as *job depth* or required font disk numbers.

Additionally, when output devices capable of accepting an instruction to output multiple copies of a job/take (such as the LaserWriter) are connected to the MagnaType computer, the option of typing in a number from 2 to 9 is allowed as an alternative to *T* for typeset.

If job/takes are requested to be sent to the output typesetting device and the MagnaType computer is not connected to a typesetter *(established in selection 5 of SYSTEM STANDARDS AND DEFAULTS),* a window menu will appear requesting the operator to indicate which disk drive has a formatted floppy disk ready to receive the transmission of the *JOBS READY TO BE OUTPUT.* This floppy disk may then be physically transported to another MagnaType computer station for import and transmission to the output device. *See following section on TRANSFER JOBS.*

Remember, once a job/take has been prepared and ready for output, it may be sent to the output device any number of times until the *WORK FILE* is reinitialized. This may be accomplished by exiting to *DOS* and calling up the initialize program. With the C> prompt, type INITL followed by RETURN. The first selection will clear the *JOBS READY FOR OUTPUT* list. It is recommended that this procedure be executed after a hundred jobs have appeared in the *READY FOR OUTPUT* list. This process removes only the temporary output versions of the job/takes, the original, most recently revised versions remain in the job directory for further action.

TRANSFER JOBS

The fifth selection of the *PRIMARY* menu allows the operator a choice of three options in the *TRANSFER JOBS* screen. This feature provides MagnaType with a doorway to the outside world of *DOS*.

Option 1, *MOVE JOB FROM AN OUTSIDE SOURCE TO MAGNA,* allows the operator to import a DOS file that may exist elsewhere on the hard disk or on a floppy disk into the MagnaType job directory for editing and/or preparation for output to a typesetter.

Type 1 RETURN

A menu appears requesting the operator to indicate which drive has the outside DOS file. If the file exists in another subdirectory of the hard disk, it must be indicated by typing a *C (assuming C is the hard disk designation)* followed by a *colon (:)* and the *subdirectory name* followed by RETURN. Next, the DOS file *ID* is typed followed by RETURN. Then the operator is required to type in the Magna job/take number to which the DOS text file is to be written followed by RETURN. The operator may then select the job disposition option:

1 is to store the imported job to the Magna job directory for later editing or review

2 is to immediately send the job to the Magna job directory as well as prepare it for output

3 is to send the job to the Magna job directory as well as to H&J it and print it out on the line printer for review.

If the job file being imported was originally stored to disk from a MagnaType computer in MagnaCode, it will already have a job name (description up to 16 characters) to appear in the job directory. If, however, the job file is created on another data processing program, a job name may be incorporated in the file so that it will automatically appear in the Magna job directory. This may be accomplished by keying into the DOS text file as the first item of data <NMx> where x is a name up to 16 alpha/numeric characters.

Option 2, *MOVE JOB FROM MAGNA TO AN OUTSIDE SOURCE,* allows the operator to export a Magna job/take to floppy disk or to another subdirectory of the hard disk.

Type 2 RETURN

A menu appears requesting the Magna job/take number for export. Next, the DOS drive is indicated that will receive the exported job. If it is to go to another subdirectory on the hard disk, type *C (assuming C is the hard disk designation)* followed by a *colon (:)* followed by the *subdirectory name* and a RETURN. Then a DOS file *ID* is requested for storage by DOS. The program will convert the Magna job number to be the DOS file name and convert the hyphen/take number to be a dot/ extension by default, however, this may be rekeyed by the operator. If all is well up to this point, the operator may then indicate which code structure should be exported, *MagnaCode* or *ASCII*. MagnaCode should be used for archival purposes. ASCII should be used if the job/take is to be processed by another DOS program.

Option 3, *MOVE TYPESET OUTPUT TO THE MAGNA SYSTEM*, allows the operator to import a collection of output files from a floppy disk (created by the *JOBS READY FOR OUTPUT* list of another MagnaType program that was not directly cabled to an output device).

Type 3 RETURN

The operator need only type in the floppy disk drive number and all jobs that have been exported to the floppy disk from the other MagnaType program will be imported and appear in the *JOBS READY FOR OUTPUT* list. See the previous section on *OUTPUT*.

Jobs composed and ready for the output device created on a composition program other than MagnaType may be imported from a floppy disk as a DOS file if it has a *·CMP* extension. This job, assumed to be in the language of the output device, will then appear in the *JOBS READY FOR OUTPUT* list and may then be sent to the typesetter as usual. Remember, however, that this *·CMP* file as well as any other imported Magna job/take ready for output is a temporary file and does not exist in the current Magna job directory.

DISPLAY LOG

The last selection in the *DISPLAY JOB DIRECTORIES* is DISPLAY LOG. This is a list of the system activities. Whenever a status line is displayed at the bottom of the screen to indicate that a job is ready to be output, or that the typesetter output is complete, this message will also be loaded into the activity log. Whenever the *TRANSFER JOBS* feature is used, a record of this activity is also entered into the log. Jobs sent to the typesetter are also loaded into the activity log. Additionally, each activity in the log carries the time and date.

The activity log may be sent to the line printer by keying a @ (shift/2) next to *DISPLAY LOG* followed by a RETURN. This activity log may be cleared or reset by calling up the *INITL* program and selecting the first option to *CREATE WORKAREA FILE*. To reset the activity log without clearing the work area, use selection 7 of *SYSTEM MANAGEMENT TASKS*.

Chapter 5
INPUT AND EDITING

Text 5-1

Command Codes 5-1

Formats 5-2

Block Storage, Insert, and Delete 5-2

Delete Copy 5-3

Programmable Keys 5-3

Search and Search/Replace 5-4

Cursor Control 5-5

Caps and Numbers Lock 5-5

Insert 5-5

H&J 5-6

Code Suppression 5-7

Line Ending Suppression 5-7

Case Shift 5-7

Auto Quotes 5-7

Precedent Keys 5-8

Accent Characters 5-9

INPUT AND EDITING

In this chapter we will discuss the variety of features designed into the MagnaType program to aid the operator in creating and editing jobs on the system. Careful study of this and the following chapter with *APPLICATIONS EXAMPLES* should prove essential to the use of the MagnaType Software. Cross references to other sections of this manual are sited throughout. The *QUICK REFERENCE* contained in Chapter 8 will serve as a summary to operations and may be used for future reference.

TEXT

The input of new or original text is accomplished much the same as keyboarding on a standard typewriter. During input, characters will continue to fill a screen line until the last word is too large to fit. The word will drop automatically to the beginning of the following line without the need to key a return and the screen will continue to display new characters as the keyboard operator continues to type.

COMMAND CODES

MagnaType command codes are typed as shown in the *COMMAND CODE* chapter. Alpha characters may be keyed in caps or lower case surrounded by the open < and close > command characters generated by using shift/comma and shift/period or by using the plus and minus keys on the numeric keypad. Several more frequently used command codes, such as the quad codes and fixed spaces may be keyed as a single ALT KEY character (see the Typewriter Key section of the *KEYBOARD* chapter). The operator may also key the four basic parameters (pointsize, lead, column measure, and font) in a streamlined form selected as an option in the *SYSTEM STANDARDS-YES/NO* of the *SYSTEM STANDARDS AND DEFAULTS* in chapter 1.

FORMATS

A format is a collection of codes and/or text that is stored for recall any number of times throughout a job. It is an economical method for reducing the number of keystrokes needed in a job by eliminating the use of redundant input (see the next chapter on *APPLICATIONS*). The contents of a format string is started with a <SFx> command where x is the identifying number and ended with an <EF> command. Format strings entered within a job/take may be accessed within that specific job/take. Format strings entered within take number F may be accessed from any take of a specific job number. Formats created within the *GLOBAL FORMAT OPTIONS* menu may be accessed from any job/take on the system. Typing SHFT/F8 will allow any format to be displayed for reference.

BLOCK STORAGE, INSERT, and DELETE

Some of the *FUNCTION KEYS* combined with the *CURSOR MOVEMENT KEYS* are used to define a block of text within a job/take for removal or copy for later and/or repeated recall.

To define a block, move cursor to the first character of the text to be defined, type SHIFT/F3. The character on the cursor will display in reversed type and a BLK message appears in the upper right corner of the screen to indicate the keyboard is in the *BLOCK DEFINE* mode. Move cursor over and/or down to the position following the last character of the block to be defined. Note that all characters between the start character and end character (cursor) are displayed in reversed type *accept those that are normally displayed in reverse type*. Type F3 and a menu window will appear on the right side of the screen. Selection 1 is indicated by menu default, but any of the 4 selections may be keyed. Selection 1 or 2 followed by RETURN will ask the operator to identify the block number for future recall. Selection 3 followed by RETURN will instantly delete the defined block without storing it for future recall. Selection 4 will terminate *BLOCK DEFINE* mode and return cursor control to job.

To insert a block, move cursor to the position which will follow the insert, type F4 and a window menu will appear where the operator may select which stored block will be inserted. Block 1 is indicated by menu default, but any of the four block numbers may be keyed followed by RETURN. Typing SHIFT/F4 will allow repeated inserts of the same block without going through the window menu each time.

To determine what copy is stored in which block, type F6. A screen display will list the four block numbers followed by the first 72 characters of each block. Type F6 again to continue working on the job.

DELETE

During text entry or editing, a single character may be deleted by placing cursor under the character and touching the DELete key. Holding the DELete key down will cause repeated character deletes to the right of the cursor. As each character disappears, the character to the right moves left to the cursor and becomes the next character to disappear as long as the DELete key is held down.

The BACK SPACE key is used to delete a character to the left of the current cursor position. Holding the key down will allow repeated character deletes.

The FUNCTION 5 key typed twice will cause the word sitting on the cursor to be deleted. A word is defined as a colletion of characters surrounded by spacebands, <, or >. The *SHIFT/FUNCTION 5 KEYS* will cause the entire line sitting on the cursor to be deleted.

PROGRAMMABLE KEYS

Type SHIFT/F6 and the cursor will appear following a P: on the screen line below the job header. Type in up to 70 characters for storage in a programmable key. Type RETURN and a window menu will appear to the right of the screen asking the operator to indicate which of the ten programmable keys to load with the typed information (note 1 is selected by default). Type in the KEY number followed by RETURN. The window will be removed and the cursor will return to the position it held in the job before the operator typed SHIFT/F6.

The programmable number key is now loaded and its contents may be entered in a job by typing ALT/(NUMBER). The contents of each programmable key may be displayed by typing F6. Type F6 again to return to the job.

SEARCH AND SEARCH/REPLACE

Type SHIFT/F9 and a window menu will appear on the right of the screen. The first question is whether the search will be forward to the end of the job or reverse to the beginning of the job. TYPE the selection followed by RETURN. The second question is whether the search will be case sensitive or not. Case sensitive indicates the search will be for the character string exactly as typed, and *no* case sensitive indicates the string searched for must be character for character, but the characters may be upper or lower case in any combination of the two. TYPE the selection followed by RETURN.

Now the cursor will appear on the screen line above the job below the job header following a S: where the search string may be keyed. The window menu gives some additional choices as to how to key in certain non-keyboard characters for a search. The *ERROR MESSAGE* symbol may be searched for by keying ALT/E, and the loose line symbol by keying ALT/Y. A *WILD CARD* choice is also possible which indicates any character in this position in the search string. The *WILD CARD* must be used in combination with other known keystrokes only. Key in ALT/W for a wild card character.

