GC26-3859-1
File No. S370-30

Systems

IBM 3800 Printing Subsystem Programmer's Guide for OS/VS2 SVS

IBM

Second Edition (April 1978)

This edition replaces the previous edition (numbered GC26-3859-0) and makes it obsolete.

This edition applies to Release 1.7 of OS/VS2 SVS and to any subsequent releases of that system unless otherwise indicated in new editions or technical newsletters. Corresponding information for OS/VS2 MVS and for OS/VS1 is in the *IBM 3800 Printing Subsystem Programmer's Guide,* GC26-3846. Significant changes are summarized under "Summary of Amendments" following the list of figures. Each technical change is marked by a vertical line to the left of the change.

Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest *IBM System/370 Bibliography,* GC20-0001, and the technical newsletters that amend the bibliography, to learn which editions and technical newsletters are applicable and current.

Requests for copies of IBM publications should be made to the IBM branch office that serves you.

Forms for reader's comments are provided at the back of the publication. If the forms have been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California 95150. All comments and suggestions become the property of IBM.

PREFACE

This book provides:

- Planning and conversion information for the IBM 3800 Printing Subsystem. The intended audience is data-processing managers, system planners, and system programmers planning to incorporate a 3800 into an OS/VS2 SVS system.
- Information on how to use the 3800. This is intended for application programmers and system programmers who will be using the 3800 Printing Subsystem.

There is no prerequisite reading for this publication, although a basic knowledge of computers and system printers is assumed.

Related Publications

The following publications are referred to in this book:

- *Forms Design Reference Guide for the IBM 3800 Printing Subsystem,* GA26-1633, which contains information on forms design and on designing and making or obtaining forms overlays.
- *IBM 3800 Printing Subsystem Character Design and Coding Form,* GX26-3713, which is a pad of forms for designing and coding characters for the 3800.
- *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information,* GC26-3858, which contains information on the programming and use of the 3800 under SVS. This book is arranged in chapters that supplement the various system library books that they pertain to. For example, the "JCL" chapter contains information to update *OS/VS JCL Services,* GT00-0141, and *OS/VS JCL Reference,* GT28-0618, with new and changed JCL parameters for the 3800.
- *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem Logic,* SY26-3854, which includes the program logic for the IEBIMAGE utility program and for other system changes to support the 3800.
- *OS/VS Service Aids,* GC28-0633-1 (obtained by ordering temporary order number GT28-0633-1), which tells how to use the AMASPZAP program, which may be needed when working with and maintaining the modules on SYS1.IMAGELIB.
- *OS/VS Utilities,* GC35-0005-1 or OS/VS2 SVS Utilities, GT35-0005-2, which tell how to use IEBUPDTE, IEHPROGM, and other utility programs needed when working with and maintaining the modules on SYS1.IMAGELIB.
- *Reference Manual for the IBM 3800 Printing Subsystem,* GA26-1635, which gives channel programming considerations for the 3800, and also the algorithm for calculating when a copy modification module is constructed such that it may cause a line overrun condition on the printer.

HASP Publications

If you are using HASP (the Houston Automatic Spooling Program) with your SVS system, you should have the following books and the technical newsletters for each of them:

- *OS/VS2 HASP II Version 4 User's Guide,* G27-0052
- *OS/VS2 HASP II Version 4 System Programmer's Guide,* GC27-6992
- *OS/VS2 HASP II Version 4 Operator's Guide,* GC27-6993

Notational Conventions

A uniform system of notation describes the format of the job control language and IEBIMAGE utility commands. This notation is not part of the language; it simply provides a basis for describing the structure of the commands.

The command-format illustrations in this book use the following conventions:

- Brackets, [ ], indicate an optional parameter.
- Braces, { }, indicate a choice of entry; unless a default is indicated, you must choose one of the entries.
- Items separated by a vertical bar, |, represent alternative items. No more than one of the items may be selected.
- An ellipsis, ..., indicates that multiple entries of the type immediately preceding the ellipsis are allowed.
- Other punctuation (parentheses, commas, apostrophes, etc.) must be entered as shown.
- **Boldface** type indicates the exact characters to be entered. Such items must be entered exactly as illustrated (in upper case).
- *Italic* type specifies fields to be supplied by the user.
- **Underscored** type indicates a default option. If the parameter is omitted, the underscored value is assumed.

CONTENTS

FIGURES

SUMMARY OF AMENDMENTS

April, 1978

OPTION OVERRUN

This revision describes the format and use of the OPTION OVERRUN statement in the IEBIMAGE program. The OPTION OVERRUN capability is provided in PTF# UY77715.

Miscellaneous technical and editorial changes have also been made.

INTRODUCTION

The IBM 3800 Printing Subsystem (Figure 1) is a nonimpact, high-speed, general-purpose system printer that uses an electrophotographic technique with a low-powered laser to print on single-ply computer output paper.

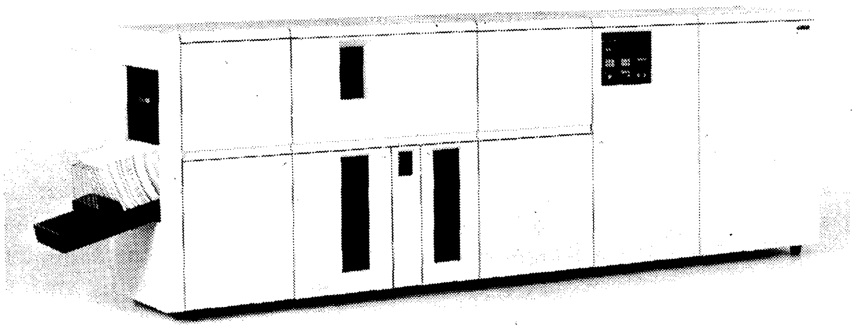


Figure 1. The IBM 3800 Printing Subsystem

The 3800 is designed to print faster and provide more features than current impact printers and to provide compatibility for user programs that have previously used impact printers. It also provides easy-to-use programming interfaces for its functions and features. Among these features are:

- Character generation storage is provided for printing 128 graphic characters (two character sets). As an option, character generation storage for an additional 127 characters is available. Printing speed is independent of the number of characters being used.
- Any two of 20 different character sets (including 10-, 12-, and 15-pitch and 15-pitch condensed sets) can be used, and these can be changed between data sets or intermixed on a page, without operator intervention. With the optional additional character generation storage, up to four character sets can be used when printing each data set.
- Graphic character modification allows the substitution or extension of graphic characters in an already-defined character set.
- Multiple copies are printed on single-ply paper, with every copy an original, eliminating the need for multiple-ply paper and subsequent deleaving.
- Copy modification permits printing of predefined data or suppressing the printing of selected data on one or more copies.
- The forms control buffer (FCB) controls the vertical format of pages, allowing printing at either 6, 8, or 12 lines per inch, or an intermix of any of these vertical line spacings on the same page.

- The optional Burster-Trimmer-Stacker bursts the fanfold output, trims 1/2 inch (12.7 mm) from each edge, and stacks the sheets in groups so that individual jobs can easily be identified and removed.
- Forms overlay permits printing of forms as data is being printed, thus reducing the need for preprinted forms.
- The mark form feature permits marking the horizontal perforations between pages to simplify job separation.

As long as there is sufficient data to print, the 3800 prints at a constant high speed. It can process, for example, 1000 11-inch long pages in approximately six minutes of continuous printing, regardless of the number of lines per page.

The 3800 Printing Subsystem operates under the OS/VS2 SVS operating system via a channel attachment, and can be added to a system configuration with an IBM System/370 Model 145, 148, 155-II, 158, 158-2 (in Japan), 158-3, 165-II, 168, or 168-3, or System/370 3031, 3032, or 3033 Processor. A 3800 Subsystem may be attached to a System/370 Model 135 or 138 via RPQ S00338.

The output writer controls the setup and use of the 3800 and can reassign the printing of data sets between impact printers and the 3800. If the printing requires special features unique to the 3800, the operating system ignores the JCL keywords specifying these features when the job is reassigned to an impact printer.

How the 3800 Prints

Data to be printed is transmitted one line at a time from the processor to the 3800 where it is stored in an internal page buffer. As each page is completed, it is exposed by the modulated beam of a low-powered laser onto the photoconductive surface of a rotating drum, thereby creating a latent image of the page to be printed. The latent image is coated with toner (a thermoplastic material impregnated with lampblack) and at the transfer station the toned image is transferred from the drum to the paper. The paper then passes through the fuser, which fuses the toned image into the paper. Meanwhile, the surface of the drum is cleaned and reconditioned for following exposures. If there is then sufficient data in the page buffer to print another page, printing continues without stopping the paper motion. Forms can be printed with the data by flashing the image of a forms overlay negative onto the drum.

Figure 2 shows the key elements in this process and the path that the paper takes from the continuous forms input station through to the continuous forms stacker or the optional Burster-Trimmer-Stacker.

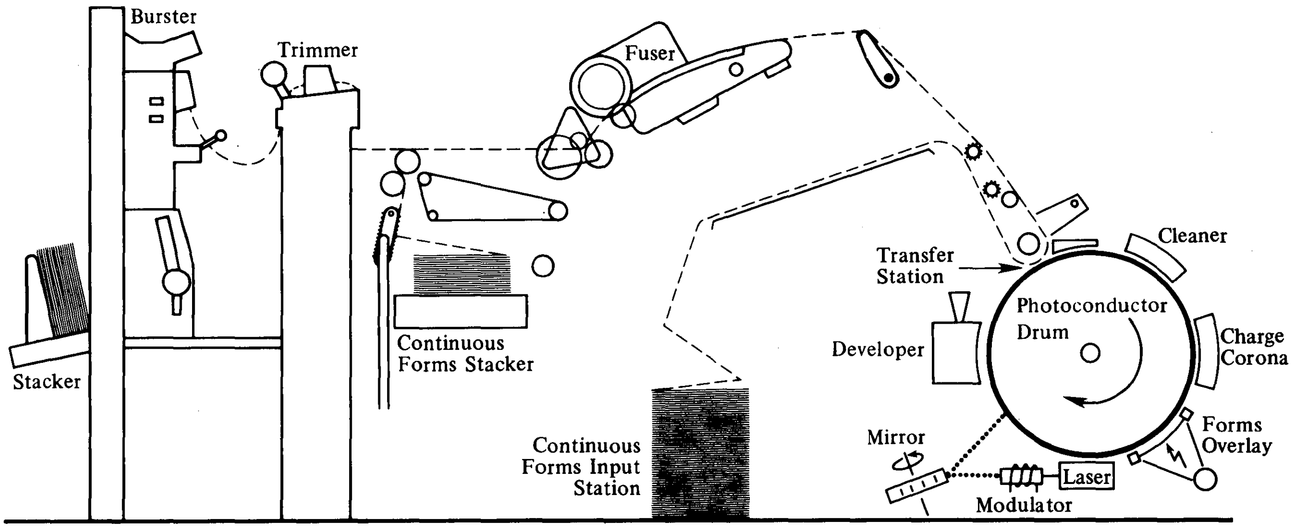


Figure 2. An Overview of How the 3800 Prints

To further understand how the 3800 prints, you should understand the terminology used and the relationship between the 8-bit character data code (usually an EBCDIC code) transmitted to the 3800 and the graphic character that prints. The following list gives some of the terms involved:

The Term and its Definition	What It Is in the 3800
character: * A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. A character is often in the form of a spatial arrangement of adjacent or connected strokes.	The scan pattern that modulates the laser beam exposing the graphic character onto the photoconductor.
character set: * A set of unique representations called characters, for example, the 26 letters of the English alphabet, 0 and 1 of the boolean alphabet, the set of signals in the Morse code alphabet, the 128 characters of the ASCII alphabet.	The scan patterns for each IBM-supplied set of a maximum of 64 graphic characters reside on a read-only flexible disk in the 3800. (In this book, only IBM-supplied character sets are referred to as "character sets.")
writable character generation module: A 64-position portion of the 3800's character generation storage that holds the scan elements of one character set. There are two WCGMs in the basic 3800, and optional added storage provides two more.	The WCGM is used for generating the printed characters from their scan patterns.
character arrangement: An arrangement composed of graphic characters from one or more modified or unmodified character sets.	Character arrangements are represented in the 3800 by character arrangement tables.
character arrangement table: A table, including a 256-byte translate table that translates the user's data code to the 3800's control electronics code, and identifiers for up to four character sets and graphic character modification modules to be used.	IBM-supplied character arrangement tables are on the SYS1.IMAGELIB system data set. You can modify or extend them, or create your own, using the IEBIMAGE utility.

*American National Standard definition

How a Character is Printed

The sequence of events that prepare the 3800 for printing is described in Figure 3 and in the numbered steps below. This sequence assumes that options are specified in the JCL rather than by default. The character arrangement table G11 was chosen for illustrative purposes, because it uses graphic character modification.

1. You request, with the CHARS parameter, that a character arrangement table be loaded.
2. SETPRT loads the character arrangement table module from SYS1.IMAGELIB into its own storage in the processor.
3. The character set identifier(s) in that module causes the desired character set(s) to be loaded from the flexible disk into the writable character generation module(s) of the 3800.
4. The translate table portion of the character arrangement table is then loaded into the 3800.
5. The graphic character modification modules (if any) identified by the character arrangement table are loaded from SYS1.IMAGELIB into SETPRT's storage.
6. The scan patterns for the characters contained in the graphic character modification module(s) replace the patterns at the specified WCGM locations.

1. **Character arrangement table G11 is specified, using the CHARS parameter.**

```
//SYSPRINT DD SYSOUT=A,CHARS=G11
```

2. **SETPRT loads the character arrangement table module from SYS1.IMAGELIB into its own storage in the CPU.**

G11 Character Arrangement Table

Translate Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2																	2
3																	3
4	00										0A	0B	0C	0D	0E	0F	4
5	10										1A	1B	1C	1D	1E	1F	5
6	20	21										2B	2C	2D	2E	2F	6
7											3A	3B	3C	3D	3E	3F	7
8																	8
9																	9
A																	A
B																	B
C		01	02	03	04	05	06	07	08	09							C
D		11	12	13	14	15	16	17	18	19							D
E	2A		22	23	24	25	26	27	28	29							E
F	30	31	32	33	34	35	36	37	38	39							F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Character Set ID → 82

00 ← WCGM ID

SPC1 ← Graphic Character Modification Module ID

3. **Using the character set identifier '82' SETPRT loads the character set from the flexible disk in the 3800 into WCGM 00.**

WCGM 00

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		A	B	C	D	E	F	G	H	I	¢	.	<	(	+	|	0
1	&	J	K	L	M	N	O	P	Q	R	!	$	*	)	;	¬	1
2	-	/	S	T	U	V	W	X	Y	Z	¤	,	%	_	>	?	2
3	0	1	2	3	4	5	6	7	8	9	:	#	@	'	=	"	3
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

4. **SETPRT loads the translate table portion of the character arrangement table into the 3800.**

Translate Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2																	2
3																	3
4	00										0A	0B	0C	0D	0E	0F	4
5	10										1A	1B	1C	1D	1E	1F	5
6	20	21										2B	2C	2D	2E	2F	6
7											3A	3B	3C	3D	3E	3F	7
8																	8
9																	9
A																	A
B																	B
C		01	02	03	04	05	06	07	08	09							C
D		11	12	13	14	15	16	17	18	19							D
E	2A		22	23	24	25	26	27	28	29							E
F	30	31	32	33	34	35	36	37	38	39							F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

5. **SETPRT loads the graphic character modification module 'SPC1' from SYS1.IMAGELIB into its own storage.**

SPC1	4A	[	5A	]	EO	\

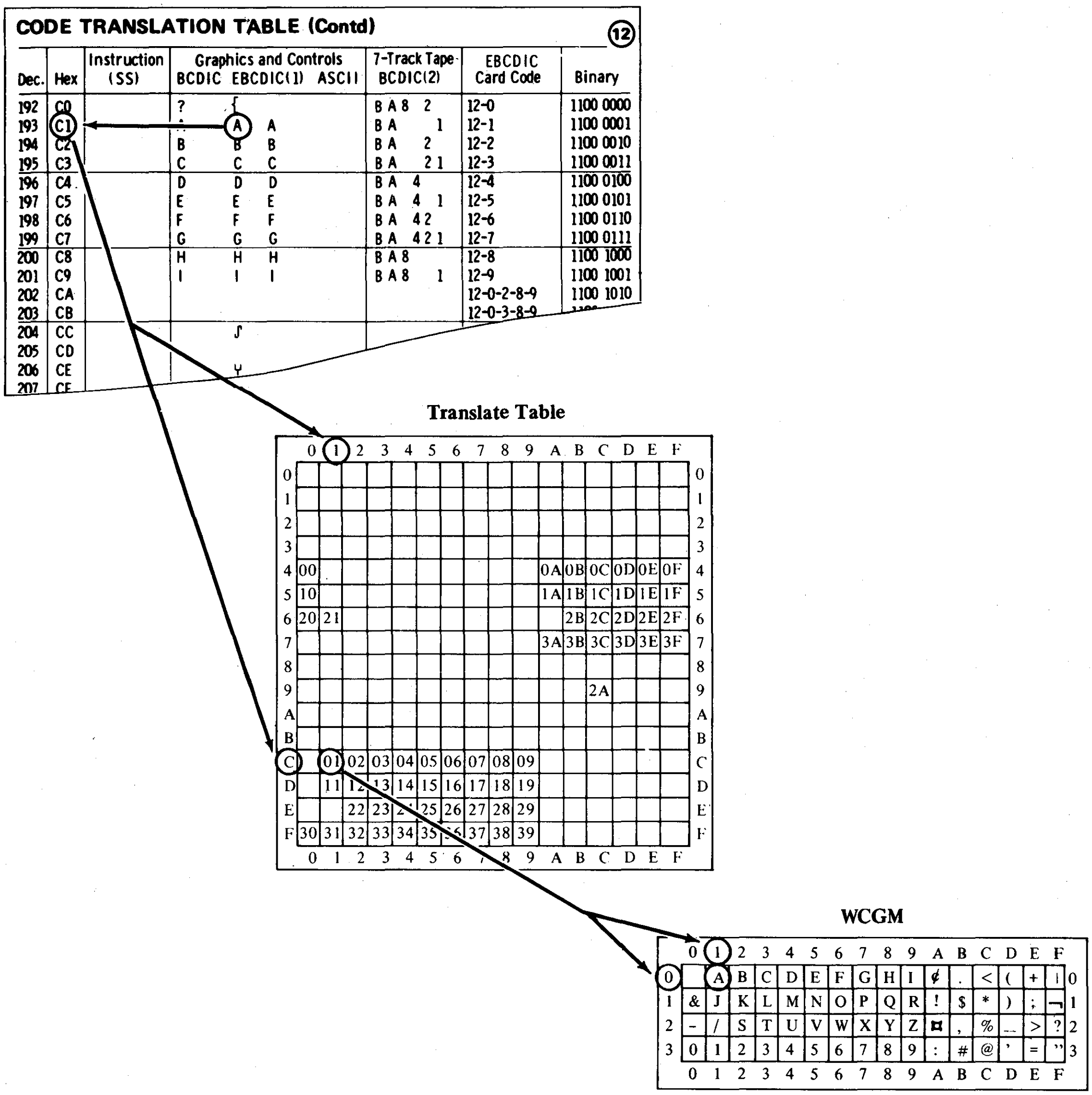
6. **The data code of each character in the module addresses the translate table location containing the address of the WCGM location where the scan pattern of the modifying character replaces the scan pattern that was there.**

WCGM 00

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		A	B	C	D	E	F	G	H	I	[	.	<	(	+	|	0
1	&	J	K	L	M	N	O	P	Q	R	]	$	*	)	;	¬	1
2	-	/	S	T	U	V	W	X	Y	Z	\	,	%	_	>	?	2
3	0	1	2	3	4	5	6	7	8	9	:	#	@	'	=	"	3
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Translate Table Contents	WCGM	
	Reference	Sequence
00-3F	00-3F	First
40-7F	00-3F	Second
80-BF	00-3F	Third
C0-FE	00-3E	Fourth

Figure 3. The Loading Sequence for the G11 Character Arrangement Table



CODE TRANSLATION TABLE (Contd) ⑫

Dec.	Hex	Instruction (SS)	Graphics and Controls BCDIC	EBCDIC(1)	ASCII	7-Track Tape BCDIC(2)	EBCDIC Card Code	Binary
192	C0		?	{		B A 8 2	12-0	1100 0000
193	C1		A	A	A	B A 1	12-1	1100 0001
194	C2		B	B	B	B A 2	12-2	1100 0010
195	C3		C	C	C	B A 2 1	12-3	1100 0011
196	C4		D	D	D	B A 4	12-4	1100 0100
197	C5		E	E	E	B A 4 1	12-5	1100 0101
198	C6		F	F	F	B A 4 2	12-6	1100 0110
199	C7		G	G	G	B A 4 2 1	12-7	1100 0111
200	C8		H	H	H	B A 8	12-8	1100 1000
201	C9		I	I	I	B A 8 1	12-9	1100 1001
202	CA						12-0-2-8-9	1100 1010
203	CB						12-0-3-8-9	
204	CC			ʃ				
205	CD							
206	CE			Y				
207	CF							

Translate Table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2																	2
3																	3
4	00										0A	0B	0C	0D	0E	0F	4
5	10										1A	1B	1C	1D	1E	1F	5
6	20	21										2B	2C	2D	2E	2F	6
7											3A	3B	3C	3D	3E	3F	7
8																	8
9													2A				9
A																	A
B																	B
C		01	02	03	04	05	06	07	08	09							C
D		11	12	13	14	15	16	17	18	19							D
E			22	23	24	25	26	27	28	29							E
F	30	31	32	33	34	35	36	37	38	39							F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

WCGM

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		A	B	C	D	E	F	G	H	I	¢	.	<	(	+	|	0
1	&	J	K	L	M	N	O	P	Q	R	!	$	*	)	;	¬	1
2	-	/	S	T	U	V	W	X	Y	Z	¤	,	%	_	>	?	2
3	0	1	2	3	4	5	6	7	8	9	:	#	@	'	=	"	3
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 4. How the 8-bit Data Code Causes Printing

Figure 4 shows how the 8-bit data code sent to the 3800 is transformed into an address in a WCGM via the translate table portion of the character arrangement table. The 8-bit EBCDIC assignment "C1," equivalent to the graphic character "A," is used by the 3800 to address the location in the translate table that contains the value "01." That value is the address of the WCGM location that contains the scan pattern for printing the "A."

OPTIONAL FEATURES OF THE 3800

Optional features for the IBM 3800 Printing Subsystem are:

- A Burster-Trimmer-Stacker
- Additional character generation storage
- A two-channel switch

Burster-Trimmer-Stacker

The Burster-Trimmer-Stacker can be added to the base machine in order to produce individual sheet output. The trimmer removes 1/2 inch (12.7 mm) from each edge of the form. The form is then fed to the burster, which separates the continuous forms into individual sheets. These sheets are then moved to the stacker.

The output from each data set is offset from that of the preceding data set in the stacker. Copies and groups of copies are also offset. The programmer can request, through JCL, that the job output go to the Burster-Trimmer-Stacker or to the continuous forms stacker (see "How to Burst and Offset Stack Output" in the "Using the 3800" chapter). The operator must thread the paper into the Burster-Trimmer-Stacker when he receives the message to do so.

Additional Character Generation Storage

The character generation storage in the basic 3800 is sufficient for 128 graphic characters—two character sets. Character generation storage for two more character sets can be added. Since the data code X'FF' is reserved to indicate a non-printable character, the last position of the added storage cannot be used and the maximum capacity with the added storage is 255 characters.

For operating purposes, the character generation storage is divided into 64-position modules called writable character generation modules (WCGMs), each to hold one character set. Up to four WCGMs may be program-selected to be active for the printing of any of the lines of a data set (see "How to Specify Character Arrangements" and "How to Intermix Character Styles and Pitches" in the "Using the 3800" chapter).

Two-Channel Switch

A two-channel switch is available as an optional feature for attaching the 3800 via two channels to two separate processors. The switching is done manually via the interface enable/disable switches on the 3800. Only one channel interface can be enabled at a time.

With the two-channel switch installed, a dynamic switch feature can be added that allows both channel interfaces to be enabled at the same time. Selection of the interface to be used is determined by programming. The dynamic switch feature allows the 3800 to be attached to two channels of a single processor and provides alternate path capability. (Note: The alternate path support is not used if the 3800 is connected to byte multiplexer channels.) The dynamic switch feature also allows the 3800 to be attached to two processors in a MP environment and allows both processors, one at a time, to access the 3800, thus providing alternate path capability. (Note: For MP systems this dynamic two-channel switch configuration is recommended because it reduces potential I/O supervisor overhead in accessing the 3800.)

Also, with the two-channel switch installed, a remote switch feature can be added to permit manual channel selection from a remote console rather than at the printer.

PLANNING FOR THE 3800

On the IBM 3800 Printing Subsystem, most of the jobs you are currently printing on your IBM 1403 or 3211 can be printed without change. Because of the differences in the way the 3800 prints, there are new things you can do with your 3800, and there are some incompatibilities with impact printers. This chapter describes the things you should consider—both the new features and the incompatibilities—when planning for the 3800.

Overprinting

Because printing on the 3800 is done by transferring images to the paper from a constantly revolving drum, overprinting cannot be done. Overprinting on an impact printer requires stopping the paper in a fixed position (that is, no carriage movement) while two or more lines of data are printed in the same line space. This results in having two or more sets of printed characters superimposed one on the other (or one set superimposed on itself, for boldface printing).

While the 3800 cannot create multiple images at the same print location (except for characters with underscores), it can merge lines of nonconflicting data into a single line before exposing the resulting line onto the moving drum. (Nonconflicting data lines contain either blanks or the same data characters as those in the corresponding positions of previously transmitted lines.)

The following two examples of frequently-used impact printer applications of overprinting are included to indicate what the results would be on the 3800.

- Boldfaced printing—the same data is printed on an impact printer more than once in the same position to emphasize it with a heavier and darker image. Since the data to be superimposed is identical, the 3800 prints it with no special emphasis in terms of contrast or thickness to make it stand out against other data. The user may, however, design boldface characters and load them into the printer by means of the graphic character modification feature.
- Printing of special composite characters (for example, a zero with a slash over it)—this requires two characters to be printed in the same space on an impact printer. The 3800 prints the first character while rejecting the second with an error indication, unless data checks are blocked, in which case no error is indicated. The output is either a zero or a slash, depending on which was transmitted first. Again, the special character might be designed by the user and loaded into the 3800 using the graphic character modification feature.

Underscoring

The 3800 prints with underscores using either of two methods:

- The 3800 has a built-in underscore capability. When an underscore and another printable character are to be printed in the same character position on a page, the 3800 generates and prints an underscore when it prints the character.
- Six underscored character sets (in Gothic and Text) are supplied.

Using the built-in underscore capability requires transmitting the output data lines containing the character to be underscored and the underscore separately over the channel to the 3800. Using the appropriate underscored character set permits transmitting one line rather than two, with a resulting saving in channel use and transmission time. The use of an underscored character set, however, may require additional character generation storage when underscored printing is intermixed with nonunderscored (for example, when Text 1 and 2 and underscored Text 1 and 2 are used together), and requires specifying the desired underscored character arrangement table name in the JCL CHARS parameter. The amount of underscoring required, then, should be the primary factor in determining which of the two methods is used.

Folding

Folding is a technique used in impact printers with the Universal Character Set (UCS) feature to allow printing in uppercase graphic characters when lowercase graphic characters are called for and not available on the print train. Activating the Fold function on an impact printer is done by issuing the Fold channel command, while deactivating the same function is done by issuing the Unfold channel command. While the 3800 does not recognize the Fold and Unfold channel commands, using the GF10, GF12, GF15, or GFC character arrangement table provides the folding effect (ignoring the first 2 bits of each character's 8-bit data code). If character arrangement tables with the folding feature for other than Gothic characters are desired, they can be provided by the user.

Compatibility with Print Trains

The 3800 provides compatibility for users of other IBM printers by providing character arrangement tables that correspond with and have the same names as the standard 1416 and 3216 print trains (as used on IBM 1403 and 3211 Printers). Desired groups of these character arrangement tables can be specified during system generation. In addition, new character arrangement tables are provided, as a Basic group, to include the characters found in various print trains and to provide for different pitches, underscoring, and the folding effect.

The 3800 programming support does not provide character arrangement tables equivalent to nonstandard 1416 and 3216 print trains. Any program using a nonstandard print train must have a user-defined character arrangement table to correspond to the train. Special characters on such a train may have to be defined by the user. Information on creating and modifying both character arrangement tables and characters is in the "Using the 3800" and "The IEBIMAGE Utility Program" chapters of this book.

Channel Command Considerations

Channel commands direct a channel, control unit, or device to perform an operation or set of operations. Channel commands for the 3800 are listed in Appendix E.

Certain channel commands are device dependent and cannot be used interchangeably among printers. The following is a complete list of channel commands that are valid for either the IBM 1403 or IBM 3211 Printer but are *not* accepted by the 3800:

1403	**3211**
Diagnostic Write	Load UCSB
Diagnostic Data Read	Fold
Diagnostic Check Read	Unfold
Raise Cover	Read PLB
UCS Load (without folding)	Read UCSB
UCS Load (folding)	Read FCB
UCS Gate	Check Read
	Diagnostic Write
	Diagnostic Gate
	Skip to Channel 0
	Raise Cover

Paper Considerations

The 3800 uses continuous single-ply fanfold forms with both edges punched for tractor feeding, and perforations between the pages. The paper may be of various light pastel colors; it can be preprinted, have ruled lines or color striping, or can be blank.

The sizes and basis weights of paper that can be used in the 3800, and the maximum number of characters than can be printed on each width at each of the three available pitches, are given in "Appendix D: Paper Sizes, Weights, and Maximum Characters per Line."

See the *Forms Design Reference Guide for the IBM 3800 Printing Subsystem* for specifications for the paper and forms that are acceptable for printing in the 3800.

Forms Design Considerations

The 3800 Printing Subsystem can use preprinted forms, or can create its own forms as it prints the data, by using:

- The forms overlay feature to print forms from previously-prepared negatives
- The Format character sets to print squares, rectangles, and column rules
- The copy modification feature to print headings, legends, explanatory remarks. and similar data on copies of a report, as well as to suppress certain data on certain copies

These features can be used singly or in any combination.

For information on how to request flashing of forms, refer to "How to Flash Forms" in the "Using the 3800" chapter. Performance considerations when using forms overlay are discussed later in this chapter under "Performance Considerations." For information on designing and making or obtaining forms overlay negatives, see the *Forms Design Reference Guide for the IBM 3800 Printing Subsystem,* listed in the "Preface."

Information on using Format characters is in the "How to Use Format Characters" section. Copy modification is discussed under "How to Modify Copies."

Forms Control Considerations

The 3800 uses a forms control buffer (FCB) to control the vertical format of the page, as does the IBM 3211 Printer. The format of the 3800 FCB, however, is different from that of the 3211.

The 3800 can print with a vertical line spacing of either 6, 8, or 12 lines per inch or an intermix of any of these on the same page. The 3800 FCB must describe the exact length of the forms being printed. Further information on this topic is under "How to Specify Forms Control."

Page Margins

The 3800 reserves the top and bottom 1/2 inch of each page for accelerating and decelerating paper when it is necessary to start or stop printing. If current programs are designed to print in these areas of a page, either the printing format or the FCB definition must be changed.

For example, only 10 inches on an 11-inch long page can be used for printing with the 3800. Therefore, if a program must print 61 lines per page on an 11-inch long page, it cannot do so using a line spacing of 6 lines per inch for the whole page. The FCB must be redefined to obtain 61 printable lines at various vertical line spacings, or all at 8 lines per inch, in order to fit in the 10-inch printable length. Jobs that in the past required printing across page boundaries (such as long graphs, or computer-produced contour or demographic maps) cannot be produced in the same way on the 3800.

Left and right page margins must allow at least 1/2 inch (12.7 mm) between the page edge and the ends of the print line, except for the widest paper size. Because the maximum printable line width is 13.7 inches (348 mm), the right margin on 14-7/8 inch paper is 0.675 inch. For a 3800 using ISO paper sizes, the right margin is 17.3 mm for 378 mm paper.

FCB Change for Users of ISO Paper Sizes

If your 3800 uses ISO (International Standards Organization) paper sizes, an additional planning consideration is that the IBM-supplied FCB named STD3 (for use when printing high-density dumps at 8 lines per inch) is not usable on your machine. STD3 specifies that 80 lines be printed at 8 lines per inch on 11-inch long paper (which is not an ISO paper size). Thus STD3 should either not be used or should be modified for use on a 3800 using ISO paper sizes. Example 5 in "The IEBIMAGE Utility Program" chapter shows how to make this modification.

Output Writer Considerations

For further information on the output writer, see the "Job Management" chapter of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

SYSOUT Classes

Any interruption for operator action (such as changing forms) reduces the overall throughput of the 3800. To minimize operator intervention, a separate SYSOUT class can be established for each type of printing that requires special operator action. With the 3800, three setup operations require explicit operator action:

- The changing of forms
- Inserting a forms overlay frame
- Threading paper in the burster or continuous forms stacker

Additional SYSOUT classes can be defined based on priority, print load, output data set sizes, etc.

SYSOUT Data Set Block Size

For any output device type, the output writer runs more efficiently when processing SYSOUT data sets that have larger blocking factors. Cataloged procedures, programs, and jobs that are run repetitively should specify relatively large blocking factors for SYSOUT data sets.

Spool Space Allocation

When the output requires an average of one or two copies, the 3800 can print SYSOUT data sets as fast as or faster than an impact printer. In that case, the spool space allocation made for impact printers is adequate for the 3800 printer. However, if a large number of copies is planned, it could take the 3800 more time to print them (since it prints one copy at a time) than it would an impact printer using multi-ply forms. Spool saturation could result if the 3800 uses more time than an impact printer to free spool space. In that case, additional direct-access space should be provided for SYSOUT data sets.

Writer Procedure Considerations

IBM supplies two writer procedures, WTR and WTR3800. Either one can be used for writing on any supported printer, punch, or tape drive. For increased function, flexibility, and efficiency, you may wish to modify them or add new ones.

Writer procedures can be used to provide defaults for certain characteristics when writing on a 3211 or 3800 printer. Temporary changes to these characteristics for the life of a writer can be made by the operator in the Start command. On a 3211, the writer procedure (as overriden by the Start command) provides defaults for FCB and UCS for data sets for which they are not specified. On a 3800, the writer procedure (as overriden by the Start command) provides defaults for FCB, CHARS, FLASH, and BURST for data sets for which they are not specified. Default characteristics are also used when printing messages and separators. You may wish to have a procedure for each combination of default characteristics required by your installation.

Other things that are specified in a writer procedure are punching vs. printing (if on tape), SYSOUT classes to be processed, the job separator routine name, region size, address space type, control character type, record length, record format, and number of buffers. For a 3800 writer procedure, you may wish to have a logical record length large enough to accommodate a print control character, table reference character, and the maximum print line length of 204 characters. You may also wish to have more buffers for a 3800 than for impact printers and punches, due to its higher speed. The various procedure parameters greatly affect performance.

For further information on writer procedures, see the "Job Management" chapter of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

Job Separation

The 3800 automatically marks the bottom of each page at the left margin. When the End-of-Transmission channel command is issued by the output writer after each job, copy (in page-number sequence), or copy group, this marking is changed from one vertical bar to two, and vice versa. If the output is going to the Burster-Trimmer-Stacker, each job, copy, or copy group also is offset from the previous one in the stacker.

If, in addition to this automatic marking, job separator pages are specified, the 3800 has a Mark Form channel command that can cause a row of vertical bars to be placed on the perforations between pages. The output writer issues this command for job separation on the 3800, since the 3800 cannot print a normal line (for example, asterisks) on the perforations. User-written separator routines can also use this command.

The Mark Form command is supported only when the RECFM parameter specifies M (machine control characters). The command generates three or more copies (depending on the page size) of the page to be marked so that the marks always appear, regardless of the fold. This makes it unnecessary to transmit multiple copies of the job separator page. If no job separator page is used, a command to skip to a new page should always precede the Mark Form command so that copies of only the blank pages and not the data are generated.

If you are creating your own job separator page, you may take advantage of the new function in the 3800 whereby characters of your own design may be used. Refer to the section, "How to Change and Create Characters" for more information on this.

Reassigning Printing to Alternate Printers

The operator can reassign printing destined for the 3800 to an impact printer. There are some restrictions, however, that apply:

- If the UCS parameter is omitted, the output writer uses the character arrangement table name specified in the CHARS parameter (assuming it is also a valid print train name) as the name of the print train for an impact printer with the UCS feature. For example, CHARS=A11 is valid for both the 3211 and the 3800 printer. If more than one character arrangement table is specified, the first one is used. If no CHARS parameter is specified, the output writer recognizes a valid name specified in the UCS parameter as being the character arrangement table name for a 3800. See Figure 32 for the names of the IBM-supplied character arrangement tables.
- Both the CHARS and the UCS parameters can be specified on the output DD statement, to be used by the appropriate printer. In the above example, specifying CHARS=GS10,UCS=A11 is valid for either a 3800 or a 3211, *but* the special characters included in GS10 that are not on an A11 print train are printed as blanks on the 3211.
- Characters that cannot be printed by the impact printer (such as user-designed characters) are printed as blanks.
- Specification of features unique to the 3800 (such as FLASH for forms overlay) is ignored.
- Any *group value* subparameters of the COPIES parameter are ignored for an impact printer.
- Table reference characters, if any, at the beginning of data lines are not printed on the impact printer if OPTCD=J is specified.
- If the print line length exceeds the limit allowed by the impact printer, the line is truncated. For example, when CHARS=DUMP is specified, the maximum output line length is 204 characters. However, when printed on an impact printer, the output lines are truncated to the maximum line length of that printer.
- For jobs with specific forms control requirements, the alternate printer should have an FCB or a carriage control tape equivalent to that used by the 3800. For instance, if you are using the IBM-supplied FCB named STD3 (which prints 80 lines at 8 lines per inch on 11-inch-long paper) you can reassign the printing to a 3211 that has an equivalent FCB image. It is not possible to create an impact-printer FCB equivalent to a 3800 FCB that intermixes different vertical line spacings on a page.

High Level Language Considerations

The considerations for users of high level languages are simply those considerations mentioned elsewhere in this chapter that apply. For example, because overprinting is not allowed on the 3800, the PL/I programmer using the 3800 cannot overprint, even though that facility is provided for him at compile time and execution time by PL/I. Likewise, the PL/I facility to control the number of lines per page is subject to the unavailability of the top and bottom half inch of the page for printing in the 3800.

Considerations for High-Density Dumps

With the 3800, paper can be saved by changing either or both of the two printing dimensions:

- Increasing the number of characters per inch (the pitch) permits the use of narrower paper, or permits more data to be printed on one line.
- Increasing the number of lines per inch (changing the FCB) permits the use of shorter paper, or allows more lines of data to be printed per page.

The programming support for the 3800 provides two options for printing system dumps in condensed formats:

- By specifying the DUMP character arrangement table (using CHARS=DUMP), dumps are printed in 15-pitch characters with each print line displaying 64 bytes of storage, rather than 32 bytes.
- By specifying FCB=STD3, 80 lines per page are printed at 8 lines per inch.

