GC26-3846-3
File No. S370-30

Systems

# IBM 3800 Printing Subsystem Programmer's Guide

OS/VS1 Release 7

OS/VS2 MVS Release 3.8

This publication was produced using the
IBM Document Composition Facility
(program number 5748-XX9)
and the master was printed on the
IBM 3800 Printing Subsystem.

**Fourth Edition (January 1980)**

This edition, as amended by technical newsletter GN26-0964, applies to Release 7 of IBM OS/VS1, and to Release 3.8 of IBM OS/VS2 MVS, and to any subsequent releases of either system until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "OS/VS1 Summary of Amendments" or "OS/VS2 MVS Summary of Amendments" following the list of figures. Information about the IBM 3800 Printing Subsystem Enhancements is for planning purposes only, until availability of the product. Since this publication has been completely reorganized, technical changes are not marked by vertical lines.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, General Products Division, Department 61C, Tucson, Arizona, U.S.A. 85744. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

## 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 computer installations.
- 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.

## STRUCTURE OF THIS PUBLICATION

This publication has been arranged in four major sections. The material in each section is divided under chapter headings and subheadings. The chapter headings are reflected in the footings on odd-numbered pages. This organization was chosen to increase retrievability of the information in the publication by creating many chapter headings, and therefore, many footings for the user to scan in searching for data.

"Section 1. Introduction" describes the IBM 3800 Printing Subsystem, how it works, and how to maintain compatibility between the 3800 and the IBM impact printers.

"Section 2. How to Use the 3800 Printing Subsystem" descibes how, as a user of the 3800 Printing Subsystem, you can take advantage of its various capabilities.

"Section 3. Supporting the 3800 Printing Subsystem" describes, for the system programmer, some special considerations for controlling the 3800 when allocating the printer directly or through JES.

"Section 4. The IEBIMAGE Utility Program" is a reference section. In it are the details of what IEBIMAGE does and how the control statements are to be coded.

Appendixes A through F contain detailed reference information about the 3800 character sets, library modules, paper standards, channel commands, and sense bytes.

The Glossary contains definitions of terms as they are used in this publication.

## RELATED PUBLICATIONS

The following publications contain additional information about the 3800 Printing Subsystem. Some of them are referenced in this book:

- IBM 3800 Printing Subsystem Enhancements System Information, GC26-3935, which contains a general description of the 3800 Enhancements, and details for ordering the Enhancement packages and related publications.

- 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 Print Line and Character Gauge, GX26-1652, which can be used to design and measure the line and character spacing for 3800 output.

- IBM 3800 Character Set Catalog, GA26-1656, which contains the scan patterns for the IBM-supplied character sets.

- 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.

- Operator's Library: OS/VS1 Reference, GC38-0110, which describes the operator's SETPRT command for OS/VS1.

- OS/VS1 Data Management Macro Instructions, GC26-3872, and OS/VS2 MVS Data Management Macro Instructions, GC26-3873, which describe the SETPRT macro instruction.

- OS/VS1 Data Management Services Guide, GC26-3874, and OS/VS2 MVS Data Management Services Guide, GC26-3875, which contain information on user exits.

- OS/VS1 Debugging Guide, GC24-5093, and OS/VS2 System Programming Library: Debugging Handbook, GC28-0708 (Volume 1), GC28-0709 (Volume 2), and GC28-0710 (Volume 3), which contain information on system dumps and how high-density dumps can be obtained with the 3800.

- OS/VS1 JCL Reference, GC24-5099, OS/VS1 JCL Services, GC24-5100, and OS/VS2 MVS JCL, GC28-0692, which describe the job control language (JCL) for using the 3800.

- OS/VS Message Library: VS1 System Messages, GC38-1001, and OS/VS Message Library: VS2 System Messages, GC38-1002, which include system messages issued for the 3800.

- OS/VS Message Library: VS1 Utilities Messages, GC26-3919, and OS/VS Message Library: VS2 MVS Utilities Messages, GC26-3920, which contain the IEBA messages issued by the IEBIMAGE utility program.

- OS/VS1 Planning and Use Guide, GC24-5090, which describes JES1 and user-written writer procedures in OS/VS1.

- OS/VS1 Storage Estimates,GC24-5094, which contains information on how to calculate auxiliary storage requirements for the system image library.

- OS/VS1 System Data Areas, SY28-0605, which contains details of DCBIFLGS and the UCB extension for 3800 printers.

- OS/VS1 Service Aids, GC28-0665, and OS/VS2 System Programming Library: Service Aids, GC28-0674, which contain information on the PRDMP dump facility and the SPZAP utility program.

- OS/VS1 Supervisor Services and Macro Instructions, GC24-5103, and OS/VS2 Supervisor Services and Macro Instructions, GC28-0683, which describe the SNAP dump facility.

- OS/VS1 System Generation Reference, GC26-3791, and OS/VS2 System Programming Library: System Generation Reference, GC26-3792, which tell how to include a 3800 in your system.