Lets assume the operator wants to search for every occurence of the word *computer* from the beginning of the job to the end regardless of whether it is capitalized or not. The window menu questions would be answered case sensitive *no* and forward search *yes* and the word *computer* would be keyed on the line below the job header followed by a RETURN. The program will begin to search the job until it finds and stops on the first occurence of *computer*. TYPE F9 to continue the search for the next occurence or until the end of the job is reached. When no more search strings appear in the job, a window message will indicate this condition. Type ESCape to continue working on the job.

Now assume the operator wants to search for every occurence of the words *to be* and automatically change it to *will be*. The operator would answer the window menu questions and then type in *to be* followed by INSERT followed by *will be* followed by HOME. TYPE RETURN and the search/replace process will be executed until the end of the job is reached.

And lets assume the operator wishes to search for each occurence of *to be* and only change select ones to *will be*. Follow the procedure above omitting the HOME key. When the first *to be* is found, type INSert if it is to be changed or type F9 if it is not to be changed and the search will continue for the next *to be* occurence.

CURSOR CONTROL

In order to effectively edit a job, it must be easy to move the cursor around in the displayed text.

The cursor ARROW KEYS will move the cursor in the direction indicated by the arrow. CTRL/END will display the last screen of the job, CTRL/HOME will display the first screen of the job, CTRL/PGDN will display the next screen of the job, and CTRL/PGUP will display the previous screen of the job. The RETURN key will move cursor to start of next screen line, HOME will move cursor to the beginning of the present screen, and END will move cursor to the end of the last line on the screen. The SHFT/F7 key will move cursor to the beginning of the next paragraph as indicated by quad codes within the text stream.

CAPS and NUMBERS LOCK

The CAPS LOCK key located to the right of the spacebar is a toggle key that will turn on or off the keyboard condition that produces all caps to be set when alpha keys are typed. A *C* is displayed in reverse video at the lower right corner of the screen when the condition is on. The NUMBERS LOCK key, located on the numeric keypad is a toggle key that will turn on or off the characters in the numeric keypad. An *N* is displayed in reverse video at the lower right corner of the screen when the condition is on.

INSERT

The INSERT toggle key located at the bottom of the numeric keypad. It is a *mode* that will remain in effect until toggled off or H&J is requested. INSERT will allow the cursor to create an opening in the text stream where additonal characters may be keyed rather than overstriking the characters following the cursor position.

IMMEDIATE H&J

A very useful feature of the MagnaType program is the availability to the operator of immediate H&J. By touching the F1 key, the program is instructed to put the current screen job through the hyphenation and justification process and display to the operator the text with lines breaking as they would on the output. The program takes into consideration all character widths, any kerning required, the track values, hanging punctuation, the fixed and variable spaces, and the hyphenation dictionary to justify each line of type. Also being considered are the vertical positions within the job of each typeset baseline depending on the leading, pointsizes, and vertical movement commands within the job. To indicate the results of these computations, line ending numbers are displayed to show the horizontal and vertical position of each line. Left indent values in picas and points if other than zero are followed by a slash and the total column measure of the text, and following the *x* is the depth from the top of the job in picas and points. Machine calculated hyphens are displayed in reverse video at the end of the appropriate line. During the justification process, if the program cannot justify a line without exceeding the maximum spaceband value requested, the line ending displays a range value of how much text could be added without altering the previous and following line break decisions. This range is displayed in picas/points or in multiples of en spaces depending on how the *SYSTEM STANDARDS-YES/NO* screen is set up (see *SYSTEM STANDARDS AND DEFAULTS* in chapter 1).

When the H&J process is intitiated by touching the F1 key, the screen will go blank for an instant and then begin to display the new lines. When the screen is full, the cursor may be moved about and editing or proofing may begin. When the entire job is processed, a new depth value will be displayed in the upper right corner of the screen.

COMMAND CODE SUPPRESSION

After H&J, to facilitate proofing and editing, the typesetting command codes may be suppressed by typing a SHIFT/F2. All MagnaType command codes or strings of command codes that are enclosed in the open and close command characters on a screen line will be reduced to a single diamond character at their position in the text. Typing SHIFT/ F2 again will bring the codes back to full display.

LINE ENDING SUPPRESSION

After H&J, to facilitate proofing and editing, the line ending values may be suppressed by typing an F2. The line endings will be replaced with left arrows to indicate to the operator machine generated line endings as opposed to screen wrapped line endings. In this mode, line looseness indicators will remain on the screen. Typing F2 again will bring the line ending values back to full display.

CASE SHIFT

The F7 key is typed to change the character indicated by the cursor from lower to upper case or upper to lower case and advance the cursor to the following character. Repeated case changes can be initiated by holding down the F7 key.

AUTO QUOTES

Open or close single and double quotes may be keyboarded using the single and double quote key as with a typewriter or word processor. During H&J and compostion for the output device, the program will automatically produce open or close quote characters partially based on the existence or nonexistence of a spaceband prior to the autoquote keystroke. To manually force an open or close quote for special situations, use the open quote or shift/6 keys (see *KEYBOARD* chapter). The autoquote feature may be turned on or off in the *SYSTEM STANDARDS-YES/NO* screen of *SYSTEM STANDARDS AND DEFAULTS*.

PRECEDENT KEYS

Because the number of codes that are generated from single keystrokes is limited, precedent keys are established to supplement the access to more codes.

The backslash key (see *KEYBOARD* chapter) is used to generate a π symbol precedent character on the screen and is used in combination with the following keystroke to generate a single code. Some characters on the output font that cannot be accessed by single keystrokes are accessed by using the π precedent character.

The TAB key (see *KEYBOARD* chapter) is used in both SHIFT and UNSHIFT to produce reverse video up and down arrows to indicate that the following character is to be set as a superior or inferior on output. These characters are automatically generated by the system based on reduction parameters contained in the *SYSTEM STANDARDS-VALUES* screen. The superior or inferior precedent may also be entered in long form, <SU> for superior and <IN> for inferior, however, the system will convert them to up and down arrows for screen display.

The semi-colon key is used as a precedent to a two character combination that serves as a streamlined mnemonic format call (see *MNEMONIC/FORMAT REFERENCE TABLE* in chapter 1). These semi-colon mnemonics may be keyed in upper or lower case.

The $ followed by two alpha characters is used to request a special character on a specific font without having the operator worry about font changes. These mnemonics are indicated in the *$PI ACCESS TABLE* of the *FONT INFORMATION*. These $PI mnemonics may be keyed in upper or lower case.

ACCENT CHARACTERS

If your output device is loaded with fonts that carry accent characters, they must be indicated as accents in the width screens of the *FONT INFORMATION* tables (see chapter 1).

If the accent characters are resident on each of the text fonts, type the accent character keystroke followed by the keystroke of the character to be accented. If, however, the accents are resident on a different font, a font change command between the accent and the next character will negate the accent routine. Therefore, to get an accent character from another font, use a $PI keystroke as the accent character followed by the keystroke of the character that is to be accented.

For example, let's assume <CFTR> is *Times Roman* and the accent character keystroke for tilde is πt on this font. To put this accent over the letter 'n' in the word 'montana', type:

<CFTR>montaπtna

and it will output as:

montaña

Now, let's assume the tilde accent character is resident on another font called PI and it is character number 21. In the *$PI ACCESS TABLE*, a mnemonic $TI could be created to equal character 21 on font PI. To put this accent over the letter 'n' as above, type:

<CFTR>monta$tina

MagnaType will automatically make the proper font change, flash the accent character, and then return to the current text font.

If a word appears at the end of a line and it requires hyphenation, the dictionary will ignore the accent characters in order to select the proper hyphenation points in the word.

Chapter 6
APPLICATIONS EXAMPLES

Begin and Print Number 6-1

Control Ragged Zone 6-3

Formats 6-4

Horizontal Escapement 6-6
- Tracking 6-6
- Kerning 6-6
- Letterspacing 6-7
- Set Width 6-7
- Word Spacing 6-7
- Fixed Spaces 6-8
- Escape Immediate 6-9
- Hanging Punctuation 6-9
- Force Justify 6-9
- No Escape 6-9
- Underscore 6-10

Indents 6-12
- Indent Take 6-12
- Indent Paragraph 6-12
- Indent First 6-13
- Indent Hang 6-13
- Indent Text 6-13
- Indent Left and Right 6-13
- Skew Text 6-13

Horizontal Rules 6-17

Leaders 6-17

XY Coordinates 6-19

Tab Number 6-21

Tab Set 6-23

Tab Proportional 6-25

Tab Stub 6-28

Tab Text 6-30

Applications Examples *continued*

Additional Tab Notes 6-32
Quit Tab 6-34
Jump Tab 6-34
Leading 6-36
Point Mark 6-36
Return Mark 6-37
Point Maximum 6-37
Return Maximum 6-37
Clear Maximum 6-37
Delay Lines/Measure 6-39
Vertical Rules 6-39

APPLICATIONS EXAMPLES

In this chapter, we will attempt to provide some helpful examples of actual production requirements where the operator may draw on the information contained throughout the previous chapters of this manual. References to other sections are contained throughout for additional information.

BEGIN AND PRINT NUMBER

The automatic number counter feature designed into the program may be used in any number of jobs on the system to produce the automatic generation of sequential numbers in place of an embedded code.

For example, assume the operator is keying in a long list of lines or blocks of text that will eventually become sequentially numbered. Rather than typing the number of each item, the operator need only key <PNx> in place of the number and when the job is composed for output, each <PNx> will be replaced with a sequential number in the required typeface and pointsize specified. By default, the sequential numbers will begin with 1 at the first occurrence of a <PNx> code, however, the <BNx,y,z> code may be used prior to the first <PNx> to indicate a start number other than 1. *See example 1.*

The advantage of using <PNx> rather than actually keying in the numbers is that numeric items may be added, deleted, or rearranged without the need for retyping all the sequential numbers.

Additionally, it is possible to have up to nine number counters working independently in any given job/take for sequencing page numbers, footnote numbers, section numbers, line numbers, paragraph numbers, and other counting elements. Also, it is possible to account for leading figure spaces or leading zeros in the automatically generated numbers.

See: <BNx,y,z>, <PNx>

EXAMPLE 1

KEYBOARD INPUT:

<CP10> <CL12> <CC20> <CFTR> <BN5,1> <BN7,2,-3>
This is BLOCK number <PN1> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is BLOCK number <PN1> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
<BN98,2,3>
This is BLOCK number <PN1> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>
This is line number <PN2> <EP>

TYPESET OUTPUT:

This is BLOCK number 5
This is line number 7
This is line number 8
This is line number 9
This is line number 10
This is BLOCK number 6
This is line number 11
This is line number 12
This is line number 13
This is line number 14
This is BLOCK number 7
This is line number 098
This is line number 099
This is line number 100
This is line number 101

CONTROL RAGGED ZONE

The Ragged Zone command <CRx,y> is used to define conditions and restrictions for the horizontal position of line endings with or without consideration of the ending position for the previous line. The first of the two variables in the command is used to define the range from the margin in which the line must terminate and if used with the second variable specified as zero <CR5,0>, each line will end at various points within the range of 5 picas. To further control the position of the line endings, a hot zone is specified with the second variable in the command. The first line of the paragraph will then end within the ragged zone and establish a hot zone within the ragged zone which the following line ending must avoid. The hot zone or second variable is a value that straddles each line ending. If <CR6,4> is specified, each line must end within the ragged zone of 6 picas, but not closer than 2 picas from the end of the previous line. Obviously, if strict demands are made on the line ending positions with the Ragged Zone Command, the system will use hyphenation, letterspacing, and/or spaceband expansion and contraction to achieve the desired *ragged look* if allowed.