The use of CHARS=DUMP requires 14-7/8 inch wide paper (or 378 mm wide for a 3800 using ISO paper sizes). The use of FCB=STD3 requires 11-inch long paper (see the section, "FCB Change for Users of ISO Paper Sizes" earlier in this chapter for information on redefining STD3 for a 3800 that uses ISO paper sizes).

Dumps can also be printed using 15-pitch characters to print 32 storage bytes per line, thus using narrower paper (9-1/2 inches wide, or 235 mm wide for a 3800 that uses ISO paper sizes), or using 8 lines per inch to print up to 60 lines on shorter paper (8-1/2 inches long, or 10 inches long for the nearest ISO paper size). Any desired combination of these formats can be used.

Information on requesting and reading high-density dumps is in *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.* ABEND dumps are described in the "Debugging Guide" chapter of that book; information on requesting SNAP dumps is in the "Supervisor Services and Macro Instructions" chapter; and stand-alone and SVC dumps (available through the PRDMP facility) are covered in the "Service Aids" chapter.

When CHARS=DUMP is specified in VS1, be sure that your writer procedure specifies a value for LRECL that is large enough for a print line of 204 characters, a print control character, and a table reference character.

Performance Considerations

The 3800 Printing Subsystem operates at a constant rate, moving paper at a nominal speed of 31.8 inches per second as long as data to be printed is available in the page buffer. The photoconductor on the drum surface is nominally 77 inches long. There is a 2-inch gap where the supply and takeup ends of the photoconductor material pass into reels mounted inside the drum. Since the image to be printed cannot be generated across the 2-inch gap, all the pages that can be printed in one revolution of the drum must fit into a 77-inch length. Data must be available in the page buffer in time to print each successive page while the drum is revolving if maximum printing speed is to be maintained. A delay in filling the page buffer to complete the data for any page can result in deferring the printing of at least one page to the next length of photoconductor.

To illustrate, assume that 11-inch-long forms are being printed. Seven such forms can be printed in each 77-inch length of photoconductor per drum revolution. If there is just one instance when all the data for a page to be printed is not in the page buffer when the image is to be generated, only six forms can be printed in that revolution of the drum. If this occurs consistently during the printing of the data set, the 3800 performs at only 6/7 of its potential.

The approximate formula PB÷N, where PB is the number of bytes of available page buffer storage and N is as given in the following tables, can be used to estimate the maximum data content of a page that can be printed without performance degradation.

Page length (common-use sizes) in inches	**N, number of required pages in the page buffer when using forms overlay**	**N, number of required pages in the page buffer when not using forms overlay**
3-1/2	8	2
5-1/2	6	2
7	5	2
8-1/2	5	2
11	4	2

Page length (ISO sizes) in inches	**N, number of required pages in the page buffer when using forms overlay**	**N, number of required pages in the page buffer when not using forms overlay**
3	9	2
4	8	2
6	6	2
8	5	2
10	5	2
12	4	2

For example, for an 11-inch page, the maximum data content is determined from the 52K-byte page buffer capacity as follows:

Forms overlay utilized: Maximum page size in bytes (PB÷4)	**No forms overlay: Maximum page size in bytes (PB÷2)**
13,312	26,624

Whenever the formula is used, the size of the page buffer should be reduced by 256 bytes for each additional character arrangement table beyond the first and by the amount of data contained in copy modification modules. A further reduction of 768 bytes is required for a 3800 with the optional Burster-Trimmer-Stacker attached. The amount of page buffer occupied by copy modification data can be closely approximated by assuming 1 byte for every non-repetitive character (including blanks) in a line, 3 bytes for every set of four or more repeating characters in a line, and 6 bytes for each unique entry appearing on one or more copies. When using hardware underscore, an additional byte of storage is requred for each underscored character, and there is no consecutive character compression for the line.

To optimize performance, these general recommendations should be followed when planning for a 3800:

- Direct output to the 3800 from a problem program is not recommended since it is unlikely that a problem program can drive the 3800 at full speed. In addition, direct output would not provide for the enhanced page recovery available with SYSOUT. Data to be printed by the 3800 should be assigned to SYSOUT data sets.
- To estimate the amount of storage required for a page of data, allow one byte per character printed. When four or more identical characters occur consecutively within a line, such as fill blanks at the end of the printable data line, only three bytes are required for each such compressed group. For example, a print line of 120 characters requires 123 bytes of page buffer storage. A page of printed output from the standard storage dump used in system installations consists of one line for the page number and 55 lines for the storage information. Because each of the storage data lines includes a few groups of repeated characters among the 120 that are printed, the page buffer storage is about 101 bytes per line rather than the 123 that might be expected. The data for an entire output page of 55 lines will occupy about 5600 bytes in the page buffer.
- It is recommended that the 3800 be attached to either a byte or a block multiplexer channel. Attachment to a selector channel is not recommended unless the channel is dedicated, because other devices on the same channel can be totally superseded when the 3800 is running. The 3800 work load should be added to the existing channel work load to help determine whether a byte or a block multiplexer channel is appropriate.
- The DASD unit(s) containing the spool volume(s) should have an access time that will not limit production of printed output. For example, an IBM 3330 or 3340 attached to the processor by a different block multiplexer channel than the 3800 should have a suitably fast access time, while a slower DASD device may not.
- An estimate of the approximate data rate can be made by using the number of characters (including imbedded blanks) to be printed on a page and the time it takes to print that length of page at 31.8 inches per second. For example, an 11-inch page takes about 0.346 seconds to pass a point in the 3800. If each such page contains 3100 bytes, the channel data rate required to maintain this printing would be 3100 divided by 0.346 or about 8960 bytes per second.
- To avoid I/O interrupts caused by using the channel 9 and 12 codes in the FCB, the application program can count lines to determine page size or line position on the page.
- When much of the printing on a system is at relatively high print densities (for instance, when storage dumps are frequently printed), a multi-volume spool is recommended.

- Further performance recommendations for using a 3800 with the system output writer are in *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*
- When printing at 12 lines per inch vertical spacing, the maximum single-copy printing speed when the 3800 is attached to a block multiplexer channel, can be maintained if the approximate data limits in the following table are observed:

Common-use paper sizes

Page length in inches	Maximum characters per page
3-1/2	5190
5-1/2	7770
7	9720
8-1/2	11700
11	14880

ISO paper sizes

Page length in inches	Maximum characters per page
3	4530
4	5830
6	8400
8	11000
10	13600
12	16230

For a 3800 attached to a byte multiplexer channel, the listed limits should be reduced by about 28%.

Data Security

The 3800 contains several features for protecting the data in the page buffer from improper alterations or disclosures:

- Page buffer data cannot be read back into the processor using available channel commands.
- Residual data in the page buffer cannot be printed because page definition is destroyed after the last copy of a page has been exposed onto the photoconductor.
- Residual images on the photoconductor are removed by the cleaner assembly, which prevents reading the images on the photoconductor.

System Generation Considerations

The 3800 Printing Subsystem independent component can be added to your system through a full system generation or an I/O device generator. It is not supported as an output console device. The SYS1.IMAGELIB system data set is required for a system using the 3800.

The groups of character arrangement tables desired at your location (as shown in the first column of Figure 31 in "Appendix B: IBM-supplied Character Arrangement Tables") can be selected at system generation.

An optional channel capability, requested by using the OPTCHAN parameter of the IODEVICE macro, can be requested only for a 3800 that has both the optional two-channel switch and the dynamic switch feature.

For more information on generating your system, see the "System Generation" chapter of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

SYSTEM PROGRAMMING FOR THE 3800

The IBM 3800 Printing Subsystem can be used either as a system output device or as a direct (that is, unit-record) output device. Figures 5 and 6 show the relationships of the various system functions that occur between the user and the 3800, for the system printer and the direct configurations, respectively.

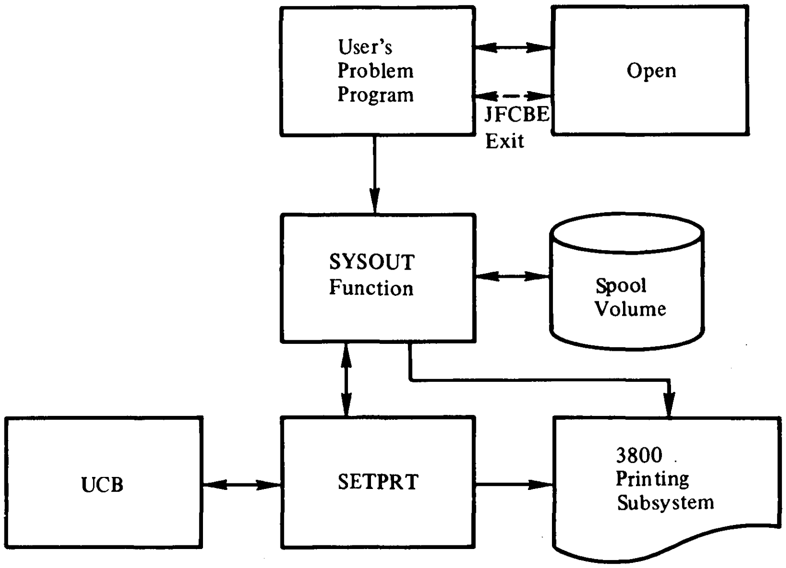


Figure 5. An Overview of SYSOUT Processing to the 3800

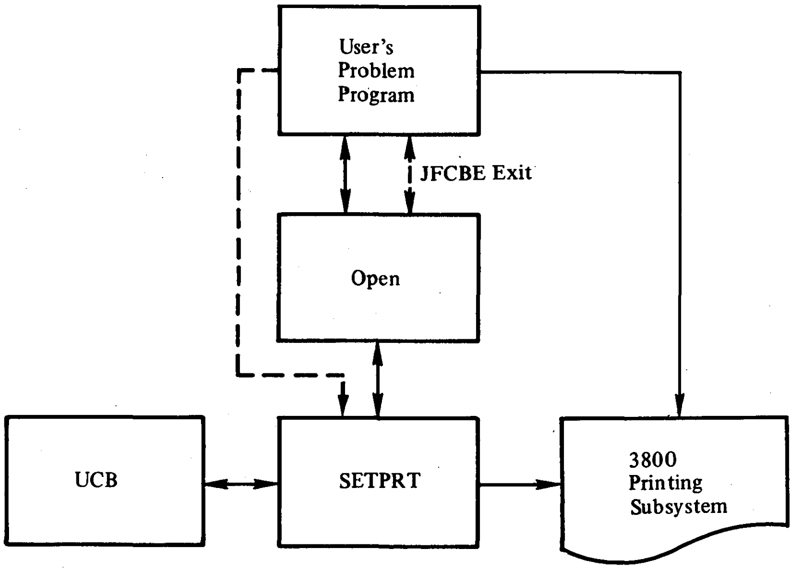


Figure 6. An Overview of Direct Processing to the 3800

The JCL parameters are discussed in the next section of this chapter and also in the next chapter, "Using the 3800." Additional details (such as defaults and examples) on these parameters can be found in the "JCL" chapter of

OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.

The output writer is a key link between the user and the 3800 for SYSOUT processing. In the previous chapter, "Planning for the 3800," specific 3800-related considerations are discussed under "Output Writer Considerations.“

Another link for both SYSOUT and direct processing to the 3800, is SETPRT, which is discussed later in this chapter.

Following that, other sections of this chapter discuss "Defaults for JCL, SETPRT, and the 3800," and "The JFCB Extension and Exit."

Job Control Language (JCL)

The following JCL parameters can be used to define the output data set's requirements to the operating system. While many of these same parameters can be specified if using the SETPRT macro, specifying them in the JCL usually gives more flexibility.

The keywords and functions they perform are as follows:

BURST= Whether or not the forms should be threaded into the Burster-Trimmer-Stacker for bursting.

CHARS= The names of the character arrangement tables, which define the characters to be used in printing.

COPIES= The number of copies of a data set to be printed, and/or the number of identical pages to be printed before printing the next group of identical pages.

FCB= The name of the FCB module to be used in the vertical formatting of the page, and if the format should be verified by printing of the FCB data.

FLASH= The name of the forms overlay frame to be used, and how many copies are to be printed with the overlay.

MODIFY= The name of the copy modification module to be used in modifying the data, and a table reference character for selecting which character arrangement table is to be used for printing of the modifying data.

OPTCD=J Specifies that each line of data contains a table reference character for selection of a character arrangement table to be used when printing that line. (This is a DCB subparameter.)

OPTCD=U Specifies unblock data check, allowing error conditions caused either by unprintable characters, or by trying to print multiple characters in one print position, to be sent to the host system. (This is a DCB subparameter.)

It may be desirable not to specify any of the keywords in either the JCL or in the SETPRT macro, and take either the output writer defaults or the 3800's machine defaults. For example, if the machine defaults are used, you get a Gothic 10-pitch character set and a folded character arrangement table. The FCB is 6 lines per inch to the length of the form, with the first print line printing on the first line allowable for printing and the last print line printing on the last allowable line. COPIES is equal to 1, BURST is equal to its last setting, MODIFY, FLASH, and OPTCD=J are ignored and Block Data Check is unchanged.

Some jobs may have simple requirements that can be taken care of by one or more keywords, for instance:

- To print data at 8 lines per inch on an 11-inch-long form:

```
//PRINTER DD SYSOUT=A,FCB=STD3
```

- To print using uppercase and lowercase Text characters:

```
//PRINTER DD SYSOUT=A,CHARS=T11
```

- To print high-density system dumps:

```
//SYSABEND DD SYSOUT=A,CHARS=DUMP
```

- To print data in 15-pitch Gothic characters, using a folded character arrangement table:

```
//PRINTER DD SYSOUT=A,CHARS=GF15
```

- To print three copies of a data set:

```
//PRINTER DD SYSOUT=A,COPIES=3
```

- To create a form by projecting an image from a forms overlay frame named "Tops":

```
//PRINTER DD SYSOUT=A,FLASH=TOPS
```

- To alter a copy of a data set with a copy modification module named "Spot":

```
//PRINTER DD SYSOUT=A,MODIFY=SPOT
```

- To alter a copy of a data set with a copy modification module named "Line" that uses an underscored Gothic 10-pitch character arrangement table named "GU10" which is the second of two character arrangement tables names:

```
//PRINTER DD SYSOUT=A,CHARS=(GS10,GU10),MODIFY=(LINE,1)
```

- To print using the OPTCD=J subparameter in your DCB, where each line of data is coded with a table reference character to select one of the character arrangement tables specified with the CHARS keyword:

```
//PRINTER DD SYSOUT=A,CHARS=(GS10,GS15),DCB=(OPTCD=J)
```

Most of these jobs will run when allocating to the 3800 directly (that is, changing the SYSOUT=A to UNIT=3800). One difference arises, however, when multiple copies that require retransmission of the data set are requested using the COPIES parameter. Such retransmission must be done by the user, when allocating direct. See "How to Request Multiple Copies" for more details.

What SETPRT Does

SETPRT is the operating system facility that sets selected control information in the 3800 Printing Subsystem and maintains information about the subsystem.

Who Uses SETPRT

The possible users of SETPRT are:

- The output writer, which uses SETPRT to set control information for each data set it prints on the 3800. The control information is obtained by the output writer from the JCL (as modified by JFCBE or DCB Open exit routines).

- Open, which uses SETPRT to reset the 3800 to the subsystem defaults and to initially set the printer with control information as presented in JCL (as modified by JFCBE or DCB Open exit routines).
- Problem programs, which can use SETPRT *only* when allocating the 3800 as a direct device (that is, specifying UNIT=3800 or *device address*).

The following explanation of SETPRT is divided into two parts: the SETPRT macro instruction and the SETPRT supervisor call (SVC 81).

The SETPRT Macro Instruction

The SETPRT macro instruction allows specification of the following:

BURST=	Whether or not forms are to be burst or stacked without bursting
CHARS=	Up to four names of character arrangement tables
COPIES=	The total number of copies to be printed
COPYNR=	The starting copy number, for copy modification reference
FCB=	The FCB module name, and whether the module needs verification
FLASH=	The forms overlay frame name, and the number of copies to be flashed with the forms overlay
INIT=	Whether or not to initialize the printer to hardware defaults
MODIFY=	The copy modification module name, and the table reference character for the copy modification data.
OPTCD=	Whether or not data checks should be blocked
REXMIT=	Whether or not data is being retransmitted

The SETPRT macro instruction builds a parameter list from the user-supplied parameters, establishes addressability to the list, and calls the SETPRT SVC. For further details on the SETPRT macro instruction and its specification, see the "Data Management" chapter of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

The SETPRT Supervisor Call

The SETPRT SVC accesses the following modules from SYS1.IMAGELIB (or the first three types from the user's storage, in the event they are there) and puts them into its storage area:

- Forms control buffer modules
- Copy modification modules
- Character arrangement table modules
- Graphic character modification modules

The SVC then builds channel programs to load data from these modules into the 3800's buffers.

When the character arrangement table modules are accessed, the SETPRT SVC looks at all the character set identifiers requested, and instructs the 3800 to load each unique character set from the read-only flexible disk into an appropriate WCGM. It may alter the two leftmost bits (the WCGM location code) of each of the 256 bytes of the translate table to correspond to the proper WCGM. (Those bytes that are X'FF' remain untouched.) The SVC then loads the translate table into the printer's control storage.

After the character sets and translate tables have been loaded, the SETPRT SVC checks the character arrangement table modules to see if any graphic character modification modules are needed. If so, it accesses the modules and builds the necessary channel programs to load the modules and modify the characters. Each modifying character carries its own index into the translate table. The 3800 then modifies the appropriate location in the proper WCGM, pointed to by the contents of the translate table location.

The SETPRT SVC loads the data from the forms control buffer module and copy modification modules into the 3800's buffers. It also, when requested, sends the messages to the operator to insert a specified forms overlay frame and to thread the Burster-Trimmer-Stacker or the continuous forms stacker.

Defaults for JCL, SETPRT, and the 3800

The output writer and the Open executors use the parameters that are specified in the JCL to control the functions requested. When these parameters are not specified in the JCL, defaults are used as specified in the output writer, the SETPRT macro, or the 3800, in that order. Figure 7 indicates what parameters can be specified in the JCL (for both SYSOUT and direct allocation) and in the SETPRT macro. The defaults and special conditions are also noted.

The output writer defaults for CHARS, BURST, FLASH, and FCB are specified in the Writer procedure started, or in the actual Start Writer command. If the default is not specified, the machine defaults are used.

For HASP, the defaults are set at HASP initialization time.

The JFCB Extension and Exit

When keywords unique to the 3800 are specified on the DD card, a JFCB Extension (JFCBE) is generated by the operating system to carry the requested setup information. This control block is conditional and does not exist if only the keywords UNIT=3800, COPIES without *group value* specifications, or FCB are indicated. When a JFCBE is created, its existence is indicated by bit JFCBEXTP in the main JFCB. The JFCBE contains the requested 3800 setup requirements as indicated in your JCL, with the exception of FCB information, which appears in the main JFCB as it does for other printers that use an FCB.

Your program has the opportunity to examine or modify the JFCBE during the Open JFCBE exit. This exit is similar to the DCB exit and is taken no matter what device is allocated or whether a JFCBE exists; you specify in the DCB the address of an exit list and in that list request a type X'15' exit (refer to the "Data Management" chapter of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information* for details concerning user exits). If you require both a DCB and a JFCBE exit, you must use the JFCBE exit request to accomplish both (both cannot be separately specified in the exit list, as only the first occurrence of either is taken during Open processing). For the format of the JFCBE, see the "System Data Areas" section of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem Logic.*

	Can be specified in JCL		**Can be specified In SETPRT macro**	**3800's Hardware Defaults**
Parameters	**SYSOUT**	**Direct**		
CHARS= *table name*	Yes	Yes	Yes	note 1
table name	Yes	Yes	Yes	-
table name	Yes	Yes	Yes	-
table name	Yes	Yes	Yes	-
COPIES= *nnn*	note 2	note 3	note 4	note 5
group value 1	Yes	Yes		-
group value 2	Yes	note 3		-
group value n	Yes			-
FLASH= *overlay name*	Yes	Yes	Yes	note 12
count	note 6	note 6	note 6a	-
MODIFY= *module name*	Yes	Yes	Yes	note 12
trc	note 7	note 7	note 7	-
FCB= *FCB id*	Yes	Yes	Yes	note 8
VERIFY	Yes	Yes	Yes	-
BURST=Y | N	Yes	Yes	note 9	note 13
OPTCD=U	note 10	note 10	note 11	-
OPTCD=J (DCB subparameter)	Yes	Yes	-	note 14

Notes:

1. The printer provides the 10-pitch Gothic character set with one folded character arrangement table referencing all 64 positions in the WCGM.

2. If *group values* are specified, the total of all *group values* replaces the value *nnn*.

3. If *nnn* is specified, it is ignored for direct output. Only the first *group value* is used. If the user specifies only *nnn* and no *group values*, a value of 1 is used.

4. Only one group can be specified at a time.

5. One copy is printed, if nothing is specified.

6. If the *overlay name* is specified and the *count* is not specified, all copies printed are flashed.

6a. In SETPRT, if the *overlay name* is specified and the *count* is specified as zero, the operator is asked to insert the overlay frame but no copies are flashed.

7. If the copy modification *module name* is specified, and the *trc* (table reference character) is not, SETPRT assumes a *trc* of 0.

8. The default FCB is 6 lines per inch, to the length of the form, with a channel 1 code at the first printable line. The first half inch at the top of the form, and the last half inch at the bottom of the form are not available for printing.

9. The default is BURST=N (no bursting), if not specified.

10. Only the unblock data check can be specified as a subparameter of the DCB parameter. If unspecified, the default is "block data check."

11. Either the block data check or the unblock data check can be specified. The default, if not specified with the SETPRT macro, is "block data check."

12. This function is inactive unless specified in JCL or SETPRT.

13. The "hardware default" for bursting is wherever the forms have been threaded by the operator.

14. The OPTCD=J subparameter is meaningless unless specified in JCL.

Figure 7. Parameters and Defaults for JCL, SETPRT, and the 3800

When control is given to your exit routine, registers are as follows:

Register	Contents
0	The address of a copy of the JFCBE in user storage area. If no JFCBE exists, this register contains zero.
1	The address of the DCB being processed.
2-15	Standard user exit contents.

The user storage area pointed to by register 0 also contains the 4-byte FCB module name, obtained from the JFCB and placed following the 176-byte JFCBE copy.

If the user copy of the JFCBE is modified during an exit routine, this should be indicated by turning on bit JFCBEOPN (X‘80’ in JFCBFLAG in the JFCBE copy). On return to Open, this bit indicates whether the system copy of the JFCBE is to be updated. The FCB module name copy in the user area, however, is used to update the JFCB, irrespective of the bit’s setting.

Restart Processing

Because the 3800 setup requirements indicated initially on JCL may have been altered as a result of an Open JFCBE exit, Restart operations interrogate the system copy of the JFCBE, if one exists, to determine if the bit signifying that you have modified the JFCBE during Open is on. If JFCBEOPN is on, then the system copy of the modified version of the JFCBE is used; otherwise the JCL-generated version is used.

USING THE 3800

The OS/VS2 SVS operating system supports the operation of the IBM 3800 Printing Subsystem via a channel attachment. You specify JCL DD statement parameters to control the functions of the 3800. The operating system initializes the printer based on these parameter specifications. The control information of the 3800 may also be initialized or changed dynamically using the SETPRT macro instruction, by a problem program that has direct control of the printer (as described in the previous chapter). The programming support includes:

- JCL parameters for specifying character arrangement tables, multiple copies, modifying copies, forms overlay printing, forms control, and bursting and stacking of output.
- Use of the names of print train arrangements (for example, AN, TN) on existing printers. All standard 1416 and 3216 print train arrangements (except for ALA) of the IBM 1403-N1, 3203 Model 4, and 3211 Printers are recognized.
- A basic group of character arrangements, not equivalent to any previous print train arrangements, that is suitable for a wider variety of printing applications. The GS10 arrangement, for instance, includes the graphic characters previously found on a number of different print trains.
- Provision for dynamically selecting the character arrangement during the printing of a data set.
- Automatic control of features and functions for SYSOUT data sets according to specifications in JCL or system defaults. Unless there is a need to override SYSOUT class values, no special provision is required in JCL to print on the 3800.
- Provision for system dumps in condensed format, using 50 to 60 percent less paper. High-density dumps are printed in 15-pitch characters, and are obtained via the ABDUMP and PRDMP facilities. ABEND dumps and SNAP dumps are available through ABDUMP. SVC and stand-alone dumps are available through PRDMP.
- The IEBIMAGE utility program to provide a means for creating:
  - Forms control buffer modules
  - Copy modification modules
  - Character arrangement table modules
  - Graphic character modification modules

How to Specify Character Arrangements

The character sets supplied with the 3800 are used by means of character arrangement tables. The CHARS parameter allows you to specify up to four character arrangement table names for printing a data set.

Any character arrangement table can specify up to four character sets and up to four graphic character modification modules. There must be sufficient character generation storage to hold the character sets specified; storage for two character sets (that is, two WCGMs) is standard in the 3800, and storage for two more sets is an optional feature.

If the optional additional character generation storage is present, the last position of any character set loaded into the fourth WCGM is unavailable for printing, because the system reserves the use of the index into that position (X'FF') to designate an unprintable character.

When two or more character arrangement tables specify the same character set, only one copy of that character set is loaded into a WCGM (thus maximizing the use of character generation storage). Note that if any of the character arrangement tables calls for graphic character modification modules that modify the same character set, each set is considered unique, and a new copy is loaded into a WCGM.

The CHARS parameter may be specified with either JCL or the SETPRT macro:

CHARS=(*table name* [, *table name*]...)

where

table name
is the name (from 1 to 4 alphanumeric or national characters) of the character arrangement table. No more than four names can be specified.

Figure 32 at the beginning of "Appendix B: IBM-Supplied Character Arrangement Tables," gives the names of the character arrangement tables supplied with the 3800 programming support. Since the various groups, other than the "Basic group" (listed first), are selected at system generation, you may not have all the listed character arrangement tables available on your system.

The IEBIMAGE utility program can be used to modify or define your own character arrangement tables. New characters (either IBM-supplied or user-designed or both) can also be added using graphic character modification. Both of these topics are treated later in this chapter.

How to Intermix Character Styles and Pitches

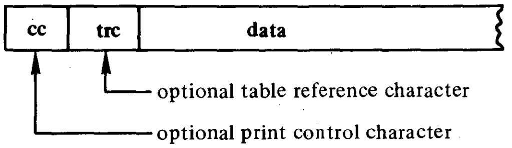
There is more than one way to intermix characters of different styles or pitches within one output data set, and even within one print line:

- One character arrangement table that points to more than one character set can be used. This is already done by some of the IBM-supplied character arrangement tables, such as TN, T11, and TU10, which use both uppercase and lowercase Text characters. This means that the data being printed in the second style or pitch must have EBCDIC assignments different from the data printed in the first style or pitch. You can modify character arrangement tables or build new ones to suit your particular printing application.
- One character arrangement table pointing to only one character set can be used, if graphic character modification is used to replace some characters with others of a different style or pitch.
- More than one character arrangement table can be specified using the CHARS parameter, and if OPTCD=J is coded as a DCB subparameter, you can select the table you want by coding a table reference character (*trc*) as the first byte of each output data line (following the optional print control character). This table reference character (0, 1, 2, or 3) selects the character arrangement table corresponding to the order in which the table names have been specified with the CHARS keyword.
- The character arrangement table used to print a copy modification can be different from the one used to print the data.

Each of these ways of intermixing characters is suitable for many applications, and often a particular printing requirement can be met in more than one way. The first two methods require techniques that are covered later in this chapter in the sections, "How to Modify Character Arrangement Tables" and "How to Change and Create Characters." A third method is explained below. A fourth method is described under "How to Modify Copies."

Intermixing Style or Pitch Using the Table Reference Character and OPTCD=J

To dynamically select character arrangements during printing, code OPTCD=J in your DCB. This informs the system that you are including a table reference character (*trc*) as the first byte of each output data record (following the optional print control character), as described in the previous section.


cc
trc
data
optional table reference character
optional print control character

This ability to select a character arrangement table can be used in different ways. For example:

- One line of data, with one *trc* selecting one character arrangement table, can print as a complete line. For instance, you can print footnotes or headings in a different style or size to subordinate or highlight them.

- Multiple output data lines, each using a character arrangement table different from the one preceding it, can be merged to make one print line. In this case, if machine code print control characters are used, each of the data lines except the last would use a "Print with No Space" control character to allow the merging into one print line. (If American National Standard (ANS) print control characters are used, each of the data lines except the *first* would use a "Print with No Space" control character.) You can use this to highlight or subordinate data within a print line, or to print different columns of data across a page in different styles or sizes. (See "Rules for Merging Data Lines into One Print Line" later in this chapter.)

The system selects the translate table by issuing the appropriate "Select Translate Table" channel command. The existence of a *trc* must be accounted for in your logical record length (LRECL) specification in the DCB. For variable spanned records, only the first segments can contain a *trc.*

Rules for Coding Table Reference Characters

When printing with the 3800, the access method (BSAM or QSAM) strips the table reference character from the output line and uses it to select the desired character arrangement table. To have output data lines with a table reference character as the first byte, you must establish some way in your problem program to specify and insert them. Rules for coding them are:

- The only valid table reference characters are 0, 1, 2, and 3. The leftmost 4 bits of the trc are ignored. Thus, X'F0' and X'00' are both valid representations for 0.
- If a table reference character is specified that references a table higher than the number loaded using the CHARS parameter, it defaults to 0 in the printer, and gives a data check. For example, if two character arrangement tables are specified with CHARS, a *trc* of 2 (referencing a third table) will default to 0, selecting the first table specified. This data check cannot be blocked.
- If an invalid number (such as 4) is specified, it defaults to 0 in the system software.

Rules for Merging Data Lines into One Print Line

Merging occurs by character position, rather than by physical position in the final printed line. The 10th character in a line, for example, will be merged with the 10th character in another line, regardless of where those characters might otherwise appear (as a result of differences in pitch) on the printed page. When merging output data lines (of the same or different pitches) into one print line the rules are as follows:

- A printable character in a following line replaces an identical character or a blank.
- A blank in a following line does not replace either a blank or a printable character.
- A printable character trying to replace a previous graphic different from itself results in a data check, and the character in the new line does not replace the character in the previous line.

The exception to these rules is the underscore, which will be generated and printed with another graphic without a data check.

When merging lines that have characters of different pitch, it is important to understand that:

- When blanks of different pitches are merged, the resulting blank has the pitch of the first one.
- When a printable graphic character is merged with a blank, the resulting character has the pitch of the printable character.

To illustrate, assume that a 10-pitch line, a 12-pitch line, and a 15-pitch line are being merged in that order to create one print line. Each line contains four positions, with three of them blanks and one a graphic in each case. The widths of the 10-, 12-, and 15-pitch characters are proportional to the 18, 15, and 12 bits, respectively, that make up the scan patterns for those pitches. The three data lines and the resulting print line, then, have the proportions shown in Figure 8.

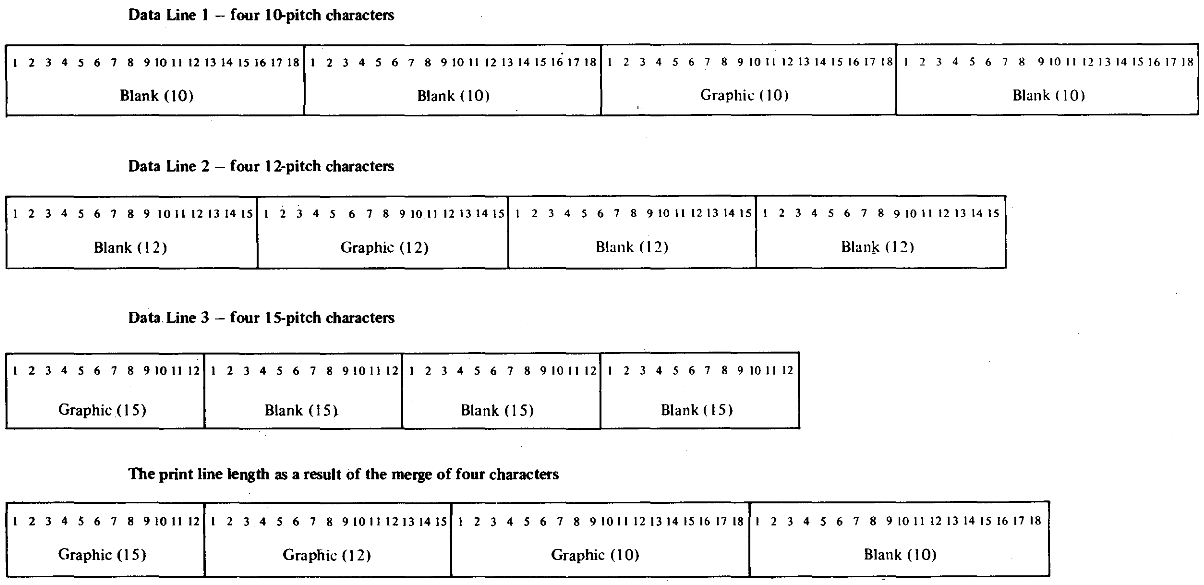


Figure 8. The Result of Merging Different Pitch Data Lines

How to Modify Character Arrangement Tables

The character arrangement table is the key to printing with the 3800. It contains the translation of the 8-bit data code for each character into the control electronics code addressing the scan pattern for that character, identifies the desired character set(s), and identifies any graphic character modification modules to be used.

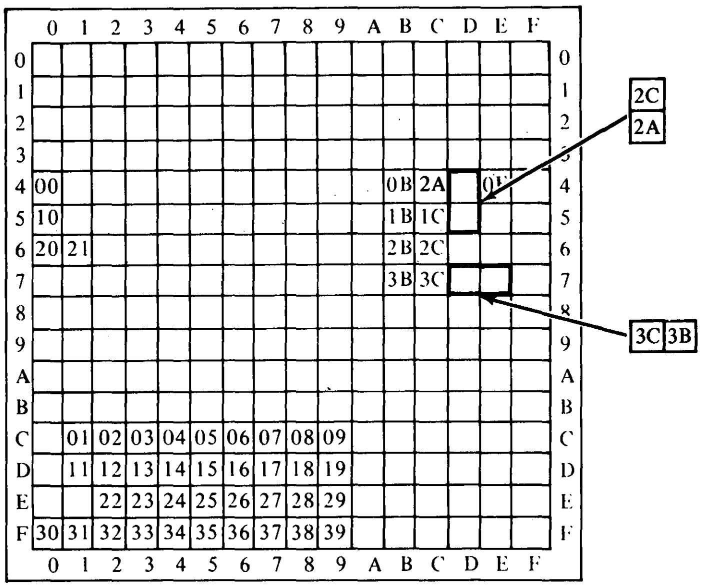
There are 48 character arrangement tables supplied by IBM. They are grouped with one "Basic group" and five others that can be specified during system generation. See "Appendix B: IBM-Supplied Character Arrangement Tables" for both a list of the tables and the details of what each table contains. The tables in the "3211 group" and "1403 group" provide character arrangements identical to those that have the same names on the IBM 3211 and 1403 Printers. Those in the "OCR group" and two in the "Katakana group" likewise provide arrangements identical to the print trains with the same names.

You can modify the IBM-supplied tables or create your own, as needed, by using the IEBIMAGE utility program (see "The IEBIMAGE Utility Program" chapter for details). Also, when you create user-designed characters or when you substitute characters from the IBM-supplied "World Trade National Use Graphics" module, you must modify one or more character arrangement tables to include them.

Modifications can be made to simulate nonstandard print trains of impact printers, as well as to extend the possibilities usually offered by impact printers. One modification sometimes made, for example, is known as "dualing of graphics." One example of dualing of graphics allows the EBCDIC assignments for the four special characters on the HN print train that are not on an AN train to print when sent to a printer with the AN train. The EBCDIC codes 5D, 4D, 7D, and 7E of the HN train are assigned to print the lozenge, percent sign, at sign, and number sign, respectively, just as the codes 4C, 6C, 7C, and 7B already do on the AN train. The modification to the translate table portion of the AN character arrangement table to simulate this is shown in Figure 9.

Any character arrangement table can reference more than one character set, and any character set can be referred to by more than one table.

The AN Translate Table



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0																	0
1																	1
2																	2
3																	3
4	00											0B	2A		0E		4
5	10											1B	1C				5
6	20	21										2B	2C				6
7												3B	3C				7
8																	8
9																	9
A																	A
B																	B
C		01	02	03	04	05	06	07	08	09							C
D		11	12	13	14	15	16	17	18	19							D
E			22	23	24	25	26	27	28	29							E
F	30	31	32	33	34	35	36	37	38	39							F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

EBCDIC Assignments	Will Contain	For the Graphic
4C and 5D	2A	¤
6C and 4D	2C	%
7C and 7D	3C	@
7B and 7E	3B	#

WCGM

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0		A	B	C	D	E	F	G	H	I	¢	.	<	(	+	|	0
1	&	J	K	L	M	N	O	P	Q	R	!	$	*	)	;	¬	1
2	-	/	S	T	U	V	W	X	Y	Z	¤	,	%	_	>	?	2
3	0	1	2	3	4	5	6	7	8	9	:	#	@	'	=	"	3
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Figure 9. Dualing of Graphics, where Two Data Codes Print the Same Character

How to Change and Create Characters

The graphic character modification feature of the 3800 provides for the substitution or extension of the graphic characters in an already-defined character arrangement. A large number of IBM-supplied substitution characters are available, and you can also design and use your own characters.

You use the IEBIMAGE utility program (see the next chapter) to build your graphic character modification module, either including characters from previous modules or specifying the design of newly-created characters, or both. You then place (also using IEBIMAGE) the name of that module in the character arrangement table you are using. Then, each time that character arrangement table is used the graphic character modification is also used.

There are two IBM-supplied graphic character modification modules. One is named SPC1 and is used by the (IBM-supplied) GN and G11 character arrangement tables to replace the cent sign (¢), the exclamation mark (!), and the lozenge(¤) with the open bracket ([), the close bracket (]), and the reverse slant (). The other supplies the World Trade National Use Graphic characters.

The World Trade National Use Graphics

The standard substitution characters known collectively as the "World Trade National Use Graphics" are available on an IBM-supplied graphic character modification module. This module, since it is larger than the usual maximum for such a module, cannot be named for use by a character arrangement table. To use it, you select individual characters (using the IEBIMAGE utility program) to become part of a new graphic character modification module (or modules) that you build.

The characters in this module, with their EBCDIC and segment number assignments, are listed in "Appendix C: World Trade National Use Graphics." Example 12 in "The IEBIMAGE Utility Program" chapter shows how to list the contents of this module. Example 13 shows how to build your own graphic character modification module containing ten of these IBM-supplied World Trade characters.

User-Designed Graphic Characters

Graphic characters in addition to those supplied with the 3800 may be desired for a particular printing task. You might want to add characters from another alphabet (such as Greek or Cyrillic), or a different font (such as boldface), or to add special characters (such as chemical or electrical symbols) for printing technical reports. To do this, you should first know some details of the character generation process and the guidelines for good character design.

The Character Cell

The 3800 prints any line of characters at either 6, 8, or 12 lines per inch, as specified in the forms control buffer. Within the printable length of the page, the printing is laid down in adjoining horizontal strips that are either 1/6, 1/8, or 1/12 inch high. Each of these strips is divided into character spaces that, depending on the pitch of the character, are either 1/10, 1/12, or 1/15 inch wide. These character spaces are called character cells.