- OS/VS2 MVS System Programming Library: JES2, GC23-0002, which contains information on JES2 and how it treats output on the 3800. It describes JES2 initialization and performance and the track cell method of despooling data to a printer.

- System Programming Library: Network Job Entry Facility for JES2, SC23-0003, which contains information on JES2 NJE and how it treats output on the 3800. It describes JES2 NJE initialization and performance and the track cell method of despooling data to a printer.

- OS/VS2 MVS System Programming Library: JES3, GC28-0608, which contains information on JES3 initialization, and on the amount of spool read-ahead that should be used with the 3800.

- OS/VS2 System Programming Library: Supervisor, GC28-0628, which contains information about the Authorized Program Facility (APF).

- OS/VS2 Message Library: JES3 Messages, GC38-1012, which includes messages issued by JES3 about the 3800.

- OS/VS1 Utilities, GC26-3901, and OS/VS2 MVS Utilities, GC26-3902, which tell how to use IEBUPDTE, IEHPROGM, and other utility programs needed when working with and maintaining the modules on SYS1.IMAGELIB.

- OS/VS Utilities Logic, SY35-0005, which includes the IEBIMAGE utility logic.

- 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 can cause a line overrun condition on the printer. It also contains information about the page IDs that are used in the 3800 Enhancements to synchronize JES with the printer processing, and details of the sense data that is returned when a Request Printer Information order is issued (3800 Enhancements only).

- OS/VS2 MVS SAM-E Installation: Storage Estimating, SH20-9144, which contains the sizes of program modules that are used for the 3800, when it is installed with SAM-Extended.

## 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).
- Medium face underscored type specifies fields to be supplied by the user.
- **Boldface underscored** type indicates a default option. If the parameter is omitted, the underscored value is assumed.

# SUMMARY OF AMENDMENTS

## OS/VS1 SUMMARY OF AMENDMENTS

**JANUARY 1980**

**Release 7**

**IBM 3800 Printing Subsystem Enhancements**

This revision describes the functions available with the IBM 3800 Printing Subsystem Enhancements, and how to use them. Also, the publication has been reorganized, and miscellaneous technical and editorial changes have been made.

**MAY 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# UX10818.

Miscellaneous technical and editorial changes have also been made.

**RELEASE 6**

The condensed character sets, the character arrangement tables that use them, and related support for printing at 12 lines per inch, are added to this edition of this book.

Miscellaneous technical and editorial changes have also been made throughout the book. "Appendix G: Additional Information on Burster-Trimmer-Stacker Support" has been deleted because the information is now included in the appropriate VS1 system library books.

## OS/VS2 MVS SUMMARY OF AMENDMENTS

**JANUARY 1980**

**Release 3.8**

**IBM 3800 Printing Subsystem Enhancements**

This revision describes the functions available with the IBM 3800 Printing Subsystem Enhancements, and how to use them. Also, the publication has been reorganized, and miscellaneous technical and editorial changes have been made.

MAY 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# UZ15813.

Miscellaneous technical and editorial changes have also been made.

## MVS IBM 3800 PRINTING SUBSYSTEM 12 LINES PER INCH SELECTABLE UNIT (VS2.03.848)

The condensed character sets, the character arrangement tables that use them, and related support for printing at 12 lines per inch, are added to this edition of this book.

Miscellaneous technical and editorial changes have also been made throughout the book. "Appendix G: Additional Information on Burster-Trimmer-Stacker Support" has been deleted because the information therein is now included in the appropriate system library books or their selectable unit newsletters.

## CONTENTS

## FIGURES

# SECTION 1. INTRODUCTION

## WHAT IS THE 3800 PRINTING SUBSYSTEM?

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.

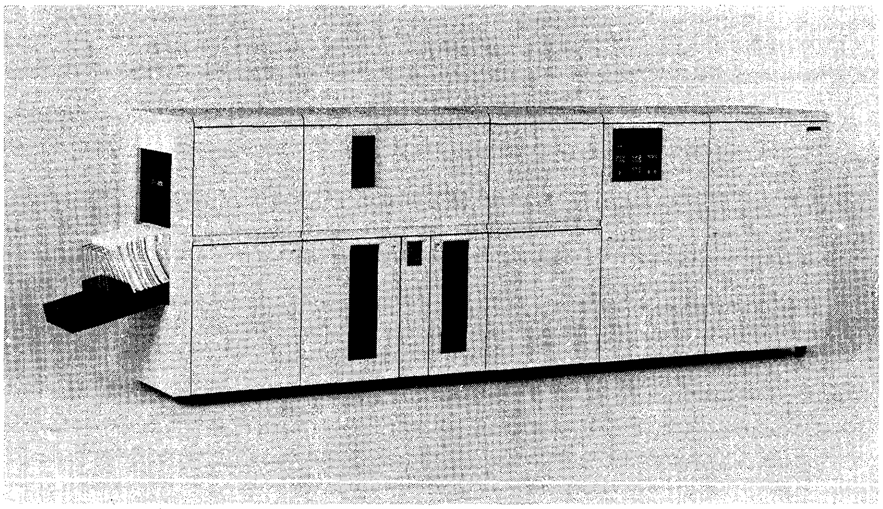


Figure 1. The IBM 3800 Printing Subsystem

It provides easy-to-use programming interfaces for its functions and features, and it offers compatibility for user programs that were designed for impact printers.