See: <CRx,y>, <CSx,y,z>, <AL>, <AH>, <HC>

FOR EXAMPLE:

<CR8,6> <CC20>
The Ragged Zone command <CRx,y> is used
to define conditions and restrictions for the horizontal position
of line endings with or without consideration
of the ending position for the previous line. The first of the
two variables in the command is used to define
the range from the margin in which the line must terminate
and if used with the second variable specified
as zero <CR5,0>, each line will end at various points within
the range of 5 picas.

<CR4,2>
The Ragged Zone command <CRx,y> is used to define conditions
and restrictions for the horizontal position of line endings with or
without consideration of the ending position for the previous
line. The first of the two variables in the command is used to define
the range from the margin in which the line must terminate
and if used with the second variable specified as zero <CR5,0>,
each line will end at various points within the range of 5 picas.

FORMATS

A format is a collection of codes and/or text that is stored for recall any number of times throughout a job. It is an economical method for reducing the number of keystrokes needed in a job by eliminating the use of redundant input.

For example, assume the same parameters are required for various bodies or blocks of text throughout a job: pointsize 10, leading 12, column measure 25, font Times Roman (TR), and a ragged center mode.

The parameter codes:

<CP10> <CL12> <CC25> <CFTR> <RC>

could be collected in a format number 25:

<SF25> <CP10> <CL12> <CC25> <CFTR> <RC> <EF>

and then before the body or block of text, the operator would key *use format 25* as:

<UF25>

When parameters or conditions change, another format could be requested or a merge copy could be used and contain all parameters in one format.

For example, assume the same parameters as in the previous example, except that after the first three words of the paragraph, the text should change to Times Roman Italic (TRI) and pointsize 9. The format could be keyed as:

<SF25> <CP10> <CL12> <CC25> <CFTR> <RC> <MC> <CFTRI> <CP9> <EF>

The operator would key:

<UF25>

followed by three words of text, and then *merge copy* as:

<MC>

When a format is requested in the text stream, all codes and text contained in the format until an <EF> or a <MC> are considered before control returns to the text stream. If a <MC> in a format causes an exit from format to text stream, a <MC> in the text stream will cause a return to the conditions in the format following the <MC> again until an <EF> or another <MC> causes a return to the text stream.

Another method for causing a temporary exit from a format is use of the delay line <DLx> or delay measure <DMx> code in the format (see the *COMMAND CODE* chapter).

Formats may also be nested up to five levels for more complex parameter change operations. For example, if a format is requested and a <MC> causes a return to the text stream, a second format may be requested. All conditions will be considered and acted upon in the second format until a <MC> is encountered in the text causing a return to consider more conditions in the first format.

Subsequently, the second format may have a <MC> incorporated in it to return control to the text stream where yet a third format could be requested. Care must be taken, however, when constructing such an arrangement that no more than five levels of formats are being maintained at any one time. If a sixth format is called into the nest, the conditions following the <MC> in the first format will never be accessed. Also, a stray <MC> in the text stream could throw the entire sequence out of sync.

Remember, the feature of nested formats is provided to give the more proficient user a method of saving operator keystrokes in more complex typesetting applications. Careful planning is a necessity.

Also, in this section on formats, we have referred to *merge copy* or *merge code* as <MC>, although ALT/X is also acceptable as an alternative keystroke sequence.

See: <SFx>, <EF>, <UFx>, <MC>, <DLx>, <DMx>, <RF>

HORIZONTAL ESCAPEMENT

MagnaType command codes used by the operator to initiate a horizontal movement within a typeset job fall into two categories. The first are those that may be controlled by system specifications with override codes available to the operator.

TRACKING - an overall reduction or addition to every character width by a fixed amount of relative units. These values may be preset for each font in the *FONT INFORMATION* and requested by using the <Tx> command where x is the track required. The usual practice is to establish the three tracks for each font where the second is a tighter set than the first, and the third is a tighter set than the second. The operator is then able to select various track values to view their effects on H&J. For example, assume the operator is attempting to get a certain amount of copy in the least amount of space. Letterspacing has been cancelled, kerning is in effect and spacebands have been reduced to an acceptable minimum. A tighter track may also be requested to see if more copy can be fitted onto each line and possibly reduce the number of lines of text. Tracking may also be adjusted by using the <SWx> command where x is a value in relative units of overall reduction or addition to each typeset character. An <SWx> command will add to or subtract from a current track value in effect. If the operator desires to experiment with tracking using the <SWx> command without impacting an established track value, use <T0> to cancel any current tracking value that maybe in effect.

See: <Tx>, <SWx>, *FONT INFORMATION*

KERNING - a reduction or addition to the space between a pair of typeset characters. These values may be preset for specific character pairs for various typefaces in the *FONT INFORMATION* tables and requested or cancelled by using the <AK> or <XK> command codes. The operator may also use a <KCx> command where x is a value in relative units to be removed or added between two characters as an override to the value that appears in the autokern table, or a <KAx> where x is a value to be considered along with the value that appears in the autokern table.

See: <AK>, <XK>, <KCx>, <KAx>, *FONT INFORMATION*

LETTERSPACING - an addition to the space between all characters on a line in an effort to force the type to justify. This addition is controlled by the <LSx> command where x is a maximum value to be used in relative units. A default value is established in the *STANDARDS AND DEFAULTS* menu (see chapter 1) and can be modified within the job/take using <LSx>.

SET WIDTH - a command used to force digital type to be expanded or condensed by requesting width values of one pointsize to be used while setting type in another pointsize. For example, expanded type characters would result by requesting <CP10> along with <CW20>. This would result in 10pt type being set with width values of 20pt type. If a pointsize is requested without a <CWx> command, normal width values will be used.

See: <CWx>, *SYSTEM STANDARDS AND DEFAULTS*

VARIABLE WORD SPACING - a range of acceptable word space values in relative units used to achieve a justified line of type. Default values established in *STANDARDS AND DEFAULTS* may be modified within any job using the <CSx,y,z> command where x is the minimum allowable word space, y is the optimum or most desirable word space, and z is the maximum word space. This is the first variable considered when the program is attempting to justify a line of type. Standard ragged copy where no control ragged zone parameters are being requested uses the optimum word space value.

See: <CSx,y,z>, *STANDARDS AND DEFAULTS*

The second category of horizontal escapement are those codes controlled specifically by the operator.

FIXED SPACES - values used for horizontal space that are relative to the current pointsize in effect. The maximum fixed space is called an em space or *mutt* and is coded in a job as <EM> or ALT/M. The width of an em space is equal in points to that of the current pointsize in effect and is defined for each typeface in the *FONT INFORMATION*. For example, an em space in 14pt type is a horizontal space of 14pts. Width values of typeset characters are specified in fractions of an em and are often referred to as relative units. Traditionally, all character widths were specified in amounts of 1/18th of an em space. With the advent of phototype, smaller increments were desired and some manufacturers started to use 1/54th of an em as a relative unit and others went so far as to use 1/100th of an em as a relative unit.

The next smaller fixed space after an *em space* is called an en space or *nutt* and is coded in a job as <EN> or ALT/N. The width of an en space is defined for each typeface in the *FONT INFORMATION*. Traditionally, the value of the en space is equal to ½ the value of an em space.

The next smaller fixed space is called a thin space and is coded in a job as <TH> or ALT/T. The width of the thin space is defined for each typeface in the *FONT INFORMATION*. Traditionally, the value of the thin space is equal to ⅓ the value of an em space.

The next smaller fixed space is called a unit space and is coded in a job as <UN> or ALT/U. The width of a unit space is defined for each typeface in the *FONT INFORMATION*. Traditionally, the value of the unit space is equal to 1/18th on an em space.

Additionally, a figure space is available and is coded in a job as <FG> or ALT/F. The width value of the figure space is equal to the width values of the numbers in the current typeface. The MagnaType system establishes the width value of the figure space based on the width value loaded for the number zero. The figure space command is generally used in tabular material to hold a space normally occupied by a number or *figure* to maintain alignment in columns of numbers.

See: <EM>,<EN>,<TH>,<UN>,<FG>, *FONT INFORMATION*

ESCAPE IMMEDIATE - a value in picas and points to indicate an immediate forward horizontal move with or without a specified weight for a baseline rule. This may be coded in a job as <EIx> where x is the amount of escapement in picas and points, or <EIx,y> where y indicates a rule weight for a baseline rule to extend the distance of x. The escape immediate command is generally used in forms work where a blank line is typeset for handwritten material to be entered.

See: <EIx,y>

HANGING PUNCTUATION - This is a feature of the system to automatically hang outside the margin certain punctuation characters, the net result being that their true character width is not considered when calculations for what will fit within the measure are being made. The feature is turned on with the code <HP> and turned off with the code <XP>. Punctuation marks at the beginning of a line that will set outside the margin are single quote, double quotes, and the asterisk. Punctuation marks at the end of a line that will hang out in the margin are period, hyphen, comma, single close quote, double close quotes, and the asterisk. If hanging punctuation is requested, the left margin of the typeset material will physically be indented by the amount specified in the *SYSTEM STANDARDS AND DEFAULTS* when sent to the output device.

See: <HP>, <XP>, *STANDARDS AND DEFAULTS*

FORCED JUSTIFY - the code used to force expansion of the interword spacing in order to fill the measure with the contents of what would normally be a short line of text. Generally used on the last line of a paragraph that is almost full measure when a block style of copy is desired. Code as <JU> or ALT/J in place of the normal quad code or end paragraph code.

See: <JU>

NO ESCAPE - the code is used to indicate that the following character is to be set but no escapement is to take place. The effect is that the character will overprint the following character. The code is set in a job as <NE>. If the last character on a typeset line is indicated to no escape, its width value will not be considered in calculating the line and the character will set outside the margin. The code is generally used to intentionally overprint two typeset characters or to hang a character outside the margin.

See: <NE>

UNDERSCORE - the code is used to indicate the start and stop points for the underlining of output text. The code is set in a job as <USx,y> where x is the number of points above or below the baseline for the rule to set and y is the weight of the rule to be set. A positive x is points below the baseline, a negative x is points above the baseline. If no y is specified, the rule weight will default to the value in *STANDARDS AND DEFAULTS*. If no x is specified, the underscore rule will be positioned in points equivalent to ⅓ the current pointsize of type. The first occurrence of the code encountered will turn the underscore feature on until it encounters another underscore command. *See example 2*. Care must be taken that no start and stop underscore codes are omitted in a job or underscores may appear where they were not required and not appear where they were required.

See: <USx,y>, *SYSTEM SPECIFICATIONS*

EXAMPLE 2

KEYBOARD INPUT:

<CP10> <CL24> <CC20> <CFTR>
This is a <US>demonstration<US> of underscore.<QL>
This is a <US6>demonstration<US> of underscore.<QL>
This is a <US3,5>demonstration<US> of underscore.<QL>
This is a <US-3,5>demonstration<US> of underscore.<QL>

TYPESET OUTPUT:

This is a demonstration of underscore.

This is a demonstration of underscore.

This is a demonstration of underscore.

This is a ~~demonstration~~ of underscore.