The pattern for each character is stored as 24 scan lines, each scan line containing 18 bit positions. The pitch designation for the character is stored with the pattern. When the character is printed, the size of the character cell determines which bit positions of the pattern are to be used.

As shown in Figure 10, a 10-pitch character printed at 6 lines per inch would occupy a character cell of 24 scan lines and 18 dots per scan line and would measure 1/6 inch high and 1/10 inch wide. In this case, all the bits in the stored pattern would be used. However, a 10-pitch character printed at 8 lines per inch would have to fit into a character cell of 18 scan lines and 18 dots per scan line. In such a case, only those pattern bits that match the dot positions in rows 5-22 (for 8 lines per inch) of Figure 10 would be used.

Matching Existing Characters

When you design a character to be used with other characters, you should match its appearance with the existing characters. All the graphic characters supplied by IBM follow certain rules of placement within the character cell. All characters, whatever their height, are based on a writing line (or *base line*) at row 19 (except Gothic-15 Condensed characters, which have their base line at row 18). The vertical centerline of 10- and 15-pitch characters is column 10, and for 12-pitch characters it is column 8. The vertical centerline of Gothic-15 Condensed characters is between columns 9 and 10. Underscores for 10-, 12-, and 15-pitch characters are two rows high, extending the full width of the cell, and are located on rows 21 and 22. The underscore for Gothic-15 Condensed characters is one row high, on row 20, and extends the full width of the 15-pitch cell (that is, from column 4 through column 15).

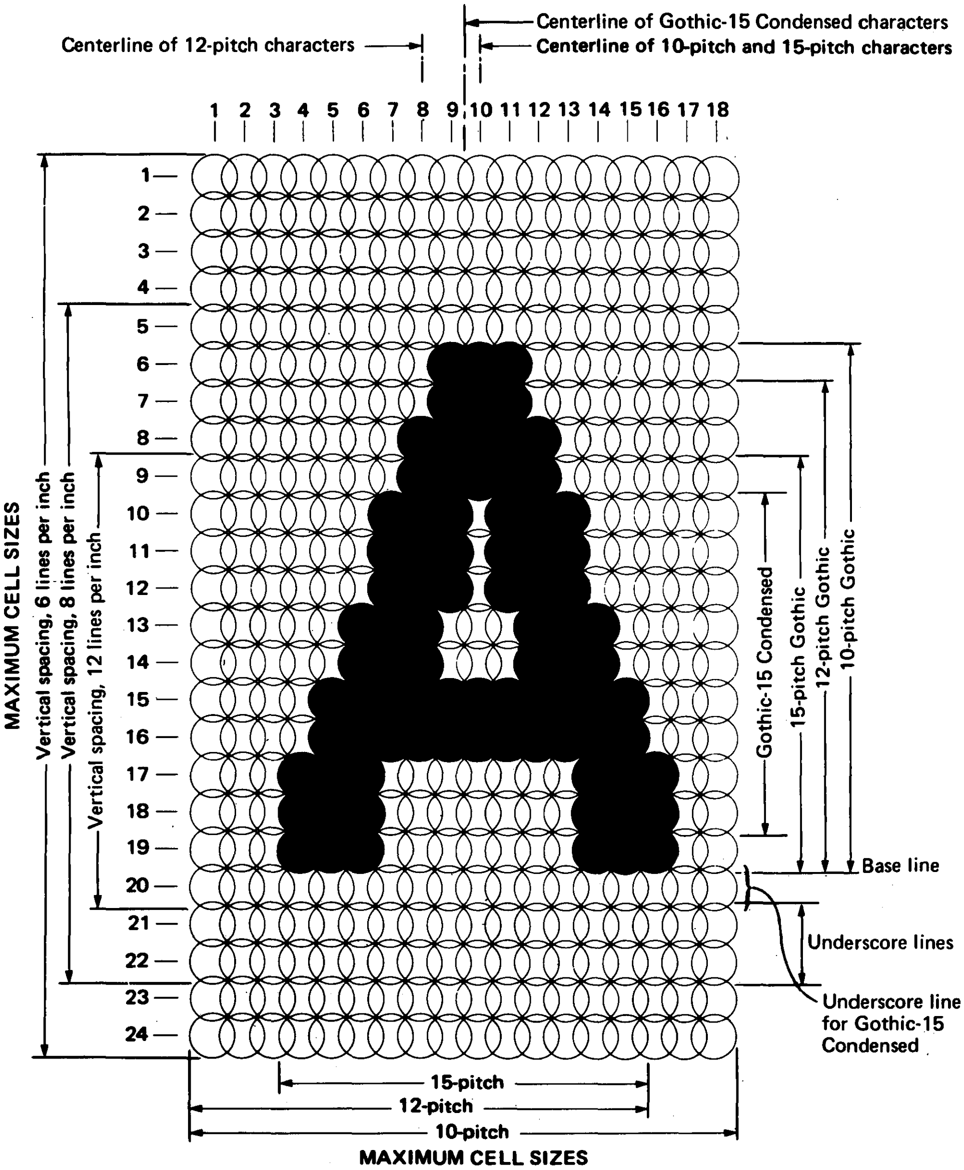

Centerline of Gothic-15 Condensed characters
Centerline of 12-pitch characters
Centerline of 10-pitch and 15-pitch characters
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
MAXIMUM CELL SIZES
Vertical spacing, 6 lines per inch
Vertical spacing, 8 lines per inch
Vertical spacing, 12 lines per inch
Gothic-15 Condensed
15-pitch Gothic
12-pitch Gothic
10-pitch Gothic
Base line
Underscore lines
Underscore line for Gothic-15 Condensed
15-pitch
12-pitch
10-pitch
MAXIMUM CELL SIZES

Figure 10. The 18-Column by 24-Row Printing Cell, with a 10-Pitch Gothic "A"

and extends the full width of the 15-pitch cell (that is, from column 4 through column 15).

Figure 11 shows, as examples, the scan patterns for some characters from the Gothic underscored, Text, and Katakana character sets. All the characters extend from the base line upward (except for descenders in lowercase Text) to a row that is appropriate for the size and pitch of the character. The rows occupied by the IBM-supplied characters are as follows:

Gothic 10-pitch and Katakana 10-pitch	row 19 through row 6
Gothic 12-pitch and Katakana 12-pitch	row 19 through row 7
Gothic 15-pitch and Katakana 15-pitch	row 19 through row 9
Gothic-15 Condensed	row 18 through row 10
Lowercase Text	main body occupies row 19 through row 10, with the ascenders rising to row 6 and the descenders dropping to row 22
Uppercase Text letters	row 19 through row 7
Text numerals and special characters	row 19 through row 6
OCR-A	variable height; row 19 through row 4 or 5, in most cases
OCR-B	variable height; row 19 through row 6 for letters, row 19 through row 5 for numerals

The position of Format characters in the character cell is shown in Figure 12. These characters extend to the edge of the cell where appropriate, to allow for the printing of continuous lines. The placement of the lines in Format characters is generally in the center of the cell but it varies with line thickness, as Figure 12 shows.

Some of the characters supplied in the graphic character modification module of World Trade National Use Graphics extend higher (or lower) than the "ordinary" Gothic or Text characters they are designed to match. This usually occurs when they have an umlaut or accent. If you are designing characters to match them or are interested in seeing their designs, you can print out the designs of the characters in that module. "Example 12: Listing the World Trade National Use Graphics Graphic Character Modification Module" in "The IEBIMAGE Utility Program" chapter shows how to do this. The designs of the characters in the IBM-supplied character sets (on the flexible disk) cannot be printed out.

Figure 13 shows how the area for the maximum character cell is partitioned for printing at various combinations of the three vertical line spacings and the three character pitches. It shows, for instance, that a 15-pitch character to be printed at 8 lines per inch must fit the rectangle formed by columns 4 and 15 and rows 5 and 22.

Character Design Procedures

The design of a character begins with a layout on the design form. Pads of the *IBM 3800 Printing Subsystem Character Design and Coding Form* are available for this purpose. Figure 14 shows, in reduced size, a copy of this design and coding form.

As the first step, you decide the size and placement of the new character. If it is to match characters in an existing set, for example, the height and center line of the new character are already defined.

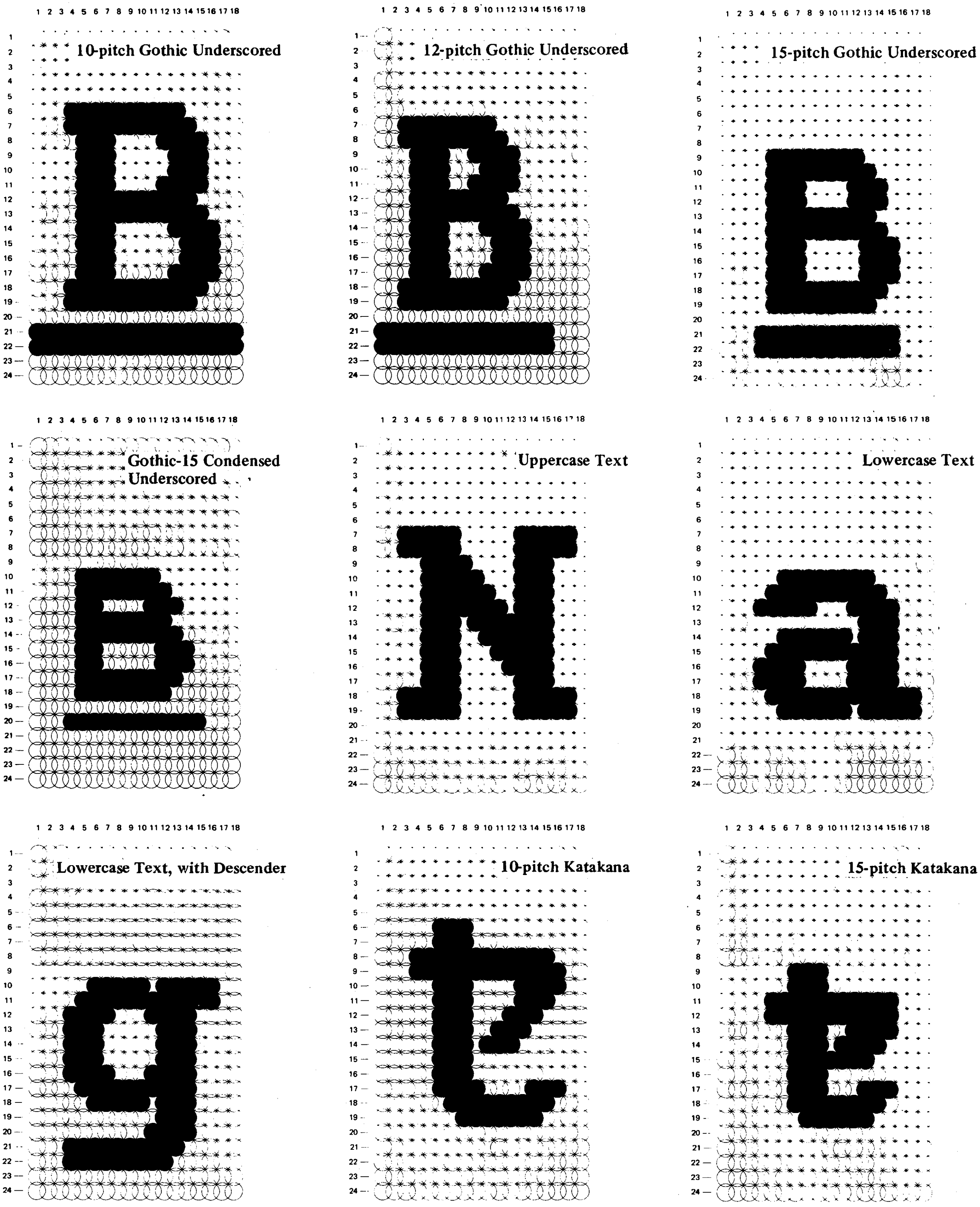


Figure 11. The Scan Patterns of some Gothic Underscored, Text, and Katakana Characters

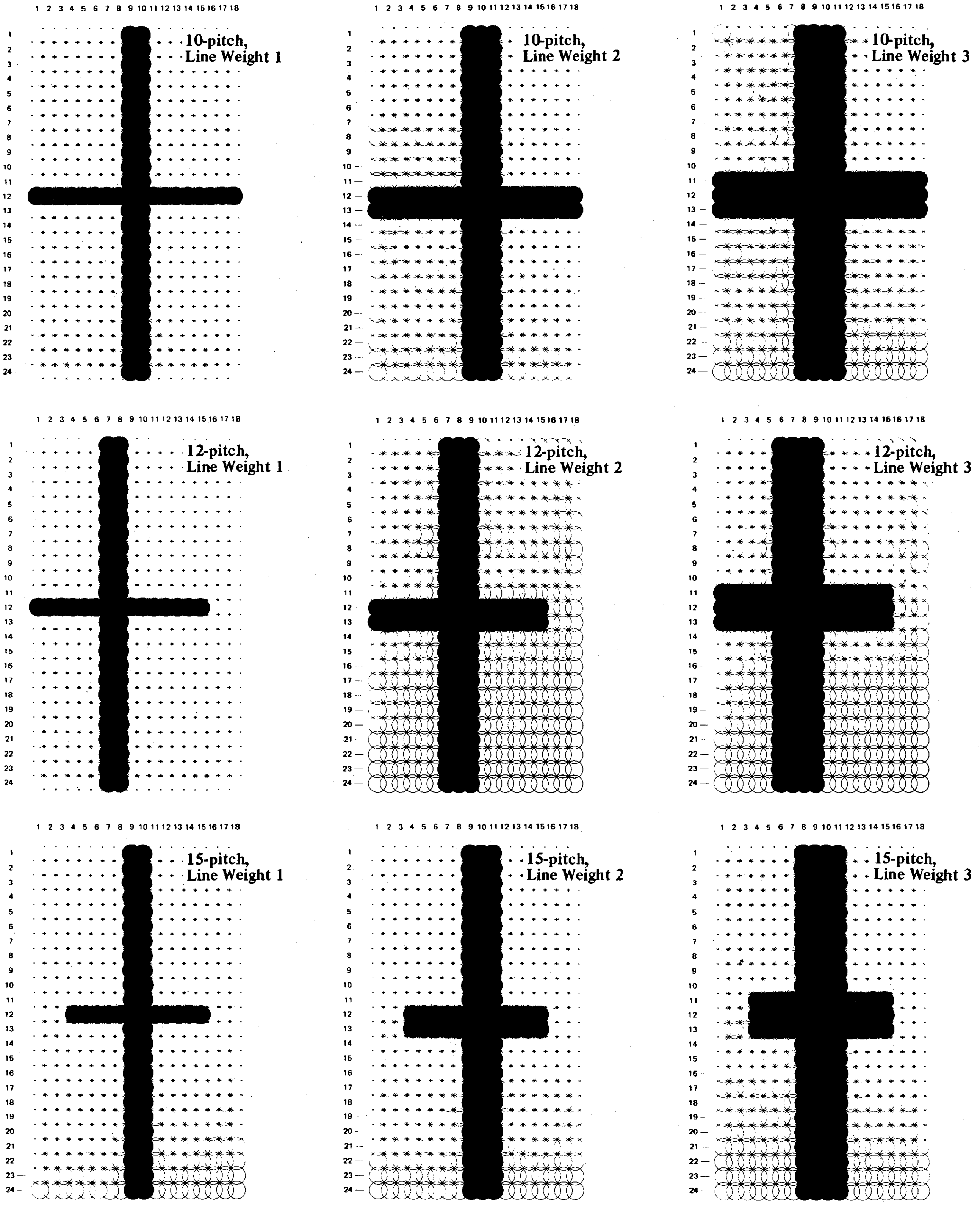


Figure 12. The Scan Patterns of the Format Intersection in Three Pitches and Line Weights

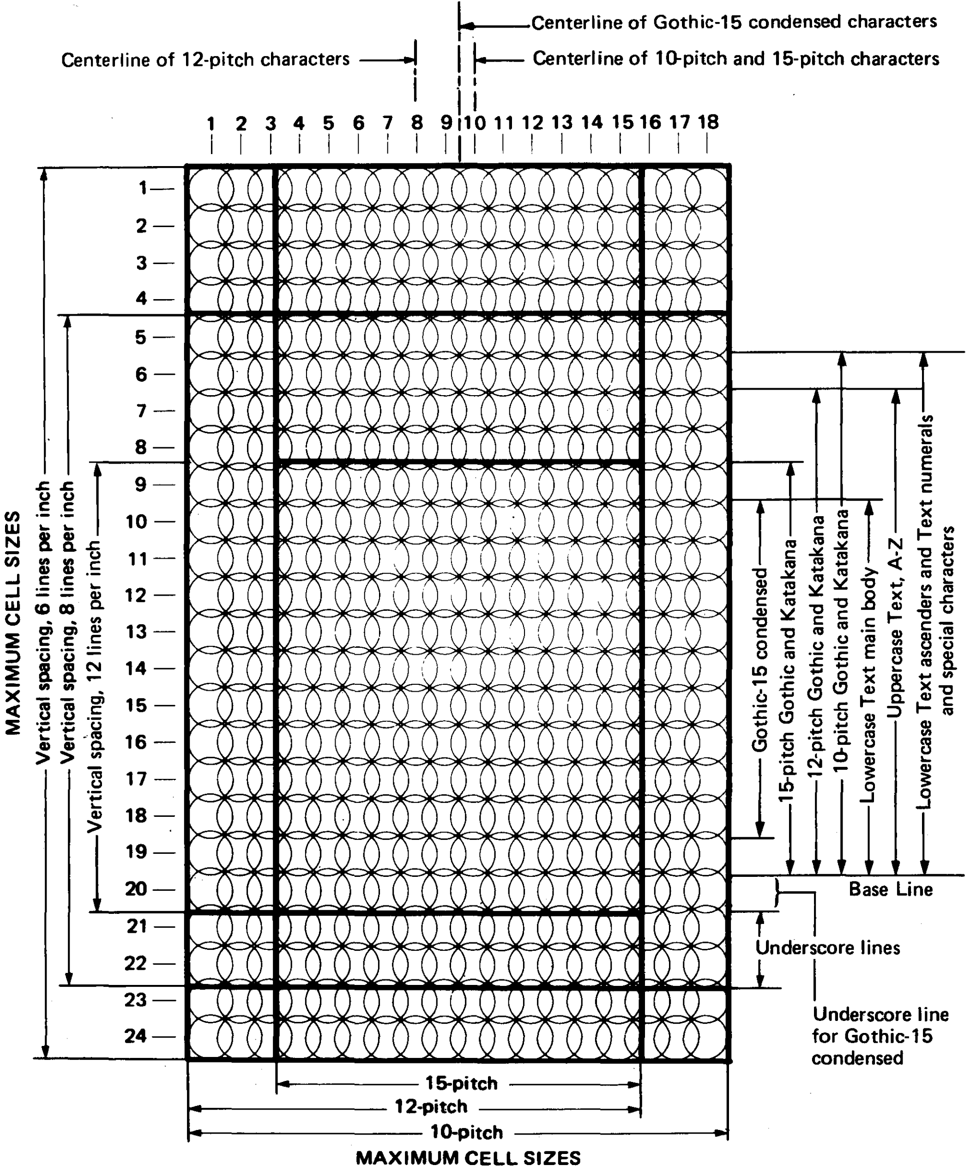
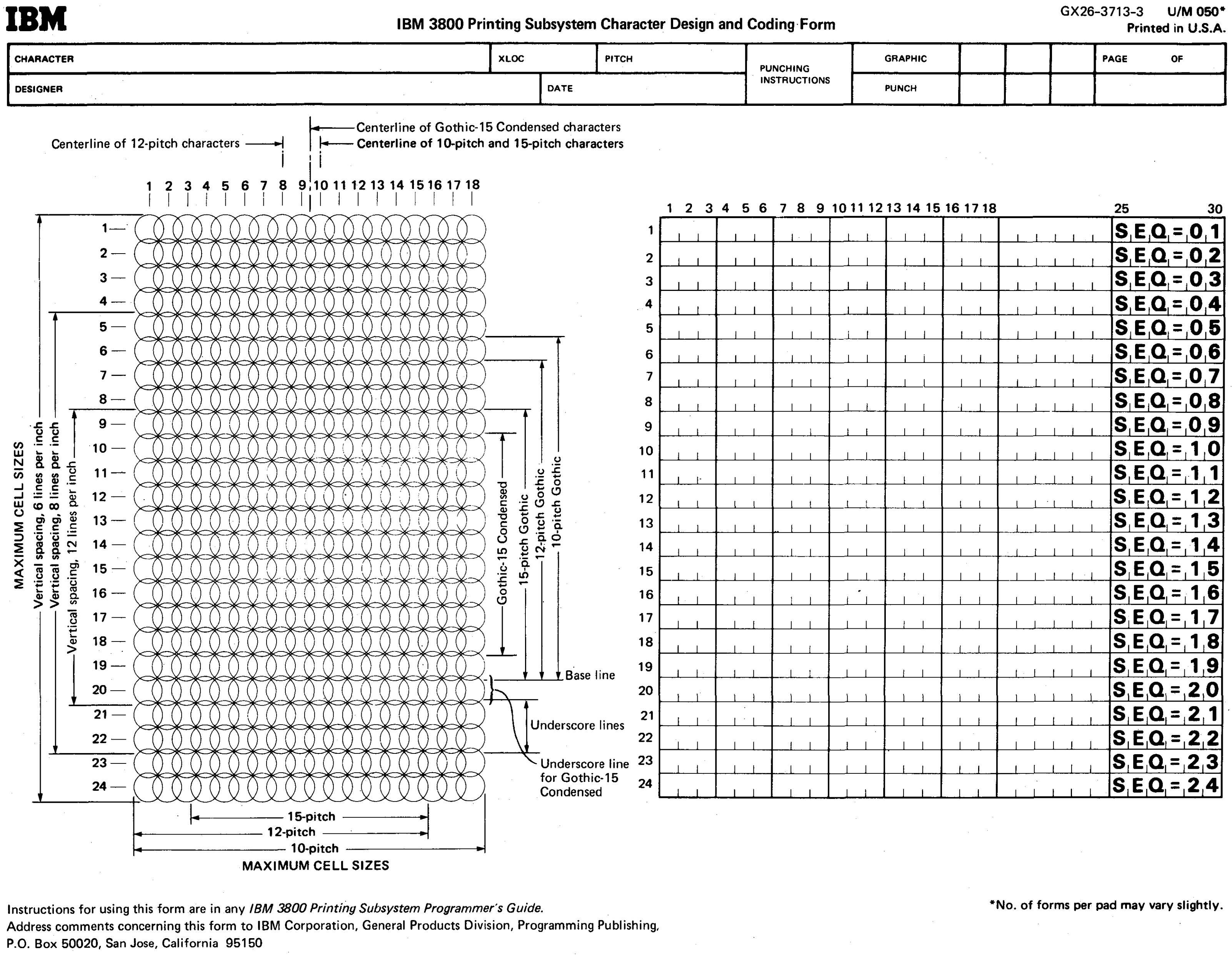

Centerline of Gothic-15 condensed characters
Centerline of 12-pitch characters
Centerline of 10-pitch and 15-pitch characters
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
MAXIMUM CELL SIZES
Vertical spacing, 6 lines per inch
Vertical spacing, 8 lines per inch
Vertical spacing, 12 lines per inch
Gothic-15 condensed
15-pitch Gothic and Katakana
12-pitch Gothic and Katakana
10-pitch Gothic and Katakana
Lowercase Text main body
Uppercase Text, A-Z
Lowercase Text ascenders and Text numerals and special characters
Base Line
Underscore lines
Underscore line for Gothic-15 condensed
15-pitch
12-pitch
10-pitch
MAXIMUM CELL SIZES

Figure 13. The Portions of the Character Cell Available for Printing at the Three Line Spacings and Three Pitches



IBM

IBM 3800 Printing Subsystem Character Design and Coding Form

GX26-3713-3 U/M 050*
Printed in U.S.A.

CHARACTER	XLOC	PITCH	PUNCHING INSTRUCTIONS	GRAPHIC				PAGE OF
DESIGNER	DATE			PUNCH				

Centerline of 12-pitch characters

Centerline of Gothic-15 Condensed characters

Centerline of 10-pitch and 15-pitch characters

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1 — 2 — 3 — 4 — 5 — 6 — 7 — 8 — 9 — 10 — 11 — 12 — 13 — 14 — 15 — 16 — 17 — 18 — 19 — 20 — 21 — 22 — 23 — 24 —

MAXIMUM CELL SIZES

Vertical spacing, 6 lines per inch

Vertical spacing, 8 lines per inch

Vertical spacing, 12 lines per inch

Gothic-15 Condensed

15-pitch Gothic

12-pitch Gothic

10-pitch Gothic

Base line

Underscore lines

Underscore line for Gothic-15 Condensed

15-pitch

12-pitch

10-pitch

MAXIMUM CELL SIZES

	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18		25 30
1			SEQ=01
2			SEQ=02
3			SEQ=03
4			SEQ=04
5			SEQ=05
6			SEQ=06
7			SEQ=07
8			SEQ=08
9			SEQ=09
10			SEQ=10
11			SEQ=11
12			SEQ=12
13			SEQ=13
14			SEQ=14
15			SEQ=15
16			SEQ=16
17			SEQ=17
18			SEQ=18
19			SEQ=19
20			SEQ=20
21			SEQ=21
22			SEQ=22
23			SEQ=23
24			SEQ=24

Instructions for using this form are in any *IBM 3800 Printing Subsystem Programmer's Guide.*
Address comments concerning this form to IBM Corporation, General Products Division, Programming Publishing,
P.O. Box 50020, San Jose, California 95150

*No. of forms per pad may vary slightly.

Figure 14. The *IBM 3800 Printing Subsystem Character Design and Coding Form*

Then sketch the character on the left half of the design form. Be sure the sketch falls within the appropriate rectangle (for pitch and lines per inch) and is centered properly. (See Figure 13.)

After you sketch the character on the design form, you select the dot pattern that best fits the character. In selecting the dot pattern to fit the character layout it is usually necessary to depart from the exact outlines of the sketch, because the 18 by 24 rectangular pattern rarely fits all the curves and angles of a character. For example, as Figure 10 shows, the sides of the letter "A" require a jagged-looking dot pattern to approximate their slope. (Note that when this dot pattern is used to print a character "A", the sides of the printed "A" appear to be smooth.)

The next step is to transfer the dot pattern to the right side of the coding form using the coordinates of the dots in the pattern on the left side of the design form. (See Figure 15.) From there, the data that represents the pattern is entered as input to the IEBIMAGE utility program using the GRAPHIC ASSIGN statement. See the "GRAPHIC Statement" section of "The IEBIMAGE Utility Program" chapter.

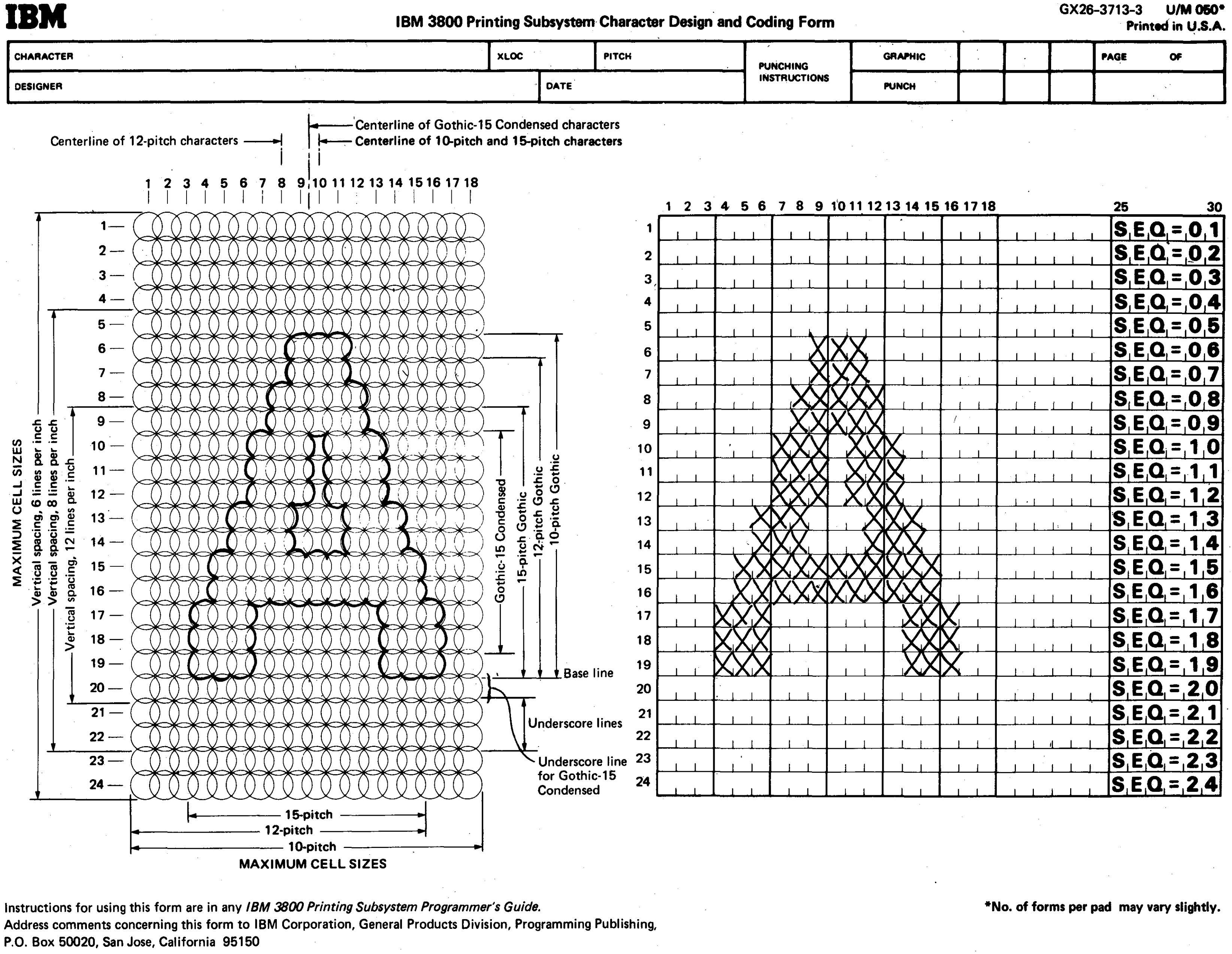
It is sometimes necessary to make other compromises in the design of a new character, some of which may only be apparent after the character has been printed. The desired effect may require modifying the design to fit the rectangular grid pattern better.

Character Design Guidelines

Although experience teaches you much about what is possible, some character design guidelines are:

- Sharp curves should be made wider if they cannot be avoided.
- Corners that are at sharp angles may sometimes be difficult to make with sharp points. You may want to modify the angle or have a less-sharp point.
- Thin lines at angles that do not match the angles of the dots in the character cell may not have the desired appearance when printed. Such lines can be made thicker, or the angle at which they are printed can be modified, or both, until the desired appearance is obtained.
- Lines or strokes in a character should be of uniform thickness, rather than tapered.
- A horizontal line made up of parallel rows of dots appears thicker than a vertical line of the same number of parallel columns of dots. When the appearance of equal line width is wanted, a usually satisfactory ratio is three columns of dots to two rows.
- Single dots should not be used, nor should a vertical column of single dots be used. They may not be visible when printed.
- It is recommended that vertical lines be no thicker than five dots and horizontal lines no thicker than four dots. Lines that exceed these recommended dimensions may not be uniformly black when printed; the centers of the lines may be lighter than the edges.

To illustrate these design guidelines, Figure 16 shows the scan patterns for a number of different characters.



IBM

IBM 3800 Printing Subsystem Character Design and Coding Form

GX26-3713-3 U/M 050*
Printed in U.S.A.

CHARACTER	XLOC	PITCH	PUNCHING INSTRUCTIONS	GRAPHIC				PAGE OF
DESIGNER	DATE			PUNCH				

SEQ=01
SEQ=02
SEQ=03
SEQ=04
SEQ=05
SEQ=06
SEQ=07
SEQ=08
SEQ=09
SEQ=10
SEQ=11
SEQ=12
SEQ=13
SEQ=14
SEQ=15
SEQ=16
SEQ=17
SEQ=18
SEQ=19
SEQ=20
SEQ=21
SEQ=22
SEQ=23
SEQ=24

Instructions for using this form are in any *IBM 3800 Printing Subsystem Programmer's Guide.*
Address comments concerning this form to IBM Corporation, General Products Division, Programming Publishing,
P.O. Box 50020, San Jose, California 95150

*No. of forms per pad may vary slightly.

Figure 15. Transferring the Dot Pattern to the Right Side of the Design Form

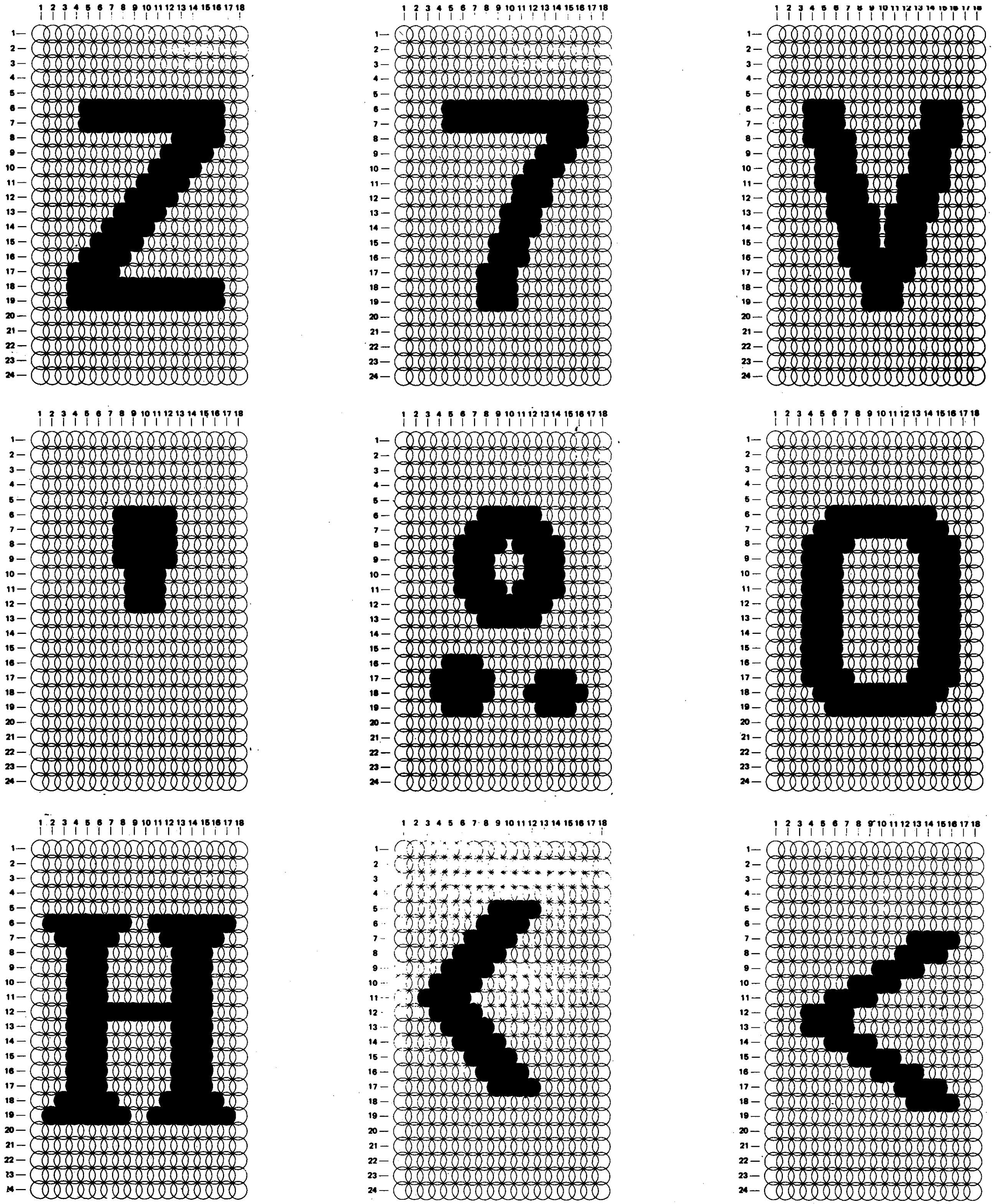
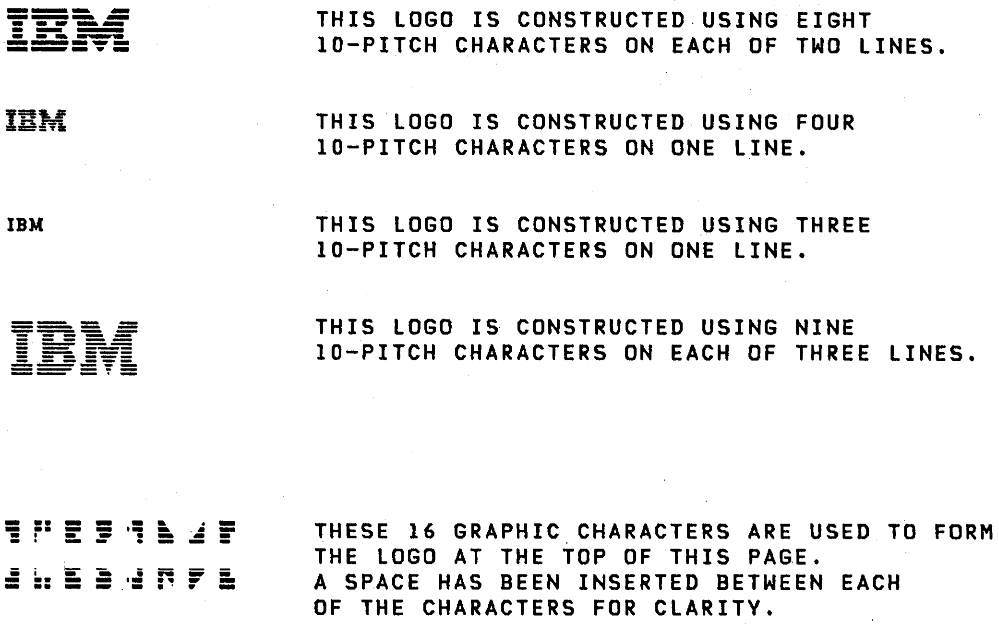


Figure 16. Some Scan Patterns to Illustrate the Design Guidelines

Composite Designs

Since printing in the IBM 3800 is in continuous strips and cells that adjoin, you can print designs that are larger than a single character cell. Figure 17 shows four such examples, composed of 10-pitch characters printed at 6 lines per inch to make composite characters.



```
THIS LOGO IS CONSTRUCTED USING EIGHT
10-PITCH CHARACTERS ON EACH OF TWO LINES.

THIS LOGO IS CONSTRUCTED USING FOUR
10-PITCH CHARACTERS ON ONE LINE.

THIS LOGO IS CONSTRUCTED USING THREE
10-PITCH CHARACTERS ON ONE LINE.

THIS LOGO IS CONSTRUCTED USING NINE
10-PITCH CHARACTERS ON EACH OF THREE LINES.

THESE 16 GRAPHIC CHARACTERS ARE USED TO FORM
THE LOGO AT THE TOP OF THIS PAGE.
A SPACE HAS BEEN INSERTED BETWEEN EACH
OF THE CHARACTERS FOR CLARITY.
```

Figure 16. Four Logos Constructed Using 10-Pitch Characters

How to Request Multiple Copies

The 3800 prints multiple copies on single-ply paper—every copy is an original. Thus there is no need for multiple-ply forms or for the subsequent deleaving and the disposal of the carbon paper.