As long as there is sufficient data to print, the 3800 prints at a constant high speed. For example, it can process 1000 11-inch long pages in approximately six minutes of continuous printing, regardless of the number of lines per page. Printing speed is also independent of the number of characters being used.

The 3800 Printing Subsystem operates under the OS/VS1 and OS/VS2 MVS operating systems through a channel attachment, and can be attached 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, 3033, or 4341 Processor.

A 3800 Subsystem can be attached to a System/370 Model 135 or 138 via RPQ S00338, and to an IBM System/370 4331 Processor, using the optional block multiplexer channel (1241) and the optional power interface (5531).

The IBM 3800 Printing Subsystem can be used either as a system output device or as a direct output device. When the 3800 is used as a system output device, the job entry subsystems (JES) control the setup and use of the 3800. When necessary, JES can reassign the printing of data sets from a 3800 to an impact printer, and vice versa. If the printing of a reassigned data set requires special features unique to the 3800, the operating system ignores the job control (JCL) keywords that specify these features when the job is processed on an impact printer. The user with direct control of the 3800 Printing Subsystem must simulate some of the job entry subsystem functions.

## STANDARD FEATURES

The following standard features are available with the 3800:

- Character generation storage is provided for printing 128 graphic characters (two character sets).
- Any two of 20 different character sets (including 10-, 12-, and 15-pitch and 15-pitch condensed sets) can be selected for printing a data set, and can be intermixed on a page without operator intervention.
- Graphic character modification allows the substitution or addition of graphic characters into an already-defined character arrangement.
- 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 6, 8, or 12 lines per inch, or a mixture of these vertical line spacings on a page.
- Forms overlay permits printing of forms at the same time 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.

## OPTIONAL FEATURES

The following optional features are available:

- Character generation storage for an additional 127 characters is available. With the additional character generation storage, as many as four character sets can be selected for printing each data set.
- The optional Burster-Trimmer-Stacker bursts the fanfold output, trims 1/2 inch (12.7 mm) from the left edge and the same amount from the right edge, and stacks the sheets in groups, so individual jobs can be identified and removed.
- A two-channel switch is available as an optional feature for attaching the 3800 to two separate processors through two channels.

## ENHANCEMENTS PACKAGE

The following additional features are available when the 3800 Enhancements package is installed:

- Twenty IBM-supplied character sets (duplicates of the character sets that are available in the standard 3800) are stored in the image library in a module type called a library character set module. The user can modify these library character sets, and can build new library character sets to be stored in the image library.

- Multiple extents can be specified for the system image library to support the addition of modules available for use by the 3800.

- Detailed printer-setup error messages simplify the recovery task for both programmers and operators. Error messages for the programmer are printed at the 3800 on a separate page from the data set output.

- JES is specifically informed when a paper jam that requires system restart occurs during printer setup, or when the Cancel key is pressed during printer setup.

- In the OS/VS2 MVS system, the user can change the printer setup for a spooled data set while creating the data set.

## HOW THE 3800 PRINTING SUBSYSTEM WORKS

The OS/VS1 and OS/VS2 operating systems support the operation of the IBM 3800 Printing Subsystem through a channel attachment. You specify JCL DD statement parameters to control the functions of the 3800, and the operating system uses the printer setup routine called SETPRT to initialize the printer based on your specifications. If the 3800 is being used as a direct output device, you can use the SETPRT macro instruction to initialize the 3800. In OS/VS2, if the Enhancements are installed, you can use SETPRT, while you are creating a spooled data set, to change the setup information for the 3800.

Figure 2 and Figure 3 show the flow of processing through the JES and data management routines.