INDENTS

The code set of MagnaType offers a variety of text indents that may be used within a job/take that are automatically triggered or cancelled when an <EP> is encountered. Some indent commands build or 'add to' other indents and some are specific measurements from the margin. With a little thought, many of these commands may be located in formats or setup at the beginning of the job/take and ultimately save time and keystrokes when the actual text material is input.

The actual position of the text on the output material, be it paper or film, is determined by the limits of the output device and is usually a fraction of an inch from the left edge of the material itself. This will be referred to as *the image area.* The only instruction from the program that would alter the starting point of *the image area* is a request for hanging punctuation where extra space prior to the left margin is required for certain start of line characters to be set.

The right margin is the ending point of all typeset material in *the image area.* Its position on the output material is determined by the column measure command <CCx> used in the job/take or defined in the *SYSTEM DEFAULTS-VALUES* menu and is to the right of the left margin in an amount of picas and points.

INDENT TAKE - an instruction to move the start of the typeset material to the right of the left margin and/or the end of the typeset material to the left of the right margin. This command remains in effect throughout the job/take until changed by another of the same code.

See:<ITx,y>

INDENT PARAGRAPH - an instruction to move the start of the typeset material to the right of the left margin and/or the end of the typeset material to the left of the right margin where the margins are either those of *the image area* or those defined by the *INDENT TAKE* command in effect. The amounts of indent specified will remain in effect for each line following the code up to and including the next line with an <EP>.

See: <IPx,y>

INDENT FIRST - an instruction to indent the first text line following an end paragraph <EP> code by an amount specified from the left and/or right margin defined by *the image area,* the indent take or indent paragraph command. The code is used instead of typing fixed space or spaces prior to the text in the first paragraph line.

See: <IFx,y>

INDENT HANG - an instruction to indent all lines of text following the first text line following an end paragraph <EP> code by an amount specified from the left and/or right margin defined by *the image area,* the indent take or indent paragraph command. Each line will indent up to and including the next line with an <EP>.

See: <IHx,y>

INDENT TEXT - an instruction to mark a horizontal point based on the position of the code <IXn> relative to the type preceding it. Once established, all text lines following will indent from the left margin starting at the horizontal point up to and including the next line with an <EP>. Additionally, at any point later in the job/take, this indent point may be returned to with an <RIn> code or re-established with another <IXn>. An ideal application for this set of indent codes would be a series of entries in an outline form. *See example 3.*

See: <IXn>,<RIn>

INDENT LEFT AND RIGHT - a series of instructions within a command code to indent a number of text lines a specified amount. A practical application would be the construction of irregularly shaped text for wrapping around artwork. The indents specified are measured from the margins of *the image area* or those defined by an indent take or indent paragraph command in effect. *See example 4.*

See: <IL...>, <IR...>

SKEW TEXT - a series of instructions within a command code to spread an amount of indent gradually over a number of text lines producing a slanted text margin. Both a left margin and right margin skew indent may be established at the same time to create shaped text. The indents specified are measured from the margins of *the image area,* indent take or indent paragraph in effect. *See example 5.*

EXAMPLE 3

KEYBOARD INPUT:

<CP10> <CL11> <CC25> <CFTR>
1.00<EM> <IX1> This data will now hang on the IX1 set up at the beginning of this line and continue to hang until it encounters an EP at which time it will allow following data to begin at the margin<EP>
1.01<EM> <RI1> This RI1 will allow following lines to hang<QL>
A.<EM> <IX2> This IX2 will establish a hanging indent point for data following this two level entry in the outline.<EP>
<RI1> B.<EM> <RI2> RI1 puts B. at correct indent and RI2 allows the wrap lines of this entry to hang in the proper place. An EP will again return type to margin <EP>
<RI2> 1.<EM> <IX3> RI2 returns third level entry to proper indent and allows IX3 to establish a hang indent for wrapped text following.<EP>
<RI2> 2.<EM> <RI3> Now all the operator needs to do to set type aligned on the proper indent is key in the appropriate RI command and all lines of type following will indent to that preset IX <EP>
<RI1> This is IX1 indent <EP>
<RI2> This is IX2 indent <EP>
<RI3> This is IX3 indent <EP>

TYPESET OUTPUT

1.00 This data will now hang on the IX1 set up at the beginning of this line and continue to hang until it encounters an EP at which time it will allow following data to begin at the margin

1.01 This RI1 will allow following lines to hang

A. This IX2 will establish a hanging indent point for data following this two level entry in the outline.

B. RI1 puts B. at correct indent and RI2 allows the wrap lines of this entry to hang in the proper place. An EP will again return type to margin

1. RI2 returns third level entry to proper indent and allows IX3 to establish a hang indent for wrapped text following.
2. Now all the operator needs to do to set type aligned on the proper indent is key in the appropriate RI command and all lines of type following will indent to that preset IX

This is IX1 indent

This is IX2 indent

This is IX3 indent

EXAMPLE 4

KEYBOARD INPUT:

<CP10> <CL12> <CC20> <CFTR>
<IL2/0,2/2,2/4,2/6,2/4,2/2> <IR4/6,4/4,2/2>
This is a demonstration of the indent left and indent right command strings. From the left, two lines will set with no left indent, two more lines will set with a two pica indent, two more will set with a four pica indent, two more lines will set with a six pica indent, two more lines will set with a four pica indent, two more lines will set with a two pica indent, and the rest of the text will have no left indent. From the right, four lines will set short six picas from the right margin at 20, four more lines will set four picas short of the right margin, two more lines will set two picas short of the right margin, and the rest of the text will have no indent from the right.<EP>

TYPESET OUTPUT

This is a demonstration of the indent left
and indent right command strings. From
the left, two lines will set with no
left indent, two more lines will set
with a two pica indent, two more
will set with a four pica indent, two
more lines will set with a six
pica indent, two more lines
will set with a four pica indent, two more
lines will set with a two pica indent, and
the rest of the text will have no left indent. From the
right, four lines will set short six picas from the right
margin at 20, four more lines will set four picas short of the
right margin, two more lines will set two picas short of the
right margin, and the rest of the text will have no indent
from the right.

EXAMPLE 5

KEYBOARD INPUT:

<CP10> <CL12> <CC25> <CFTR>
<SL1/0,7/6,-7/6>
<SR1/0,4/5,-4/5,5/3>
This is a demonstration of the MagnaType skewing command strings. The skew left codes indicate one line starting at the left margin, seven more lines will gradually indent from the left to six picas and then seven more lines will gradually return from a six pica indent to the left margin with the rest of the text lines starting at the left margin. The skew right codes indicate one line to set all the way to the right margin, the next four lines gradually indenting to five picas short of the right margin, four more lines gradually returning to the right margin from an indent of five picas followed by five lines gradually reaching an indent of three picas from the right margin with the rest of the text set to full right margin. With the use of the skew command strings, the operator could establish wraps around shaped artwork. Remember, each skew command string may contain up to 20 pairs of values and will be acted upon as long as the amount of set text continues. Repeated patterns of shaped indents could be formatted for easier keyboarding.

TYPESET OUTPUT

This is a demonstration of the MagnaType skewing command strings. The skew left codes indicate one line starting at the left margin, seven more lines will gradually indent from the left to six picas and then seven more lines will gradually return from a six pica indent to the left margin with the rest of the text lines starting at the left margin. The skew right codes indicate one line to set all the way to the right margin, the next four lines gradually indenting to five picas short of the right margin, four more lines gradually returning to the right margin from an indent of five picas followed by five lines gradually reaching an indent of three picas from the right margin with the rest of the text set to full right margin. With the use of the skew command strings, the operator could establish wraps around shaped artwork. Remember, each skew command string may contain up to 20 pairs of values and will be acted upon as long as the amount of set text continues. Repeated patterns of shaped indents could be formatted for easier keyboarding.

HORIZONTAL RULES - the setting of a horizontal rule may be executed on the current baseline with a single command code, <WR> which uses the default rule thickness or weight indicated in the STANDARDS AND DEFAULTS, or <WRx> where x is the rule thickness variable dependent on the output device. With the second variable, the rule can be positioned so many points below the current baseline, but all text following until a quad code is encountered will be set on the newly displaced baseline.

The rule will begin to set to the right from the horizontal point indicated by the position of the code and continue until more text or a quad code is encountered. A rule command found between text elements on the line will act as a quad middle command forcing the text on the left out to the left margin and the text on the right out to the right margin, setting a rule of specified thickness on the baseline in between the text elements. *See example 6.*

LEADERS - the setting of leader dots may be executed on the current baseline with a single command code, <WL> which uses the default value in the *STANDARDS AND DEFAULTS* to determine the amount of spacing between dots (or periods) of the current font and pointsize. <WLx> where x is a value in relative units will set leader dots with x amount of space between dots. The leader dots will begin at the horizontal point marked by the position of the code and continue to the right until more text is encountered or a quad code. The proper selection of a quad code on a line with leaders depends on how the leaders are being used. On a line that begins with leaders and ends with text the end of line command should be quad right, a line that begins with text and ends with leaders should be quad left, and a line with leaders between text elements should be quad center. This insures that consecutive lines of leader dots of the same spacing will align vertically.

The user may also elect to set leadered characters using the <WXn,m> command where n equals the character to be leadered and m equals the amount of relative units between the center of one leader character and the center of the next leader character. *See example 6.*

EXAMPLE 6

KEYBOARD INPUT:

<CP10> <CL11> <CC25> <CFTR>
<WR> <QL>
<WR5> <QL>
<WR10> <QL>
TEXT <WR> COPY <QL>
TEXT <WR> TEXT <WR> TEXT <QL>
TEXT <WR> <WR> TEXT <WR> TEXT <QL>
<EL6> Thin Space Leaders <WL18> <QL>
En Space Leaders <WL27> <QL>
Em Space Leaders <WL54> <QL>
<EL6> Leader Character A <WXA> <QL>
Leader Character M <WXM> <QL>
Leader Character z <WXz> <QL>
<EL6> TEXT <WL> <QL>
<WL> TEXT <QR>
TEXT <WL> TEXT <QC>
TEXT COPY <WL> TEXT COPY <QC>
DOTS <WL> DOTS <QC>

TYPESET OUTPUT:

TEXT COPY
TEXT TEXT TEXT
TEXT TEXT TEXT

Thin Space Leaders .....
En Space Leaders .....
Em Space Leaders . . . . . .

Leader Character A AAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Leader Character M MMMMMMMMMMMMMMMMMMMMMMMMMM
Leader Character zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

TEXT .....
..... TEXT
TEXT ..... TEXT
TEXT COPY ..... TEXT COPY
DOTS ..... DOTS

X-Y COORDINATES

The first variable in the XY coordinate command <XYx,y> is an instruction to move the starting point of all horizontal escapement instructions that normally measure from the left margin. The second variable repositions the point from which all vertical escapement (or leading) instructions are measured. Using the XY coordinates, text blocks may be placed at various positions in the job. This allows the operator to then keyboard blocks of copy in a logical order, but not necessarily in the order they will appear on the finished page. *See example 7.*

EXAMPLE 7

KEYBOARD INPUT:

<CP10> <CL12> <CC7> <CFTR>
<XY0,0> This is block ONE. This is block ONE. This is block ONE. This is block ONE. This is block ONE. <EP>
<XY10,14> This is block TWO. This is block TWO. This is block TWO. This is block TWO. <EP>
<XY10,0> This is block THREE. This is block THREE. This is block THREE. This is block THREE. <EP>
<XY0,7> This is block FOUR. This is block FOUR. This is block FOUR. This is block FOUR. <EP>
<XY10,7> This is block FIVE. This is block FIVE. This is block FIVE. This is block FIVE. <EP>
<XY0,14> This is block SIX. This is block SIX. This is block SIX. This is block SIX. <EP>

TYPESET OUTPUT:

This is block ONE.
This is block ONE.
This is block ONE.
This is block ONE.
This is block ONE.