The number of copies wanted and, optionally, whether copies are to be grouped with more than one copy of each page together, are specified using COPIES parameter:

COPIES=(*nnn* **[,(** *group value,group value ...* **)])**

where

nnn
is the total number of copies to be printed, the maximum not exceeding 255. If the COPIES keyword is omitted, one copy is printed.

group value
describes how the printed copies are to be grouped. Each *group value* specifies the number of copies of each individual page that is to be printed in a group before starting the printing of the next page. Up to eight *group values* can be specified. No single *group value* can exceed 255, nor can the sum of those specified exceed 255. When *group values* are specified for SYSOUT data sets, their sum determines the number of copies printed (that is, if one or more *group values* are coded, *nnn* is ignored).

Retransmission of the data set over the channel is necessary when more than one group is specified in the JCL DD statement, or when multiple copies without *group values* are specified. For SYSOUT data sets, the retransmission is done by the output printer.

For information on defaults and rules for coding, see "The COPIES Parameter" in the "JCL" section of *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

The following are examples of the use of the COPIES keyword for SYSOUT data sets:

COPIES=14

In this example, 14 copies of each page in the data set are printed, and each copy is in page number sequence. When the printing of the data set is completed, there are 14 separate but identical copies of the same data set. The complete data set is transmitted over the channel 14 times. If there were three pages in the data set, for example, the output would be as shown in Figure 18.

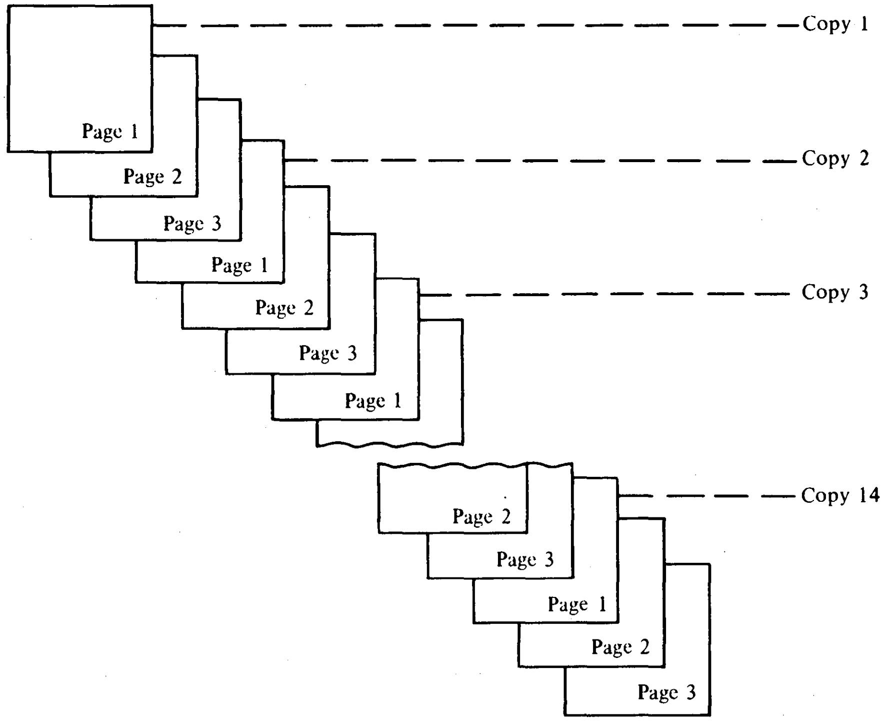


Figure 18. Output in Response to COPIES=14

In this example, if your data set is not a SYSOUT data set (that is, if the printer is directly allocated) the number of copies defaults to 1. To get 14 copies, you must retransmit each copy of the data set (using the REXMIT parameter of the SETPRT macro).

COPIES=(6,(1,3,2))

In this example, 6 copies of the data set are to be printed with three groups being produced. The first group contains one copy of the data set, the second group contains three copies of each page of the data set, and the third contains two copies of each page of the data set. The complete data set is transmitted over the channel three times, once for each group. If there were three pages in the data set, for example, the output would be as shown in Figure 19.

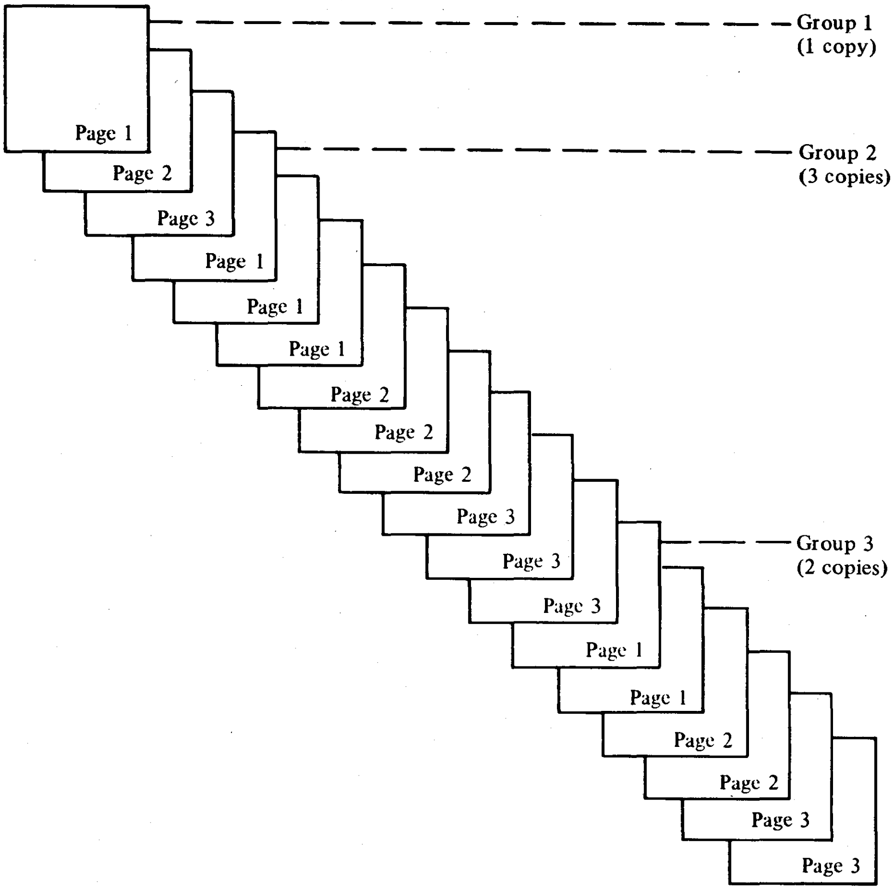


Figure 19. Output in Response to COPIES=(6,(1,3,2))

If your data set is directly allocated rather than a SYSOUT data set, the first *group value* determines the number of copies. In this example, only one copy of the data set is printed, unless you retransmit additional copies. If the first *group value* specification was 2, two copies of each page would be printed before printing two copies of the next page.

If other features such as copy modification or forms overlay flashing are used when requesting multiple copies, the number of copies requested using the COPIES parameter is still the number printed. The number of copies being modified or flashed cannot change the number being printed. More copies can be printed than are flashed or modified.

How to Modify Copies

The copy modification feature allows printing of predefined data on a specified copy or copies of all pages of a data set. Examples of such data are legends, explanatory remarks and column headings, which might vary from copy to copy of the same printed page of data. Copies may also be personalized with the recipient's name, department, and other desired information.

Blanks or printable characters, such as asterisks, can be used as the copy modification data to suppress the printing of variable data on particular copies of a page. An example of this is a manufacturer's unit cost, which is required on a file copy of an invoice but not on the customer's copy. This is a function done in other printers by means of short or spot carbon in the forms set.

The predefined data for copy modification is created as a copy modification module on SYS1.IMAGELIB using the IEBIMAGE utility program (see "The IEBIMAGE Utility Program" chapter for details). You specify the copy numbers, line numbers, and print positions to be modified as well as the modifying data. The module you create is then used by specifying its name in your JCL with the MODIFY parameter:

MODIFY=(*module name* [, *trc*])

where

module name
is the 1- to 4-character name of the copy modification module stored in the SYS1.IMAGELIB system data set.

trc
is the table reference character (0, 1, 2, or 3) used to select one of the character arrangement tables specified with the CHARS keyword to be used for the copy modification text. The values of 0, 1, 2, or 3 correspond to the order in which the table names have been specified. If *trc* is not specified, reference character 0 is assumed.

For further coding rules and examples, see the "MODIFY Parameter" section in *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

The COPIES parameter in your JCL controls the number of copies printed, so if a copy modification module specifies modification of more copies than are being printed, the request is ignored for those copies that are not printed.

When using copy modification, the copy modification text completely replaces the data text in the specified positions. You can inadvertently overlay valid data and not realize it. You should ensure that the data to be printed in a field does not overflow the size of that field, into a position where it will be overlaid by a copy modification. There is no notification when this happens. For example, if you use copy modification with Format characters to print a box around a data field, then you cannot allow the data in that field to expand and overflow the number of available print positions.

Good documentation is important to ensure the usability of copy modification modules by future users. Besides recording the purpose of a module you create and the program it is to be used with, you should record the name of the character arrangement table that is to be used with the module, especially if it is different from that used to print the other data. The modifying text of a module intended for use with Format characters, for example, might look rather strange if printed using Gothic characters.

How to Flash Forms

The forms overlay feature provides the ability to print (in black) a form, grid, design, or constant data on paper as it is being processed through the 3800. This allows the use of blank forms rather than forms pre-printed for specific requirements, and reduces job set-up time. A frame holding the overlay negative for the form must be inserted into the printer before printing begins. There is no verification by the system that the requested forms overlay frame is inserted. For information on designing and making or obtaining forms overlays, see *Forms Design Reference Guide for the IBM 3800 Printing Subsystem.*

The FLASH parameter is used in JCL to identify the overlay to be used and to allow the specification of the number of copies on which the overlay is to be printed. A message to the operator is printed asking him to insert that particular forms overlay frame into the 3800.

FLASH=(*overlay name* [, *count* **])**

where

overlay name
is the 1- to 4-character name of the forms overlay frame.

count
beginning with the first copy printed, is the number of consecutive copies on which the overlay will be printed. The maximum value of *count* is the value of *nnn* or the sum total of *group values* in the COPIES keyword. If the *count* value is not specified with the FLASH keyword, all copies have the overlay printed.

For coding rules and examples, see the "FLASH Parameter" section in *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

How to Use Format Characters

The Format character sets consist of lines, corners, junctions, and intersections in three different line weights and in 10, 12, and 15 pitch. Format characters can be used to print horizontal and vertical lines, boxes around data or blank space, bar graphs, and so on.

Each Format character uses its own print position and thus cannot be combined with another character in that same position. The lines of the Format characters are centered within the print position, either horizontally or vertically or both. The combination of the corners and vertical and horizontal lines forms a box. If junctions and intersections are also used, multiple adjoining boxes can be formed.

Format characters are generally used together on a page with other characters. You can do this in a number of ways. For instance, specifying the name of the desired Format character arrangement table (FM10, FM12, or FM15) as one of two or more that you name with the CHARS keyword, you can:

- Print Format characters as the data in a copy modification module (as described above under "How to Modify Copies")

- Intermix and merge multiple data lines using table reference characters with OPTCD=J in your DCB (as described under "Intermixing Style or Pitch Using the Table Reference Character and OPTCD=J")

For ease in entering them, the Format characters have 8-bit data code assignments that are the same as the EBCDIC for the alphanumerics A-Z and 0-9. The last figure in "Appendix B: IBM-Supplied Character Arrangement Tables" shows the Format characters and their 8-bit data code and WCGM assignments.

When compared with forms overlay, the use of the Format characters has both advantages and disadvantages. For each Format character printed, a print position is required as well as character generation storage. Usually, some programming effort is also required. However, relatively simple forms can be generated using Format characters without the time and expense to prepare a forms overlay negative. Also, no operator intervention is required, as it is when inserting a forms overlay frame. And, forms produced using Format characters can be changed between pages of a data set, or printed on some pages and not others, while forms overlays cannot be.

How to Specify Forms Control

The forms control buffer (FCB) in the 3800 specifies control for the vertical format of the page, replacing the paper carriage-control tape used in some impact printers. It permits printing at either 6, 8, or 12 lines per inch, or an intermix of these line spacings on the same page.

The FCB module is created in the SYS1.IMAGELIB system data set using the IEBIMAGE utility program (see "The IEBIMAGE Utility Program" chapter for details). For use, it is loaded into the 3800 by specifying its name with the FCB parameter on your output data set's JCL DD statement. For details on use of the FCB parameter, see "FCB Parameter" in *OS/VS2 SVS Independent Component: IBM 3800 Printing Subsystem System Information.*

If you do not specify an FCB in your JCL, the default is the FCB specified in the then-active writer procedure. If the writer procedure does not specify an FCB or no writer procedure is active, the default is 6 lines per inch on whatever paper size is then in the 3800, with a channel-1 code defined in the first printable line.

The FCB specifications are loaded before any data is transferred from the processor to the 3800. Before printing begins, the printer verifies that the paper length specified in the FCB agrees with the length of the paper that is then in the 3800.

Print Control Characters and Channel Codes

A print control character (either a machine code control character or an ANS control character) is used to initiate a write and/or skip (or space) operation. To terminate the skip operation, a channel code corresponding to the print control character must be coded in the FCB to mark the appropriate line of the form. For example, if a "Skip to Channel 1" print control character is coded in the logical record (the output data line), the FCB should contain a channel 1 code to terminate the operation.

Print control characters are optional. Their presence is indicated by coding either A, for American National Standard (ANS) control characters, or M, for machine code control characters, together with the record format specification in the RECFM subparameter of the DCB parameter. For format-F and format-U records, the control character is the first byte of the logical record. For format-V records, it must be the fifth byte of the logical record, immediately following the record descriptor word.

The control characters that can be used are:

ACTION	Machine Code Control Characters		ANS Control Characters
	Print then Action	**Action Only (Immediate)**	**Action Before Printing**
Print only (no space)	01	-	4E(+)
Space 1 line	09	0B	40 (ƀ)
Space 2 lines	11	13	F0 (0)
Space 3 lines	19	1B	60 (–)
Skip to Channel 1	89	8B	F1 (1)
Skip to Channel 2	91	93	F2 (2)
Skip to Channel 3	99	9B	F3 (3)
Skip to Channel 4	A1	A3	F4 (4)
Skip to Channel 5	A9	AB	F5 (5)
Skip to Channel 6	B1	B3	F6 (6)
Skip to Channel 7	B9	BB	F7 (7)
Skip to Channel 8	C1	C3	F8 (8)
Skip to Channel 9	C9	CB	F9 (9)
Skip to Channel 10	D1	D3	C1 (A)
Skip to Channel 11	D9	DB	C2 (B)
Skip to Channel 12	E1	E3	C3 (C)

All codes are shown in hexadecimal. The graphic representations of the ANS codes are also shown in parentheses.

With the 3800, the presence of a channel 9 code or a channel 12 code in the FCB can be sensed during the execution of a line space operation (but not a line skipping operation). A channel 9 code sets a unit check status bit in the channel status word (CSW) and a channel 9 bit in sense byte 0. A channel 12 code sets a unit exception status bit in the CSW.

If a channel code is not specified in the FCB module and a "Skip" channel command to that channel is issued, the 3800 advances the form to the first printable line on the next page, and sets I/O error indicators: "Unit Check" and "Data Check." This data check cannot be blocked using the OPTCD=B parameter of the SETPRT macro.

How to Burst and Offset Stack Output

The BURST parameter is used to specify whether the paper output is to go to the optional Burster-Trimmer-Stacker or not. The keyword causes a message to be printed on the system console telling the operator to thread the paper into the Burster-Trimmer-Stacker or the continuous forms stacker, unless he was so requested by the last such message.

BURST={Y | N}

where

Y
indicates that the printed output is to be burst into separate sheets.

N
indicates that the output is to be continuous fanfold.

When you are directing your output to the Burster-Trimmer-Stacker and using the 3800 as a system printer, the output writer causes the stacked sheets of each data set to be offset from those of the preceding data set. Copies and groups of copies are also offset. When using the 3800 as a direct output printer, you get offset stacking by issuing an End of Transmission channel command.

THE IEBIMAGE UTILITY PROGRAM

The IEBIMAGE utility program creates and maintains the following types of modules in the SYS1.IMAGELIB system data set:

- Forms control buffer modules, which specify controls for the vertical line spacing, and any one of 12 channel codes per line
- Copy modification modules, which specify data that is to be printed on every page for specified copies of the output data set
- Character arrangement table modules, which translate the input data into printable characters, and identify the associated character set(s) and graphic character modification module(s)
- Graphic character modification modules, which contain the scan patterns of user-designed characters and/or of characters from IBM-supplied modules

For the structure of each of the four types of modules, see "The Structure of Modules Produced by IEBIMAGE".

For the notational conventions used in this chapter, see "Notational Conventions" in the "Preface."

When IEBIMAGE processing is completed, the program generates messages that indicate successful completion or errors. Each message is described in "Appendix G: IEBIMAGE Program Messages."

Forms Control Buffer (FCB) Modules

The FCB module is variable-length and contains vertical line spacing information (6, 8, or 12 lines per inch). The FCB module can also identify one of 12 carriage-control channel codes for each line. The module is identified with a 1- to 4-character name specified when it is built. For example, if the name you assign is PAY1 then you code FCB=PAY1 when you use JCL to load the FCB module into the 3800.

The FCB module is created and stored in SYS1.IMAGELIB with the IEBIMAGE program, using the FCB and NAME utility control statements. The FCB statement is used to describe the FCB module's contents. The NAME statement is used to identify the FCB module and to indicate whether it is new or is to replace an existing module with the same name. You can also use the IEBIMAGE program to display an existing FCB module's contents.

Copy Modification Modules

The copy modification module contains predefined data for modifying some or all copies of an output data set. Segments of the module contain predefined text, its position on each page of the output data set, and the copy or copies the text applies to. The copy modification module is identified with a 1- to 4-character name specified when it is built.

The copy modification module is created with the IEBIMAGE program, using the INCLUDE, OPTION, COPYMOD, and NAME utility control statements. The INCLUDE statement identifies a module that is to be copied and used as a basis for the newly-created module. The OPTION statement with the OVERRUN parameter allows the user to suppress the printing of line overrun condition messages for those vertical line spacings that are not applicable to the job. The COPYMOD statement is used to describe the contents of one of the new module's segments. The NAME statement is used to identify the new module and to indicate whether it is new or is to replace an existing module with the same name. You can code more than one COPYMOD statement within an operation group; all COPYMOD statements so coded apply to the same copy modification module. You can also use the IEBIMAGE program to display an existing copy modification module.

Character Arrangement Tables

The character arrangement table module is fixed-length and consists of three sections:

- System control information, which contains the module's name and length.
- The translate table, which contains 256 one-byte translate table entries, corresponding to the 8-bit data codes (X'00' through X'FF'). A translate table entry can identify one of 64 character positions in any one of four writable character generation modules. The code X'FF' is reserved to indicate an unprintable character. When an entry of X'FF' is detected by the printer as a result of attempting to translate an invalid 8-bit data code, the printer prints a blank and sets the data-check indicator (unless the block-data-check option is in effect).
- Identifiers, which identify the character sets and the graphic character modification modules associated with the character arrangement table.

The character arrangement table is created with the IEBIMAGE program, using the INCLUDE, TABLE, and NAME utility control statements. The INCLUDE statement identifies an existing character arrangement table that is to be copied and used as a basis for the new module. The TABLE statement describes the new or modified module's contents. The NAME statement identifies the character arrangement table, and indicates whether it is new or is to replace an existing module with the same name. The IEBIMAGE program can also be used to print a character arrangement table.

"Appendix B: IBM-Supplied Character Arrangement Tables" describes each IBM-Supplied character arrangement table in detail. The character sets associated with each character arrangement table are described in "Appendix A: IBM-Supplied Character Sets."

Note: All characters in a character set *might not* be referred to by the character arrangement table you select. The character arrangement table corresponds to a print train, which is sometimes a subset of one or more complete character sets. When the character set is loaded, all characters of the set (up to 64) are loaded into the printer's WCGM; only those characters that are referenced by a translate table can be printed.

Graphic Character Modification Modules

The graphic character modification module is variable-length and contains up to 64 segments, each of which contains a graphic character's 8-bit data code and its 72-byte scan pattern.

The graphic character modification module is created using the IEBIMAGE program's INCLUDE, GRAPHIC, and NAME utility control statements. The INCLUDE statement identifies an existing graphic character modification module to be copied and used as a basis for the new module. A GRAPHIC statement, when followed by one or more *data statements,* specifies the design of a user-designed character. A GRAPHIC statement can also select a character segment from another graphic character modification module. The NAME statement identifies the module, and indicates whether it is new, or is to replace an existing module. More than one GRAPHIC statement can be coded between the INCLUDE and NAME statements; all GRAPHIC statements so coded apply to the same graphic character modification module. The IEBIMAGE program can also be used to print a graphic character modification module.

Input and Output for IEBIMAGE

The IEBIMAGE utility program uses as input the utility control statements: INCLUDE, OPTION, FCB, COPYMOD, TABLE, GRAPHIC, and NAME. The GRAPHIC statement can be followed by one or more *data statements.* Modules of the SYS1.IMAGELIB system data set can also be part of the IEBIMAGE program's input.

The IEBIMAGE utility program produces one or more modules that are stored in the SYS1.IMAGELIB data set. The IEBIMAGE program alsc produces an output data set (a listing) that includes the program's identification, contents of the utility control statements, results of the program's processing, and messages and return codes.

Messages produced by the IEBIMAGE program are described in "Appendix G: IEBIMAGE Program Messages." Note that a single input statement could cause multiple error messages.

IEBIMAGE produces a return code which represents the severest error condition encountered during the program execution. This return code is printed in the output listing and placed in register 15. The significance of the code values is as follows:

Return Code	**Description of Results**
00(00)	Successful completion; operation(s) performed as requested
04(04)	Operation(s) performed; investigate messages for exceptional circumstances
08(08)	Operation(s) not performed; investigate messages
12(0C)	Severe exception; utility may terminate
16(10)	Catastrophic exception; utility terminated
20(14)	SYSPRINT data set could not be opened; utility terminated
24(18)	User parameter list invalid; utility is terminated

Operation Groups

IEBIMAGE utility control statements are grouped together to create or print a SYS1.IMAGELIB module. Each group of statements is called an *operation group.* Your job's input stream might include many operation groups. The utility control statements (shown below without operands coded) that can be coded are:

- To create or print an FCB module:

  FCB
  NAME

- To create or print a copy modification module:

  [INCLUDE]
  [OPTION]
  COPYMOD
  [additional **COPYMOD** statements]
  NAME

- To create or print a character arrangement table module:

  [INCLUDE]
  TABLE
  NAME

- To create or print a graphic character modification module:

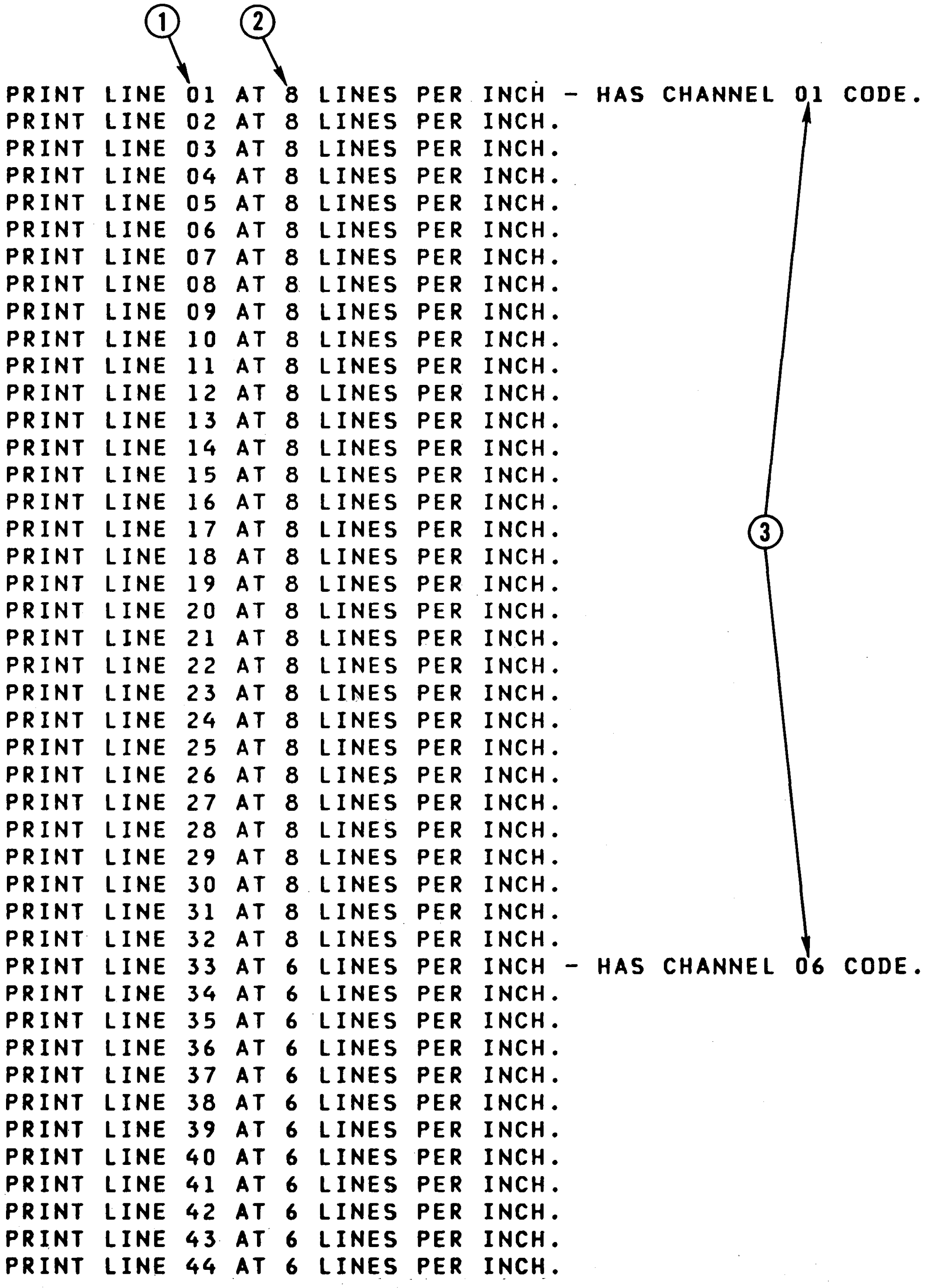
  [INCLUDE]
  {**GRAPHIC** | **GRAPHIC**, followed immediately by data statements}
  [additional **GRAPHIC** statements]
  NAME

To print any module, you need only supply the function statement (that is, **FCB, COPYMOD, TABLE,** or **GRAPHIC**) with no operands specified, followed by the **NAME** statement naming the module.

The following sections describe each utility control statement in detail.

Output For Each Operation Group

When the IEBIMAGE utility program builds a SYS1.IMAGELIB module, the program lists the module's contents. Figure 20 shows the IEBIMAGE listing of an FCB module. Figure 21 shows the listing of three segments of a copy modification module. This listing shows only the positioning of the modifying text. To print out the text itself, you can use the IEBPTPCH utility program. Figure 22 shows the listing of a character arrangement table module. Figure 23 shows the listing of two segments of a graphic character modification module. In each figure, the numbered notes describe the item indicated with the same circled number on the listing.



```
PRINT LINE 01 AT 8 LINES PER INCH - HAS CHANNEL 01 CODE.
PRINT LINE 02 AT 8 LINES PER INCH.
PRINT LINE 03 AT 8 LINES PER INCH.
PRINT LINE 04 AT 8 LINES PER INCH.
PRINT LINE 05 AT 8 LINES PER INCH.
PRINT LINE 06 AT 8 LINES PER INCH.
PRINT LINE 07 AT 8 LINES PER INCH.
PRINT LINE 08 AT 8 LINES PER INCH.
PRINT LINE 09 AT 8 LINES PER INCH.
PRINT LINE 10 AT 8 LINES PER INCH.
PRINT LINE 11 AT 8 LINES PER INCH.
PRINT LINE 12 AT 8 LINES PER INCH.
PRINT LINE 13 AT 8 LINES PER INCH.
PRINT LINE 14 AT 8 LINES PER INCH.
PRINT LINE 15 AT 8 LINES PER INCH.
PRINT LINE 16 AT 8 LINES PER INCH.
PRINT LINE 17 AT 8 LINES PER INCH.
PRINT LINE 18 AT 8 LINES PER INCH.
PRINT LINE 19 AT 8 LINES PER INCH.
PRINT LINE 20 AT 8 LINES PER INCH.
PRINT LINE 21 AT 8 LINES PER INCH.
PRINT LINE 22 AT 8 LINES PER INCH.
PRINT LINE 23 AT 8 LINES PER INCH.
PRINT LINE 24 AT 8 LINES PER INCH.
PRINT LINE 25 AT 8 LINES PER INCH.
PRINT LINE 26 AT 8 LINES PER INCH.
PRINT LINE 27 AT 8 LINES PER INCH.
PRINT LINE 28 AT 8 LINES PER INCH.
PRINT LINE 29 AT 8 LINES PER INCH.
PRINT LINE 30 AT 8 LINES PER INCH.
PRINT LINE 31 AT 8 LINES PER INCH.
PRINT LINE 32 AT 8 LINES PER INCH.
PRINT LINE 33 AT 6 LINES PER INCH - HAS CHANNEL 06 CODE.
PRINT LINE 34 AT 6 LINES PER INCH.
PRINT LINE 35 AT 6 LINES PER INCH.
PRINT LINE 36 AT 6 LINES PER INCH.
PRINT LINE 37 AT 6 LINES PER INCH.
PRINT LINE 38 AT 6 LINES PER INCH.
PRINT LINE 39 AT 6 LINES PER INCH.
PRINT LINE 40 AT 6 LINES PER INCH.
PRINT LINE 41 AT 6 LINES PER INCH.
PRINT LINE 42 AT 6 LINES PER INCH.
PRINT LINE 43 AT 6 LINES PER INCH.
PRINT LINE 44 AT 6 LINES PER INCH.
```

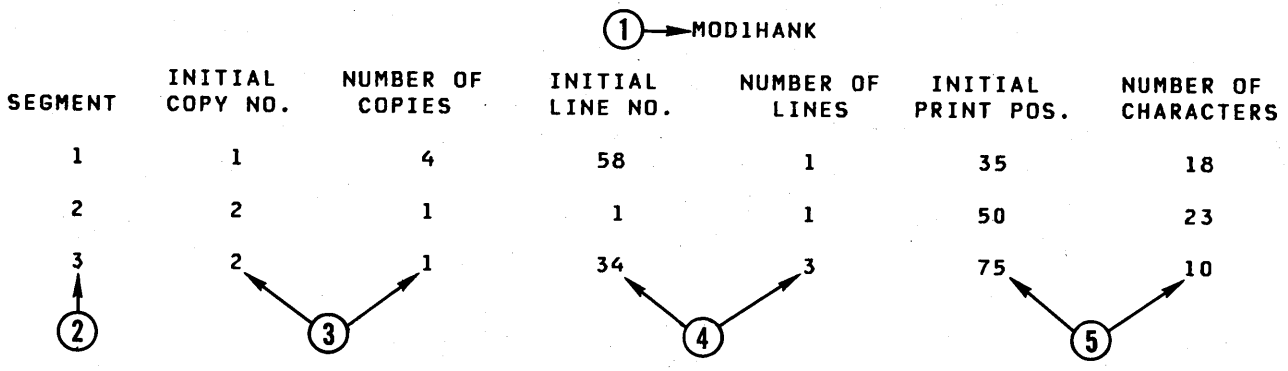
Notes:

1. The line number. Each line of the form is listed in this fashion.

2. The vertical spacing of the line, in lines per inch.

3. The channel code, printed for each line that includes a channel code.

Figure 20. IEBIMAGE Listing of a Forms Control Buffer Module



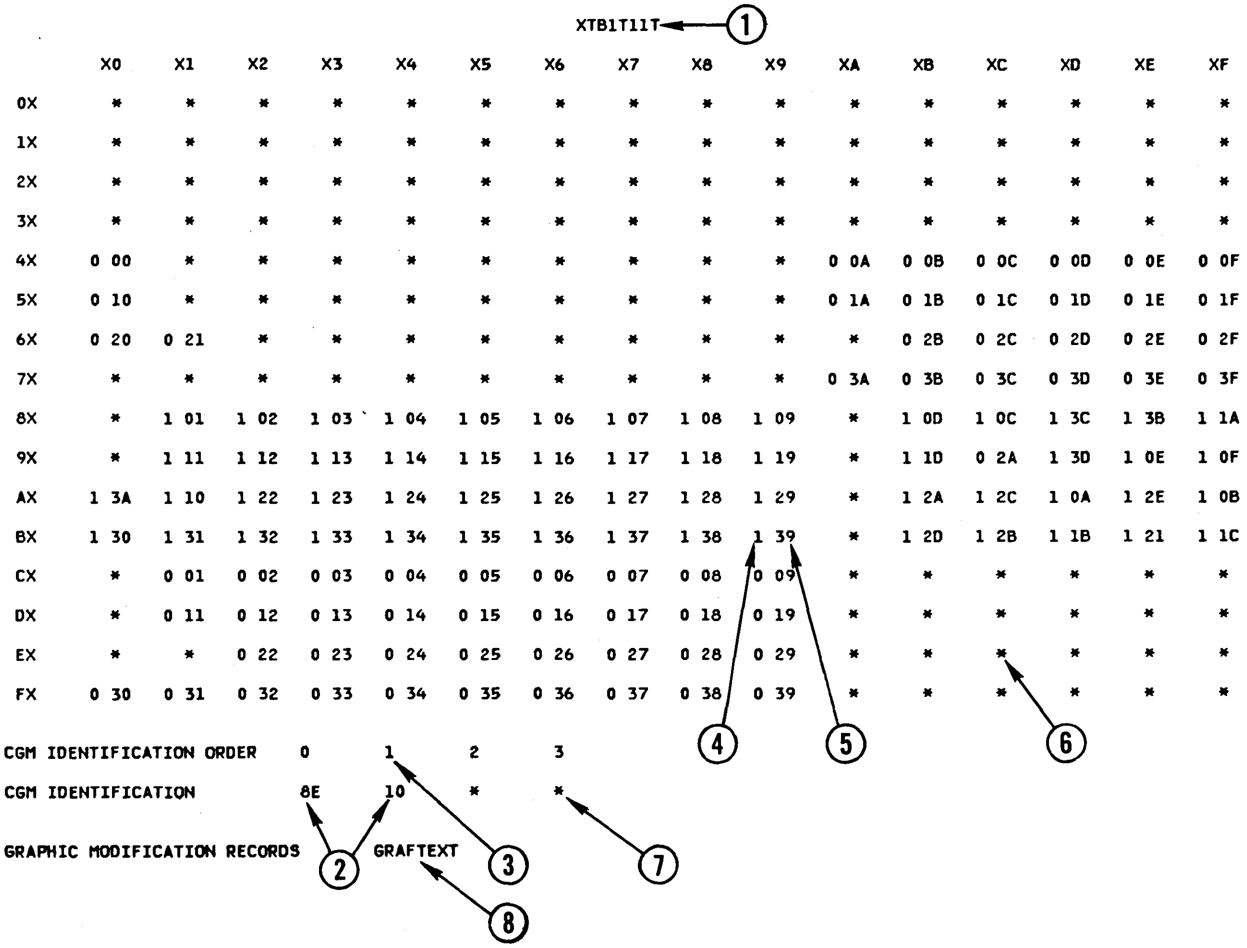
```
                                        MOD1HANK

           INITIAL    NUMBER OF    INITIAL     NUMBER OF    INITIAL      NUMBER OF
SEGMENT    COPY NO.   COPIES       LINE NO.    LINES        PRINT POS.   CHARACTERS

   1          1          4            58           1            35           18

   2          2          1             1           1            50           23

   3          2          1            34           3            75           10
```

Notes:

In this example, each note refers to the module's third segment.

1. The name of the copy modification module, as it exists in the SYS1.IMAGELIB data set's directory (including the 4-byte system-assigned prefix).
2. The segment number of the modification segment.
3. This segment applies only to the second copy of the output data set.
4. The text of the segment is located on lines 34, 35, and 36.
5. The text on each line starts at the 75th character, and occupies 10 character spaces.