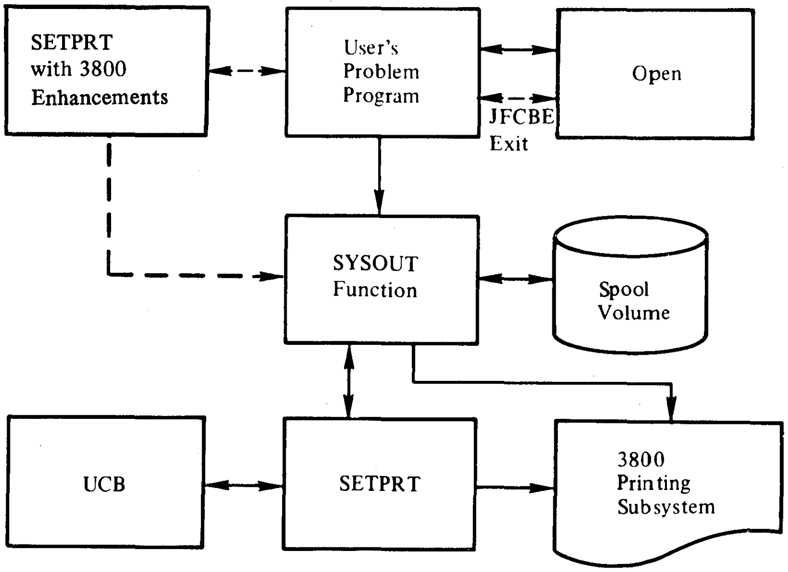


Figure 2. An Overview of SYSOUT Processing to the 3800

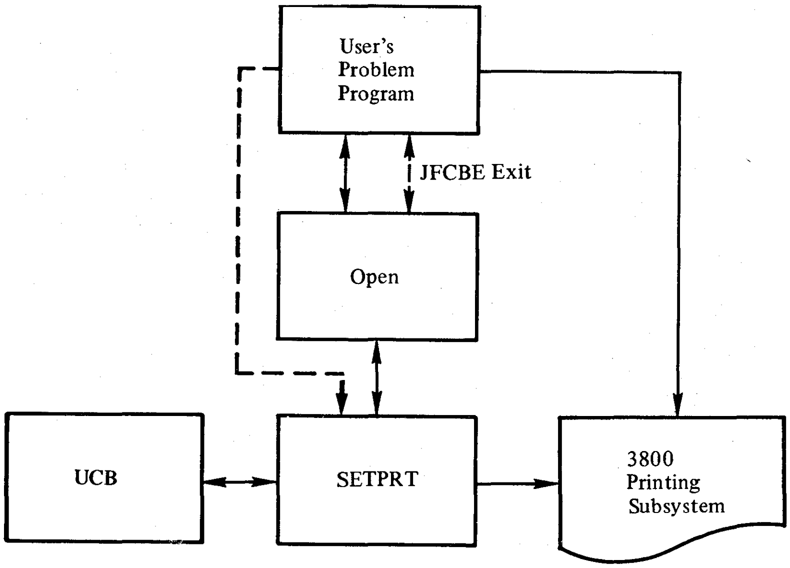


Figure 3. An Overview of Direct Processing to the 3800

JES and the Open executors use the parameters that are specified in the JCL or in the SETPRT macro to control the functions of the 3800. When parameters are not specified, defaults are used as specified by JES, the SETPRT macro, or the 3800 printer itself, in that order.

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 4 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.

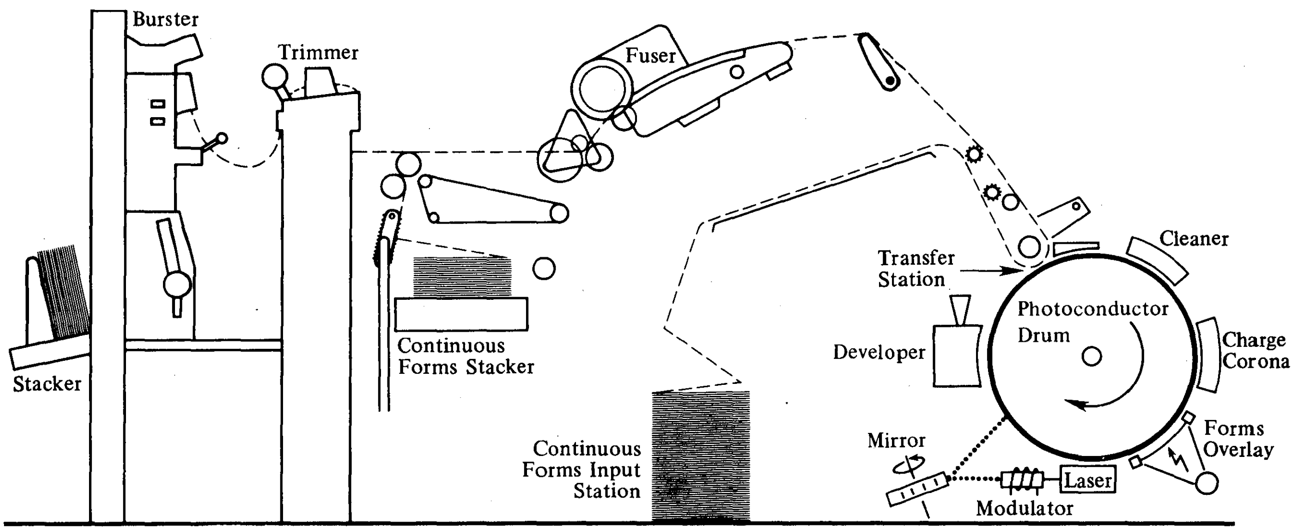


Figure 4. An Overview of How the 3800 Prints

## 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.

## CONTROLLING THE PRINTER SETUP

The 3800 printer setup functions are supported by IBM-supplied or user-defined modules that are created and stored in an image library using the IEBIMAGE utility program. (IEBIMAGE is described in "Section 4. The IEBIMAGE Utility Program.") JCL and SETPRT parameters are available for controlling the following functions:

- You can specify the character arrangements to be used to print a data set by naming character arrangement table modules. See the section entitled "Character Arrangements" in this chapter for a description of character arrangements and character arrangement tables.
- You can control the vertical format of a page by naming a format control buffer module to be used for printing the page.