This is block
THREE. This is
block THREE. This
is block THREE.
This is block
THREE.

This is block
FOUR. This is
block FOUR. This
is block FOUR.
This is block
FOUR.

This is block FIVE.
This is block FIVE.
This is block FIVE.
This is block FIVE.

This is block SIX.
This is block SIX.
This is block SIX.
This is block SIX.

This is block TWO.
This is block TWO.
This is block TWO.
This is block TWO.

SETTING TABULAR MATERIAL

There is usually more than one way to establish a tab condition for any particular application. The best way to approach tabular setups is to first consider the style of table to be produced, examine the known variables, and then select the tab setup command that seems easiest to apply.

TAB NUMBER - The easiest tab setup command to understand and work with is the <TNx> Tab Number command where x is a number of equal tab columns to be established across the current column measure. With a <CC20> and a <TN4>, the program will automatically establish 4 columns each having a width of 5 picas. In other words, column 1 will extend from the left margin to 5 picas, column 2 will extend from 5 to 10 picas, column 3 from 10 to 15 picas, and column 4 from 15 to 20 picas.

Now lets assume 8 point gutters are required with vertical rules of weight 1, the setup command will be: <TN4,G.8/1> where G.8 indicates an 8pt gutter, /1 indicates automatic vertical rules of weight 1, and 4 indicate 4 equal columns. With this command, the program will establish 3 gutters of 8 points each (total of 2 picas) and 4 columns of 4 picas and 6 points each (total of 18 picas) over a 20 pica column measure.

In other words, column 1 will extend from the left margin to 4 picas and 6 points where the first gutter will start, column 2 will extend from the end of the first gutter at 5 picas and 2 points to 9 picas and 8 points where the second gutter will start, column 3 will extend from the end of the second gutter at 10 picas and 4 points to 14 picas and 10 points where the third gutter will start, and column 4 will extend from the end of the third gutter at 15 picas and 6 points to the right margin at 20 picas.

As you can see, these calculations are all established by the program based on the simple parameters indicated by the operator: four equal columns with 8 point gutters and automatic vertical rules over a 20 pica column measure. *See example 8.*

EXAMPLE 8

KEYBOARD INPUT:

```
<CP10> <CL12> <CC20> <CFTR>
<TN4,G.8/1>
TABLE USING TAB NUMBER <QC>
<WR1> <QL>
<MA>
This is <QC>
This is <QC>
This is <QC>
This is <QC>
<MA>
Column 1 <QC>
Column 2 <QC>
Column 3 <QC>
Column 4 <QC>
<MA>
copy block <QC>
copy block <QC>
copy block <QC>
copy block <QC>
<QT>
<TR1,6> <QL>
```

TYPESET OUTPUT:

TABLE USING TAB NUMBER

This is Column 1 copy block	This is Column 2 copy block	This is Column 3 copy block	This is Column 4 copy block

TAB SET - This tab setup command works the same as a traditional typewriter tab setup where the values specified in the <TSp1,p2,p3,...> Tab Set command indicate horizontal points on the column measure that define the start of a tab column. The first column has an implied start at the left margin and need not be included in the command.
With a <CC20> and <TS5,11,17>, the program will recognize 4 columns with widths of 5, 6, 6, and 3 picas respectively. In other words, column 1 will extend from the left margin to 5 picas, column 2 from 5 picas to 11 picas, column 3 from 11 picas to 17 picas, and column 4 from 17 picas to 20 picas.
Now lets assume 6 point gutters are required with vertical rules of weight 1, the setup command will be:
<TSG.6/1,5,11,17> where G.6 indicates a 6 point gutter, /1 indicates automatic vertical rules of weight 1, and 5,11,17 indicate column starts at 5 picas, 11 picas, and 17 picas across the measure of 20 picas.
The program will then establish column 1 from the left margin to 4 picas and 6 points, column 2 from 5 picas to 10 picas and 6 points, column 3 from 11 picas to 16 picas and 6 points, and column 4 from 17 to 20 picas. *See example 9.*

EXAMPLE 9

KEYBOARD INPUT:

<CP10> <CL12> <CC20> <CFTR>
<TSG.6/1,5,11,17>
TABLE USING TAB SET <QC> <WR1> <QL>
<MA>
NO.1 <QC>
NO.2 <QC>
NO.3 <QC>
NO.4 <QC>
<MA>
Copy <QC>
Copy <QC>
Copy <QC>
Copy <QC>
<QT>
<TR1,6> <QL>

TYPESET OUTPUT

TABLE USING TAB SET

NO.1 Copy	NO.2 Copy	NO.3 Copy	NO.4 Copy

TAB PROPORTIONAL - This tab setup command may be used when columns are required to be proportionally related to each other over a given total measure. <TP1,2,1,4> indicates the setup of 4 columns where the second column is twice the width of columns one and three, and column four is 4 times the width of columns one or three and twice the width of column two. Assume a total measure of 24 picas <CC24>, column one and three are each 3 picas wide, column two is 6 picas wide, and column four is 12 picas wide (a total of 24 picas). Changing the total measure to 16 picas <CC16> will instruct the program to recalculate the column measures. Column one and three will be 2 picas wide, column two will be 4 picas wide, and column four will be 8 picas wide (a total of 16 picas).

Lets assume a 1 pica gutter is required between each column with an automatic vertical rule of weight 1 over a total tab measure of 19 picas. The gutter specifier need only be keyed once. The setup command will be: <CC19> <TP1,G1/1,2,1,4> where G1 indicates 1 pica gutters, /1 indicates a vertical rule of weight 1 automatically generated in each gutter, and 4 columns of proportional width as described above. For the purpose of this example, all column and gutter points on the horizontal measure calculate to even picas, however, if the total measure is changed to any other value, the program will recalculate all the start and stop points for each column and gutter automatically. *See examples 10 and 11.*

EXAMPLE 10

KEYBOARD INPUT:

<CP10> <CL12> <CC19> <CFTR>
<TP1,G1/1,2,1,4>
TABLE USING TAB PROPORTIONAL<QC> <WR1> <QL>
<MA>
copy <QC>
copy <QC>
copy <QC>
<MA>
text <QC>
text <QC>
text <QC>
<QT>
<TR1,6> <QL>

TYPESET OUTPUT

TABLE USING TAB PROPORTIONAL

copy text	copy text	copy text	copy text

EXAMPLE 11

KEYBOARD INPUT:

```
<TP1,G1/1,2,1,4>
<CP10> <CL12> <CC23> <CFTR>
TABLE USING TAB PROPORTIONAL<QC> <WR1> <QL>
<MA>
copy <QC>
copy <QC>
copy <QC>
<MA>
text <QC>
text <QC>
text <QC>
<QT>
<TR1,6> <QL>
```

TYPESET OUTPUT

TABLE USING TAB PROPORTIONAL

copy text	copy text	copy text	copy text

TAB STUB - This tab setup command is used for producing traditional statistical tables where the first column is larger than the others because it contains text while the balance of the table consists of several smaller equal columns containing numeric data. The command takes the form of <TBp,n> where p is the width of the first column in picas and points and n is a number of equal columns occupying the balance of the total measure. Assume a total measure of 26 picas, the first column of the table is 8 picas with the balance of the measure divided equally among 6 columns. The tab setup will be: <CC26> <TB8,6> where 8 is the 8 pica first column and 6 refers to the 6 equally spaced columns.
Now let's assume a 6 point gutter with automatic vertical rules of weight 1 in each one. For the purpose of this example, we will decrease the total measure to 22 picas. The tab set command will be: <CC22> <TB8,G.6/1,4> where the total measure will be divided into one column of 8 picas and 4 more columns of 3 picas each with a 6 point gutter and a centered vertical rule of weight 1 between each pair of columns. *See example 12.*

EXAMPLE 12

KEYBOARD INPUT:

<CP10> <CL12> <CC22> <CFTR>
<TB8,G.6/1,4>
TABLE USING TAB STUB COMMAND<QC> <WR1> <QL>
<MA> Stub Column Text <QL>
123<QC>
123<QC>
123<QC>
123<QC>
<MA> Second Line of Stub <QL>
123<QC>
123<QC>
123<QC>
123<QC>
<MA> Third Line of Stub <QL>
123<QC>
123<QC>
123<QC>
123<QC>
<QT>
<TR1,6> <QL>

TYPESET OUTPUT:

TABLE USING TAB STUB COMMAND

Stub Column Text	123	123	123	123
Second Line of Stub	123	123	123	123
Third Line of Stub	123	123	123	123

TAB TEXT - This tab set command is used as a floating tab marker on the horizontal measure based on its position relative to text material on the line. For example, the <TT> command may be used to position tab markers based on the longest word or block of characters in each column with the extra space on the total measure divided equally between columns. The tabs may be established in a no flash <NF> line of copy followed by a zero lead <ZL> to inhibit vertical escape before the body of the table is set. Assume a three column table with the following data in column one: THIS IS A WONDERFUL, the following data in column two: EXAMPLE OF THE TAB, and the following data in column three: SET COMMAND TAB TEXT. With the longest word in each column being WONDERFUL, EXAMPLE and COMMAND, set a no flash tab set line as:

<NF> WONDERFUL <QM> <TT> EXAMPLE <QM> <TT> COMMAND <QM> <QL> <ZL>

Let's also assume automatic vertical rules are required between each column of weight 1 using 3 pts as a gutter value (enough to contain the rule). The set up would be:

<NF>WONDERFUL<QM> <TTG.3/1> EXAMPLE<QM> <TTG.3/1> COMMAND<QM> <QL> <ZL>

See example 13.

EXAMPLE 13

KEYBOARD INPUT:

<CP10> <CL12> <CC20> <CFTR>
<NF> WONDERFUL <QM> <TTG.3/1> EXAMPLE <QM>
<TTG.3/1> COMMAND <QM> <QL>
<ZL>
TABLE USING TAB TEXT COMMAND<QC>
<WR1> <QL>
<MA>
THIS <QC>
EXAMPLE <QC>
SET <QC>
<MA>
IS <QC>
OF <QC>
COMMAND <QC>
<MA>
A <QC>
THE <QC>
TAB <QC>
<MA>
WONDERFUL<QC>
TAB <QC>
TEXT <QC>
<QT>
<TR1,6> <QL>

TYPESET OUTPUT:

TABLE USING TAB TEXT COMMAND

THIS	EXAMPLE	SET
IS	OF	COMMAND
A	THE	TAB
WONDERFUL	TAB	TEXT

ADDITIONAL NOTES ON SETTING TABS - If you have studied the previous examples on the various methods of constructing tabular setups, you will have noticed the use of a few additional commands. The *SET HORIZONTAL RULE* command <WRx> used a value for x that was equal to the weight of the automatic vertical rule requested in the tab setup command. This was done merely to make the horizontal rules equal in weight to the automatic vertical rules. The *MARK TAB BEGINNING* command <MA> is used at the beginning of each horizontal line of text that is to fall into the various tab columns across the measure. A *QUAD* code is used to indicate how each entry is to set in the column plus it advances the following text entry into the next column. If no data is to appear in a column, an additional *QUAD* code will skip that column and advance to the next. You may also have noticed that each example set the data horizontally, that is, the first line in column one followed by the first line in column two, followed by the first line in column three, and so on. You may also set the data vertically using a *POINT MARK* <PM> command at the beginning of the first line in the first column, enter all the data in column one followed by a *QUAD* code to get to the next column and a *RETURN MARK* <RM> to return to the first line of the column on the same horizontal position of the first line in column one.
For example, lets rekey the table in example 13 using the *TAB TEXT* command setup and note the changes in bold in *example 14*.