Figure 21. IEBIMAGE Listing of Three Segments of a Copy Modification Module

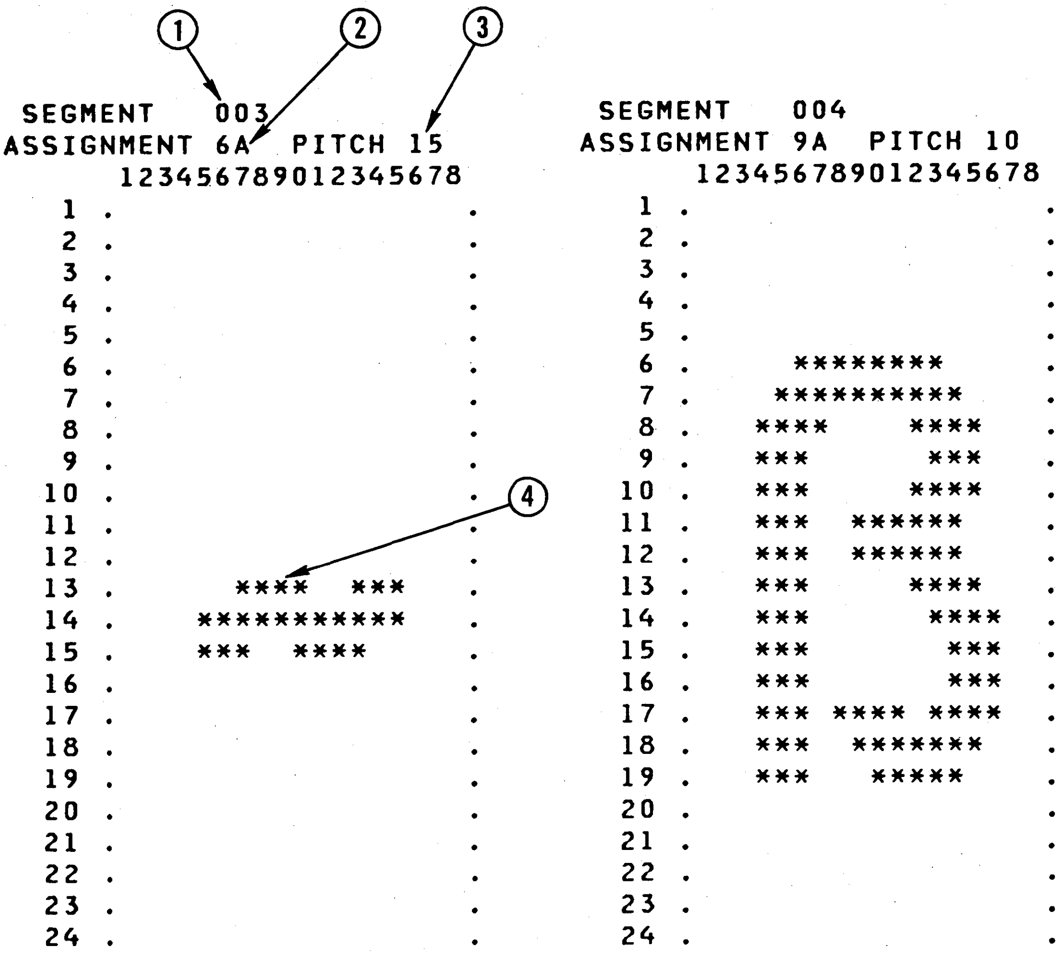


```
                                                  XTB1T11T
       X0     X1     X2     X3     X4     X5     X6     X7     X8     X9     XA     XB     XC     XD     XE     XF
0X     *      *      *      *      *      *      *      *      *      *      *      *      *      *      *      *
1X     *      *      *      *      *      *      *      *      *      *      *      *      *      *      *      *
2X     *      *      *      *      *      *      *      *      *      *      *      *      *      *      *      *
3X     *      *      *      *      *      *      *      *      *      *      *      *      *      *      *      *
4X    0 00    *      *      *      *      *      *      *      *      *     0 0A   0 0B   0 0C   0 0D   0 0E   0 0F
5X    0 10    *      *      *      *      *      *      *      *      *     0 1A   0 1B   0 1C   0 1D   0 1E   0 1F
6X    0 20   0 21    *      *      *      *      *      *      *      *      *     0 2B   0 2C   0 2D   0 2E   0 2F
7X     *      *      *      *      *      *      *      *      *      *     0 3A   0 3B   0 3C   0 3D   0 3E   0 3F
8X     *     1 01   1 02   1 03   1 04   1 05   1 06   1 07   1 08   1 09    *     1 0D   1 0C   1 3C   1 3B   1 1A
9X     *     1 11   1 12   1 13   1 14   1 15   1 16   1 17   1 18   1 19    *     1 1D   0 2A   1 3D   1 0E   1 0F
AX    1 3A   1 10   1 22   1 23   1 24   1 25   1 26   1 27   1 28   1 29    *     1 2A   1 2C   1 0A   1 2E   1 0B
BX    1 30   1 31   1 32   1 33   1 34   1 35   1 36   1 37   1 38   1 39    *     1 2D   1 2B   1 1B   1 21   1 1C
CX     *     0 01   0 02   0 03   0 04   0 05   0 06   0 07   0 08   0 09    *      *      *      *      *      *
DX     *     0 11   0 12   0 13   0 14   0 15   0 16   0 17   0 18   0 19    *      *      *      *      *      *
EX     *      *     0 22   0 23   0 24   0 25   0 26   0 27   0 28   0 29    *      *      *      *      *      *
FX    0 30   0 31   0 32   0 33   0 34   0 35   0 36   0 37   0 38   0 39    *      *      *      *      *      *

CGM IDENTIFICATION ORDER     0      1      2      3
CGM IDENTIFICATION          8E     10      *      *

GRAPHIC MODIFICATION RECORDS       GRAFTEXT
```

Notes:

1. The name of the character arrangement table module, as it exists in the SYS1.IMAGELIB data set's directory (including the 4-byte system-assigned prefix).
2. The 1-byte identifier of an IBM-supplied character set (in this example, the Text 1 and Text 2 character sets, whose identifiers are X'8E' and X'10').
3. Sequence number of the WCGM that is to contain the character set indicated below it (in this example, the second WCGM, whose identifier is 1).
4. Sequence number of the WCGM that contains the scan pattern for the 8-bit data code that locates this translate table entry.
5. Your 8-bit data code X'B9' transmitted to the 3800 addresses this, the B9 location in the translate table, where the value X'39' in turn is the index into the WCGM that contains the scan pattern to be used (in this example, the Text 2 superscript 9).
6. An asterisk is shown in the listing for each translate table entry that contains X'FF'. This indicates that the 8-bit data code that addresses this location does not have a graphic defined for it and is therefore unprintable.
7. An asterisk in the character set identifiers list indicates that no character set is specified to use the corresponding WCGM. The specification of FF as a character set identifier (to allow accessing a WCGM without loading it) causes FF to print here.
8. The name of a graphic character modification module, as the name exists in the SYS1.IMAGELIB data set's directory (including the system-assigned prefix).

Figure 22. IEBIMAGE Listing of a Character Arrangement Table Module



```
 SEGMENT   003                SEGMENT   004
ASSIGNMENT 6A  PITCH 15      ASSIGNMENT 9A  PITCH 10
      123456789012345678           123456789012345678
   1 .                  .       1 .                  .
   2 .                  .       2 .                  .
   3 .                  .       3 .                  .
   4 .                  .       4 .                  .
   5 .                  .       5 .                  .
   6 .                  .       6 .     ********     .
   7 .                  .       7 .    **********    .
   8 .                  .       8 .   ****    ****   .
   9 .                  .       9 .   ***      ***   .
  10 .                  .      10 .   ***     ****   .
  11 .                  .      11 .   ***  ******    .
  12 .                  .      12 .   ***  ******    .
  13 .      ****  ***   .      13 .   ***     ****   .
  14 .    ***********   .      14 .   ***      ****  .
  15 .    ***  ****     .      15 .   ***       ***  .
  16 .                  .      16 .   ***       ***  .
  17 .                  .      17 .   *** **** ****  .
  18 .                  .      18 .   ***  *******   .
  19 .                  .      19 .   ***   *****    .
  20 .                  .      20 .                  .
  21 .                  .      21 .                  .
  22 .                  .      22 .                  .
  23 .                  .      23 .                  .
  24 .                  .      24 .                  .
```

Notes:

1. The segment number of the character segment within the module.
2. The 8-bit data code for the character.
3. The pitch of the character.
4. The scan pattern for the character. A dollar sign ($) is printed instead of an asterisk if the bit specified is out of the pitch range.

Figure 23. IEBIMAGE Listing of Two Segments of a Graphic Character Modification Module

Control

The IEBIMAGE utility program is controlled by job control statements and utility control statements. The IEBIMAGE utility program can be executed by the operating system in different ways:

- Job control statements can be included in the input stream.
- Job control statements, placed in a procedure library or defined as an inline procedure, can be included by using the EXEC job control statement.
- The IEBIMAGE program can be invoked by a calling program.

If job control statements are placed in a procedure library, they should satisfy the requirements for most applications of the program. A procedure can be modified or supplemented for applications that require additional parameters, data sets, or devices. The IEBUPDTE utility program can be used to enter a procedure into a procedure library (see *OS/VS Utilities* for details about the IEBUPDTE utility program).

The modules that the IEBIMAGE utility program builds are stored as members of a partitioned system data set called SYS1.IMAGELIB. One or more modules can be built or printed with a single execution of the IEBIMAGE program.

Job Control Statements

The JCL used to invoke the IEBIMAGE utility program is:

```
//JOB1       JOB  ...
//            EXEC PGM=IEBIMAGE
//SYSPRINT    DD   SYSOUT=A
//SYSUT1      DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSIN       DD   *
   utility  control  statements
/*
```

Notes:

- DISP=OLD is specified to ensure that the SYS1.IMAGELIB data set is not updated by other jobs at this time.
- *Utility control statements* are grouped into operation groups, each of which applies to one SYS1.IMAGELIB module. Utility control statements are described in subsequent sections.

Restrictions

- PARM parameters cannot be specified with the EXEC statement.
- Blocksize for the SYSPRINT data set is 121 or a multiple of 121. The first character of each 121-byte output record is an ANSI control character.
- Blocksize for the SYSIN data set is 80 or a multiple of 80.

Invoking the IEBIMAGE Program from a Problem Program

The IEBIMAGE utility program can be invoked by a problem program through the use of the ATTACH or LINK macro instructions. The format of the LINK or ATTACH macro instruction is:

[*name*]   **LINK | ATTACH**   **EP=IEBIMAGE,**
PARAM=(*optionaddr*
[, *ddnameaddr* [, *hdingaddr*]]**),**
VL=1

where:

optionaddr
specifies the address of an option list. For the IEBIMAGE utility program, the option list is a halfword containing X'0000'.

ddnameaddr
specifies the address of a list of alternate ddnames for the data sets used during IEBIMAGE processing. The list is in the following format:

Bytes 1-2	specify number of bytes following in ddname list
Bytes 3-10	are binary zero, or ddname entry
Bytes 11-18	are binary zero, or ddname entry
Bytes 19-26	are binary zero, or ddname entry
Bytes 27-34	are binary zero, or ddname entry
Bytes 35-42	are system input ddname, normally SYSIN
Bytes 43-50	are system output ddname, normally SYSPRINT
Bytes 51-58	are binary zero
Bytes 59-66	are library input ddname, normally SYSUT1
Bytes 67-74	are library output ddname, normally SYSUT1

Each ddname entry must be either binary zero (indicating no change) or the ddname left-justified and padded with blanks (indicating change). The list may be truncated and only the ddnames in the list are changed. If *hdingaddr* is coded, *ddnameaddr* must be coded.

hdingaddr
specifies the address of a six-byte list which contains an EBCDIC initial page number for the output device. The six-byte list is X'0004nnnnnnnn', where nn is a hexadecimal value between X'F0' and X'F9'. If *hdingaddr* is omitted, the initial page number defaults to 1.

See *OS/VS Utilities* for details on the LINK and ATTACH macro instructions and their parameters.

Maintaining the SYS1.IMAGELIB Data Set

Maintenance of SYS1.IMAGELIB is done by using several OS/VS utilities in conjunction with IEBIMAGE. For example, you may find it necessary to rename or delete modules, or to compress or list the entire contents of the data set. Utility programs such as IEBCOPY, IEBPTPCH, IEHLIST, IEHMOVE, and IEHPROGM (as described in *OS/VS Utilities*) and AMASPZAP (as described in *OS/VS Service Aids*) should be used to help maintain SYS1.IMAGELIB.

When using utilities other than IEBIMAGE, the full module name must be specified. The module's full name consists of a 4-character prefix followed by its 1- to 4-character user-assigned name. It is thus a 5- to 8-character member name in the form:

FCB3*xxxx*, which identifies an FCB module

MOD1*xxxx*, which identifies a copy modification module

XTB1*xxxx*, which identifies a character arrangement table module

GRAF*xxxx*, which identifies a graphic character modification module

where *xxxx* is the 1- to 4-character user-specified name of the module, specified with the IEBIMAGE program's NAME utility control statement when the module is built.

Alias names are not supported by IEBIMAGE, so you should be careful if you use them. For example, if you change a module by specifying its alias name, the alias name becomes the main name of the new module, and the old module is no longer accessible via the alias but is still accessible via its original main name.

Utility Control Statements

Utility control statements are used to identify a particular function to be performed by the IEBIMAGE utility program. The control statements have the following standard format:

[*label*] *operation operand*

The *label* symbolically identifies the control statement and can be omitted. When included, the label must begin in the first position of the statement and must be followed by one or more blanks. It can contain from one to eight alphameric characters.

The *operation* identifies the type of control statement: INCLUDE, OPTION, FCB, COPYMOD, TABLE, GRAPHIC, or NAME. It must be preceded and followed by one or more blanks.

The *operand* is made up of one or more keyword parameters separated by commas. The commas that separate keyword parameters are called *delimiting commas*. The operand field must be preceded and followed by one or more blanks. Commas, parentheses, and blanks can be used only as delimiting characters, unless they are part of a field enclosed in single quotes.

Comments can be written in a utility control statement, but they must be separated from the last operand field by one or more blanks. Comments may not be on a statement that specifies an operation and no operands.

Continuation of Control Statements

Utility control statements are coded on cards or as card images and are contained in columns 1 through 71. A statement that exceeds 71 characters must be continued on one or more additional cards. A nonblank character is placed in column 72 to indicate continuation. A utility statement can be interrupted either in column 71 or after any delimiting comma.

The continued portion of the utility control statement must begin in column 16 of the following card. (Job control language continuations can begin in any column from 4 through 16, and do not require a nonblank character in column 72 for continued operand fields.) Comments can be placed on any card containing a complete or partial job control or utility control statement. However, when a card is included for the sole purpose of continuing a comment, the continuation must begin in column 16.

INCLUDE Statement

When an IEBIMAGE operation group is used to create a new module, the INCLUDE statement can identify an existing SYS1.IMAGELIB module to be copied and used as a basis for the new module. When the operation group is used to update a SYS1.IMAGELIB module, the INCLUDE statement identifies the module to be referenced and must be specified. The format of the INCLUDE statement is:

[*label*]	**INCLUDE**	*module name* **[,DELSEG=(** *segno* **[,** *segno* **...])]**

where

module name
names the module to be copied, using its 1- to 4-character name. The named module must be the same type as the module being created.

DELSEG=(*segno* **[,** *segno* **...])**
specifies the segments of the copied module that are to be deleted when the module is copied. Segment numbers can be specified in any order. When you code the DELSEG parameter, you should use a current listing of the module's contents to ensure that you are correctly identifying the unwanted segments.

Note: You can code the DELSEG parameter only when the named module is a copy modification module or a graphic character modification module.

Restrictions:

- When the INCLUDE statement is coded in an operation group, it must precede any COPYMOD, TABLE, or GRAPHIC statements.
- Only one INCLUDE statement should be coded per operation group. If more than one is coded for an operation group, the last is used; the others are ignored.
- You can code an INCLUDE statement only when the copied module is a copy modification module, a character arrangement table module, or a graphic character modification module. You cannot copy an FCB module.

Examples of an INCLUDE Statement

The following examples illustrate an INCLUDE statement that is used to:

- Obtain the entire copy of an existing module:

```
STEP1   INCLUDE   CAT5
```

- Obtain a copy of the first three segments of an existing module with seven segments:

```
STEP2   INCLUDE   GRF2,DELSEG=(4,5,6,7)
```

OPTION Statement

The OPTION statement with the OVERRUN parameter is used in a COPYMOD operation group. The value in the OVERRUN parameter specifies a print density in lines per inch. If a line overrun is anticipated at the specified lines per inch or a smaller number of lines per inch, the warning message IEBA33I is printed. The warning message is suppressed for anticipated line overrun at a greater number of lines per inch than specified in the OVERRUN parameter. The OPTION statement can be placed before or after any INCLUDE statement for the operation group.

The format of the OPTION statement is:

[*label*] **OPTION** **[OVERRUN={0 | 6 | 8 | 12}]**

where

OVERRUN={0 | 6 | 8 | 12}
specifies the greatest number of lines per inch for which message IEBA33I is to be printed for a COPYMOD operation. For example, OVERRUN=8 allows the message for 6 and 8 lines per inch, but suppresses it for 12 lines per inch. Specifying OVERRUN=0 suppresses message IEBA33I for every case.

Restrictions:

- If the OPTION statement is omitted, the default value is 12, and messages are not suppressed.
- If the OVERRUN parameter is omitted, the default value is 12.
- The OPTION statement applies only for the operation group in which it appears. OPTION operands are reset after each operation group.
- If the parameter specification is invalid (for instance, if OVERRUN=16 is specified), the entire operation group does not complete successfully.
- If the OPTION statement is coded for any IEBIMAGE operation other than COPYMOD, it is ignored.

Examples of the OPTION Statement

The following example illustrates an OPTION statement that is used to suppress the printing of message IEBA33I except for copy modifications at 6 lines per inch:

```
STEP1 OPTION OVERRUN=6
```

FCB Statement

The FCB statement specifies the contents of a forms control buffer (FCB) module: spacing codes (lines per inch), channel codes (simulated carriage-control channel punches), and the length of the form. The FCB statement must always be followed by a NAME statement, and cannot be preceded by an INCLUDE statement.

An FCB statement with no operands specified, followed by a NAME statement that identifies an FCB module in the SYS1.IMAGELIB system data set, causes the module to be formatted and printed. To build an FCB module, you code the FCB statement with at least one operand. The format of a printed FCB module is described in the "Input and Output for IEBIMAGE" section.

The format of the FCB statement is:

```
[ label ]   FCB   [LPI=(( l [ , n ])[,( l [ , n ])...]) | 6]
                  [ ,CHx =( line [ , line ...])[ ,CHx =( line ...)...]]
                  [ ,SIZE= length  | 110]
```

where

LPI=((*l* **[,** *n* **]) [,(** *l* **[,** *n* **]) ...]) | 6**

specifies the number of lines per inch and the number of lines to be printed at that line spacing.

l

specifies the number of lines per inch, and can be 6, 8, or 12.

n

specifies the number of lines at a line spacing of *l*. When the 3800 uses common-use paper sizes, *n* is a decimal value from 1 to 60 when *l* is 6, from 1 to 80 when *l* is 8, or from 1 to 120 when *l* is 12. When the printer uses ISO paper sizes, *n* is a value from 1 to 66 when *l* is 6, from 1 to 88 when *l* is 8, or from 1 to 132 when *l* is 12. See Appendix D for the paper sizes.

Restrictions:

- The total number of lines specified must result in a length that is a multiple of 1/2 inch.
- The total number of lines cannot result in a value that exceeds the usable length of the form. Do not specify coding for the top and bottom 1/2 inch of the form; IEBIMAGE does this for you.
- When you specify more than one (*l*,*n*) pair, *l* must be specified for each pair and *n* must be specified for each pair except the last.

Defaults:

- When LPI is not specified, all lines on the page are at 6 lines per inch.
- When only *l* is specified, or when *l* is the last parameter in the LPI list, all remaining lines on the page are at *l* lines per inch.
- If the total number of lines specified is less than the maximum number that can be specified, the remaining lines default to 6 lines per inch.

CH*x* =(*line* **[,** *line* **...])[, CH*x* =(** *line* **...) ...]**

specifies the channel code (or codes) and the line number (or numbers) to be skipped to when that code is specified.

CH*x*

specifies a channel code, where *x* is a decimal integer from 1 to 12.

line

specifies the line number of the print line to be skipped to, and is expressed as a decimal integer. The first printable line on the page is line number 1.

Restrictions:

- The value of *line* cannot be larger than the line number of the form's last printable line.
- Only one channel code can be specified for a print line. However, more than one print line can contain the same channel code.

Conventions:

- Channel 1 is used to identify the form's first printable line. The output writer, and the Close routines for direct allocation to a 3800 with BSAM or QSAM, require a channel code even when the data being printed contains no skip to channel 1.
- Channel 9 is used to identify a special line. To avoid I/O interrupts that are caused by use of channel 9, count lines to determine the line position.
- Channel 12 is used to identify the form's last print line to be used. To avoid I/O interrupts that are caused by use of channel 12, count lines to determine the page size.
- Use of an FCB that lacks a channel code to terminate a skip operation causes a data check at the printer when the corresponding skip is issued. This data check cannot be blocked.

SIZE= *length* | **110**

specifies the vertical length of the form, in tenths of an inch. See "Appendix D: Paper Sizes, Weights, and Maximum Characters per Line" for the allowable lengths. The complete length of the form is specified (for example, SIZE=110 for an 11-inch form) even though the amount of space available for printing is reduced by the 1/2 inch top and bottom areas where no printing occurs.

Restriction:

When both SIZE and LPI keywords are specified in the FCB statement, each keyword is checked against the other to ensure against conflicting page-length specifications. For example, SIZE=35 specifies a 3-1/2 inch length; acceptable LPI values cannot define more than the printable 2-1/2 inches of this length.

Default:

When SIZE is not specified, the form length is assumed to be 11 inches.

Examples of the FCB Statement

The following examples illustrate an FCB statement that is used to:

- Format a 7-inch page with all lines at 8 lines per inch (the default is no channel codes and 48 print lines):

```
FORM FCB SIZE=70,LPI=8
```

- Intermix different line-per-inch values on a page (the default is no channel codes, an 11-inch page length, and the last 45 lines on the page at 6 lines per inch):

```
MIX FCB LPI=((6,6),(8,2),(6,3),(8,6))
```

- Allow skipping to lines 1 and 35 at channel 1 code, and to line 15 at channel 3 code on an 8-1/2 inch form (the default is 45 print lines at 6 lines per inch):

```
SKIP FCB CH1=(1,35),CH3=15,SIZE=85
```

COPYMOD Statement

A copy modification module consists of header information followed by one or more modification segments. The header information includes the module's name and length. Each modification segment contains the text to be printed, identifies the copy (or copies) the text applies to, and specifies the position of the text on each page of the copy.

A COPYMOD statement specifies the contents of one of the modification segments of a copy modification module. More than one COPYMOD statement can be coded in an operation group; all COPYMOD statements so coded apply to the same copy modification module.

The IEBIMAGE program analyzes the modification segments specified for a copy modification module to anticipate line overrun conditions that might occur when the module is used in the printer. A line overrun condition occurs when the modification of a line is not completed in time to print that line. The time available for copy modification varies with the vertical line spacing (lines per inch) at which the printer is being operated.

Factors used in determining a line overrun condition are:

- Number of modification modules
- Number of segments per module

Combining COPYMOD segments reduces the possibility of a line overrun condition.

The IEBIMAGE program issues a message when it detects that a copy modification module might cause a line overrun condition when printed at one or more vertical line spacings. The message is printed for every line that could cause an overrun.

You can use the OPTION statement with the OVERRUN parameter to suppress the printing of that message for some or all vertical line spacings. For example, you could choose to suppress the printing for 8 and 12 lines per inch, but allow the printing at 6 lines per inch.

For information on using your copy modification module, see the section "How to Modify Copies" in the "Using the 3800" chapter. The copy modification text can be printed using the same character size or style, or one different from the size or style used to print the data in the output data set.

For the algorithm for calculating when a copy modification module might cause a line overrun condition, see the *Reference Manual for the IBM 3800 Printing Subsystem.*

The COPYMOD statement must always be followed by a NAME statement or another COPYMOD statement, and can be preceded by an INCLUDE statement. When more than one COPYMOD statement is coded, IEBIMAGE sorts the statements into order by line number within copy number. A COPYMOD statement with no operands specified, followed by a NAME statement that identifies a copy modification module, is used to format and print the module. The format of the printed module is described in the "Input and Output for IEBIMAGE" section.

The format of the COPYMOD statement, when used to create a copy modification module's segment, is:

[*label*] **COPYMOD** **COPIES=(** *starting-copy* **[,** *copies* **| 1]),**
LINES=(*starting-line* **[,** *lines* **| 1]),**
POS= *position* **,**
TEXT=(([*d* **]** *t* **,'** *text* **') [, ([** *d* **]** *t* **,'** *text* **')...])**

where

COPIES=(*starting-copy* **[,** *copies* **| 1])**

specifies the starting copy number, and the total number of copies to be modified.

starting-copy

specifies the starting copy number, and is expressed as a decimal integer from 1 to 255. The *starting-copy* value is required.

copies

specifies the number of copies that are to contain the modifying text, and is expressed as a decimal integer from 1 to 255. When *copies* is not specified, the default is 1 copy.

Restriction:

The sum of *starting-copy* and *copies* cannot exceed 256.

LINES=(*starting-line* **[,** *lines* **| 1])**

specifies the starting line number, and the total number of lines to be modified.

starting-line

specifies the starting line number, and is expressed as a decimal integer from 1 to 132. The *starting-line* value is required.

lines

specifies the number of lines that are to contain the modification segment's text, and is expressed as a decimal integer from 1 to 132. When *lines* is not specified, the default is 1 line.

Restrictions:

The sum of *starting-line* and *lines* cannot exceed 133. If the sum exceeds the number of lines specified for the form size (see the "FCB Statement" section), the modifying text is not printed on lines past the end of the form.

POS= *position*

specifies the starting print position (the number of character positions from the left margin) of the modifying text.

position

specifies the starting print position and is expressed as an integer from 1 to 204. See the restriction noted for the TEXT parameter below.

Note: The maximum number of characters per printed line depends on the pitch of each character and the width of the form. See "Appendix D: Paper Sizes, Weights, and Maximum Characters per Line" for the maximum number of characters per print line that can be printed on each form width.

TEXT=(([*d*] *t* ,‘ *text* ’)[,([*d*] *t* ,‘ *text* ’) ...])

specifies the modifying text. The text is positioned on the form based on the LINES and POS parameters, and replaces the output data set's text in those positions.

d

specifies a duplication factor (that is, the number of times the text is to be repeated). The *d* is expressed as a decimal integer from 1 to 204. If *d* is not specified, the default is 1.

t

specifies the form in which the text is entered: **C** for character, or **X** for hexadecimal. The *t* is required.

text

specifies the text and is enclosed in single quotes.

If the text type is **C**, you can specify any valid character. Blanks are valid characters. A single quote is coded as two single quotes. You are not allowed to specify a character that results in a X‘FF’.

If the text type is **X**, the text is coded in increments of two characters that specify values between X‘00’ and X‘FE’. You are not allowed to specify X‘FF’.

Restriction:

The sum of the starting print position (see the POS parameter) and the total number of text characters cannot exceed 205. If the width of the form is less than the amount of space required for the text (based on character pitch, starting position, and number of characters), characters are not printed past the right margin of the form.

If a text character specifies a character whose translate table entry contains X‘FF’ the printer sets the “Data Check” error indicator when the copy modification module is loaded.

Examples of the COPYMOD Statement

The following examples illustrate COPYMOD statements that are used to modify copies of an output data set with the following characteristics:

- The form has 80 print lines
- Each line can contain 90 characters, all 12-pitch
- The character arrangement table used with the copy modification module translates the characters exactly as coded

Each sample COPYMOD statement results in one modification segment of the copy modification module.

- Put a heading on all four copies:

```
EX10   COPYMOD  COPIES=(1,4),LINES=1,POS=71,           X
                TEXT=(C,'ABC COMPANY INVOICE')
```

- Put a label on the line following the heading line. The label is to uniquely identify each copy:

```
EX21   COPYMOD   COPIES=1,LINES=2,POS=81,                      X
                 TEXT=(C,'FILE COPY')
EX22   COPYMOD   COPIES=2,LINES=2,POS=80,                      X
                 TEXT=(C,'SALES COPY')
EX23   COPYMOD   COPIES=3,LINES=2,POS=75,                      X
                 TEXT=(C,'CUSTOMER''S COPY')
EX24   COPYMOD   COPIES=4,LINES=2,POS=70,                      X
                 TEXT=((3X,'5C'),(C,'  BANK COPY  '),          X
                 (X,'5C5C5C'))
```

- Blank out confidential information on the customer's copy:

```
EX30   COPYMOD   COPIES=3,LINES=(65,10),POS=1,                 X
                 TEXT=(90C,' ')
```

TABLE Statement

The TABLE statement is used to build a character arrangement table module. When a character arrangement table is built by the IEBIMAGE utility program and an INCLUDE statement is specified, the contents of the copied character arrangement table are used as a basis for the new character arrangement table. If an INCLUDE statement is not specified, each translate table entry in the new character arrangement table module is initialized to X'FF', the graphic character modification module name fields are set with blanks (X'40'), and the first character set identifier is set to X'82' (which is the Gothic 10-pitch set). The remaining identifiers are set to X'00'.

After the character arrangement table is initialized, the IEBIMAGE utility program modifies the table with data specified in the TABLE statement: character set identifiers, names of graphic character modification modules, and specified translate table entries. The character arrangement table, when built, must contain a reference to at least one printable character. Only one TABLE statement per operation group can be specified. The TABLE statement can be preceded by an INCLUDE statement, and must always be followed by a NAME statement.

A TABLE statement with no operands specified, followed by a NAME statement that identifies a character arrangement table module in the SYS1.IMAGELIB data set, causes the module to be formatted and printed. The format of the printed character arrangement table module is described in the "Input and Output for IEBIMAGE" section.

The format of the TABLE statement is:

```
[ label ]   TABLE     [CGMID=( set0 [, set1 ...])]
                      [,GCMLIST=(gcm1 [, gcm2 ...]) | DELETE]
                      [,LOC=(( xloc [, cloc [, setno | 0] | FF])
                      [ ,( xloc ...) ...])]
```

where

CGMID=(*set0* **[,** *set1* **...])**
identifies the IBM-supplied character sets that are to be used with the character arrangement table. (Each character set is described in "Appendix A: IBM-Supplied Character Sets.") When CGMID is specified, all character set identifiers are changed. If only one character set is specified, the other three identifiers are set to X'00'.

setx

is a 1-byte identifier of one of the IBM-supplied character sets. Up to four character set identifiers can be specified; *set0* identifies the character set that is to be loaded into the first writable character generation module (WCGM); *set1* is loaded into the second WCGM; etc. You should ensure that the character set identifiers are specified in the proper sequence, so that they are coordinated with the translate table entries. See "Appendix A: IBM-Supplied Character Sets" for the character set identifiers. When the only WCGM characters that will be used are those that are from a graphic character modification module, specifying FF as the character set identifier eliminates the unnecessary step of loading a character set from the flexible disk.

GCMLIST=(*gcm1* **[,** *gcm2* **...]) | DELETE**

names or deletes the names of up to four graphic character modification modules to be associated with the character arrangement table. When GCMLIST is specified, all graphic character modification module name fields are changed (if only one module name is specified, the other three name fields are set to blanks).

gcmx

is the 1- to 4-character name of the graphic character modification module. Up to four module names can be specified. The name is put into the character arrangement table, whether or not a graphic character modification module currently exists with that name. However, if the module doesn't exist, the IEBIMAGE program issues a warning message to the user. The character arrangement table should not be used unless all graphic character modification modules it refers to are stored in the SYS1.IMAGELIB system data set.

DELETE

specifies that all graphic character modification module name fields are to be set to blanks.

LOC=((*xloc* **[,** *cloc* **[,** *setno* **| 0] | FF])**
[,(*xloc* **...) ...])**

specifies values for some or all of the 256 translate table entries. Each translate table entry identifies one of 64 character positions within one of the WCGMs.

xloc

is an index into the translate table, and is specified as a hexadecimal value from X'00' to X'FF'; *xloc* identifies a translate table entry, not the contents of the entry; *cloc* and *setno* specify the contents of the translate table entry located by *xloc*.

cloc

identifies one of the 64 character positions within a WCGM, and is specified as a hexadecimal value between X'00' and X'3F'. When *cloc* isn't specified, the default is X'FF', an invalid character. You can specify the same *cloc* and *setno* values for more than one *xloc*.

setno

identifies one of the WCGMs, and is specified as a decimal integer from 0 to 3. When *setno* is not specified, the default is 0. The *setno* cannot be specified unless *cloc* is also specified.

Examples of the TABLE statement

The following examples illustrate a TABLE statement that is used to create a new character arrangement table by copying an existing one (using the INCLUDE statement) and modifying its contents:

- Add special characters to the existing character arrangement table:

  ```
  CAT1  TABLE  GCMLIST=CR5,                                    X
  ```

  The INCLUDE statement that must precede this TABLE statement specifies a character arrangement table that references two character sets (loaded into two WCGMs). CR5 is a user-created graphic character modification module containing three characters whose 8-bit data codes are X'FA', X'FB', and X'FC'. The LOC parameter assigns these three characters to positions X'3D', X'3E', and X'3F'—the last three positions in the second WCGM.

- Change the requested character set from 10-pitch to 12-pitch:

  ```
  CAT2 TABLE CGMID=1C
  ```

  In this example, the INCLUDE statement that precedes this TABLE statement specifies a character arrangement table that identifies the 10-pitch Katakana character set. The CGMID parameter specifies the 12-pitch Katakana character set. All characters in the 12-pitch set correspond exactly to their equivalent 10-pitch characters, so the translate table entries remain unchanged.

GRAPHIC Statement

The GRAPHIC statement specifies the contents of one of the character segments of a graphic character modification module. A graphic character modification module consists of header information followed by from 1 to 64 character segments. Each character segment contains the character's 8-bit data code, its scan pattern, and its pitch. Using the INCLUDE statement, an entire module can be copied, minus any segments deleted using the DELSEG keyword. In addition, character segments can be selected from any module named with the GCM keyword on the GRAPHIC statement. The GRAPHIC statement can also specify the scan pattern and characteristics for a new character.

The GRAPHIC statement must always be followed by a NAME statement, another GRAPHIC statement, or one or more data statements. The GRAPHIC statement can be preceded by an INCLUDE statement. More than one GRAPHIC statement can be coded in the operation group. The operation group can include GRAPHIC statements that select characters from existing modules and GRAPHIC statements that create new characters. The GRAPHIC statement, preceded by an INCLUDE statement, can be used to delete one or more segments from the copy of an existing module to create a new module.

A GRAPHIC statement with no operands specified, followed by a NAME statement that identifies a graphic character modification module, is used to format and print the module. The format of the printed module is described in the "Input and Output for IEBIMAGE" section.

The format of the GRAPHIC statement, when it is used to select a character segment from another graphic character modification module, is:

[*label*] **GRAPHIC** **REF=((** *segno* **[,** *xloc* **])[,(** *segno* **[,** *xloc* **])...])**
[,GCM= *name* **]**

The format of the GRAPHIC statement, when it is used to specify the scan pattern and characteristics of a newly-created character, is:

```
[ label ]   GRAPHIC   ASSIGN=( xloc [, pitch | 10 ])
                      data statements
```

where

REF=((*segno* **[,** *xloc* **]) [,(** *segno* **[,** *xloc* **])...])**

identifies one or more character segments within an existing graphic character modification module. Each character segment contains the scan pattern for a character, and its 8-bit data code (used to locate its translate table entry). This 8-bit data code can be respecified with the *xloc* subparameter. The REF parameter cannot be used to change a character's pitch or scan pattern.

segno

is the segment number, a decimal integer between 1 and 999. When a character segment is copied from the IBM-supplied "World Trade National Use Graphics" graphic character modification module, *segno* can be greater than 64. When the character segment is copied from a graphic character modification module built with the IEBIMAGE program, *segno* is a number from 1 to 64.

xloc

specifies an 8-bit data code for the character, and can be any value between X'00' and X'FF'. You should ensure that *xloc* identifies a translate table entry that points to a character position in the WCGM (that is, the translate table entry doesn't contain X'FF'). If *xloc* is not specified, the character's 8-bit data code remains unchanged when the segment is copied.

Note: The REF parameter can be coded in a GRAPHIC statement that includes the ASSIGN parameter.

GCM= *name*

can be coded when the REF parameter is coded and identifies the graphic character modification module that contains the character segments identified by the REF parameter.

name

specifies the 1- to 4-character user-specified name of the graphic character modification module.

Restriction: If GCM is coded, REF must also be coded.

Default: When GCM is not coded, the segments are copied from the IBM-supplied "World Trade National Use Graphics" graphic character modification module. To get segments copied from this module, GCM must not be coded.

ASSIGN=(*xloc* **[,** *pitch* **|** **10])**

identifies a newly-created character and its characteristics. The ASSIGN parameter specifies the new character's 8-bit data code and its pitch. The *data statements* that follow the GRAPHIC statement specify the new character's scan pattern. When the IEBIMAGE utility program detects the ASSIGN parameter, the program assumes that all following statements, until a statement without the characters "**SEQ=**" in columns 25 through 28 is encountered, are *data statements* that specify the character's scan pattern.

xloc
specifies the character's 8-bit data code, and can be any value between X'00' and X'FF'. You should ensure that *xloc* identifies a translate table entry that points to a character position in a WCGM (that is, the translate table entry doesn't contain X'FF'). The *xloc* is required when ASSIGN is coded.

pitch
specifies the character's horizontal size, and is one of the following decimal numbers: 10, 12, or 15. If *pitch* is not specified, the default is 10.

Restriction: At least one *data statement* must follow a GRAPHIC statement containing the ASSIGN parameter.

data statement
describes the design of the character, as it is represented on a character design form. For details of how to design a character and how to use the character design form, see the "User-Designed Graphic Characters" section. Each *data statement* represents a line on the design form. Each nonblank line on the design form must be represented with a *data statement*; a blank line can also be represented with a *data statement*. You can code up to 24 *data statements* to describe the new character's pattern.

On each statement, card columns 1 through 18 can contain nonblank grid positions when the character is 10-pitch. Any nonblank character can be punched in each card column that represents a nonblank grid position.

Columns 1 through 15 can contain nonblank grid positions when the character is 12-pitch.

Columns 4 through 15 can contain nonblank grid positions when the character is 15-pitch.

SEQ=*nn*
specifies the sequence number that must appear in columns 25-30 of the data statement, and identifies the card as a data statement; *nn* specifies a line number (corresponding to a line on the character design form), and is a 2-digit decimal number from 01 to 24.

Examples of the GRAPHIC Statement

The following examples illustrate GRAPHIC statements used to:

- Copy two character segments from the IBM-supplied "World Trade National Use Graphics" graphic character modification module and respecify their 8-bit data codes:

```
GREX1 GRAPHIC REF=((26,4A),(27,5A))
```

- Create the mathematical symbol "approximately equal" and specify its 8-bit data code and pitch:

```
GREX2  GRAPHIC  ASSIGN=(4A,15)
      XXXX XXX          SEQ=10
    XXXXXXXXXXX         SEQ=11
    XXX  XXXX           SEQ=12
                        SEQ=13
    XXXXXXXXXXX         SEQ=14
    XXXXXXXXXXX         SEQ=15
                        SEQ=16
    XXXXXXXXXXX         SEQ=17
    XXXXXXXXXXX         SEQ=18
```

NAME Statement

The NAME statement can name a new SYS1.IMAGELIB module to be built by the IEBIMAGE utility program. The NAME statement can also specify the name of an existing SYS1.IMAGELIB module. The NAME statement is required, and is the last statement in each operation group. The format of the NAME statement is:

[*label*]   **NAME** *module name* [(**R**)]

where

module name
names or identifies a SYS1.IMAGELIB module. The *module name* is 1 to 4 alphanumeric and national ($, #, and @) characters, which can be specified in any order.

(R)
specifies that this new module is to replace a module with the same name, if it exists, and reference to the original copy of the named module is to be deleted.

Examples of a NAME Statement

The following examples illustrate a NAME statement that is used to:

- Name or identify a module with the name IO40:

```
FORM1   NAME   IO40
```

- Identify a module named CSET, to be replaced by a new module with the same name:

```
OLDREC   NAME   CSET(R)
```

- Identify the "World Trade National Use Graphics" graphic character modification module for the purpose of printing it:

```
WTGRAPH   NAME   *
```

IEBIMAGE Examples

Example 1: Building a Forms Control Buffer Module

In this example, the vertical spacing and channel codes for an 11-inch form are specified, and the module is added to the SYS1.IMAGELIB data set as a new member.

```
//           JOB    ...
//           EXEC   PGM=IEBIMAGE
//SYSUT1     DD     DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT   DD     SYSOUT=A
//SYSIN      DD     *
    FCB  CH1=1,                                                   X
               CH12=80,                                           X
               LPI=8
    NAME IJ
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- CH1=1 specifies channel 1 code for line 1, allowing for positioning at line 1.
- CH12=80 specifies channel 12 code for line 80, allowing for positioning at line 80 and a unit exception indication at line 80 (the last printable line on the page.)
- LPI=8 specifies that the entire form is to be at a vertical spacing of 8 lines per inch.
- Because the SIZE parameter is omitted, the form length defaults to 11 inches. Since there are 10 inches of printable space in an 11-inch form, 80 lines are printed at 8 lines per inch.
- The name of the new FCB module is IJ, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 2: Building a Forms Control Buffer Module

In this example, the size and channel codes for a 5-1/2 inch form are specified, and the module is added to the SYS1.IMAGELIB data set as a replacement for an existing member. The new module is added to the end of the data set; the name in the data set's directory is updated so that it points to the new module; the old module can no longer be accessed through the data set's directory.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    FCB  CH1=(1,7,13,20),                                X
               CH12=26,                                  X
               SIZE=55
    NAME S55(R)
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- CH1=(1,7,13,20) specifies channel 1 code for printable line 1, line 7, line 13, and line 20.
- CH12=26 specifies channel 12 code for printable line 26.
- SIZE=55 specifies the length of the form as 55 tenths of an inch, or 5-1/2 inches.
- Because the LPI parameter is omitted, the vertical spacing defaults to 6 lines per inch. Since there are 4-1/2 inches of printable lines in a 5-1/2 inch form, there are 27 print lines on this form.
- The name of the FCB module is S55, and it replaces an existing FCB module of the same name. The new FCB module is stored as a member of the SYS1.IMAGELIB data set.

Example 3: Building a Forms Control Buffer Module

In this example, the vertical spacing, channel codes, and size for a form are specified, and the module is added to the SYS1.IMAGELIB data set as a replacement for an existing member. The new module is added to the end of the data set; the name in the data set's directory is updated so that it points to the new module; the old module can no longer be accessed through the data set's directory.