- You can specify that a forms overlay is to be flashed onto a page to be printed. See the chapter entitled "How to Flash Forms" for a description of how to use the forms overlay function.

- You can specify that the data set to be printed contains codes for line-by-line selection of the character arrangements to be used. See the chapter entitled "How to Specify Character Arrangements" for details on how to do this.

- You can specify the number of copies of a data set you want printed. The copies can be grouped by full data set or by multiple copies of each page.

- You can request the bursting, trimming, and stacking of the output if your installation has the optional equipment.

- You can specify that you want predefined data to be printed on certain copies of a page by naming a copy modification module to be used. See the chapter entitled "How to Modify Copies" for details.

- You can request that certain error condition indicators be sent to the host system. See the section entitled "Using Job Control Language" in the chapter entitled "How to Specify Printer Settings."

- You can request initialization of the printer to hardware defaults.

- If you have direct control of the 3800, you can request retransmission of a data set to the 3800 to obtain additional copies of that data set.

- If the 3800 Enhancements are installed, you can specify whether or not error messages that are generated as a result of printer setup should be written to a printer for the programmer's information.

- If the 3800 Enhancements are installed, you can specify a message area for SETPRT to use to store a copy of an error message.

- If the 3800 Enhancements are installed, you can cause a spooled data set to be split into two or more units of printed output. You can also specify different printer setups for the output units.

- If the 3800 Enhancements are installed, the user with direct control of the 3800 can specify a user-defined library to replace SYS1.IMAGELIB.

## CHARACTER ARRANGEMENTS

The standard 3800 subsystem uses only the 20 IBM-supplied character sets that reside on the flexible disk in the 3800. Each character set contains a maximum of 64 graphic character representations. After the character set has been loaded into the 3800 character generation storage, you can use graphic character modification modules created with the IEBIMAGE program to modify the character set.

With the Enhancements installed, the 3800 subsystem uses both the IBM-supplied character sets that reside on the flexible disk and library character sets that are stored in the system image library (SYS1.IMAGELIB) or, for the direct u<sup></sup>er of the 3800, in a user-defined library.

The 20 library character sets that are supplied by IBM are duplicates of those character sets that reside on the flexible disk. The character sets that reside on the 3800's flexible disk have even-numbered IDs, while their comparable IBM-supplied library character sets have odd-numbered IDs.

If you want to use the 20 IBM-supplied library character sets, the installation must:

- Have loaded them into the image library using IEBIMAGE. For details on how to do this, see the program directory that accompanies the 3800 Enhancements package.
- Have modified existing character arrangement tables to reflect the odd-numbered IDs, or create new character arrangement tables to point to the library character sets.

You can customize the IBM-supplied library character sets using IEBIMAGE control statements. You can also use IEBIMAGE control statements to create entirely new library character sets to be stored in SYS1.IMAGELIB, or in a user-defined library.

To use a 3800 character set, you must point to the character set by means of a character arrangement table. Multiple character arrangement tables can point to one character set. Character arrangement tables can be created using IEBIMAGE control statements.

A character arrangement is a collection of characters that have been selected from one or more character sets or have been newly designed or redesigned. A character arrangement table contains three kinds of data:

- IDs for as many as four character sets that are to be used as sources of data codes and graphic character representations for the character arrangement being designed
- Names for as many as four graphic character modification modules to be used to introduce new graphic characters into the arrangement or to replace characters in the character sets referenced by the character arrangement table
- A 256-byte translate table (that accommodates a maximum of four 64-byte character sets) that translates 8-bit data codes into WCGM locations where the graphic representations (scan patterns) of those data codes are stored.

A WCGM is a writable character generation module. It contains 64 segments of character generation storage, which hold the scan patterns of characters being used by the 3800 to print a data set. There are two WCGMs in the standard 3800, and two more can be added as an option. The WCGMs are numbered 0-3.