EXAMPLE 14

KEYBOARD INPUT:

<CP10> <CL12> <CC20> <CFTR> <**RC**>
<NF> WONDERFUL <QM> <TTG.3/1> EXAMPLE <QM>
<TTG.3/1> COMMAND <QM> <QL>
<ZL>
TABLE USING TAB TEXT COMMAND<QC> <WR1> <QL>
<MA><**PM**> **THIS** <**EP**> **IS** <**EP**> **A** <**EP**> **WONDERFUL** <**QC**>
<**RM**> **EXAMPLE** <**EP**> **OF** <**EP**> **THE** <**EP**> **TAB** <**QC**>
<**RM**> **SET** <**EP**> **COMMAND** <**EP**> **TAB** <**EP**> **TEXT** <**QC**>
<QT>
<TR1,6> <QL>

TYPESET OUTPUT:

TABLE USING TAB TEXT COMMAND

THIS	EXAMPLE	SET
IS	OF	COMMAND
A	THE	TAB
WONDERFUL	TAB	TEXT

As you can see, the same results were achieved, only the material was set in logical reading order rather than logical typeset order without any reverse lead. Actually, whichever method is used to keyboard this example, the program will compose it in such a way that the output device will not need to physically reverse the output material. The method of input is left to the operator's choice.

You will also notice in this last example that *END PARAGRAPH* commands <EP> were used to break lines in the column without advancing to the next column until the *QUAD* code <QC> was keyed. The <EP> was used to force break the line after each word. Without the <EP>s, the copy would have broken as the program decided what would actually fit within the column width. Try one without using the <EP>s, TYPE F1 for H&J, and see if the text in each column breaks differently.

The *QUIT TABULAR* command <QT> was also used in each of the tab examples to end a tab line and restore the set to full column measure for the instruction of setting the last horizontal rule the full width of the table without regard to the tab column markers. This code is technically required at the end of each tab line before beginning the next but the <MA> automatically implies a <QT> code, so the <QT> is only necessary at the end of the tabular material or to temporarily exit the influence of the tab setup command.

The *TAB RULE* command <TR1,6> following the last <MA> code in each example was necessary to advance the automatic vertical rules sufficiently before setting the bottom horizontal rule of the table. The vertical rules will only set automatically while the data is under the influence of the tab set command indicated by the <MA> commands, or with a tab rule command.

The *JUMPTAB* command <JTx> where x is a number of additional columns is used to temporarily combine the current defined tab column with a number of following columns for the setting of straddle material. The condition is terminated by a *QUAD* code and also temporarily suspends the setting of automatic vertical rules. Note the use of *JUMPTAB in example 15* to combine the middle three columns for a straddle head and combine all columns for a horizontal rule.

EXAMPLE 15

KEYBOARD INPUT:

```
<CP10> <CL12> <CFTR> <CC24> <TN5,G1/1>
THIS IS A TABLE USING JUMPTAB<QC>
<WR1,4> <QL>
<MA>
HEAD<QC>
<JT1>
STRADDLE<QC>
HEAD<QC>
HEAD<QC>
<QT>
<TR1,4> <QL>
<MA>
TEXT<QC>
TEXT<QC>
TEXT<QC>
TEXT<QC>
TEXT<QC>
<MA>
COPY<QC>
COPY<QC>
COPY<QC>
COPY<QC>
COPY<QC>
<QT>
<TR1,4> <QL>
```

TYPESET OUTPUT

THIS IS A TABLE USING JUMPTAB

HEAD	STRADDLE		HEAD	HEAD
TEXT COPY	TEXT COPY	TEXT COPY	TEXT COPY	TEXT COPY

VERTICAL ESCAPEMENT

MagnaType offers several command codes to execute a vertical movement within a job. However, regardless of the vertical order in which typeset characters are input, the software reverse lead feature of the composition program arranges the output of data to the typesetter in logical vertical order so the output device never physically reverses the movement of the photomedia within.

LEADING: forward vertical advance within a job is executed on a line by line basis depending on the current leading <CLx> command in effect. Each new line of type in a job will lead that amount until the <CLx> command is changed.
Reverse lead <RVx> is an immediate shift up of the current baseline on which type is being set and extra lead <ELx> is an immediate shift down of the current baseline.

If *AUTOLEADING* is set as a standard in the *SYSTEM STANDARDS* and no change leading command is specified following a change pointsize, the amount of lead from one line to the next is automatically calculated based on 1/3 the pointsize above plus 2/3 the pointsize below. The increment autolead command <IAx> specifies x as an amount to be added to this autolead value calculated.

POINT MARK: with the <PMx> codes, the program has made available to the user a set of commands that allows the marking and returning to various vertical depth points within a job. *POINT MARK* placed in the text stream will load into memory the current baseline depth of the previously set character. In other words, if the code appears at the beginning of a line prior to any typeset characters, the <PMx> will remember the previous baseline. If the code appears on a line somewhere after the first typeset character, the current baseline will be remembered. Six different *POINT MARKS* may be held in memory at any given time.

RETURN MARK: <RMx> placed in the text indicates that the following typeset characters will set on a baseline positioned below the *POINT MARK* of same x by an amount equal to the current lead in effect. In other words, if <PM2> is holding in memory a baseline depth of 16 picas from the top of the job and the current lead is 13 pts when <RM2> is encountered, the following characters will set on a baseline at 17 picas and 1 point from the top of the job. The horizontal position of the typeset characters is controlled by the current indent and column measure commands in effect.

Let's assume the operator desires that the following copy be set at the same baseline depth as the <PM2>, the *RETURN MARK* command followed by a zero lead command <RM2> <ZL> would position the following text at 16 picas and the current lead of 13 pts would not take effect until the next line wrap or quad code.

POINT MAXIMUM: <PXn> placed in the text will hold in memory the current baseline depth as a maximum for <PMx> where x = n and will only be replaced in memory by another <PXn> of same n if it is on a baseline of greater depth. This command may be used by the operator to keep track of a maximum depth which may later be returned to for setting material below that maximum point.

RETURN MAXIMUM: <RXn> placed in the text stream is an instruction that the following material is to be typeset beyond the depth of <PXn> of same n. The <RXn> compares the current baseline depth with that baseline depth held in memory for <PXn> and the following copy will set on a baseline postioned below the remembered maximum depth by an amount equal to the current lead.

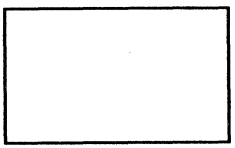
CLEAR MAXIMUM <XM>: placed in the text will compare the current baseline depth to all six *POINT MAXIMUMS* that may be held in memory and if it is less than the greatest *POINT MAXIMUM*, the following copy will set on a baseline positioned below that greatest *POINT MAXIMUM* by an amount equal to the current lead. If the current baseline of <XM> is greater than the greatest *POINT MAXIMUM*, the following copy will continue to set on the current baseline. In any event, <XM> will remove all current *POINT MAXIMUMS* in memory. *See example 16.*

EXAMPLE 16

KEYBOARD INPUT:

<CP9> <CL10> <CC23> <CFTR> <IT0,16> <PM1>
<BX7,4.2> <QL>
<PM2> <RM1> <IT8,8>
This is a block of data starting at the top of column two and will continue to a depth beyond the bottom of the box at the top of column one. We will return to that point and to this point. <EP>
<PM3> <RM1> <IT16,0>
This is a block of data starting at the top of column three and will continue to a point higher than the end of data in column two. <EP>
<PM4> <RM3> <EL4> <IT8,0>
<BX15,3.4> <QL>
<RM2> <IT0,16>
This is text following the box in column one and should extent beyond top of box in column two. Now return to end of text in column three. <EP>
<RM4> <IT16,0>
END COPY <QC>
<XM> <IT0,0> <WR> <QL>

TYPESET OUTPUT:

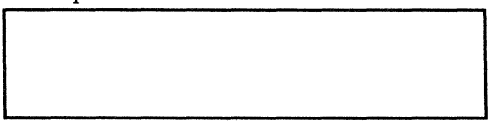


This is text following the box in column one and should extent beyond top of box in column two. Now return to end of text in column three.

This is a block of data starting at the top of column two and will continue to a depth beyond the bottom of the box at the top of column one. We will return to that point and to this point.

This is a block of data starting at the top of column three and will continue to a point higher than the end of data in column two.

END COPY



DELAY LINES/MEASURE: these commands are used within formats and are considered temporary or conditional merge codes. *DELAY LINES* <DLx> where x is a number of text lines. It may be used at a point within a format which indicates to use parameters in the format up to the point of the *DELAY LINES* code, return to the text stream and process text until a number of lines specified in <DLx> is achieved, and then return to the format to consider the parameters following the *DELAY LINES* command.

Similarly, *DELAY MEASURE* <DMx>, where x is a measured depth, may be used within the format to indicate a merge from the format to the text stream until a specified depth is achieved and then return to format for further instructions. The simplest application of these commands is to code a specific indent amount for a cut-out over a number of lines or specific depth and then return the text to full measure. A more complex application would be to establish depths and indent changes for creating a multicolumn page layout and allow the text to stream out by format control. *See example 17.*

VERTICAL RULES: The construction of vertical rules in a job is achieved by first defining the start of the rule vertically positioning a <DVx> command (where x is a rule number from 1 to 40) on the current baseline and horizontally positioning it based on the other characters or codes on the current baseline. When the vertical depth of the rule is reached in the job, a <SVx,y,z> where x equals the rule number as in the <DVx>, y equals the rule weight, and z equals the amount of horizontal move to the left to properly allow the corners of a right vertical and a horizontal rule to meet. The amount of z will depend on the way the output device draws the vertical rules and experimentation maybe required to discover the proper amount of adjustment necessary. *See example 18.*

EXAMPLE 17

KEYBOARD INPUT:

<CP10> <CL12> <CC20>
<SF1> <CFTR> <DL3> <IT4,4> <DL3> <IT2,2> <CFTB> <DL2>
<IT0,0> <CFTI> <DL2> <IT2,2> <CFTR> <EF>
<UF1>
This is a demonstration of how delay line codes in a format may be used to change conditions of the typeset material based on the number of lines of text set. This is a demonstration of how delay line codes in a format may be used to change conditions of the typeset material based on the number of lines of text set. This is a demonstration of how delay line codes in a format may be used to change conditions of the typeset material based on the number of lines of text set. This is a demonstration of how delay line codes in a format may be used. <EP>

TYPESET OUTPUT:

This is a demonstration of how delay line codes in a format may be used to change conditions of the typeset material based on the number of lines of text set. This is a demonstration of how delay line codes in a format may be used to change conditions of the typeset **material based on the number of lines of text set. This is a demonstration of how delay line** *codes in a format may be used to change conditions of the typeset material based on the number of lines of text set.* This is a demonstration of how delay line codes in a format may be used.