```
//          JOB  ...
//          EXEC PGM=IEBIMAGE
//SYSUT1    DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN     DD   *
    FCB  CH1=1,                                                    X
               CH2=4,                                              X
               CH5=11,                                             X
               LPI=((6,2),(8,3),(6,4),(8,9)),                      X
               SIZE=35
    NAME HL(R)
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- CH1=1 specifies channel 1 code for printable line 1.
- CH2=4 specifies channel 2 code for line 4.
- CH5=11 specifies channel 5 code for line 11.
- LPI=((6,2),(8,3),(6,4),(8,9)) specifies vertical spacing for the first 18 printable lines in the form:

  (6,2) specifies lines 1 through 2 are at a vertical spacing of 6 lines per inch, and take up 2/6 inch.

  (8,3) specifies lines 3 through 5 are at a vertical spacing of 8 lines per inch, and take up 3/8 inch.

  (6,4) specifies lines 6 through 9 are at a vertical spacing of 6 lines per inch, and take up 4/6 inch.

  (8,9) specifies lines 10 through 18 are at a vertical spacing of 8 lines per inch, and take up 1-1/8 inch.

- SIZE=35 specifies the length of the form as 35 tenths of an inch, or 3-1/2 inches. Since there are 2-1/2 inches of printable space on a 3-1/2 inch form, and since the LPI parameter specifies vertical spacing for 2-1/2 inches of lines, the vertical spacing of all lines in the form is accounted for.
- The name of the FCB module is HL, and it replaces an existing module of the same name. The new FCB module is stored as a member of the SYS1.IMAGELIB data set.

Example 4: Building a Forms Control Buffer Module

In this example, the vertical spacing, channel codes, and length of a form are specified, and the module is added to the SYS1.IMAGELIB data set as a new member.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    FCB  CH1=1,                                          X
               CH6=33,                                   X
               LPI=((8,32),(12,2)),                      X
               SIZE=70
    NAME TGT
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- CH1=1 specifies channel 1 code for printable line 1.
- CH6=33 specifies channel 6 code for line 33.
- LPI=((8,32), (12,2)) specifies that the first 32 printable lines of the form are to be at a vertical spacing of 8 lines per inch, and the next 2 printable lines are to be at a vertical spacing of 12 lines per inch.
- SIZE=70 specifies that the length of the form is 70 tenths of an inch, or 7 inches. Since there are 6 inches of printable lines in a 7-inch form, and the LPI parameter specifies 32 lines at 8 lines per inch, or 4 inches, and 2 lines at 12 lines per inch, or 1/6 inch, the vertical spacing for the remaining 1-5/6 inches defaults to 6 lines per inch.

  Therefore, the form consists of lines 1 through 32 at 8 lines per inch, lines 33 through 34 at 12 lines per inch, and lines 35 through 45 at 6 lines per inch, with channel codes at line 1 and line 33.
- The name of the new FCB module is TGT, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 5: Building a Forms Control Buffer Module to Replace STD3 for ISO Paper Sizes

In this example, an FCB module is defined for a 3800 that uses ISO paper sizes, replacing the IBM-supplied module named STD3. This must be done before the dump-formatting routines that print high-density dumps can print them at 8 lines per inch on that 3800.

```
//          JOB  ...
//          EXEC PGM=IEBIMAGE
//SYSUT1    DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN     DD   *
    FCB  CH1=1,CH12=88,
              LPI=(8,88),
              SIZE=120
    NAME STD3(R)
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- CH1=1 specifies channel 1 code for printable line 1; CH12=88 specifies channel 12 code for line 88.
- LPI=(8,88) specifies that all 88 printable lines of the form are to be at a vertical spacing of 8 lines per inch.
- SIZE=120 specifies that the length of the form is 120 tenths of an inch, or 12 inches, which is the longest ISO paper size.
- The name of the new FCB module is STD3, and it is to replace the existing module of that same name on SYS1.IMAGELIB.

Example 6: Building a Copy Modification Module

In this example, a copy modification module that contains four modification segments is built. The module is added to the SYS1.IMAGELIB data set as a new member.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
COPY1  COPYMOD  COPIES=(1,1),                                   X
               LINES=(1,1),POS=50,                              X
               TEXT=(C,'CONTROLLER''S COPY')
COPY2A COPYMOD  COPIES=(2,1),                                   X
               LINES=(1,1),POS=50,                              X
               TEXT=(C,'SHIPPING MANAGER''S COPY')
COPY2B COPYMOD  COPIES=(2,1),                                   X
               LINES=(34,3),POS=75,                             X
               TEXT=(10C,' ')
COPYALL COPYMOD COPIES=(1,4),                                   X
               LINES=(58,1),POS=35,                             X
               TEXT=((C,'***'),(C,'CONFIDENTIAL'),              X
               (3X,'5C'))
           NAME  RTO1
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The COPY1 COPYMOD statement specifies text that applies to each page of the first copy of the output data set:

  LINES=(1,1) and POS=50 specify that the text is to be on the first printable line of each page, starting at the 50th print position from the left.

  The TEXT parameter identifies each page of the copy as being the "Controllers's Copy."
- The COPY2A COPYMOD statement specifies text that applies to each page of the second copy of the output data set. The text is to be on the first line of each page, at the 50th print position from the left, with each page of the copy being the "Shipping Manager's Copy."
- The COPY2B COPYMOD statement specifies that part of the second copy's output data set text is to be blanked out, so that the first, third, and subsequent copies contain information that is not printed on the second copy. The blank area is to be on lines 34, 35, and 36, beginning at the 75th print position from the left. The "text" on lines 34, 35, and 36, between print positions 75 and 84 is to be blank (that is, the character specified between the TEXT parameter's single quotes is a blank).
- The COPYALL COPYMOD statement specifies text that applies to the first four copies of the output data set. This example assumes that no more than four copies are printed each time the job that produces the output data set is executed. The text is to be on the 58th line on each page, at the 35th print position from the left. The legend "***CONFIDENTIAL***" is to be on each page of the copy. Note that the text can be coded in both character and hexadecimal format.
- The name of the copy modification module is RTO1, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 7: Building a Copy Modification Module

In this example, a copy of an existing copy modification module, RTO1, is used as the basis for a new copy modification module. The new module is added to the SYS1.IMAGELIB data set as a new member. The existing module, RTO1, remains unchanged and available for use.

```
//        JOB  ...
//        EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    INCLUDE  RTO1,                                      X
               DELSEG=1
    OPTION   OVERRUN=8
    COPYMOD  COPIES=(2,3),                              X
               LINES=(52,6),POS=100,                    X
               TEXT=(X,'40404040404040405C5C')
    NAME  AP
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the copy modification module named RTO1 is used as a basis for the new module, and that the first modification segment of RTO1 is to be deleted from the copy.
- OVERRUN=8 in the OPTION statement specifies that the IEBIMAGE program is to print a warning message if the copy modification could cause a line overrun condition when printing at 6 and 8 lines per inch. The program is to suppress any warning messages that apply to printing at 12 lines per inch.
- The COPYMOD statement specifies text that applies to each page of the second, third, and fourth copies of the output data set:

  LINES=(52,6) and POS=100 specify that the text is to be on the 52nd line and repeated for the 53rd through 57th lines of each page, starting at the 100th print position from the left.

  The TEXT statement specifies the text in hexadecimal form: eight blanks followed by two asterisks (in this example, the assumption is made that X'40' prints as a blank, and that X'5C' prints as an asterisk; in actual practice, the character arrangement table used with the copy modification module might translate X'40' and X'5C' to other printable characters).
- The name of the new copy modification module is AP, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 8: Modifying a Character Arrangement Table Module

In this example, an IBM-supplied character arrangement table module is modified to include another character, and then added to the SYS1.IMAGELIB data set as a replacement for the existing module.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
     INCLUDE  GF10
     TABLE    LOC=((2A,2A),(6A,2A),(AA,2A),(EA,2A))
     NAME     GF10(R)
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the character arrangement table named GF10 is to be used as a basis for the new module.
- The TABLE statement specifies updated information for four translate table entries: X‘2A’, X‘6A’, X‘AA’, and X‘EA’. (These four locations are unused in the IBM-supplied GF10 table.) Each of the four translate table entries is to point to the ‘2A’ (43rd character) position in the first WCGM, which contains the scan pattern for a lozenge.
- The name of the character arrangement table is GF10, and it is stored as a new module in the SYS1.IMAGELIB data set. The data set’s directory is updated so that the name GF10 points to the new module; the old GF10 module can no longer be accessed through the data set’s directory.

Example 9: Building a Character Arrangement Table Module

In this example, an existing character arrangement table module is copied and used as a basis for a new module. The new character arrangement table is identical to the old one except that it uses the Gothic 15-pitch character set instead of Gothic 10-pitch.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    INCLUDE  A11
    TABLE    CGMID=86
    NAME     A115
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the character arrangement table named A11 is to be used as a basis for the new module. The A11 character arrangement table translates 8-bit data codes to printable characters in the Gothic 10-pitch character set.
- The TABLE statement specifies a new character set identifier, X‘86’, which is the identifier for the Gothic 15-pitch character set. No other changes are made to the character arrangement table. The new table calls for characters in the Gothic 15-pitch character set.
- The name of the new character arrangement table is A115, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 10: Building a Character Arrangement Table Module

In this example, an existing character arrangement table module is copied and used as the basis for a new module that will include user-designed characters of a graphic character modification module. The new module is then added to the SYS1.IMAGELIB data set.

```
//           JOB  ...
//           EXEC PGM=IEBIMAGE
//SYSUT1     DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD     SYSOUT=A
//SYSIN      DD   *
    INCLUDE ONB
    TABLE   GCMLIST=ONB1,                                        X
               LOC=((6F,2F,1),(7C,3C,1),(6A,2A,0))
    NAME    ONBZ
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the character arrangement table named ONB is to be used as a basis for the new module. ONB references two WCGMs.
- The TABLE statement identifies a graphic character modification module and stipulates the translate table entries for each of its segments:

  GCMLIST=ONB1 identifies the graphic character modification module named ONB1. The LOC parameter specifies the translate table entry location, character position, and WCGM number for each segment of the module:

  > The first segment corresponds to the 8-bit data code X'6F'. The segment's scan pattern is to be loaded at character position X'2F' (that is, the 48th character position) in the second WCGM.
  >
  > The second segment corresponds to the 8-bit data code X'7C'. The segment's scan pattern is to be loaded at character position X'3C' (that is, the 61st character position) in the second WCGM.
  >
  > The third segment corresponds to the 8-bit data code X'6A'. The segment's scan pattern is to be loaded at character position X'2A' (that is, the 43rd character position) in the first WCGM.

- The name of the new character arrangement table is ONBZ, and it is stored as a new module in the SYS1.IMAGELIB data set.

Example 11: Building a Character Arrangement Table Module

In this example, an existing character arrangement table module is copied and used as a basis for a new one. The new character arrangement table deletes references to all graphic character modification modules and resets the translate table entries that were used to point to character positions for the segments of a graphic character modification module.

```
//          JOB   ...
//          EXEC  PGM=IEBIMAGE
//SYSUT1    DD    DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
      INCLUDE  ZYL
      TABLE    GCMLIST=DELETE,                                      X
                 LOC=((6A),(6B))
      NAME     ZYLA
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the character arrangement table named ZYL is to be used as a basis for the new module.
- The TABLE statement deletes references to graphic character modification modules and resets two translate table entries:

  GCMLIST=DELETE specifies that all names of graphic character modification modules included with the module when the ZYL character arrangement table was copied are to be reset to blanks (X'40').

  The LOC parameter identifies two locations in the translate table, X'6A' and X'6B', that are to be set to X'FF' (the default value, when no character position or WCGM values are specified).
- The name of the new character arrangement table is ZYLA, and it is stored as a member of the SYS1.IMAGELIB data set.

Example 12: Listing the World Trade National Use Graphics Graphic Character Modification Module

In this example, each segment of the IBM-supplied graphic character modification module containing the "World Trade National Use Graphics" is printed. Each segment is unique, although the scan patterns for some segments are identical to other segment's scan patterns with only the 8-bit data code being different.

```
//          JOB   ...
//          EXEC  PGM=IEBIMAGE
//SYSUT1    DD    DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT  DD    SYSOUT=A
//SYSIN     DD    *
        GRAPHIC
          NAME  *
/*
```

Notes:

- The "World Trade National Use Graphics" graphic character modification module is identified with the pseudonym of "*". The scan pattern of each of the characters in the module is printed.

Example 13: Building a Graphic Character Modification Module

In this example, a graphic character modification module is built. Its characters are segments copied from the "World Trade National Use Graphics" graphic character modification module. (See Appendix C for the listing of all the segments of that module, including the EBCDIC assignments for the characters.) The new module is stored in the SYS1.IMAGELIB system data set.

```
//          JOB  ...
//          EXEC PGM=IEBIMAGE
//SYSUT1    DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
    GRAPHIC  REF=((24),(25),(26),(27),(28),                       X
                (31),(33),(35),(38),(40))
    NAME     CSTW
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- By not specifying the GCM keyword, the GRAPHIC statement identifies the "World Trade National Use Graphics" graphic character modification module. Ten of its segments are to be copied and used with the new module.
- The name of the graphic character modification module is CSTW, and it is stored as a new module in the SYS1.IMAGELIB data set.

Example 14: Building a Graphic Character Modification Module and Modifying a Character Arrangement Table to Use It

In this example, a graphic character modification module is built. The module contains one user-designed character, a reverse 'E', whose 8-bit data code is designated as X'E0', and whose pitch is 10. An existing character arrangement table is then modified to include the reverse E.

```
//         JOB  ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
        GRAPHIC  ASSIGN=(E0,10)
   XXXXXXXXXXXX         SEQ=07
   XXXXXXXXXXXX         SEQ=08
            XXX         SEQ=09
            XXX         SEQ=10
            XXX         SEQ=11
    XXXXXXXXXXX         SEQ=12
    XXXXXXXXXXX         SEQ=13
            XXX         SEQ=14
            XXX         SEQ=15
            XXX         SEQ=16
            XXX         SEQ=17
   XXXXXXXXXXXX         SEQ=18
   XXXXXXXXXXXX         SEQ=19
    NAME  BODE
    INCLUDE  GS10
    TABLE    CGMID=(82,FF),                               X
               GCMLIST=BODE,                              X
               LOC=(E0,03,1)
    NAME     RE10
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The GRAPHIC statement's ASSIGN parameter establishes the 8-bit data code, X'E0', and the width, 10-pitch, for the user-designed character. The *data statements* that follow the GRAPHIC statement describe the character's scan pattern.
- The name of the graphic character modification module is BODE, and it is stored as a new module in the SYS1.IMAGELIB data set.
- The INCLUDE statement specifies that a copy of the GS10 character arrangement table is to be used as the basis for the new table.
- The TABLE statement specifies the addition of the reverse E to that copy of the GS10 table.

  CGMID=(82,FF) specifies the character set identifier X'82' for the Gothic-10 set (which is the set already used by the GS10 table) and specifies X'FF' as a character set identifier to allow accessing of the second WCGM without loading it.

  GCMLIST=BODE identifies the graphic character modification module containing the reverse E for inclusion in the table.

  LOC=(E0,03,1) specifies that the reverse E, which has been assigned the 8-bit data code X'E0', is to be loaded into position X'03' in the second WCGM. Since this second WCGM is otherwise unused, any position in it could have been used for the reverse E.
- The new character arrangement table is named RE10 and stored as a new module in SYS1.IMAGELIB.

Example 15: Building a Graphic Character Modification Module

In this example, a graphic character modification module is created. Its contents come from three different sources: nine segments are copied from an existing module with the INCLUDE statement; the GRAPHIC statement is used to select another segment to be copied; the GRAPHIC statement is also used to establish characteristics for a user-designed character. The new graphic character modification module, when built, is added to the SYS1.IMAGELIB.

```
//         JOB    ...
//         EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
    INCLUDE  CSTW,DELSEG=3
    GRAPHIC  REF=(1,6A),GCM=BODE,ASSIGN=9A
     ********           SEQ=06
    **********          SEQ=07
   ****    ****         SEQ=08
   ***      ***         SEQ=09
   ***     ****         SEQ=10
   ***  ******          SEQ=11
   ***  ******          SEQ=12
   ***     ****         SEQ=13
   ***      ****        SEQ=14
   ***       ***        SEQ=15
   ***       ***        SEQ=16
   *** **** ****        SEQ=17
   ***  *******         SEQ=18
   ***   *****          SEQ=19
    NAME  JPCK
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The INCLUDE statement specifies that a copy of the graphic character modification module named CSTW is to be included with the new module. All segments of CSTW, except the third segment (as a result of DELSEG=3), are to be copied into the new module, and become the module's first through ninth modification segments.
- The GRAPHIC statement specifies the module's tenth and eleventh segments:

  REF=(1,6A) and GCM=BODE specify that the tenth segment of the new module is to be obtained by copying the first segment from the graphic character modification module named BODE. In addition, the segment's 8-bit data code is to be changed so that its character is identified with the code X'6A'.

  ASSIGN=9A specifies that the new module's eleventh segment is a user-designed character whose 8-bit data code is X'9A' and whose width is 10-pitch (the default when no pitch value is specified). The GRAPHIC statement is followed by *data statements* that specify the character's scan pattern.
- The name of the graphic character modification module is JPCK, and it is stored as a new module in the SYS1.IMAGELIB data set.

Example 16: Defining a Character and Using It

In this example, a graphic character modification module containing a user-designed character is built. Next, a Format character arrangement table is modified to include that new character. Then, a copy modification module is created to print the new character enclosed in a box of Format characters. Finally, the result is tested to allow comparison of the output with the input.