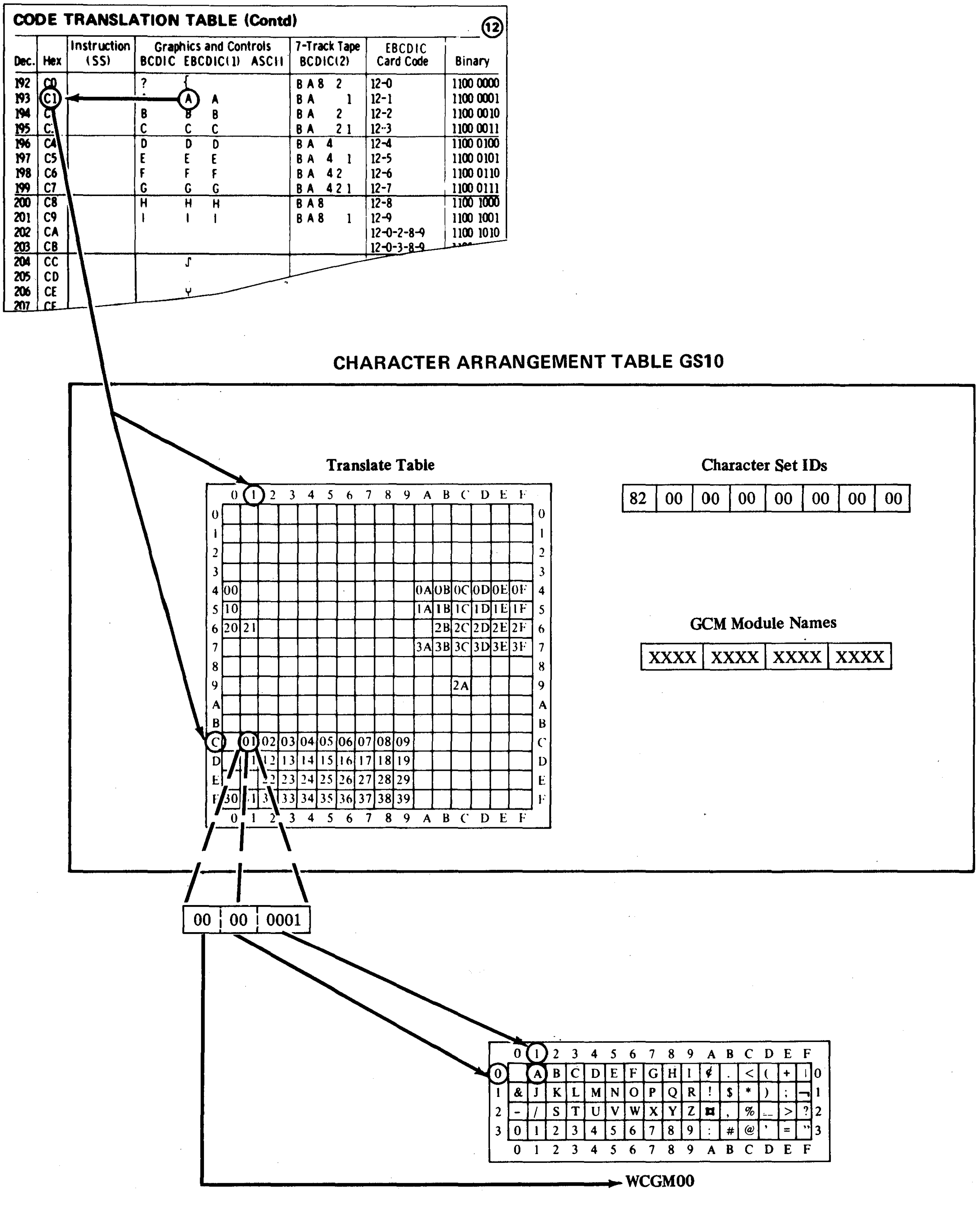
The last position of the fourth WCGM, that is, the WCGM position that is addressed by X'FF', is not available for printing because the X'FF' code is reserved for indicating an unprintable character. Therefore, a maximum of 255 scan patterns can be used for printing a data set.

Figure 5 shows how an 8-bit data code sent to the 3800 is transformed into an address in a WCGM using 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 a location in the translate table. That location contains the value '01', which is the address of the WCGM location that contains the scan pattern for printing the 'A'.

The following table shows which location of which WCGM is referenced by the contents of a translate table location.

| Translate Table Contents | WCGM # | WCGM Location |
|---|---|---|
| 00-3F | 0 | 00-3F |
| 40-7F | 1 | 00-3F |
| 80-BF | 2 | 00-3F |
| C0-EF | 3 | 00-3E |

Note that translate table contents of 'FF' indicates an invalid character, and does not address a WCGM location. Therefore, location '3F' of WCGM 3 is not addressable.



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 | 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 | | | | | | | |

CHARACTER ARRANGEMENT TABLE GS10

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 | |

Character Set IDs

| 82 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
|---|---|---|---|---|---|---|---|

GCM Module Names

| XXXX | XXXX | XXXX | XXXX |
|---|---|---|---|

| 00 | 00 | 0001 |
|---|---|---|

| | 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 5. How an 8-bit Data Code is Translated into a WCGM Location

The sequence of events that causes the loading of the WCGM in preparation for printing is described in the numbered steps below. This sequence assumes that options are specified in the JCL. The character arrangement table G11 was chosen for illustrative purposes, because it uses graphic character modification. Graphic character modification allows the substitution or addition of one or more characters into an already defined character arrangement. Steps 5 and 6 of the following sequence explain the graphic character modification process.