NOTE: <CFTR> is times roman
<CFTI> is times italic
<CFTB> is times bold

EXAMPLE 18

KEYBOARD INPUT:

<CP9> <CL10> <CC15> <CFTR>
<DV1> <WR> <DV2> <QL>
<IP1,1>
This is a block of text that will be enclosed in a ruled box that will expand vertically to contain any amount of type keyed by the operator. When the text is completed with a quad code, a zero lead command followed by an extra lead approximately one third the amount of the pointsize will position the bottom rule of the box such that the copy will appear optically centered. <EP>
<ZL> <EL3> <WR> <SV1> <SV2> <QL>
<EL12><DV1> <WR2> <DV2> <QL>
<IP1,1>
This is a block of text that will be enclosed in a ruled box that will expand vertically to contain any amount of type keyed by the operator. When the text is completed with a quad code, a zero lead command followed by an extra lead approximately one third the amount of the pointsize will position the bottom rule of the box such that the copy will appear optically centered. <EP>
<ZL> <EL3> <WR2> <SV1,2> <SV2,2> <QL>
<EL12>

TYPESET OUTPUT:

This is a block of text that will be enclosed in a ruled box that will expand vertically to contain any amount of type keyed by the operator. When the text is completed with a quad code, a zero lead command followed by an extra lead approximately one third the amount of the pointsize will position the bottom rule of the box such that the copy will appear optically centered.

This is a block of text that will be enclosed in a ruled box that will expand vertically to contain any amount of type keyed by the operator. When the text is completed with a quad code, a zero lead command followed by an extra lead approximately one third the amount of the pointsize will position the bottom rule of the box such that the copy will appear optically centered.

CONCLUSION

Hopefully, these applications examples will give you enough practical information to begin full utilization of the MAGNATYPE software program. We strongly recommend that you experiment with these examples, combine some of them, and become familiar with all the codes available. Remember, there may be more than one way to achieve the required results, the choice is up to the individual operator based on with what he or she feels most comfortable.

APPENDIX A
ERROR MESSAGES

COMMAND indicates that an illegal command code has been generated or a legal command has been altered.

FORMAT indicates that an illegal format string has been produced, perhaps a looping format instruction. This may also indicate that a format has been requested that doesn't exist.

DICT a dictionary has been requested that doesn't exist, or too many characters have been input to the temporary exception dictionary with the ED command.

FONT a font has been requested that doesn't exist.

SIZE an illegal pointsize has been requested.

COLUMN an illegal column measure has been requested. Column Measure exceeds output device capability. XY coordinates may have forced current measure off the right margin.

SPACEBAND indicates an illegal or illogical value in the change spaceband command. Usually will result from minimun being larger than optimum or maximum being less than optimum.

RANGE indicates a pointsize/setwidth combination exceeds limitation of the output device.

LEAD indicates an illegal lead command forcing text to set above the top of the job.

INDENT an indent has been specified in excess of the available measure for typeset copy.

VRULE vertical rule error indicates a vertical rule has been incorrectly defined or requested, problem could be in the <SV or <DV command or an illegal rule weight specified for the automatic gutters.

RAGZONE error usually indicates the change ragged command has a value that exceeds the allowable measure or the hotzone is specified larger than the ragged zone.

WRAP error indicates an amount of characters and codes on one line of typeset output that exceeds a maximum allowable count of approximately 500.

MEASURE indicates that the requested column width exceeds the limit of the output device entered in the system specifications.

$PI dollar sign pi character requested that has not been defined in the PI TABLE of the font module.

CHAR a character has been requested that doesn't exist in the current font.

#TABS number of tabular columns requested exceeds limit of the program.

TAB SPEC indicates a tab set instruction with incorrect specs or markers out of order.

#HYPHEN indicates that H&J has allowed a number of hyphens in a row that exceeds the user specifications.

POINTMARK indicates a pointmark has been requested that is not legal, or a return mark has been requested where no point mark exists.

REVERSAL a reverse leading instruction has been requested that moves above the start of job.

THICKNESS indicates an illegal rule weight has been requested.

WHITESPACE indicates a set whitespace has been requested that exceeds the limit of the output device in conjunction with the current track value in effect.

GUTTER OVERFLOW indicates that a specified gutter value exceeds the width of a tab column.

APPENDIX B

MagnaType
QUICK REFERENCE

After booting up the program, the first screen asks for the operator to enter two initials followed by a RETURN. The primary menu appears. Enter selection number followed by RETURN.

DISPLAY THE JOB DIRECTORIES allows the operator to view part or all of any job list in the system. The four options available when this selection is requested are:

> JOBS ON DISK FOR EDITING will display the list of jobs residing on the disk. Here, a number of tasks may be performed on a job as well as calling it up to the screen for proofing.
>
> JOBS IN PREPARATION FOR OUTPUT will display jobs waiting in line to compose for the output device.
>
> JOBS READY TO BE OUTPUT will display the list of jobs composed and ready for the typesetter.
>
> JOBS THAT REQUIRE ATTENTION will display a list of jobs that have been brought into the system from outside sources.

SELECT AN EXISTING JOB allows the operator to perform a number of tasks on a specific job.

CREATE A NEW JOB allows the operator to create a new job/take for storage in the system.

SYSTEM MANAGEMENT TASKS allows the operator to sign off and allow a new operator to sign onto the system; to access the system standards and defaults screens for processing or reference; to enter the mnemonic/format screen for setting up mnemonic format codes; to enter global format options; to work with the hyphenation dictionaries; or to work with the font information.

TRANSFER JOBS allows the operator to import and export job/takes to outside media as well as import composed jobs ready for output.

A JOB NUMBER may be up to 8 alpha-numeric characters. Any job may have associated with it up to 999 individually numbered *TAKES* plus a format take called F.

TO CREATE A JOB, at the main menu type 3 RETURN. Type a job number followed by a hyphen followed by a take number, RETURN. Type a job description or name up to 16 ASCII keyboard characters for identification purposes, RETURN. Now begin to keyboard job text and coding.

TO GET OUT OF A JOB, type ESCape. The *EDITING COMPLETE* menu will appear with several options. The current version could be copied to another *JOB NUMBER* with a new name or a new name could be given to the current *JOB NUMBER*. The job could be sent to a line printer for hard copy proof, to prepare code for the typesetter, or to the disk and stored for later recall. The *EDITING COMPLETE* menu also gives the option to return immediately to the job or ignore all that has just been done to the job.

THE FUNCTION KEYS

F1 - H&J
SHFT/F1 - stores current take, allows access to another
F2 - suppress or display line endings
SHFT/F2 - suppress or display command codes
F3 - defines end of block
SHFT/F3 - defines start of block
F4 - insert a block
SHFT/F4 - insert a block previously inserted with F4
F5 - word delete
SHFT/F5 - line delete
F6 - display blocks & programmable keys
SHTF/F6 - define programmable keys
F7 - change case
SHFT/F7 - next paragraph
F8 - display diamond
SHFT/F8 - display format
F9 - continue search
SHFT/F9 - define search
F10 - display help screens

THE ALT KEYS

C Quad Center	T Thin Space
L Quad Left	U Unit Space
R Quad Right	S Standard Font
P End Paragraph	I Italic Font
J Force Justify	B Bold Font
M Em Space	O Other Font
N En Space	X Merge Copy
F Figure Space	H Discretionary Hyphen

OTHER KEYS

PLUS is open command
MINUS is close command
BACK SLASH is PI precedent
TAB is down arrow for inferior
SHFT/TAB is up arrow for superior
SHFT/6 is force close quote
OPEN QUOTE is force open quote

TO OUTPUT JOB

Assuming the job been *prepared for output*, it will appear in the list of *JOBS READY FOR OUTPUT* in the *JOB DIRECTORIES* screen. Move the cursor to that list and TYPE RETURN. The jobs ready for output will be displayed as selected in the menu window. Move cursor to job to be output, TYPE T and RETURN next to each job to be sent to the output device, and then TYPE ESCape to begin the transmission. A message will appear at the bottom of the screen to indicate when the transmission is complete to the typesetter.

BOOTING THE PROGRAM

Once you have established the MagnaType software in a DOS directory on the hard disk of the PC, go into the directory and type the first five characters of your xCOMP.EXE program. For example, if you have an L/202 (CCOMP.EXE), type:

CCOMP (RETURN)

Wait for the program to load into memory (this should take only a few seconds). When the first screen appears, enter your initials and the *PRIMARY MENU* screen will appear. Then go to chapter 1 of this manual and begin.

APPENDIX C
GLOSSARY

A

ALPHA CHARACTER - any letter of the alphabet from a to z

ALT KEY - a precedent key used in combination with another key to perform a specific function

AUTO FRACTION - see BUILD FRACTIONS

AUTOKERN WIDTHS - amount of space to be removed from between a specific pair of characters, see KERNING

AUTO LEADING - the feature that prevents a line of type in a larger pointsize from printing over a portion of the type in the line above

AUTOMATIC NUMBER COUNTER - see NUMBER COUNTER

AUTO QUOTES - feature that automatically turns a generic quote mark (such as on a typewriter) into an open or a close quote

B

BASELINE - the line on which the type rests; from where leading is calculated

BLOCKS - a defined body of text and/or command codes to be remembered for later or repeated insert

BOLD FONT - the third font in a font family assignment, usually the BOLD version of a specific typeface

BOOT-UP - the process of putting a program into computer memory for operation

BUILD FRACTIONS - the feature that automatically produces a numerator in a smaller pointsize, raised from the baseline followed by a fraction bar followed by a denominator in a smaller pointsize on the baseline

CASE - distinction between capital letters and small letters

CHANGE FONT ID - feature of the FONT INFORMATION which allows the user to change the ID of an existing font

CHARACTER - any single symbol that is available to typeset including alpha characters and numeric characters

CHARACTER WIDTH - see WIDTH VALUE

CLOSE QUOTE - like an apostrophe, used to end a quotation

CODE - a special combination of characters within a job to indicate a function or operation

COLUMN WIDTH - the measurement indicating how wide a block or line of type is to be set

COMBINATION COMMAND - a streamlined form for inputting a set of commands, for example, pointsize, leading, column measure, and font indicated as seperate commands would look like this: <CP10> <CL12> <CL25> <CFTR>, where the combination command would look like this: <CP10/12x25,TR>

COMMAND CODE - see CODE

COMPOSE - the process of converting the H&Jed version of a job/take into the output language of the typesetter

CONTROL KEY - precedent key used in combinaton with anothe in effect, and vertically justify it to 20 picas. Last line hyphenation is allowed.