```
//         JOB  ...
//BUILD    EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
STEP1    GRAPHIC ASSIGN=5C
XXXXXXXX  XXXXXXXX      SEQ=03
XXXXXXXX  XXXXXXXX      SEQ=04
      XX  XX    XX      SEQ=05
      XX  XX    XX      SEQ=06
  XXXXXX  XXXXXXXX      SEQ=07
  XXXXXX  XXXXXXXX      SEQ=08
      XX  XX    XX      SEQ=09
      XX  XX    XX      SEQ=10
XXXXXXXX  XXXXXXXX      SEQ=11
XXXXXXXX  XXXXXXXX      SEQ=12
                        SEQ=13
                        SEQ=14
XXXXXXXX  XXXXXXXX      SEQ=15
XXXXXXXX  XXXXXXXX      SEQ=16
XX    XX  XX    XX      SEQ=17
XX    XX  XX    XX      SEQ=18
XX    XX  XX    XX      SEQ=19
XX    XX  XX    XX      SEQ=20
XX    XX  XX    XX      SEQ=21
XX    XX  XX    XX      SEQ=22
XXXXXXXX  XXXXXXXX      SEQ=23
XXXXXXXX  XXXXXXXX      SEQ=24
         NAME AIBM
STEP2    INCLUDE FM10
         TABLE GCMLIST=AIBM,LOC=(5C,2C)
         NAME BIBM
STEP3    COPYMOD COPIES=1,LINES=58,POS=5,                X
               TEXT=(C,'W6X')
         COPYMOD COPIES=1,LINES=59,POS=5,                X
               TEXT=(C,'7*7')
         COPYMOD COPIES=1,LINES=60,POS=5,                X
               TEXT=(X,'E9F6E8')
         NAME CIBM
/*
//TEST     EXEC PGM=IEBIMAGE
//SYSUT1   DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD   SYSOUT=A,CHARS=(GF10,BIBM),
//    MODIFY=(CIBM,1)
//SYSIN    DD   *
        GRAPHIC
        NAME AIBM
/*
```

Notes:

- The SYSUT1 DD statement includes DISP=OLD to ensure that no other job can modify the data set while this job is executing.
- The GRAPHIC statement's ASSIGN parameter specifies that the 8-bit data code for the user-designed character is X'5C' and the width is 10-pitch (the default when no pitch is specified). The GRAPHIC

statement is followed by *data statements* that specify the character's scan pattern.

- The name of the graphic character modification module is AIBM, and it is stored as a new module in SYS1.IMAGELIB.
- At STEP2, the INCLUDE statement specifies that a copy of the FM10 character arrangement table is to be used as a basis for the new module.
- The TABLE statement identifies the graphic character modification module named AIBM, created in the previous step. The TABLE statement's LOC parameter specifies the translate table entry location (the character's 8-bit data code) of X'5C' and the position (X'2C') where that character is to be loaded into the WCGM.
- The name of the new character arrangement table, which is added to SYS1.IMAGELIB, is BIBM.
- At STEP3, the three COPYMOD statements specify text that is to be placed on lines 58, 59, and 60 of the first copy of the output data set, starting at print position 5 on each line. When used with the BIBM character arrangement table, the characters W, 6, and X print as a top left corner, horizontal line segment, and top right corner, all in line weight 3. The characters 7, *, and 7 print as a weight-3 vertical line segment on both sides of the user-designed character built at STEP1 (the asterisk has the EBCDIC assignment 5C which addresses that character). The hexadecimal E9, F6, and E8 complete the line-weight-3 Format box around the character.
- The name of the copy modification module is CIBM, and it is stored as a new module on SYS1.IMAGELIB.
- At TEST, the EXEC statement calls for another execution of the IEBIMAGE program to test the modules just created. On the SYSPRINT DD statement the BIBM character arrangement table is the second of two specified, and the CIBM copy modification module is specified with a table reference character of 1, to use that BIBM table.
- The GRAPHIC statement with no operand specified calls for printing of the module, AIBM, specified with the NAME statement that follows it. Each page of the output listing for this IEBIMAGE run has the following modification printed in the lower left corner:

THE STRUCTURE OF MODULES PRODUCED BY IEBIMAGE

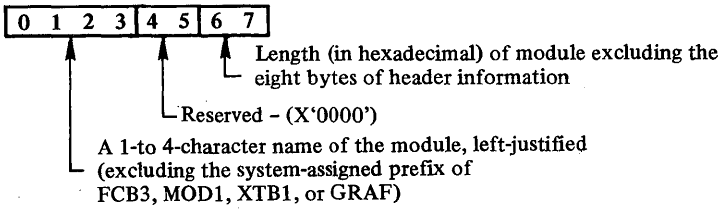
The four types of modules produced by the IEBIMAGE program and used by the SETPRT SVC are:

- Forms control buffer modules
- Copy modification modules
- Character arrangement table modules
- Graphic character modification modules

The module structures shown below are the structures that are created on SYS1.IMAGELIB as *output* from the utility. If you code these modules in your own storage, the structure must be the same.

The Header Information

Each module contains eight bytes of header information preceding the data.


0 1 2 3
4 5
6 7
Length (in hexadecimal) of module excluding the eight bytes of header information
Reserved - (X'0000')
A 1-to 4-character name of the module, left-justified (excluding the system-assigned prefix of FCB3, MOD1, XTB1, or GRAF)

The SETPRT SVC uses the name to:

- Identify the module on SYS1.IMAGELIB
- Store the name in the UCB extension

The SETPRT SVC uses the length to:

- Obtain sufficient storage for the module
- Move the data from user's storage to its own area
- Build channel programs to load the data into the printer

Naming Conventions for Modules

Each module placed on SYS1.IMAGELIB by the IEBIMAGE utility has a 4-character system-assigned prefix as the first part of its name. These prefixes are:

FCB3 for forms control buffer modules

MOD1 for copy modification modules

XTB1 for character arrangement table modules

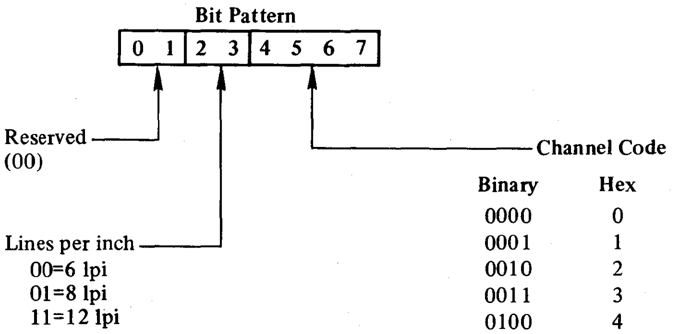
GRAF for graphic character modification modules

While the 3800's programming support refers only to the 1- to 4-character name (the suffix) that is appended to the prefix, the full name must be used when using other utilities (such as IEBPTPCH or IEHPROGM). The 1- to 4-character suffix is alphanumeric and national ($, @, and #) characters in any order.

The FCB Module Structure

The FCB data following the header information is a series of 1-byte line control codes for each physical line of the form. There are 18 to 144 of these bytes, depending on the length of the form.

Each byte is a bit pattern describing one of 12 channel codes for vertical forms positioning and one of three lines-per-inch codes for vertical line spacing.

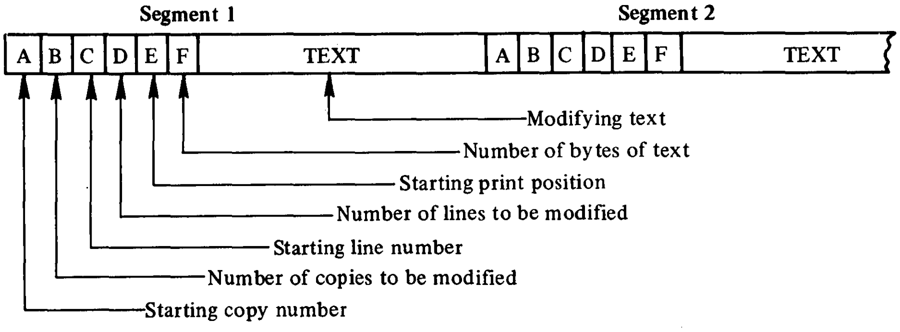


Binary	Hex	Channel
0000	0	- (null)
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12

- The top and bottom 1/2 inch of each page are unprintable, and the bytes corresponding to these positions must be void of any channel codes. Three bytes of binary zeros are supplied by the IEBIMAGE utility for the top and bottom 1/2 inch.
- The total number of lines defined in the module must be equal to the length of the form. The printable lines defined must start 1/2 inch below the top and stop 1/2 inch from the bottom of the form.

The Copy Modification Module Structure

The copy modification data following the header information is a series of segments. Each segment is of variable length and is composed of the following components:



Drop-in

A, B, C, D, E and F are each 1-byte fields.

- If the module contains more than one segment, the starting copy number will be equal to or greater than the starting copy number in the previous segment.
- Any string of the same character within the text may be compressed into 3 bytes. The first such byte is X'FF', the second byte is the number of compressed characters and the third byte is the data code for the character. The IEBIMAGE utility uses this compression algorithm.
- The size of the module is limited to 8K bytes.

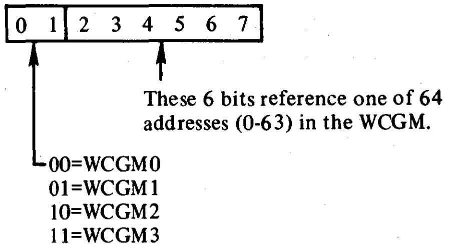
The Character Arrangement Table Module Structure

The character arrangement table data following the header information is composed of the following components:

- A 256-byte translate table
- Four 2-byte fields for codes identifying character sets and their sequence
- Four 4-byte fields for graphic character modification module names

The translate table consists of 256 1-byte entries, each pointing to one of 64 positions within one of four WCGMs:

- Bits 0 and 1 of each translate table byte reference one of four WCGMs and bits 2 through 7 point to one of 64 addresses (0-63) within the WCGM. If SETPRT loads a character set into a WCGM other than the WCGM called for, SETPRT alters bits 0 and 1 for each non-X'FF' byte in the translate table to correspond with the WCGM loaded.


0 1 2 3 4 5 6 7
These 6 bits reference one of 64 addresses (0-63) in the WCGM.
00=WCGM0
01=WCGM1
10=WCGM2
11=WCGM3

- A byte value of X'FF' indicates an invalid character and prints as a blank and gives a data check. The data check is suppressed if the "block data check" option is selected.

- One translate table can address multiple WCGMs, and multiple translate tables can address one WCGM. The translate tables supplied by IBM address either one or two WCGMs.

The next two components provide the linkage to character sets and graphic character modification modules. They consist of four 2-byte sets of character set IDs with their corresponding character set sequence, followed by four 4-character names of graphic character modification modules. The format is as follows:

- Each CGMID is a 1-byte ID of the character set (as listed in Figure 24 in "Appendix A: IBM-Supplied Character Sets"). Each WCGMNO refers to the corresponding WCGM sequence (X'00' to X'03'). Each "Name" is the 4-character name of a graphic character modification module.

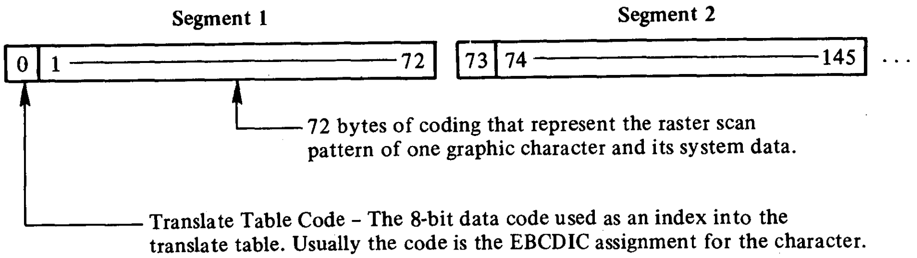
CGMID0	WCGMNO0	CGMID1	WCGMNO1
CGMID2	WCGMNO2	CGMID3	WCGMNO3
Name1			
Name2			
Name3			
Name4			

- Most of the standard character arrangement tables do not need graphic character modification. The "Names" are blank (X'40's) if no modules are referenced.

- The CGMID *x* and the WCGMNO *x* are both X'00' when there are no character sets referred to after the first one.

The Graphic Character Modification Module Structure

The graphic character modification data following the header information is a series of 73-byte segments. A maximum of 64 such segments is allowed per module.


Segment 1
0
1
72
Segment 2
73
74
145
. . .
72 bytes of coding that represent the raster scan pattern of one graphic character and its system data.
Translate Table Code - The 8-bit data code used as an index into the translate table. Usually the code is the EBCDIC assignment for the character.

When a graphic character is to be modified, the 3800 uses the translate table code to index into the translate table, and uses the contents found at that location (a 1-byte WCGM code) to determine the WCGM location for placing the 72 bytes of scan pattern and system data.

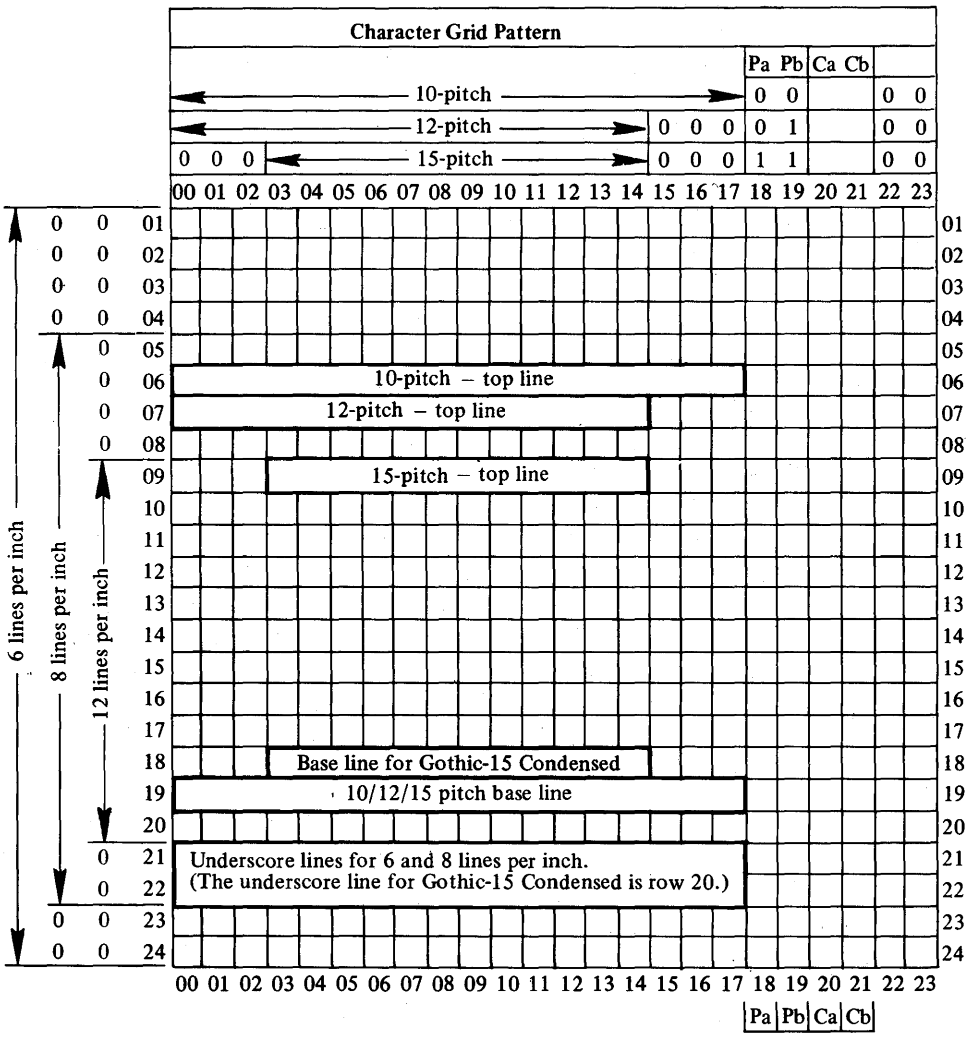
The 72-byte graphic definition that makes up the scan pattern and system data for one character is divided into twenty-four 3-byte groups. Each 3-byte group is expressed below as a horizontal row of twenty-four 1-bit elements:

Bits 0-17 give the scan pattern for that row in the character.

Bits 18-19 contain the character pitch code.

Bits 20-21 are check bits for internal parity checking.

Bits 22-23 are always zeros.



Pa & Pb = pitch (which must be constant
within a character)

00=10-pitch
01=12-pitch
11=15-pitch

Ca=Check bit (odd parity) for bits 0-8 and Pa
Cb=Check bit (odd parity) for bits 9-17 and Pb

APPENDIX A: IBM-SUPPLIED CHARACTER SETS

Twenty character sets are supplied with the 3800. Figure 24 lists the supplied sets. The WCGM assignments of the characters in each of the sets are shown in Figures 25 through 31.

Character set	Characters per inch (pitch)	Number of nonblank characters in the set	Character set identifier, CGMID[3]
Gothic-10	10	63	82
Gothic-12	12	63	84
Gothic-15	15	63	86
Gothic-15 Condensed[1]	15	63	92
Gothic-10 underscored	10	63	38
Gothic-12 underscored	12	63	3A
Gothic-15 underscored	15	63	3C
Gothic-15 Condensed underscored[1]	15	63	36
Katakana-10[2]	10	64	1A
Katakana-12[2]	12	64	1C
Katakana-15[2]	15	64	1E
OCR-A	10	52	16
OCR-B	10	54	18
Text 1	10	63	8E
Text 2	10	63	10
Text 1 underscored	10	63	3E
Text 2 underscored	10	63	40
Format-10	10	36	08
Format-12	12	36	0A
Format-15	15	36	0C

[1] The condensed character sets are for use at 12 lines per inch. If other character sets are printed at 12 lines per inch, the tops of the characters may be clipped off.

[2] Katakana is used together with another character set that contains a blank. The combined sets use character generation storage for 128 characters (including a blank).

[3] See "TABLE Statement" for an explanation of the use of FF as a character set identifier (CGMID).

Figure 24. Character Sets Supplied with the 3800

WCGM Assignment	Graphic	Description
01	A	A
02	B	B
03	C	C
04	D	D
05	E	E
06	F	F
07	G	G
08	H	H
09	I	I
0A	¢	CENT SIGN
0B	.	PERIOD OR DECIMAL POINT
0C	<	LESS THAN
0D	(	LEFT PARENTHESIS
0E	+	PLUS SIGN
0F	|	LOGICAL OR
10	&	AMPERSAND
11	J	J
12	K	K
13	L	L
14	M	M
15	N	N
16	O	O
17	P	P
18	Q	Q
19	R	R
1A	!	EXCLAMATION POINT
1B	$	DOLLAR SIGN
1C	*	ASTERISK
1D	)	RIGHT PARENTHESIS
1E	;	SEMICOLON
1F	¬	LOGICAL NOT
20	-	MINUS SIGN OR HYPHEN
21	/	SLASH
22	S	S
23	T	T
24	U	U
25	V	V
26	W	W
27	X	X
28	Y	Y
29	Z	Z
2A	¤	LOZENGE
2B	,	COMMA
2C	%	PERCENT SIGN
2D	_	UNDERSCORE
2E	>	GREATER THAN
2F	?	QUESTION MARK
30	0	ZERO
31	1	ONE
32	2	TWO
33	3	THREE
34	4	FOUR
35	5	FIVE
36	6	SIX
37	7	SEVEN
38	8	EIGHT
39	9	NINE
3A	:	COLON
3B	#	NUMBER SIGN
3C	@	AT SIGN
3D	'	PRIME OR APOSTROPHE
3E	=	EQUAL SIGN
3F	"	QUOTATION MARK

Figure 25. WCGM Assignments for Gothic and Gothic Underscored (in All Three Pitches) and Gothic-15 Condensed Characters

WCGM Assignment	Graphic	Description
00	¥	YEN
01	｡	KANA PERIOD
02	｢	OPEN BRACKET
03	｣	CLOSE BRACKET
04	､	KANA COMMA
05	･	CENTER MARK
06	ｦ	WO
07	ｧ	SMALL A
08	ｨ	SMALL I
09	ｩ	SMALL U
0A	ｺ	KO
0B	ｹ	KE
0C	ｻ	SA
0D	ｼ	SHI
0E	ｽ	SU
0F	ｾ	SE
10	ｿ	SO
11	ﾀ	TA
12	ﾁ	CHI
13	ﾂ	TSU
14	ﾃ	TE
15	ﾄ	TO
16	ﾅ	NA
17	ﾆ	NI
18	ﾇ	NU
19	ﾈ	NE
1A	ﾉ	NO
1B	ｰ	PROLONGED SOUND
1C	ｱ	A
1D	ﾊ	HA
1E	ﾋ	HI
1F	ﾌ	FU
20	ｲ	I
21	ｳ	U
22	ﾍ	HE
23	ﾎ	HO
24	ﾏ	MA
25	ﾐ	MI
26	ﾑ	MU
27	ﾒ	ME
28	ﾓ	MO
29	ﾔ	YA
2A	ﾕ	YU
2B	ｴ	E
2C	ﾖ	YO
2D	ﾗ	RA
2E	ﾘ	RI
2F	ﾙ	RU
30	ｵ	O
31	ｶ	KA
32	ｷ	KI
33	ｸ	KU
34	ｪ	SMALL E
35	ｫ	SMALL O
36	ｬ	SMALL YA
37	ｭ	SMALL YU
38	ｮ	SMALL YO
39	ｯ	SMALL TSU
3A	ﾚ	RE
3B	ﾛ	RO
3C	ﾜ	WA
3D	ﾝ	N
3E	ﾞ	VOICED
3F	ﾟ	SEMI VOICED

Figure 26. WCGM Assignments for Katakana Characters (in All Three Pitches)

WCGM Assignment	Graphic	Description
00		BLANK
01	A	A
02	B	B
03	C	C
04	D	D
05	E	E
06	F	F
07	G	G
08	H	H
09	I	I
0A	⑀	HOOK
0B	.	PERIOD OR DECIMAL POINT
0C[2]	·	
0D	▁	BOTTOM FIDUCIAL[1]
0E[2]	·	
0F	⑂	FORK
10	&	AMPERSAND
11	J	J
12	K	K
13	L	L
14	M	M
15	N	N
16	O	O
17	P	P
18	Q	Q
19	R	R
1A	⑁	CHAIR
1B	$	DOLLAR SIGN
1C	*	ASTERISK
1D	┫	CORNER FIDUCIAL[1]
1E	▐	RIGHT-HAND FIDUCIAL[1]
1F	▌	LEFT-HAND FIDUCIAL[1]
20	-	MINUS SIGN OR HYPHEN
21	/	SLASH
22	S	S
23	T	T
24	U	U
25	V	V
26	W	W
27	X	X
28	Y	Y
29	Z	Z
2A[2]	·	
2B	,	COMMA
2C[2]	·	
2D	—	TIMING MARK
2E[2]	·	
2F[2]	·	
30	0	ZERO
31	1	ONE
32	2	TWO
33	3	THREE
34	4	FOUR
35	5	FIVE
36	6	SIX
37	7	SEVEN
38	8	EIGHT
39	9	NINE
3A[2]	·	
3B[2]	·	
3C	|	LONG VERTICAL MARK[1]
3D[2]	·	
3E[2]	·	
3F[2]		

[1] These characters are not addressed by any IBM-supplied character arrangement table. Their recommended EBCDIC assignments are:

74 Bottom fiducial
75 Corner fiducial
76 Right-hand fiducial
77 Left-hand fiducial
FA Long vertical mark

[2] The symbol 🄽, denoting an unassigned character, is in the indicated locations in this character set.

Figure 27. WCGM Assignments for OCR-A Characters

WCGM Assignment	Graphic	Description
00		BLANK
01	A	A
02	B	B
03	C	C
04	D	D
05	E	E
06	F	F
07	G	G
08	H	H
09	I	I
0A[2]		
0B	.	PERIOD OR DECIMAL POINT
0C	<	LESS THAN
0D	▁	BOTTOM FIDUCIAL[1]
0E	+	PLUS SIGN
0F[2]		
10	&	AMPERSAND
11	J	J
12	K	K
13	L	L
14	M	M
15	N	N
16	O	O
17	P	P
18	Q	Q
19	R	R
1A[2]		
1B	$	DOLLAR SIGN
1C	*	ASTERISK
1D	┤	CORNER FIDUCIAL[1]
1E	▐	RIGHT-HAND FIDUCIAL[1]
1F	▌	LEFT-HAND FIDUCIAL[1]
20	-	MINUS SIGN OR HYPHEN
21	/	SLASH
22	S	S
23	T	T
24	U	U
25	V	V
26	W	W
27	X	X
28	Y	Y
29	Z	Z
2A[2]		
2B	,	COMMA
2C[2]		
2D	—	TIMING MARK
2E	>	GREATER THAN
2F[2]		
30	0	ZERO
31	1	ONE
32	2	TWO
33	3	THREE
34	4	FOUR
35	5	FIVE
36	6	SIX
37	7	SEVEN
38	8	EIGHT
39	9	NINE
3A[2]		
3B[2]		
3C	|	LONG VERTICAL MARK[1]
3D	'	PRIME OR APOSTROPHE
3E	=	EQUAL SIGN
3F[2]		

[1] These characters are not addressed by any IBM-supplied character arrangement table. Their recommended EBCDIC assignments are:

74	Bottom fiducial
75	Corner fiducial
76	Right-hand fiducial
77	Left-hand fiducial
FA	Long vertical mark

[2] The symbol 🄽 denoting an unassigned character, is in the indicated locations in this character set.

Figure 28. WCGM Assignments for OCR-B Characters

WCGM Assignment	Graphic	Description
00		BLANK
01	A	A
02	B	B
03	C	C
04	D	D
05	E	E
06	F	F
07	G	G
08	H	H
09	I	I
0A	¢	CENT SIGN
0B	.	PERIOD OR DECIMAL POINT
0C	<	LESS THAN
0D	(	LEFT PARENTHESIS
0E	+	PLUS SIGN
0F	|	LOGICAL OR
10	&	AMPERSAND
11	J	J
12	K	K
13	L	L
14	M	M
15	N	N
16	O	O
17	P	P
18	Q	Q
19	R	R
1A	!	EXCLAMATION POINT
1B	$	DOLLAR SIGN
1C	*	ASTERISK
1D	)	RIGHT PARENTHESIS
1E	;	SEMICOLON
1F	¬	LOGICAL NOT
20	-	MINUS SIGN OR HYPHEN
21	/	SLASH
22	S	S
23	T	T
24	U	U
25	V	V
26	W	W
27	X	X
28	Y	Y
29	Z	Z
2A	¤	LOZENGE
2B	,	COMMA
2C	%	PERCENT SIGN
2D	_	UNDERSCORE
2E	>	GREATER THAN
2F	?	QUESTION MARK
30	0	ZERO
31	1	ONE
32	2	TWO
33	3	THREE
34	4	FOUR
35	5	FIVE
36	6	SIX
37	7	SEVEN
38	8	EIGHT
39	9	NINE
3A	:	COLON
3B	#	NUMBER SIGN
3C	@	AT SIGN
3D	'	PRIME OR APOSTROPHE
3E	=	EQUAL SIGN
3F	"	QUOTATION MARK

Figure 29. WCGM Assignments for Text 1 and Text 1 Underscored Characters

WCGM Assignment	Graphic	Description
00		BLANK
01	a	LOWERCASE A
02	b	LOWERCASE B
03	c	LOWERCASE C
04	d	LOWERCASE D
05	e	LOWERCASE E
06	f	LOWERCASE F
07	g	LOWERCASE G
08	h	LOWERCASE H
09	i	LOWERCASE I
0A	[	OPEN SQUARE BRACKET
0B	•	BULLET
0C	≤	EQUAL OR LESS THAN
0D	{	OPENING BRACE
0E	±	PLUS OR MINUS
0F	■	HISTOGRAM
10	°	DEGREE
11	j	LOWERCASE J
12	k	LOWERCASE K
13	l	LOWERCASE L
14	m	LOWERCASE M
15	n	LOWERCASE N
16	o	LOWERCASE O
17	p	LOWERCASE P
18	q	LOWERCASE Q
19	r	LOWERCASE R
1A	+	D A CROSS
1B	]	CLOSE SQUARE BRACKET
1C	—	EXTENDED DASH
1D	}	CLOSING BRACE
1E	↑	UP ARROW[1]
1F	§	SECTION SIGN[1]
20	¶	PARAGRAPH SIGN[1]
21	≠	NOT EQUAL
22	s	LOWERCASE S
23	t	LOWERCASE T
24	u	LOWERCASE U
25	v	LOWERCASE V
26	w	LOWERCASE W
27	x	LOWERCASE X
28	y	LOWERCASE Y
29	z	LOWERCASE Z
2A	└	LOWER LEFT CORNER
2B	┐	UPPER RIGHT CORNER
2C	┌	UPPER LEFT CORNER
2D	┘	LOWER RIGHT CORNER
2E	≥	EQUAL OR GREATER THAN
2F	\	REVERSE SLANT[1]
30	^{0}	SUPERSCRIPT ZERO
31	^{1}	SUPERSCRIPT ONE
32	^{2}	SUPERSCRIPT TWO
33	^{3}	SUPERSCRIPT THREE
34	^{4}	SUPERSCRIPT FOUR
35	^{5}	SUPERSCRIPT FIVE
36	^{6}	SUPERSCRIPT SIX
37	^{7}	SUPERSCRIPT SEVEN
38	^{8}	SUPERSCRIPT EIGHT
39	^{9}	SUPERSCRIPT NINE
3A	$^{-}$	SUPERSCRIPT MINUS
3B	$^{+}$	SUPERSCRIPT PLUS
3C	$^{(}$	SUPERSCRIPT LEFT PARENTHESIS
3D	$^{)}$	SUPERSCRIPT RIGHT PARENTHESIS
3E	†	DAGGER[1]
3F	‡	DOUBLE DAGGER[1]

[1] These characters are not addressed by any IBM-supplied character arrangement table. Their recommended EBCDIC assignments are:

8A	Up arrow	E0	Reverse slant
DC	Section sign	9A	Dagger
DD	Paragraph sign	90	Double dagger

Figure 30. WCGM Assignments for Text 2 and Text 2 Underscored Characters

WCGM Assignment	Graphic	Description
00		BLANK
01	┌	TOP LEFT CORNER - WEIGHT 1
02	┌	TOP LEFT CORNER - WEIGHT 2
03	┏	TOP LEFT CORNER - WEIGHT 3
04	┐	TOP RIGHT CORNER - WEIGHT 1
05	┐	TOP RIGHT CORNER - WEIGHT 2
06	┓	TOP RIGHT CORNER - WEIGHT 3
07	└	BOTTOM LEFT CORNER - WEIGHT 1
08	└	BOTTOM LEFT CORNER - WEIGHT 2
09	┗	BOTTOM LEFT CORNER - WEIGHT 3
0A	┘	BOTTOM RIGHT CORNER - WEIGHT 1
0B	┘	BOTTOM RIGHT CORNER - WEIGHT 2
0C	┛	BOTTOM RIGHT CORNER - WEIGHT 3
0D	├	LEFT JUNCTION - WEIGHT 1
0E	├	LEFT JUNCTION - WEIGHT 2
0F	┣	LEFT JUNCTION - WEIGHT 3
10	┤	RIGHT JUNCTION - WEIGHT 1
11	┤	RIGHT JUNCTION - WEIGHT 2
12	┫	RIGHT JUNCTION - WEIGHT 3
13	┬	TOP JUNCTION - WEIGHT 1
14	┬	TOP JUNCTION - WEIGHT 2
15	┳	TOP JUNCTION - WEIGHT 3
16	┴	BOTTOM JUNCTION - WEIGHT 1
17	┴	BOTTOM JUNCTION - WEIGHT 2
18	┻	BOTTOM JUNCTION - WEIGHT 3
19	┼	INTERSECTION - WEIGHT 1
1A	┼	INTERSECTION - WEIGHT 2
1B	╋	INTERSECTION - WEIGHT 3
1C	─	HORIZONTAL LINE SEGMENT - WEIGHT 1
1D	─	HORIZONTAL LINE SEGMENT - WEIGHT 2
1E	━	HORIZONTAL LINE SEGMENT - WEIGHT 3
1F	│	VERTICAL LINE SEGMENT - WEIGHT 1
20	│	VERTICAL LINE SEGMENT - WEIGHT 2
21	┃	VERTICAL LINE SEGMENT - WEIGHT 3
22	╎	VERTICAL BROKEN LINE - WEIGHT 1
23	╏	VERTICAL BROKEN LINE - WEIGHT 2
24	⋮	VERTICAL DOTTED LINE - WEIGHT 2

Note: The symbol 🄽 denoting an unassigned character is in locations X'25' through X'3F' of the Format character sets.

Figure 31. WCGM Assignments for Format Characters (in All Three Pitches)

APPENDIX B: IBM-SUPPLIED CHARACTER ARRANGEMENT TABLES

Figure 32 lists the character arrangement tables supplied with the 3800 programming support. Figures 33 through 40 show the EBCDIC and WCGM assignments of the characters in each table.

System Generation Group	Character Arrangement Table Names	Character Set	Pitch[1]	Number of Graphic Characters[2]
Basic group	GS10	Gothic-10	10	63
	GS12	Gothic-12	12	63
	GS15	Gothic-15	15	63
	GSC	Gothic-15 Condensed	15	63
	GF10[3]	Gothic-10 – folded	10	62
	GF12[3]	Gothic-12 – folded	12	62
	GF15[3]	Gothic-15 – folded	15	62
	GFC[3]	Gothic-15 condensed – folded	15	62
	GU10	Gothic-10 underscored	10	63
	GU12	Gothic-12 underscored	12	63
	GU15	Gothic-15 underscored	15	63
	GUC	Gothic-15 Condensed underscored	15	63
	TU10[4]	Text 1 & 2 underscored	10	120
	DUMP[4]	Gothic-15 and underscored Gothic-15	15	79
3211 group	A11	Gothic-10	10	48
	G11	Gothic-10	10	63
	H11	Gothic-10	10	48
	P11	Gothic-10	10	60
	T11[4]	Text 1 & 2	10	120
1403 group	AN	Gothic-10	10	48
	GN	Gothic-10	10	63
	HN	Gothic-10	10	48
	PCAN	Gothic-10	10	48
	PCHN	Gothic-10	10	48
	PN	Gothic-10	10	60
	QN	Gothic-10	10	60
	QNC	Gothic-10	10	60
	RN	Gothic-10	10	52
	XN	Gothic-10	10	40
	YN	Gothic-10	10	42
	SN[4]	Text 1 & 2	10	84
	TN[4]	Text 1 & 2	10	120
OCR group	AOA[4]	Gothic-10, OCR-A	10	48
	AOD[4]	Gothic-10, OCR-A	10	48
	AON[4]	Gothic-10, OCR-A	10	48
	OAA[4]	Gothic-10, OCR-A	10	48
	ODA[4]	Gothic-10, OCR-A	10	48
	ONA[4]	Gothic-10, OCR-A	10	48
	BOA[4]	Gothic-10, OCR-B	10	48
	BON[4]	Gothic-10, OCR-B	10	48
	OAB	OCR-B	10	48
	ONB[4]	Gothic-10, OCR-B	10	48
Katakana group	2773[4]	Gothic-10, Katakana-10	10	62
	2774[4]	Gothic-10, Katakana-10	10	108
	KN1[4]	Gothic-10, Katakana-10	10	127
Format group	FM10	Format-10	10	36
	FM12	Format-15	12	36
	FM15	Format-15	15	36

[1] For any table using 10-pitch Gothic or Katakana, the pitch can be changed to 12 or 15 by changing the character set identifier using the IEBIMAGE utility. (For the GN and G11 tables, a 12-pitch or 15-pitch graphic character modification module equivalent to SPC1 is also required.)

[2] Not including the blank.

[3] The GF10, GF12, GF15, and GFC tables provide the folding effect to allow the printing of uppercase characters when lowercase are called for in the print data.

[4] This character arrangement table uses two WCGMs.

Figure 32. Character Arrangement Tables Supplied with the 3800

EBCDIC Assignment	WCGM Location	Graphic	Description
40	00		Blank
4A	0A	¢	Cent sign
4B	0B	.	Period or Decimal Point
4C	0C	<	Less than
4D	0D	(	Left parenthesis
4E	0E	+	Plus sign
4F	0F	|	Logical Or
50	10	&	Ampersand
5A	1A	!	Exclamation point
5B	1B	$	Dollar sign
5C	1C	*	Asterisk
5D	1D	)	Right parenthesis
5E	1E	;	Semicolon
5F	1F	¬	Logical not
60	20	-	Minus sign
61	21	/	Slash
6B	2B	,	Comma
6C	2C	%	Percent sign
6D	2D	_	Underscore
6E	2E	>	Greater than
6F	2F	?	Question mark
7A	3A	:	Colon
7B	3B	#	Number sign
7C	3C	@	At sign
7D	3D	'	Prime or Apostrophe
7E	3E	=	Equal sign
7F	3F	"	Quotation mark
9C	2A	¤	Lozenge
C1	01	A	A
C2	02	B	B
C3	03	C	C
C4	04	D	D
C5	05	E	E
C6	06	F	F
C7	07	G	G
C8	08	H	H
C9	09	I	I
D1	11	J	J
D2	12	K	K
D3	13	L	L
D4	14	M	M
D5	15	N	N
D6	16	O	O
D7	17	P	P
D8	18	Q	Q
D9	19	R	R
E2	22	S	S
E3	23	T	T
E4	24	U	U
E5	25	V	V
E6	26	W	W
E7	27	X	X
E8	28	Y	Y
E9	29	Z	Z
F0	30	0	Zero
F1	31	1	One
F2	32	2	Two
F3	33	3	Three
F4	34	4	Four
F5	35	5	Five
F6	36	6	Six
F7	37	7	Seven
F8	38	8	Eight
F9	39	9	Nine

Figure 33. The GS10, GS12, GS15, and GSC; and GU10, GU12, GU15, and GUC (Underscored) Character Arrangement Tables

EBCDIC Assignment				WCGM Location	Graphic	Description
00	40	80	C0	00		Blank
01	41	81	C1	01	A	A
02	42	82	C2	02	B	B
03	43	83	C3	03	C	C
04	44	84	C4	04	D	D
05	45	85	C5	05	E	E
06	46	86	C6	06	F	F
07	47	87	C7	07	G	G
08	48	88	C8	08	H	H
09	49	89	C9	09	I	I
0A	4A	8A	CA	0A	¢	Cent sign
0B	4B	8B	CB	0B	.	Period or Decimal point
0C	4C	8C	CC	0C	<	Less than
0D	4D	8D	CD	0D	(	Left parenthesis
0E	4E	8E	CE	0E	+	Plus sign
0F	4F	8F	CF	0F	|	Logical Or
10	50	90	D0	10	&	Ampersand
11	51	91	D1	11	J	J
12	52	92	D2	12	K	K
13	53	93	D3	13	L	L
14	54	94	D4	14	M	M
15	55	95	D5	15	N	N
16	56	96	D6	16	O	O
17	57	97	D7	17	P	P
18	58	98	D8	18	Q	Q
19	59	99	D9	19	R	R
1A	5A	9A	DA	1A	!	Exclamation point
1B	5B	9B	DB	1B	$	Dollar sign
1C	5C	9C	DC	1C	*	Asterisk
1D	5D	9D	DD	1D	)	Right parenthesis
1E	5E	9E	DE	1E	;	Semicolon
1F	5F	9F	DF	1F	¬	Logical not
20	60	A0	E0	20	-	Minus sign or Hyphen
21	61	A1	E1	21	/	Slash
22	62	A2	E2	22	S	S
23	63	A3	E3	23	T	T
24	64	A4	E4	24	U	U
25	65	A5	E5	25	V	V
26	66	A6	E6	26	W	W
27	67	A7	E7	27	X	X
28	68	A8	E8	28	Y	Y
29	69	A9	E9	29	Z	Z
2B	6B	AB	EB	2B	,	Comma
2C	6C	AC	EC	2C	%	Percent sign
2D	6D	AD	ED	2D	_	Underscore
2E	6E	AE	EE	2E	>	Greater than
2F	6F	AF	EF	2F	?	Question mark
30	70	B0	F0	30	0	Zero
31	71	B1	F1	31	1	One
32	72	B2	F2	32	2	Two
33	73	B3	F3	33	3	Three
34	74	B4	F4	34	4	Four
35	75	B5	F5	35	5	Five
36	76	B6	F6	36	6	Six
37	77	B7	F7	37	7	Seven
38	78	B8	F8	38	8	Eight
39	79	B9	F9	39	9	Nine
3A	7A	BA	FA	3A	:	Colon
3B	7B	BB	FB	3B	#	Number sign
3C	7C	BC	FC	3C	@	At sign
3D	7D	BD	FD	3D	'	Prime or Apostrophe
3E	7E	BE	FE	3E	=	Equal sign
3F	7F	BF	FF	3F	"	Quotation mark

Note: The machine default for the 3800 is the same as the GF10 character arrangement table except that EBCDIC assignments 2A, 6A, AA, and EA address WCGM location 2A to cause the lozenge to print.

Figure 34 The GF10, GF12, GF15, and GFC Folded Character Arrangement Tables

Gothic-10 WCGM		Character Arrangement Table								
		Number of characters in table								
		48	48	60	63	48	52	42	40	
				QNC						
			PCHN	QN						
EBCDIC	WCGM		HN	PN	GN[1]	PCAN				
Assignment	Location	A11	H11	P11	G11[1]	AN	RN	YN	XN	Description
40	00	Sp	Sp	Sp	Sp	Sp	Sp	Sp	Sp	Blank
4A	0A				[					Open bracket
4B	0B	.	.	.	.	.	.	.	.	Period or Decimal Point
4C	0C	<		<	<					Less than
4C	2A					¤	¤			Lozenge
4D	0D		(	(	(		(			Left parenthesis
4E	0E	+	+	+	+	+	+			Plus sign
4F	0F			|	|					Logical Or
50	10	&	&	&	&	&	&			Ampersand
5A	1A				]					Close bracket
5B	1B	$	$	$	$	$	$	$	$	Dollar sign
5C	1C	*	*	*	*	*	*	*	*	Asterisk
5D	1D		)	)	)		)			Right parenthesis
5E	1E			;	;					Semicolon
5F	1F			¬	¬					Logical not
60	20	-	-	-	-	-	-	-		Minus sign or Hyphen
61	21	/	/	/	/	/	/			Slash
6B	2B	,	,	,	,	,	,	,	,	Comma
6C	2C	%		%	%	%	%			Percent sign
6D	2D			_	_					Underscore
6E	2E			>	>					Greater than
6F	2F			?	?					Question mark
7A	3A			:	:					Colon
7B	3B	#		#	#	#	#	#		Number sign
7C	3C	@		@	@	@	@			At sign
7D	3D		'	'	'		'			Prime or Apostrophe
7E	3E		=	=	=		=			Equal sign
7F	3F			"	"					Quotation mark
C1-C9	01-09	A-I	A-I	A-I	A-I	A-I	A-I	A-I	A-I	A-I
D1-D9	11-19	J-R	J-R	J-R	J-R	J-R	J-R	J-R	J-R	J-R
E0	2A				\					Reverse slant
E2-E9	22-29	S-Z	S-Z	S-Z	S-Z	S-Z	S-Z	S-Z	S-Z	S-Z
F0-F9	30-39	0-9	0-9	0-9	0-9	0-9	0-9	0-9	0-9	Zero-Nine

[1] The GN and G11 character arrangement tables use the graphic character modification module named SPC1 to supply the open bracket ([), close bracket (]), and reverse slant (\).

Figure 35. The 3211 Group and 1403 Group of Gothic Character Arrangement Tables

EBCDIC Assignment	WCGM Number[1]	WCGM Location	Character Arrangement Table TN T11 TU10	SN	Description
40	0	00	Sp	Sp	Blank
4A	0	0A	¢	¢	Cent sign
4B	0	0B	.	.	Period or Decimal point
4C	0	0C	<		Less than
4D	0	0D	(	(	Left parenthesis
4E	0	0E	+	+	Plus sign
4F	0	0F	|	|	Logical Or
50	0	10	&	&	Ampersand
5A	0	1A	!	!	Exclamation point
5B	0	1B	$	$	Dollar sign
5C	0	1C	*	*	Asterisk
5D	0	1D	)	)	Right parenthesis
5E	0	1E	;	;	Semicolon
5F	0	1F	¬		Logical not
60	0	20	–	–	Minus sign or Hyphen
61	0	21	/	/	Slash
6B	0	2B	,	,	Comma
6C	0	2C	%	%	Percent sign
6D	0	2D	_	—	Underscore
6E	0	2E	>		Greater than
6F	0	2F	?	?	Question mark
7A	0	3A	:	:	Colon
7B	0	3B	#		Number sign
7C	0	3C	@	@	At sign
7D	0	3D	'	'	Prime or Apostrophe
7E	0	3E	=		Equal sign
7F	0	3F	"	"	Quotation mark
81-89	1	01-09	a-i	a-i	a-i
8B	1	0D	{		Opening brace
8C	1	0C	≤		Equal or Less than
8D	1	3C	(		Superscript left parenthesis
8E	1	3B	+		Superscript plus
8F	1	1A	+		D A Cross
91-99	1	11-19	j-r	j-r	j-r
9B	1	1D	}		Closing brace
9C	0	2A	¤	¤	Lozenge
9D	1	3D	)		Superscript right parenthesis
9E	1	0E	±		Plus or Minus
9F	1	0F	■		Histogram
A0	1	3A	-		Superscript Minus
A1	1	10	°		Degree
A2-A9	1	22-29	s-z	s-z	s-z
AB	1	2A	└		Lower left corner
AC	1	2C	┌		Upper left corner
AD	1	0A	[		Open square bracket
AE	1	2E	≥		Equal or Greater than
AF	1	0B	•		Bullet
B0-B9	1	30-39	0-9		Superscript zero-nine
BB	1	2D	┘		Lower right corner
BC	1	2B	┐		Upper right corner
BD	1	1B	]		Close square bracket
BE	1	21	≠		Not equal
BF	1	1C	—		Extended dash
C1-C9	0	01-09	A-I	A-I	A-I
D1-D9	0	11-19	J-R	J-R	J-R
E2-E9	0	22-29	S-Z	S-Z	S-Z
F0-F9	0	30-39	0-9	0-9	Zero-Nine

[1] WCGM 0 contains the Text 1 character set and WCGM 1 contains the Text 2 character set.

Figure 36. The TU10, T11, TN, and SN Character Arrangement Tables

EBCDIC Assignment	WCGM Number[1]	WCGM Location	Graphic	Description
01-06	1	01-06	A̲-F̲	Underscored A-F
30-39	1	30-39	0̲-9̲	Underscored Zero-Nine
40	0	00		Blank
4A	0	0A	¢	Cent sign
4B	0	0B	.	Period or Decimal point
4C	0	0C	<	Less than
4D	0	0D	(	Left parenthesis
4E	0	0E	+	Plus sign
4F	0	0F	|	Logical Or
50	0	10	&	Ampersand
5A	0	1A	!	Exclamation point
5B	0	1B	$	Dollar sign
5C	0	1C	*	Asterisk
5D	0	1D	)	Right parenthesis
5E	0	1E	;	Semicolon
5F	0	1F	¬	Logical not
60	0	20	-	Minus sign or Hyphen
61	0	21	/	Slash
6B	0	2B	,	Comma
6C	0	2C	%	Percent sign
6D	0	2D	_	Underscore
6E	0	2E	>	Greater than
6F	0	2F	?	Question mark
7A	0	3A	:	Colon
7B	0	3B	#	Number sign
7C	0	3C	@	At sign
7D	0	3D	'	Prime or Apostrophe
7E	0	3E	=	Equal sign
7F	0	3F	"	Quotation mark
9C	0	2A	¤	Lozenge
C1-C9	0	01-09	A-I	A-I
D1-D9	0	11-19	J-R	J-R
E2-E9	0	22-29	S-Z	S-Z
F0-F9	0	30-39	0-9	Zero-Nine

[1] WCGM 0 contains the Gothic 15-pitch character set and WCGM 1 contains the Gothic Underscored 15-pitch character set.

Figure 37. The DUMP Character Arrangement Table

Character Arrangement Tables
Gothic and OCR-A

EBCDIC	WCGM	AOA OAA		AON ONA		ODA		AOD		
Assignment	Location	Gothic	OCR-A	Gothic	OCR-A	Gothic	OCR-A	Gothic	OCR-A	Description
40	00	Sp		Sp		Sp		Sp		Blank
4B	0B		.	.		.		.		Period or Decimal point
4C	0C	<						<		Less than
4C	2A					⌑				Lozenge
4E	0E	+				+		+		Plus sign
50	10		&	&		&		&		Ampersand
5B	1B		$	$		$		$		Dollar sign
5C	1C		*	*		*		*		Asterisk
60	20		–	–		–		–		Minus sign or Hyphen
61	21		/	/		/		/		Slash
6B	2B		,	,		,		,		Comma
6C	2C	%				%		%		Percent sign
7B	3B	#		#		#		#		Number sign
7C	3C	@		@		@		@		At sign
C1-C9	01-09		A-I	A-I		A-I		A-I		A-I
CC	0A				⑀					Hook
CE	0F				⑂					Fork
D1-D9	11-19		J-R	J-R		J-R		J-R		J-R
E2-E9	22-29		S-Z	S-Z		S-Z		S-Z		S-Z
EC	1A				⑁					Chair
F0-F9	30-39		0-9		0-9		0-9		0-9	Zero-Nine

Character Arrangement Tables
Gothic and OCR-B

EBCDIC	WCGM	BOA		BON ONB		OCR-B OAB	
Assignment	Location	Gothic	OCR-B	Gothic	OCR-B	OCR-B	Description
40	00	Sp		Sp		Sp	Blank
4B	0B		.	.		.	Period or Decimal point
4C	0C		<		<	<	Less than
4E	0E		+		+	+	Plus sign
50	10		&	&		&	Ampersand
5B	1B		$	$		$	Dollar sign
5C	1C		*	*		*	Asterisk
60	20		–	–		–	Minus sign or Hyphen
61	21		/	/		/	Slash
6B	2B		,	,		,	Comma
6E	2E		>		>	>	Greater than
7D	3D	'		'		'	Prime or Apostrophe
7E	3E	=		=		=	Equal sign
C1-C9	01-09		A-I	A-I		A-I	A-I
D1-D9	11-19		J-R	J-R		J-R	J-R
E2-E9	22-29		S-Z	S-Z		S-Z	S-Z
F0-F9	30-39		0-9		0-9	0-9	Zero-Nine

Note: Each of the character arrangement tables is a combination of Gothic and OCR characters, except for the OAB table which is entirely OCR. For those other than OAB, the Gothic character set is loaded into the first of the two WCGMs used.

Figure 38. The OCR Group of Character Arrangement Tables

EBCDIC	WCGM	WCGM	Character Arrangement Table			
Assignment	Number[1]	Location	2773	2774	KN1	Description
40	0	00	Sp	Sp	Sp	Blank
41	1	01			｡	Kana period
42	1	02			｢	Open bracket
43	1	03			｣	Close bracket
44	1	04			､	Kana comma
45	1	05			･	Center mark
46	1	06	ｦ	ｦ	ｦ	Wo
47	1	07			ｧ	Small A
48	1	08			ｨ	Small I
49	1	09			ｩ	Small U
4A	0	0A			¢	Cent sign
4B	0	0B	.	.	.	Period
4C	0	0C		<	<	Less than
4D	0	0D		(	(	Left parenthesis
4E	0	0E		+	+	Plus sign
4F	0	0F		|	|	Logical Or
50	0	10		&	&	Ampersand
51	1	34			ｪ	Small E
52	1	35			ｫ	Small O
53	1	36			ｬ	Small Ya
54	1	37			ｭ	Small Yu
55	1	38			ｮ	Small Yo
56	1	39			ｯ	Small Tsu
58	1	1B			ｰ	Prolonged sound
5A	0	1A			!	Exclamation point
5B	1	00		¥	¥	Yen
5C	0	1C	*	*	*	Asterisk
5D	0	1D		)	)	Right parenthesis
5E	0	1E		;	;	Semicolon
5F	0	1F		¬	¬	Logical not
60	0	20	-	-	-	Minus sign or Hyphen
61	0	21		/	/	Slash
6B	0	2B	,	,	,	Comma
6C	0	2C		%	%	Percent sign
6D	0	2D		_	_	Underscore
6E	0	2E		>	>	Greater than
6F	0	2F		?	?	Question mark
7A	0	3A		:	:	Colon
7B	0	3B		#	#	Number sign
7C	0	3C		@	@	At sign
7D	0	3D		'	'	Prime or Apostrophe
7E	0	3E		=	=	Equal sign
7F	0	3F		"	"	Quotation mark
81	1	1C	ｱ	ｱ	ｱ	A
82	1	20	ｲ	ｲ	ｲ	I
83	1	21	ｳ	ｳ	ｳ	U
84	1	2B	ｴ	ｴ	ｴ	E
85	1	30	ｵ	ｵ	ｵ	O

[1] WCGM 0 contains the Gothic-10 character set and WCGM 1 contains the Katakana-10 character set.

Figure 39 (Part 1 of 2). The Katakana Group of Character Arrangement Tables

EBCDIC Assignment	WCGM Number[1]	WCGM Location	Character Arrangement Table			Description
			2773	2774	KN1	
86	1	31	ｶ	ｶ	ｶ	Ka
87	1	32	ｷ	ｷ	ｷ	Ki
88	1	33	ｸ	ｸ	ｸ	Ku
89	1	0B	ｹ	ｹ	ｹ	Ke
8A	1	0A	ｺ	ｺ	ｺ	Ko
8C	1	0C	ｻ	ｻ	ｻ	Sa
8D	1	0D	ｼ	ｼ	ｼ	Shi
8E	1	0E	ｽ	ｽ	ｽ	Su
8F	1	0F	ｾ	ｾ	ｾ	Se
90	1	10	ｿ	ｿ	ｿ	So
91	1	11	ﾀ	ﾀ	ﾀ	Ta
92	1	12	ﾁ	ﾁ	ﾁ	Chi
93	1	13	ﾂ	ﾂ	ﾂ	Tsu
94	1	14	ﾃ	ﾃ	ﾃ	Te
95	1	15	ﾄ	ﾄ	ﾄ	To
96	1	16	ﾅ	ﾅ	ﾅ	Na
97	1	17	ﾆ	ﾆ	ﾆ	Ni
98	1	18	ﾇ	ﾇ	ﾇ	Nu
99	1	19	ﾈ	ﾈ	ﾈ	Ne
9A	1	1A	ﾉ	ﾉ	ﾉ	No
9C	0	2A			¤	Lozenge
9D	1	1D	ﾊ	ﾊ	ﾊ	Ha
9E	1	1E	ﾋ	ﾋ	ﾋ	Hi
9F	1	1F	ﾌ	ﾌ	ﾌ	Fu
A2	1	22	ﾍ	ﾍ	ﾍ	He
A3	1	23	ﾎ	ﾎ	ﾎ	Ho
A4	1	24	ﾏ	ﾏ	ﾏ	Ma
A5	1	25	ﾐ	ﾐ	ﾐ	Mi
A6	1	26	ﾑ	ﾑ	ﾑ	Mu
A7	1	27	ﾒ	ﾒ	ﾒ	Me
A8	1	28	ﾓ	ﾓ	ﾓ	Mo
A9	1	29	ﾔ	ﾔ	ﾔ	Ya
AA	1	2A	ﾕ	ﾕ	ﾕ	Yu
AC	1	2C	ﾖ	ﾖ	ﾖ	Yo
AD	1	2D	ﾗ	ﾗ	ﾗ	Ra
AE	1	2E	ﾘ	ﾘ	ﾘ	Ri
AF	1	2F	ﾙ	ﾙ	ﾙ	Ru
BA	1	3A	ﾚ	ﾚ	ﾚ	Re
BB	1	3B	ﾛ	ﾛ	ﾛ	Ro
BC	1	3C	ﾜ	ﾜ	ﾜ	Wa
BD	1	3D	ﾝ	ﾝ	ﾝ	N
BE	1	3E	ﾞ	ﾞ	ﾞ	Voiced
BF	1	3F	ﾟ	ﾟ	ﾟ	Semi-voiced
C1-C9	0	01-09		A-I	A-I	A-I
D1-D9	0	11-19		J-R	J-R	J-R
E0	0	1B			$	Dollar sign
E2-E9	0	22-29		S-Z	S-Z	S-Z
F0-F9	0	30-39	0-9	0-9	0-9	Zero-Nine

[1] WCGM 0 contains the Gothic-10 character set and WCGM 1 contains the Katakana-10 character set.

Figure 39 (Part 2 of 2). The Katakana Group of Character Arrangement Tables

Data Code Assignment	Keypunch Equivalent	WCGM Location	Graphic	Description
40	Sp	00		Blank
C1	A	01	┌	Top left corner–weight 1
D3	L	02	┌	Top left corner–weight 2
E6	W	03	┏	Top left corner–weight 3
C2	B	04	┐	Top right corner–weight 1
D4	M	05	┐	Top right corner–weight 2
E7	X	06	┓	Top right corner–weight 3
C4	D	07	└	Bottom left corner–weight 1
D6	O	08	└	Bottom left corner–weight 2
E9	Z	09	┗	Bottom left corner–weight 3
C3	C	0A	┘	Bottom right corner–weight 1
D5	N	0B	┘	Bottom right corner–weight 2
E8	Y	0C	┛	Bottom right corner–weight 3
C7	G	0D	├	Left junction–weight 1
D9	R	0E	├	Left junction–weight 2
F3	3	0F	┣	Left junction–weight 3
C8	H	10	┤	Right junction–weight 1
E2	S	11	┤	Right junction–weight 2
F4	4	12	┫	Right junction–weight 3
C5	E	13	┬	Top junction–weight 1
D7	P	14	┬	Top junction–weight 2
F1	1	15	┳	Top junction–weight 3
C6	F	16	┴	Bottom junction–weight 1
D8	Q	17	┴	Bottom junction–weight 2
F2	2	18	┻	Bottom junction–weight 3
C9	I	19	┼	Intersection–weight 1
E3	T	1A	┼	Intersection–weight 2
F5	5	1B	╋	Intersection–weight 3
D1	J	1C	─	Horizontal line segment–weight 1
E4	U	1D	─	Horizontal line segment–weight 2
F6	6	1E	━	Horizontal line segment–weight 3
D2	K	1F	│	Vertical line segment–weight 1
E5	V	20	│	Vertical line segment–weight 2
F7	7	21	┃	Vertical line segment–weight 3
F8	8	22	╎	Vertical broken line–weight 1
F9	9	23	╏	Vertical broken line–weight 2
F0	0	24	⋮	Vertical dotted line–weight 2

Figure 40. The Format Group of Character Arrangement Tables

APPENDIX C: WORLD TRADE NATIONAL USE GRAPHICS

The IBM-supplied graphic character modification module named GRAFƀ ƀ ƀ ƀ contains the modification data for the standard substitution characters known collectively as the "World Trade National Use Graphics." This is the only graphic character modification module that is not restricted to a maximum of 64 segments. It cannot be referred to by a TABLE statement (when a character arrangement table is built) or by an INCLUDE statement (when a graphic character modification module is built). To use it, you select segments (that is, characters) to become part of the graphic character modification module that you build using the GRAPHIC statement.

The characters making up GRAFƀ ƀ ƀ ƀ, with their EBCDIC assignments and segment numbers, are shown in Figure 41. The designs of these characters can be printed out using the IEBIMAGE utility (Example 12 in "The IEBIMAGE Utility Program" chapter shows how to do this).

Gothic Characters			Gothic Underscored Characters					
Segment Number			Segment Number					
10-pitch	12-pitch	15-pitch	10-pitch	12-pitch	15-pitch	EBCDIC Assignment	Graphic	Description
001	041	081	194	234	274	5B	Ü	U UMLAUT
002	042	082	195	235	275	7B	Ä	A UMLAUT
003	043	083	196	236	276	7C	Ö	O UMLAUT
004	044	084	197	237	277	5B	Ç	C CEDILLA
005	045	085	198	238	278	E0	Ç	C CEDILLA
006	046	086	199	239	279	7B	Æ	DIGRAPH
007	047	087	200	240	280	7C	Ø	O SCANDINAVIAN
008	048	088	201	241	281	5B	₧	PESETA
009	049	089	202	242	282	7B	Ñ	N TILDE
010	050	090	203	243	283	7B	£	POUND STERLING
011	051	091	204	244	284	7B	Õ	O TILDE
012	052	092	205	245	285	7C	Õ	O TILDE
013	053	093	206	246	286	7B	Ã	A TILDE
014	054	094	207	247	287	7C	Ã	A TILDE
015	055	095	208	248	288	5B	Å	A SCANDINAVIAN
016	056	096	209	249	289	5B	¥	YEN
017	057	097	210	250	290	E0	É	E ACUTE
018	058	098	211	251	291	4A	É	E ACUTE
019	059	099	212	252	292	A1	ß	ESS TSET
020	060	100	213	253	293	4A	§	SECTION SIGN
021	061	101	214	254	294	E0	§	SECTION SIGN
022	062	102	215	255	295	5A	§	SECTION SIGN
023	063	103	216	256	296	7C	§	SECTION SIGN
024	064	104	217	257	297	4A	[	OPEN SQUARE BRACKET
025	065	105	218	258	298	5A	]	CLOSE SQUARE BRACKET
026	066	106	219	259	299	C0	{	OPENING BRACE
027	067	107	220	260	300	D0	}	CLOSING BRACE
028	068	108	221	261	301	E0	\	REVERSE SLANT
029	069	109	222	262	302	4A	•	CIRCLE ABOVE
030	070	110	223	263	303	5A	¤	CURRENCY
031	071	111	224	264	304	7B	#	NUMBER SIGN
032	072	112	225	265	305	4A	#	NUMBER SIGN
033	073	113	226	266	306	5B	$	DOLLAR SIGN
034	074	114	227	267	307	5A	$	DOLLAR SIGN
035	075	115	228	268	308	7C	@	AT SIGN
036	076	116	229	269	309	4A	¢	CENT SIGN
037	077	117	230	270	310	5A	!	EXCLAMATION POINT
038	078	118	231	271	311	5F	¬	LOGICAL NOT
039	079	119	232	272	312	A1	‾	OVERSCORE
040	080	120	233	273	313	6A	¦	VERTICAL BROKEN LINE

Figure 41 (Part 1 of 4). The World Trade National Use Graphics

Text 1 Characters	Text 1 Underscored Characters			
Segment Number	Segment Number	EBCDIC Assignment	Graphic	Description
121	314	5B	Ü	U UMLAUT
122	315	7B	Ä	A UMLAUT
123	316	7C	Ö	O UMLAUT
124	317	5B	Ç	C CEDILLA
125	318	E0	Ç	C CEDILLA
126	319	7B	Æ	DIGRAPH
127	320	7C	Ø	O SCANDINAVIAN
128	321	5B	₧	PESETA
129	322	7B	Ñ	N TILDE
130	323	7B	£	POUND STERLING
131	324	7B	Õ	O TILDE
132	325	7C	Õ	O TILDE
133	326	7B	Ã	A TILDE
134	327	7C	Ã	A TILDE
135	328	5B	Å	A SCANDINAVIAN
136	329	5B	¥	YEN
137	330	E0	É	E ACUTE
138	331	4A	É	E ACUTE
139	332	5B	$	DOLLAR SIGN
140	333	5A	$	DOLLAR SIGN
141	334	7C	@	AT SIGN
142	335	4A	¢	CENT SIGN
143	336	5A	!	EXCLAMATION POINT
144	337	E0	\	REVERSE SLANT

Text 2 Characters	Text 2 Underscored Characters			
Segment Number	Segment Number	EBCDIC Assignment	Graphic	Description
145	338	A1	ü	U UMLAUT
146	339	D0	ü	U UMLAUT
147	340	C0	ä	A UMLAUT
148	341	6A	ö	O UMLAUT
149	342	A1	ç	C CEDILLA
150	343	6A	ç	C CEDILLA
151	344	E0	ç	C CEDILLA
152	345	C0	æ	DIGRAPH
153	346	6A	ø	O SCANDINAVIAN
154	347	6A	ñ	N TILDE
155	348	C0	õ	O TILDE
156	349	6A	õ	O TILDE
157	350	C0	ã	A TILDE
158	351	79	ã	A TILDE
159	352	D0	å	A SCANDINAVIAN
160	353	79	é	E ACUTE
161	354	D0	é	E ACUTE
162	355	C0	é	E ACUTE
163	356	5A	é	E ACUTE
164	357	A1	ß	ESS TSET
165	358	C0	à	A GRAVE
166	359	7C	à	A GRAVE
167	360	D0	è	E GRAVE
168	361	A1	ì	I GRAVE