1. Using the CHARS parameter, you request that character arrangement table G11 be loaded.

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

2. SETPRT loads the character arrangement table module from SYS1.IMAGELIB into the processor. The character arrangement table G11 specifies that the Gothic-10 character set, which is identified by '82', is to be loaded into WCGM 0.

3. Using the character set identifier '82', SETPRT loads the Gothic-10 character set from the flexible disk in the 3800 into WCGM 0 (see Figure 6).

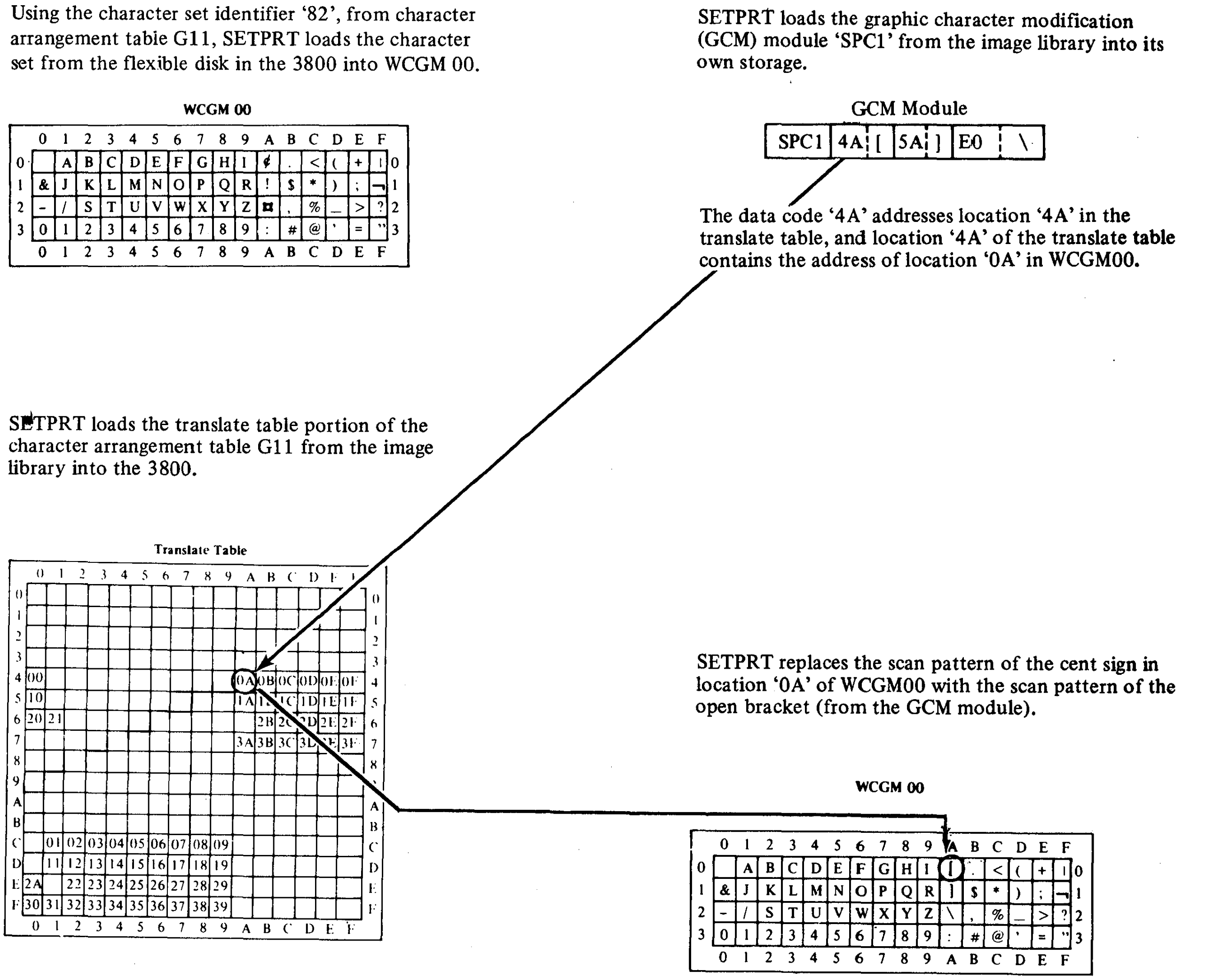
   Note that, if the Enhancements package is installed, SETPRT first checks to see if the requested character set is already in a WCGM. If so, the Gothic-10 character set is not reloaded unless the WCGM in which the set resides is WCGM 3. If it is in WCGM 3, the character set is reloaded into one of the other WCGMs, in order to accommodate the Gothic-10 reference to location '3F' of the WCGM. Figure 6 shows the Gothic-10 character set in WCGM 0.