CURSOR - a position indicated on the screen by a flashing underbar controlled by the arrow keys on the numeric keypad

D

DEFAULT VALUE - a value used if no other value is specified with a code within the job

DELETE - to remove a specified character, word, or line of type

DELAY LINES/MEASURE - a temporary exit from a format to the text stream until a number of lines is set or a specified depth is reached before returning back to format control

DIAMOND - symbol generated on screen in a job to indicate presence of a code when codes are suppressed with SHIFT/F2

DISCRETIONARY HYPHEN - a manually keyed indication of how to break a specific word if it falls at the end of a line

DISPLAY DIRECTORIES - menu selection that allows the selection of lists to be displayed on the screen

DOS - see MS-DOS

DOWNLOADING - process by which values from one media are moved over to another media

E

EDIT - process in which changes to text or codes are implemented

EDITING COMPLETE - the menu accessed when escaping a job/take on the system

ELECTRONIC ITALIC - process in which a roman type face is electronically slanted to produce an italic set

EM SPACE - a relative amount of horizontal space on a line equal in points to the current pointsize of type

END PARAGRAPH - a code used at the end of a paragraph that has a quadding effect equal to the ragged mode in effect, or if justified type, equal to a quad left

EN LEADER - a series of periods or full points that are spaced one en space apart

EN SPACE - a relative amount of horizontal space on a line equal in points to one half the current pointsize of type

ENTER KEY - used to indicate an instruction or menu selection has been given to the program, sometimes referred to as the RETURN key

ESCAPE - in typeset material, escape is a horizontal movement on a line of type, on the keyboard, escape is used to exit a job or move through the menu screens

F

FIGURE SPACE - a relative amount of horizontal space in a typeset line traditionally equal to the width of a numeric character in the current pointsize

FIXED SPACE - an amount of space that is fixed in width as opposed to a word space which varies to justify a line of type

FLASH - used to describe the actual exposure of a character on the typesetting paper

FLOPPY DISK - the media on which data is recorded or read into a computer, a circular piece of magnetic material contained in a square envelope

FONT - a collection of typographic characters and symbols in a specific style of artwork

FONT DIRECTORY - a list of all the typefaces loaded in the program

FONT FAMILY - a collection of four fonts assigned to a family name that may later be referred to as STANDARD, ITALIC, BOLD, or OTHER with codes or ALT KEY characters

FONT ID - abbreviated form by which fonts are specified within a job

FONT INFORMATION - the portion of the program that controls the justification half of the H&J process

FONT NAME - the typographic name given to a font entry in the program that allows easy reference from the directory

FONT NUMBER - the number used to address a specific font loaded on the output device (typesetter).

FONT INFORMATION TABLE - selection in the FONT INFORMATION that allows access to a specific font and its associated data, such as width table, kern table, etc.

FORMAT - a collection of text and/or command codes that may be accessed by a request to use the format rather than rekeying the contents of the format

FRACTION BAR - similar to a slash, the character in a fraction that seperates numerator from denominator

FRACTION HEIGHT - percentage of current pointsize to be used in generating numerator and denominator in BUILD FRACTION feature

FRACTION WIDTH - percentage of current pointsize to be used as width when generating numerator and denominator in BUILD FRACTION feature

FUNCTION KEYS - set of ten keys to the left of the keyboard dedicated to specific functions by the software

G

GLOBAL FORMAT - format that is defined in the program for use in any job on the disk

GUTTER - the space between columns of typeset material in a tab set, may have a vertical rule

H

H&J - shorthand way of referring to the process where a body of text is hyphenated and justified

HANGING PUNCTUATION INDENT - amount of space preceding the left typeset margin where punctuation falling at the beginning of a line will set

HORIZONTAL ESCAPEMENT - move to the right on a typeset line

HOT ZONE - a specification in the control ragged zone feature which defines the range in which a line must end

HYPHENATE WIDOW - when a portion of the last word in a paragraph appears on a line by itself

I

INDENT HANG - when the first line of a paragraph is set the full measure and each subsequent line is indented by a specified amount of "hang"

INDENT TEXT - an indent amount which effectively moves the right and/or left margin from which all other indents are measured

INFERIOR HEIGHT - a percentage of the current pointsize to be used when generating inferior characters

INFERIOR WIDTH - the percentage of the current pointsize ot be used when generating inferior character widths

INPUT - the act of entering data into a system; the actual data that has been entered into the system

ITALIC FONT - a version of a typeface that has the characters set slanted to the right

INTERGALLEY LEADING - the vertical measure amount that seperates one galley of output from the next

INTER-WORD SPACING - the range in which the amount of space occuring between words on a line may vary

J

JOB HEADER - the block of information that describes a job, contains the number, operator initials, revision level and date, job size by characters, etc.

JOB/TAKE - smallest individually numbered block of material in the system, job number may be up to 8 alpha/numerics with each job having up to 999 seperate *takes*

JUMP TAB - a command issued in tabular mode that indicates following material is to occupy current column plus a number of additional columns

JUSTIFICATION - the process in which interword spacing is adjusted so the material contained on a line of type fills the required measure

JUSTIFY LINE - a code which forces a line of type to fill the measure by expanding the interword spacing beyond the specified limit, or by letterspacing words on the line

K

KERN CHARACTER - one of a pair of characters that is either forced closer together or farther apart

KERNING - the reduction or addition of space between two specific characters

KERN NUMERIC ONE - reduction of the space around a number 1 because it is narrower than the other numbers, generally not used in tabular material where numbers in a column should align

KEYBOARD - the collection of data entry keys that sits in front of the VDT (video display terminal)

KEYSTROKES - refers to the act of striking keys to input data

L

LEADING - amount of vertical space from one baseline of type to another

LEFT ARROW - a key on the numeric keypad that moves the cursor on the screen one character space to the left

LIGATURE CHARACTER - a single character that has the artwork of two or three characters, used in place of the two or three characters such as fi or ffi

LINE PRINTER - a monotype printing device used to produce hard copy of material in the system

M

MARK TAB - command used to indicate beginning of a horizontal line of type in a tabular construction

MAXIMUM WIDTH - largest horizontal amount of space for typeset material on output

MENU - a VDT screen that gives the user a choice of functions or operations to perform

MEASURE - usually indicates the width of the image area in which type must appear

MERGE COPY - a code that indicates program control is to switch over from format to text stream or text stream to format

MODE - a preferred way of doing something, in the system specifications, modes are usually stated as yes or no

MONOTYPE - when all printed characters occupy the same horizontal width space, such as on a typewriter

MS-DOS - the part of the program that controls the operation of the computer with the disk drive, referred to as DOS

MUT - old expression to indicate an EM SPACE as opposed to an EN SPACE (nut)

N

NESTING - a way of linking formats in that before one format is completed, another is called into use

NUMBER LOCK - toggle key on numeric keypad to shift the character output from one set to another

NUMERIC CHARACTER - zero (0) through nine (9)

NUMERIC HYPHENATION - breaking a string of numbers over a line ending

NUT - old expression to indicate an EN SPACE as opposed to an EM SPACE (mut)

NUMBER COUNTER - automatic feature of the system that keeps track of a count, prints a number on command and increments the counter for the next print

O

OPEN QUOTE - character indicating the beginning of a quotation

OTHER FONT - fourth font in a font family assignment, usually the *BOLD ITALIC* version of a typeface

OVERRIDE - an immediate change in a specification that may have been previously defined

OVERSTRIKING - the act of typing in new data on the screen over old data already displayed

P

PGDN - the page down key on the numeric keypad, used to scroll into the depth of the screen text

PGUP - the page up key on the numeric keypad, used to scroll up into the screen text

PICAS - a measure used by printers, slightly less than an inch is six picas

PI FONT - a collection of special characters other than what usually appears on a standard on a standard alpha/numeric font

PI TABLE - a cross-reference of mnemonics to specific characters on specific fonts, allows the user to key in a short mnemonic rather than changing font to special characters and then back to text font

PI TABLE OVERRIDES - table for each font to modify standard pi table mnemonics

POINTMARK - code used to remember a vertical position in a job that can later be returned to with another code (RETURN MARK)

POINTSIZE - indicates the size of the typeset characters

PRECEDENT KEYS - keys used in combination with another key to perform a function or task

PRIMARY LANGUAGE - the language logic used by the dictionary to perform hyphenation breaks unless specified otherwise

PRINT FONT - feature that allows a hard copy version of font information screens to be produced for study or filing

PRINT SCREEN - a key used to perform a DOS operation where the contents of the VDT screen is printed to the lineprinter

POINTS - a measure used by printers, there are twelve points to a pica

PROGRAMMABLE KEYS - allows the storage of a string of keystrokes into a single key for easier repeated entries

PROOF - displaying a job to the screen without first passing the text through the H&J process, shows the contents of the job in one continuous stream

Q

QUAD CENTER - an end of line code that indicates copy is to be centered within the specified measure

QUAD LEFT - and end of line code that indicates copy is to set flush against the left margin with excess space pushed to the right

QUAD RIGHT - an end of line code that indicates copy is to set flush against the right margin with excess space pushed to the left

QUOTES - see OPEN QUOTE and CLOSE QUOTE

R

RAGGED - refers to a block of text that is set unjustisfied, that is each line begins, ends, or both in a different position on the measure; ragged left indicates unjustified lines begin in different positions and end against the right margin, ragged right indicates lines begin on the left margin and end at different positions, and ragged center indicates lines begin and end in different positions with copy centered on specified measure

RAGGED ZONE - specified horizontal area at the margin where a ragged line must begin and/or end

RELATIVE UNITS - a method of measure and comparing character widths with a value that varies depending on the pointsize of type

RETURN - key used to indicate to the program that a screen menu choice has been made; sometimes called the *enter* key; returns cursor to beginning if next line during data entry or editing; another name for a *MERGE CODE*

RETURN MARK - code used to indicate following copy is to begin at same depth remembered by the previous POINT MARK

RIGHT ARROW - key on numeric key pad that moves cursor one character position to the right

ROMAN FACE - the standard artwork of a type face as opposed to italic, bold, or bold italic

ROOTS - the base portion of a word that may be shared by other forms of the word or other words

RULE THICKNESS - way of specifying the weight of a rule to be set, method varies from one output device to another

RULE WEIGHT - see RULE THICKNESS

S

SET WIDTH - method of describing the width of a character to be set that may be different from the pointsize of the character, used to indicate the set of expanded or condensed type

SHIFT - key used to indicate the set of upper case versions of characters or keys typed

SKEW - taking a specified indent value and spreading it equally over a number of lines where each line indent is greater than the previous line indent; may also create the opposite effect with a negative skew

SLUG - block of information ahead of a job that gives the number, name, operator initials, revision level, etc

SOFTWARE REVERSE LEAD - during the compose process, the typeset material of the job/take is sent to the output device in vertical order so the output device never actually needs to physically reverse the output media

SPACEBAND - a key used to generate a word space; refers to the space between words

STANDARD FONT - the first font in a font family assignment, usually the roman version of a typeface

STRING COMMAND - a method for keying a series of command codes in an abbreviated form

SUPERIOR - a character that is smaller than the current pointsize set on a higher baseline

SYSTEM STANDARDS - values and choices that make one system setup unique based on the user's preferences

T

TAB KEY - key used in the MAGNA program to type in a superior or inferior character

TAB RULE - allows the setting of a horizontal rule in a tab construction where the vertical rules will automatically connect to it

TAB STUB - a way of specifying a tabular construction where the first variable describes the width of column one followed by the second variable indicating a number of equal columns spread over the balance of the margin

TAKE NUMBER - division of a job, -000 through -999 and -F

TEXT STREAM - collection of data entered into a job that will become typeset material

THIN SPACE - traditionally one third the width of an EM SPACE; a fixed space relative to the pointsize

TRACKING - an overall reduction or expansion of the space between every character

TYPEFACE - a style of printed characters

U

UNDERSCORE - a rule appearing below characters to add emphasis

UNIT SPACE - a fixed space relative to the pointsize of type, user definable in fractions of an EM SPACE

V

VIDEO REVERSE - setting type characters in a negative form where what is normally black is white and normally white is black

W

WHITESPACE - expression used to indicate amounts of tracking

WIDTH VALUES - measurement for comparing the relative width of one character to another

X

X-Y COORDINATES - combination code that specifies a start point for a block of text by giving indent and depth from upper left corner of the job

Z

ZERO LEADING - code used to temporarily terminate the leading value in effect