169	362	6A	ò	O GRAVE
170	363	6A	ù	U GRAVE
171	364	79	ù	U GRAVE

Figure 41 (Part 2 of 4). The World Trade National Use Graphics

OCR-A (10-pitch) Characters

Segment Number	EBCDIC Assignment	Graphic	Description
172	5B	Ü	U UMLAUT
173	7B	Ä	A UMLAUT
174	7C	Ö	O UMLAUT
175	5B	Å	A SCANDINAVIAN
176	7B	Æ	DIGRAPH
177	7C	Ø	O SCANDINAVIAN
178	5B	$	DOLLAR SIGN
179	5A	$	DOLLAR SIGN
180	7B	Ñ	N TILDE
181	7B	£	POUND STERLING
182	5B	¥	YEN

OCR-B (10-pitch) Characters

Segment Number	EBCDIC Assignment	Graphic	Description
183	5B	Ü	U UMLAUT
184	7B	Ä	A UMLAUT
185	7C	Ö	O UMLAUT
186	5B	Å	A SCANDINAVIAN
187	7B	Æ	DIGRAPH
188	7C	Ø	O SCANDINAVIAN
189	5B	$	DOLLAR SIGN
190	5A	$	DOLLAR SIGN
191	7B	Ñ	N TILDE
192	7B	£	POUND STERLING
193	5B	¥	YEN

Figure 41 (Part 3 of 4). The World Trade National Use Graphics

Gothic-15 Condensed Characters	Gothic-15 Condensed Under-scored Characters			
Segment Number	Segment Number	EBCDIC Assignment	Graphic	Description
365	405	5B	Ü	U UMLAUT
366	406	7B	Ä	A UMLAUT
367	407	7C	Ö	O UMLAUT
368	408	5B	Ç	C CEDILLA
369	409	E0	Ç	C CEDILLA
370	410	7B	Æ	DIGRAPH
371	411	7C	Ø	O SCANDINAVIAN
372	412	5B	₧	PESETA
373	413	7B	Ñ	N TILDE
374	414	7B	£	POUND STERLING
375	415	7B	Õ	O TILDE
376	416	7C	Õ	O TILDE
377	417	7B	Ã	A TILDE
378	418	7C	Ã	A TILDE
379	419	5B	Å	A SCANDINAVIAN
380	420	5B	¥	YEN
381	421	E0	É	E ACUTE
382	422	4A	É	E ACUTE
383	423	A1	ß	ESS TSET
384	424	4A	§	SECTION SIGN
385	425	E0	§	SECTION SIGN
386	426	5A	§	SECTION SIGN
387	427	7C	§	SECTION SIGN
388	428	4A	[	OPEN SQUARE BRACKET
389	429	5A	]	CLOSE SQUARE BRACKET
390	430	C0	{	OPENING BRACE
391	431	D0	}	CLOSING BRACE
392	432	E0	\	REVERSE SLANT
393	433	4A	•	CIRCLE ABOVE
394	434	5A	¤	CURRENCY
395	435	7B	#	NUMBER SIGN
396	436	4A	#	NUMBER SIGN
397	437	5B	$	DOLLAR SIGN
398	438	5A	$	DOLLAR SIGN
399	439	7C	@	AT SIGN
400	440	4A	¢	CENT SIGN
401	441	5A	!	EXCLAMATION POINT
402	442	5F	¬	LOGICAL NOT
403	443	A1	‾	OVERSCORE
404	444	6A	¦	VERTICAL BROKEN LINE

Figure 41 (Part 4 of 4). The World Trade National Use Graphics

APPENDIX D: PAPER SIZES, WEIGHTS, AND MAXIMUM CHARACTERS PER LINE

Figure 42 lists the common-use paper sizes and basis weights for the 3800. The widths, lengths, and basis weights shown can be used in any combination. Figure 43 lists the ISO paper sizes and basis weights; these widths, lengths, and weights are also usable in any combination in a 3800 equipped to use ISO paper sizes. A 3800 can use either common-use or ISO paper sizes, not both, and those using ISO paper sizes are not available in the United States or Canada.

Widths in inches (overall)	Lengths in inches	Basis weights
6-1/2	3-1/2	15 to 24 pound (14.25 pound minimum to 25.25 pound maximum). This is equivalent to 56 to 90 grams per square metre (53.3 minimum to 94.4 maximum grams per square metre).
8-1/2	5-1/2	
9-1/2	7	
9-7/8	8-1/2	
10-5/8	11	
11		
12		
13-5/8		
14-3/10		
14-7/8		

Figure 42. Common-Use Paper Sizes Usable in the 3800

Widths in millimetres (overall)	Lengths in inches	Basis weights
165	3	15 to 24 pound (14.25 pound minimum to 25.25 pound maximum). This is equivalent to 56 to 90 grams per square metre (53.3 minimum to 94.4 maximum grams per square metre).
180	4	
215	6	
235	8	
250	10	
270	12	
280		
305		
322		
340		
363		
375		
378		

Figure 43. ISO Paper Sizes Usable in the 3800 (Not Available in the United States and Canada)

Figures 44 and 45, for common-use paper sizes and ISO paper sizes, respectively, show the maximum number of printable characters per horizontal line for each form width and character pitch used.

Paper width in inches	Maximum number of printable characters		
	10-pitch	12-pitch	15-pitch
6-1/2	55	66	82
8-1/2	75	90	112
9-1/2	85	102	127
9-7/8	89	106	133
10-5/8	96	115	144
11	100	120	150
12	110	132	165
13-5/8	126	151	189
14-3/10	133	159	199
14-7/8	136	163	204

Figure 44. Maximum Characters per Line on Common-Use Paper Sizes

Paper width in millimetres	Maximum number of printable characters		
	10-pitch	12-pitch	15-pitch
165	55	66	82
180	61	73	91
215	74	89	111
235	82	99	123
250	88	106	132
270	96	115	144
280	100	120	150
305	110	132	165
322	116	139	174
340	123	148	185
363	133	159	199
375	136	163	204
378	136	163	204

Figure 45. Maximum Characters per Line on ISO Paper Sizes

APPENDIX E: CHANNEL COMMANDS

The following are the channel commands for the IBM 3800, and the hexadecimal codes for those commands:

CHANNEL COMMANDS

Type Command	Command Name	Hex Code
Load	Load Forms Control Buffer	63
	Load Translate Table	83
	Load Character Module (WCGM)	53
	Load Copy Number	23
	Load Forms Overlay Sequence Control	43
	Load Graphic Character Modification	25
	Load Copy Modification	35
Write	Write with No Space	01
	Write and Space 1 Line	09
	Write and Space 2 Lines	11
	Write and Space 3 Lines	19
	Write and Skip to Channel 1	89
	Write and Skip to Channel 2	91
	Write and Skip to Channel 3	99
	Write and Skip to Channel 4	A1
	Write and Skip to Channel 5	A9
	Write and Skip to Channel 6	B1
	Write and Skip to Channel 7	B9
	Write and Skip to Channel 8	C1
	Write and Skip to Channel 9	C9
	Write and Skip to Channel 10	D1
	Write and Skip to Channel 11	D9
	Write and Skip to Channel 12	E1
Forms	Space 1 Line Immediately	0B
	Space 2 Lines Immediately	13
	Space 3 Lines Immediately	1B
	Skip to Channel 1 Immediately	8B
	Skip to Channel 2 Immediately	93
	Skip to Channel 3 Immediately	9B
	Skip to Channel 4 Immediately	A3
	Skip to Channel 5 Immediately	AB
	Skip to Channel 6 Immediately	B3
	Skip to Channel 7 Immediately	BB
	Skip to Channel 8 Immediately	C3
	Skip to Channel 9 Immediately	CB
	Skip to Channel 10 Immediately	D3
	Skip to Channel 11 Immediately	DB
	Skip to Channel 12 Immediately	E3
Status	Test I/O	00
	Sense I/O	04
	Sense Type Model	E4
	Sense Intermediate Buffer	14
	Sense Error Log	24
Control	No Operation	03
	Block Data Check	73
	Allow Data Check	7B
	Initialize Printer	37
	Select Translate Table 0	47
	Select Translate Table 1	57
	Select Translate Table 2	67
	Select Translate Table 3	77
	Clear Printer	87
	End of Transmission	07
	Mark Form (End of Job)	17

APPENDIX F: SENSE BYTES

The following are the first 3 of the 24 sense bytes for the IBM 3800 Printing Subsystem. For a full description of the sense bytes, see the *Reference Manual for the IBM 3800 Printing Subsystem.*

SENSE BYTES 0 - 2 (Condition defined by byte 0 is further defined by bit(s) turned on in bytes 1 and 2.)

			Byte 0							
Hex ▶			80	40	20	10	08	04	02	01
Bit ▶			0	1	2	3	4	5	6	7
	Bit ▼	Hex ▼	Command Reject	Intervention Required	Bus Out Parity	Equipment Check	Data Check	(Reserved) 0	Load Check	Channel 9
Byte 1	0	80	Invalid Command	Not Ready	Command Code	Hardware Error	Unprintable Character		Incorrect Length	(Reserved)
	1	40	(Reserved)	Operation Check	Data Byte	Permanent Error	(Reserved)		Incorrect Multiple of 6, 8, or 12 Lines	
	2	20		Toner Collector Full	(Reserved)	Internal Log Full	No Translate Table		FCB $^1/_2$ Inch Error	
	3	10		Toner Supply Empty		Cancel Key	No FCB Channel Code Match		Invalid FCB Channel Codes	
	4	08		Developer Replacement Required		(Reserved)	Multiple Characters		FCB Length Check	
	5	04		End of Forms			(Reserved)		WCGM Not Loaded	
	6	02		Output Full					Unassigned Graphic Character	
	7	01		(Reserved)					(Reserved)	
Byte 2	0	80	(Reserved)	Forms Overlay Check	(Reserved)	(Reserved)	(Reserved)		Invalid WCGM ID	(Reserved)
	1	40		Transfer Check					No ID for WCGM 00	
	2	20		Fuser Check					Invalid Copy Modification	
	3	10		CFS Check					Invalid Forms Overlay Sequence	
	4	08		Process Check					Invalid Graphic Modification	
	5	04		Burster Trimmer Stacker Check					WCGM Data Parity Error	
	6	02		(Reserved)					(Reserved)	
	7	01		Line Overrun					(Reserved)	

Appendix G: IEBIMAGE Program Messages

Notes:

- This appendix can be taken out of this book and inserted in your copy of *OS/VS Message Library: Utilities Messages,* if desired.
- The "library data set" mentioned in some of these messages can be on SYS1.IMAGELIB or, temporarily, on another library for later transfer to SYS1.IMAGELIB.

IEBA01I IEBIMAGE UTILITY COMPLETE. RETURN CODE = *return code*

Explanation: The utility has completed execution. The *return code* indicates the greatest severity of any of the requested operations:

00 = Successful completion, operation(s) performed as requested.

04 = Operation(s) performed, investigate messages for exceptional circumstances.

08 = Operation(s) not performed. Investigate messages.

12 = Severe exception. Utility may terminate.

16 = Catastrophic exception. Utility terminated.

20 = SYSPRINT data set could not be opened. Utility terminated.

24 = User parameter list invalid. Utility terminated.

System Action: IEBIMAGE processing is terminated.

Programmer Response: Examine the *return code* to determine the degree of successful execution. If the return code is '08' or greater, the preceding messages must be investigated and resolved and the job rerun.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA02I MEMBER (*member name***) {ADDED | REPLACED} IN DIRECTORY.**

Explanation: The specified member has been placed into the library and a directory entry added/replaced in the PDS directory.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: None

Problem Determination: Table I, items 4, 13, 22, 25c, 29, 31.

IEBA03I NO SPACE IN DIRECTORY.

Explanation: A STOW macro has been issued but all directory blocks for the data set specified by the SYSUT1 DD card have been used.

System Action: The IEBIMAGE program continues processing the next control statement, but subsequent writing to the library data set is inhibited.

Programmer Response: Using utility programs such as IEBCOPY and IEHPROGM the data set may be recreated, allowing for more directory entries.

Problem Determination: Table I, items 1, 3, 4, 22, 25c, 29, 31.

IEBA04I PERMANENT I/O ERROR IN {PDS. | DIRECTORY} *unit address, device type, ddname, operation, error description, last seek address, access method*

Explanation: During writing to the specified library an I/O error occurred.

System Action: IEBIMAGE processing is terminated.

Programmer Response: Rerun the job.

Problem Determination: Table I, Items 1, 13, 22, 30, 31.

IEBA05I FCB SPECIFICATION BEYOND RANGE OF FORM.

Explanation: The keywords supplied on the FCB control statement have specified channel controls beyond the last byte of the FCB image.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the FCB control statement and rerun the job.

Problem Determination: Table I, Items 4, 13, 22, 29.

IEBA06I PARAMETERS SPECIFIED IN (LOC) KEYWORD RESULT IN X(FF), AN UNPRINTABLE CHARACTER.

Explanation: During processing of a TABLE operation, the LOC keyword contained a parameter group in the following specification (23, 3F, 03). The "3F" index into WCGM '03' is specifically reserved to represent an unprintable character. The message does not inhibit the construction of the character arrangement table, but this character will not be printed.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: If an unprintable character is desired, specify only the character arrangement table location, that is (LOC=23). To specify a printable character the WCGM or the CGMID reference must be changed and the job rerun.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA07I MODULE EXCEEDS MAXIMUM SIZE.

Explanation: Control statement specifications requested a module larger than the maximum be generated. Maximum size for COPYMOD is 8196 bytes, and for GRAPHIC is 64 segments. For FCB, the LPI keyword specifies lines beyond that specified by the SIZE keyword.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Respecify COPYMOD statements to optimize storage usage. If more than 64 segments are needed for 'GRAPHIC', up to four 'GRAFMODS' can be created and their names included into this appropriate character arrangement table. For FCB, SIZE keyword determines maximum number of lines. After respecification of control cards rerun the job.

Problem Determination: Table I, items 4, 13, 22, 26c, 29, 31.

IEBA08I NO MEMBER NAME SPECIFIED.

Explanation: No NAME control statement or no name was specified on the NAME or INCLUDE statement.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Supply the missing 1- to 4-character name and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA09I DUPLICATE MEMBER NAME FOR (*membername*), MEMBER NOT ADDED.

Explanation: A request was made to add a member to the specified library whose directory currently contains the name of a member equal to that of the name of the member to be added.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Change the name of the member to be added or if the member currently on the library is no longer needed, specify the replace option (R) on the NAME control statement.

Problem Determination: Table I, items 4, 13, 22, 25c, 29, 31.

IEBA10I MEMBER (*membername*) NOT FOUND.

Explanation: While processing the utility control statements, the name specified could not be found on the library directory. If this message precedes message IEBA42I, control card specifications are erroneous; otherwise it is a warning only.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Ensure that the name specified is correct and resubmit the job if necessary.

Problem Determination: Table I, items 4, 13, 22, 25c, 29, 31.

IEBA11I FCB NOT MULTIPLE OF 1/2 INCH.

Explanation: The LPI keyword has specified a mixture of 6 and 8 lines per inch and the total lines specified for each is not a multiple of one half inch. The number of lines specified at 8 LPI must be divisible by 4, and the number specified at 6 LPI must be divisible by 3. Any lines not specified default to 6 LPI up to the length of the form.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the LPI keyword specifications and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA12I CONTINUATION INVALID.

Explanation: A control statement was found that is not in proper format for a continuation statement. Columns 1 through 15 must be blank and the continuation information must start in column 16.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Change either the control statement that indicates continuation (it is non-blank in column 72), or ensure that the subsequent control statement is specified properly.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA13I PREVIOUS INCLUDE STATEMENT INVALID FOR THIS OPERATION, IT IS IGNORED.

Explanation: If an INCLUDE control statement precedes a FCB control statement or a print-only request, it is invalid and is ignored. If multiple INCLUDE statements are present, only the last INCLUDE statement is used.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: Remove unnecessary INCLUDE statements and rerun job, if necessary.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA14I PARAMETER(S) INVALID FOR KEYWORD (*keyword*).

Explanation: Parameters for the indicated keyword have been incorrectly specified. There are too many characters, or too many parameters, or incorrect parameter specification.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the parameter specifications and resubmit the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA15I KEYWORD (*keyword*) INVALID FOR THIS OPERATION.

Explanation: The indicated keyword is not valid for the current operation.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and resubmit the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA16I OPERATION INVALID - *operation*.

Explanation: The indicated operation is either incorrectly specified or is not in the proper sequence.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the indicated control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA17I INVALID CONTROL CARD FORMAT.

Explanation: Either the control statement has an incorrect label or no data is on the statement or a blank card.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and resubmit the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA18I INVALID SEQUENCE FOR INITIAL COPY NUMBER, IN SEGMENT *segment number*.

Explanation: In an existing module that was either built or modified by other than the IEBIMAGE utility, the indicated segment has an initial copy member that is lower than the previous segment.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Using the INCLUDE with the DELSEG keyword, delete the appropriate segments.

Problem Determination: Table I, items 4, 13, 22, 26c, 29.

IEBA19I DATA SET NOT FOUND, DSNAME = *dsname*

Explanation: The indicated data set could not be found during the issuance of an OPEN macro.

System Action: IEBIMAGE processing is terminated.

Programmer Response: Ensure that the DD statement is correctly specified. Rerun the job.

Problem Determination: Table I, items 4, 13, 22, 25a, 29, 31.

IEBA20I FAILURE DURING OPEN FOR DD = *ddname*

Explanation: The indicated *ddname* statement was incorrectly specified or not included in the JCL for this job step.

System Action: IEBIMAGE processing is terminated.

Programmer Response: Correct the DD statement error and rerun the job.

Problem Determination: Table I, items 1, 4, 13, 22, 29.

IEBA21I: INSUFFICIENT SPACE IN DATA SET - *dsname*

Explanation: The indicated data set is full. A D37 ABEND has occurred which is ignored by the utility.

System Action: The IEBIMAGE program continues processing the next control statement, but subsequent writing to the library data set is inhibited.

Programmer Response: Using utility programs such as IEBCOPY and IEHPROGM, the data set may be recreated, allowing for more space.

Problem Determination: Table I, items 1, 3, 4, 13, 22, 25a, 29, 31.

IEBA22I INVALID CHARACTER IN CARD COLUMN *number*

Explanation: An unexpected character was encountered in the indicated column.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and resubmit the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA23I DUPLICATE OR MUTUALLY EXCLUSIVE KEYWORD (*keyword*) FOR OPERATION.

Explanation: The indicated keyword has been specified more than once on this control statement, or conflicts with other keywords or with the operation.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA24I SEGMENT NUMBER (*number*) NONEXISTANT.

Explanation: The indicated segment number is greater than the number of segments in the module referenced.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the segment number specification in the DELSEG or the REF keyword and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29, 31, execute the IEBIMAGE utility to print the applicable member.

IEBA25I INSUFFICIENT KEYWORD INFORMATION TO COMPLETE OPERATION.

Explanation: Necessary keywords have not been specified, preventing the requested operation from completing successfully.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Specify the necessary keywords and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA26I INVALID USE OF DELIMITER IN KEYWORD (*keyword*).

Explanation: A comma or a parenthesis has been encountered on a control statement which does not have the proper relationship to other delimiters within the indicated keyword.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA27I NO OPERATION DEFINED.

Explanation: A NAME control statement has been encountered that does not immediately follow an operation control statement (that is, FCB, COPYMOD, TABLE, or GRAPHIC).

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement sequence and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA28I INVALID DATA SEQUENCE NUMBER.

Explanation: The sequence number on a data card (column 29 and 30) for a GRAPHIC operation is invalid.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the sequence number and rerun the job.

Problem Determinrtion: Table I, items 4, 13, 22, 29, 31.

IEBA29I MEMBER NAME OR OPTION INVALID.

Explanation: The name specified exceeds four characters or the replace option (R) was incorrectly specified.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the error and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA30I GRAPHIC DATA EXCEEDS RANGE OF PITCH IN SEGMENT *segment number*.

Explanation: Data specified on the GRAPHIC data cards has requested that a character be built which has bits outside the allowable limits of the pitch specified in the ASSIGN keyword. In the formatted print of that character the erroneous bit(s) will be printed as a dollar sign ($).

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the error and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA31I NO VALID INPUT DATA FOR GRAPHIC ASSIGN.

Explanation: The GRAPHIC operation with an ASSIGN keyword is not followed by a data card which has the data 'SEQ=' in column 25 through 28.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Supply appropriate data cards and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA32I SIZE OF FORM IS *number* INCHES.

Explanation: The FCB form size is printed as a number in inches and tenths of inches.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: None

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA33I SEGMENT *segno* COPY *copyno* LINE *lineno* MAY CREATE A LINE OVERRUN CONDITION IF PREVIOUS LINE IS PRINTED AT {6 | 8 | 12} LINES PER INCH.

Explanation: **The indicated segment, in combination with all of the previous segments, could cause a line overrun condition to occur at the printer when the specified copy modification module is used. If only 12 lines per inch is specified in the message, the segment will work with 6 or 8 lines per inch. If 8 lines per inch is specified, the segment will only work at 6 lines per inch. The copy of the referenced segment that is in the formatted listing is flagged with the phrase NOTE *(n)*, where *n* has the meanings:**

1 **There has been at least one 12 LPI overrun indication for this segment.**

2 **There has been at least one 8 LPI overrun indication for this segment.**

3 **There has been at least one 6 LPI overrun indication for this segment.**

System Action: **The IEBIMAGE program continues processing the next control statement.**

Programmer Response: Respecify COPYMOD statements such that each segment in combination with preceding segments qualifies for use. See the ***Reference Manual for the IBM 3800 Printing Subsystem*** for the algorithm for the limiting condition. The OVERRUN parameter in the OPTION statement can be coded to suppress unwanted IEBA33I messages.

Problem Determination: Table I, items 4, 13, 22, 25c, 29, 31.

IEBA34I CHARACTERS SPECIFIED IN TEXT KEYWORD RESULT IN HEXADECIMAL (FF).

Explanation: A specification in the TEXT keyword which results in a hexadecimal (FF) is invalid. 'FF' has been reserved for use within copy modification text to denote a duplication factor.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the error and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA35I UNBALANCED PARENTHESIS IN KEYWORD (*keyword*)

Explanation: The number of right parentheses does not equal the number of left parentheses in the indicated keyword.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA36I DUPLICATE CHANNEL SPECIFICATION FOR PRINT LINE ***print line number.***

Explanation: More than one channel specification has been requested for the same print line (for example, CH1=10, CH2=10).

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA37I REFERENCES TO NONEXISTANT CGMID WITHIN TABLE.

Explanation: One or more locations in the character arrangement table reference a CGMID that has not been defined in the trailer portion of the table.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA38I TABLE REFERENCES NO PRINTABLE CHARACTERS.

Explanation: A character arrangement table has been constructed or modified such that the resulting table has all 256 locations referencing the non-printable character (FF).

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29,31.

IEBA39I INVALID PARITY IN SEGMENT ***segment number.***

Explanation: In a graphic character modification module that was either created or modified by other than the IEBIMAGE utility, the indicated segment was found to have invalid parity.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Use AMASPZAP to correct the error, or rebuild the module using an INCLUDE control statement with the DELSEG keyword, and a GRAPHIC control statement with the ASSIGN keyword and appropriate data cards.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA40I INCONSISTENT PITCH WITHIN SEGMENT ***segment number.***

Explanation: In a graphic character modification module that was either created or modified by other than the IEBIMAGE utility, the indicated segment was found to have inconsistent pitch.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Use AMASPZAP to correct the error, or rebuild the modules using an INCLUDE control statement with the DELSEG keyword, and a GRAPHIC control statement with the ASSIGN keyword and the appropriate data cards.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA41I PREVIOUSLY USED EBCDIC ASSIGNMENT IN SEGMENT ***segment number.***

Explanation: A segment was found to have an assignment value equal to that of one previously used in the module. This error does not prohibit completion of the operation. Should this module be used in the printer, only the last of the duplicated assignments will be used.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: Use an INCLUDE control statement with the DELSEG keyword and a GRAPHIC control statement with the REF keyword to change the assignment value.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

IEBA42I MEMBER NOT ADDED TO LIBRARY, ERROR(S) LISTED.

Explanation: Errors previous to this statement have prevented the utility from updating the library with the module specified in the current operation.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the errors and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 25c, 26c, 29, 31.

IEBA43I IN SEGMENT *segment number*, {COPY | LINE | TEXT} SPECIFICATION BEYOND RANGE OF MAXIMUM VALUE.

Explanation: One of the following values has been exceeded:

COPY - 255

LINE - 132*

TEXT - 204

*The IEBIMAGE utility allows the line specification to be 132, which is valid for a 3800 using ISO paper sizes. If this value is greater than 120 for a 3800 that uses common-use paper sizes, the entry will never be used.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the error and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA44I SEQUENCE NUMBER INVALID OR PREVIOUSLY USED.

Explanation: A data card sequence number of less than 1 or greater than 24, or one that has been used previously, has been encountered.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the data card and resubmit the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA45I LIBRARY RESIDES ON MULTIPLE EXTENTS.

Explanation: After updating the library specified by the SYSUT1 DD statement, it was found that the last record was written on other than the first extent. The SYS1.IMAGELIB system data set must have only one extent.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: Using utility programs such as IEBCOPY and IEHPROGM, the data set may be recreated and copied, allowing for only one extent.

Problem Determination: Table I, items 1, 4, 13, 25c, 26c, 29, 31.

IEBA46I REQUEST TO UPDATE LIBRARY DENIED, DSNAME = *data set name*

Explanation: The OPEN macro has caused a message to be sent to the operator requesting an update to a library whose expiration has not been exceeded. The operator replied M, denying the request.

System Action: IEBIMAGE processing continues with the next control statement but updating of the library data set is inhibited.

Programmer Response: Specify a different volume and/or obtain authorization to update the subject data set.

Problem Determination: Table I, items 4, 13, 22, 25c, 29.

IEBA47I INSUFFICIENT VIRTUAL STORAGE TO PERFORM STOW FUNCTION.

Explanation: A STOW SVC was issued and its function was prohibited due to lack of virtual storage.

System Action: The IEBIMAGE program continues processing the next control statement, but subsequent writing to the library data set is inhibited.

Programmer Response: Rerun the job. The error may be a temporary one caused by fragmentation of virtual storage.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA48I DUPLICATE TABLE REFERENCE FOR LOCATION (*location*).

Explanation: Specification for the same location in a character arrangement table occurred more than once. Each reference must be unique.

System Action: The IEBIMAGE program continues processing the next control statement, but writing to the library data set for the current operation is inhibited.

Programmer Response: Correct the control statement and rerun the job.

Problem Determination: Table I, items 4, 13, 22, 29.

IEBA49I NO CHANNEL 1 CODE HAS BEEN SPECIFIED.

Explanation: It has been determined that the FCB contains no channel 1 code. Be careful when using this FCB, because a skip command issued to a 3800 that is using an FCB that does not contain the required channel code causes a unit check at the printer.

System Action: The IEBIMAGE program continues processing the next control statement.

Programmer Response: None.

Problem Determination: Table I, items 4, 13, 22, 29, 31.

Problem Determination

Problem determination is the activity required to identify a failing hardware unit or program and determine who is responsible for maintenance.

Problem determination is accomplished by using the procedures specified by IBM. In some cases, these procedures may be initiated by a message or code which requires operator or programmer response. The response may include the requirement for additional problem-related data to be collected and will attempt, where possible, to indicate "probable" failure responsibility.

Problem determination information is included for applicable messages under the heading "Problem Determination." Standard problem determination actions are identified as items of Table I. Unique actions are identified following the list of standard actions to be taken. In any case, it is intended that the specified actions be taken before calling IBM for support.

Table I

Note: If the problem recurs, follow the problem determination aids specified by the associated message or code before calling IBM for support.

1. Make sure that MSGLEVEL=(1,1) was specified in the JOB statement.

2. Not used.

3. Save the job stream associated with the job.

4. Save the system output (SYSOUT) associated with the job.

5-12. Not used

13. Save all the associated output.

14-21. Not used.

22. Save the source input associated with the job.

23-24. Not used.

25. Execute the IEHLIST system utility program to obtain a list of the:

    a. Volume table of contents of the associated volume, specifying the FORMAT option

    c. Directory of the associated data set

26. Execute the IEBPTPCH data set utility to:

    c. Print the applicable member

27-28. Not used.

29. Contact IBM for programming support.

30. Contact IBM for hardware support.

31. Use IEBCOPY to unload SYS1.IMAGELIB to tape.

GLOSSARY

The following terms are defined as they are used in this book. If you do not find the term you are looking for, refer to the index or to the IBM *Data Processing Glossary,* GC20-1699.

basis weight: The weight in pounds of a ream (500 sheets) of paper cut to a given standard size for that grade: 25x38 inches for book papers, 17x22 inches for bond, and other sizes for other grades. The basis weight of continuous forms for computer output is based on the size for bond papers.

CGMID: Character generation module identifier. The same as character set identifier.

chain printer: * A printer in which the type slugs are carried by the links of a revolving chain.

channel command: An instruction that directs a channel, control unit, or device to perform an operation or set of operations.

character: * A letter, digit, or other symbol that is used as part of the organization, control, or representation of data. A character is often in the form of a spatial arrangement of adjacent or connected strokes.

character arrangement: An arrangement composed of graphic characters from one or more modified or unmodified character sets.

character arrangement table: A table that contains identifiers for one to four character sets, identifiers for zero to four graphic character modification modules, and a 256-byte translate table used to locate the scan pattern that corresponds to the data code of a character to be printed.

character cell: The rectangular area that can be occupied by a character on the printed page. The size of the rectangular area varies with the pitch of the character and the number of lines per inch at which it is printed.

character set: As used in this book, the scan patterns for an IBM-supplied set of a maximum of 64 graphic characters, all of one size and style.

character set identifier: A 1-byte code identifying the particular character set on the flexible disk within the 3800. Abbreviated CGMID.

condensed: Character sets having a face that is smaller than that of a set not so characterized.

control character: * A character whose occurrence in a particular context initiates, modifies, or stops a control operation. A control character may be recorded for use in a subsequent action. A control character is not a graphic character, but may have a graphic representation in some circumstances.

copy modification: A feature that allows printing of predefined data on all pages of specified copies of a data set.

copy modification module: A program module in SYS1.IMAGELIB that can be used for copy modification by specifying the name of the module with the MODIFY keyword.

data code, 8-bit: A byte that represents a character. This byte-character pair can be a member of a coded character set (usually EBCDIC) or a user-assigned code.

EBCDIC: Extended binary-coded decimal interchange code. A coded character set consisting of 8-bit coded characters.

EXCP: Execute channel program.

FCB: Forms control buffer.

folding: A technique used with the Universal Character Set (UCS) feature on an impact printer to allow each of the 256 possible character codes to print some character on a chain or train with fewer graphics. The first 2 bits of each 8-bit code are ignored. For example, it allows the printing of uppercase graphic characters when lowercase are not available in the character array on the chain or train.

Format: Character sets that provide graphics such as lines, corners, and intersections in different line weights, which can be used, for example, to print lined columns or boxes around data.

forms control buffer (FCB): A buffer for controlling the vertical format of printed output; analogous to the punched paper carriage control tape used on IBM 1403 Printers.

forms control buffer module: A program module in SYS1.IMAGELIB that is loaded into the forms control buffer when specified with the FCB keyword.

forms overlay: A feature providing the ability to print a form, grid, design, or other constant data from a negative as the variable data is being printed.

fuser: The unit that fuses the toned image into the paper.

Gothic: Character sets (available in 10-, 12-, and 15-pitch) with 63 sans serif graphic characters.

graphic: * A symbol produced by a process such as handwriting, drawing, or printing.

graphic character: * A character, other than a control character, that is normally represented by a graphic.

graphic character modification: A feature that allows the substitution or extension of the graphic characters in an already-defined character arrangement.

graphic character modification module: A program module in SYS1.IMAGELIB that can be used for graphic character modification when it is named in a character arrangement table.

HASP: Houston Automatic Spooling Program. A computer program that provides supplementary job management, data management, and task management functions such as control of job flow, ordering of tasks, and spooling.

impact printer: A printer that mechanically strikes the paper through an inked ribbon.

ISO: International Standards Organization. Used in this book to identify a set of paper sizes (available on the 3800 outside the United States and Canada only) selected from those that have been standardized by that organization for use in data processing.

*American National Standard definition

JFCBE: JFCB Extention. Job File Control Block Extention, carries setup information for the 3800.

Katakana: Character sets of symbols for writing in one of the two common Japanese phonetic alphabets.

line overrun: An indication that copy modification was not completed in time for printing the line.

OCR: * Optical character recognition.

OCR-A, OCR-B: Character sets that are designed for optical character recognition use.

optical character recognition (OCR): * The machine identification of printed characters through use of light-sensitive devices.

output writer: A part of the job scheduler that transcribes specified output data sets onto a system output device independently of the program that produced the data sets.

pitch: A unit of width of type based on the number of times a letter can be set in a linear inch. For example, 10-pitch has 10 characters per inch.

print chain/train: The revolving chain or train in which the type slugs of an impact printer are carried.

print control character: * A control character for print operations such as line spacing, page ejection, or carriage return.

table reference character (trc): A numeric character (0, 1, 2, or 3) corresponding to the order in which the character arrangement table names have been specified with the CHARS keyword. It is used for selection of a character arrangement table during printing.

Text: Character sets of upper and lowercase graphic characters with serifs.

toner: A thermoplastic material impregnated with lampblack. Toner adheres to the exposed areas on the photoconductor during the developing process and is then transferred to the paper to form the developed image on the paper.

transfer station: The location where the toned image on the photoconductor is transferred to the paper.

translate table: That 256-byte portion of the character arrangement table that translates the user's data code for a character to the code required by the 3800's control electronics.

trc: Table reference character.

UCS: Universal Character Set.

Universal Character Set (UCS): A feature on some IBM impact printers that permits the use of a variety of character arrays.

WCGM: Writable character generation module.

writable character generation module (WCGM): A 64-position portion of the 3800's character generation storage that holds the scan elements of one character set. There are two WCGMs in the basic 3800, and optional added storage provides two more.

*American National Standard definition

INDEX

GC26-3859-1

IBM 3800 Printing Subsystem Programmer's Guide for OS/VS2 SVS (File No. S370-30) Printed in U.S.A. GC26-3859-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

IBM 3800 Printing Subsystem
Programmer's Guide for OS/VS2 SVS
GC26-3859-1

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ____________________

Previous TNL ________________

Previous TNL ________________

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

Reader's Comment Form

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and Tape

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

IBM 3800 Printing Subsystem
Programmer's Guide for OS/VS2 SVS
GC26-3859-1

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL ________________

Previous TNL ________________

Previous TNL ________________

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

GC26-3859-1

Reader's Comment Form

Fold and Tape

Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

IBM 3800 Printing Subsystem Programmer's Guide for OS/VS2 SVS (File No. S370-30) Printed in U.S.A. GC26-3859-1