4. The translate table portion of the character arrangement table is then loaded into the 3800. Figure 5 shows the translate table entries for the WCGM location assignments for the Gothic-10 character set data codes. In Figure 6, the translate table entries have been modified using graphic character modification.

5. The graphic character modification (GCM) modules (if any) identified by the character arrangement table are loaded from SYS1.IMAGELIB into SETPRT's storage.

   In this instance, G11 uses the graphic character modification module SPC1, which contains data codes and scan patterns for the open bracket, close bracket, and reverse slant.

6. The data codes of the characters in the GCM module address translate table locations that contain the addresses of WCGM locations. The scan patterns for the characters in the graphic character modification module(s) replace the patterns at the specified WCGM locations. Figure 6 shows the graphic character modification process. The open bracket replaces the cent sign, the close bracket replaces the exclamation point, and the reverse slant replaces the lozenge.



Using the character set identifier '82', from character arrangement table G11, 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 |   |

SETPRT loads the graphic character modification (GCM) module 'SPC1' from the image library into its own storage.

GCM Module

| | | | |
|---|---|---|---|
| SPC1 | 4A [ | 5A ] | E0 \ |

The data code '4A' addresses location '4A' in the translate table, and location '4A' of the translate table contains the address of location '0A' in WCGM00.

SETPRT loads the translate table portion of the character arrangement table G11 from the image library 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 |   |

SETPRT replaces the scan pattern of the cent sign in location '0A' of WCGM00 with the scan pattern of the open bracket (from the GCM module).

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 |   |

Figure 6. How Graphic Character Modification Works

## BURSTING-TRIMMING-STACKING

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, where the output from each data set is offset from that of the preceding data set. Copies and groups of copies can also be offset. The programmer can request 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" section).

In the standard system, you can initialize for BURST using JCL. With the enhancement features, you can also issue the SETPRT macro for a SYSOUT data set to change the printer settings during the printing process. If you are a direct user of the 3800 subsystem, you can use SETPRT to specify BURST.

## FORMS

The 3800 uses continuous single-ply fanfold forms with both edges punched for tractor feeding, and perforations between the pages. The paper 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 that 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.

The 3800 Printing Subsystem can use preprinted forms; or it 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 forms flashing, refer to the chapter "How to Flash Forms" in "Section 2. How to Use the 3800 Printing Subsystem." Performance considerations for using forms overlay are discussed later in the chapter "Performance" in "Section 3. Supporting the 3800 Printing Subsystem." For information on designing, making, and obtaining forms overlay negatives, see the Forms Design Reference Guide for the IBM 3800 Printing Subsystem.

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

## TWO-CHANNEL SWITCH

The two-channel switch attaches the 3800 via two channels to two separate processors. The switching is done manually using the interface enable/disable switches on the 3800. Only one channel interface can be enabled at a time, unless the dynamic switch feature is installed.

The dynamic switch feature 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 an 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.

# 3800 COMPATIBILITY WITH OTHER IBM PRINTERS

Most of the jobs you are currently printing on an IBM 1403 or 3211 can be printed without change on the IBM 3800 Printing Subsystem. However, there are some incompatibilities between the 3800 and impact printers, and there are some 3800 features that can be used to compensate for these incompatibilities. This chapter describes the things you should consider—both the new features and the incompatibilities—when planning for the 3800.

## OVERPRINTING

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. However, the user can design boldface characters and load them into the printer by means of the graphic character modification feature or by using library character sets.

- 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 can be designed by the user and loaded into the 3800 using the graphic character modification feature or a library character set.

## UNDERSCORING

On the 3800, you can print with underscores using one of these methods:

- The 3800 has a built-in underscore capability. It is invoked by referring to the code X'2D' in a character set that has the high-order bit of its ID on. When an underscore and another printable character are to be merged (that is, printed in the same character position) using this method, the 3800 generates the underscore to fit the cell-size of the other character, and prints the underscore at the same time as the character with which it is paired.

- You can use one or more of the six underscored character sets (in Gothic and Text) that are supplied by IBM. In these character sets, each of the characters is designed to contain its own underscore.

Using the built-in underscore capability requires two transmissions of the output line. The data line containing the character to be underscored must be sent to the 3800 first, then the underscore is sent. If you use an underscored character set, you transmit the output line only once, with a resulting saving in channel use and transmission time. The use of an underscored character set, however, might 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); it also requires specifying the desired underscored character arrangement table name in the CHARS parameter. The amount of underscoring required, then, should be the primary factor in determining which of the two methods is used.

**Note:** You can design your own underscore character and use it as you use any other character. However, see "Rules for Merging Data Lines into One Print Line" if you plan to use your underscore in combination with other characters.

## 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 "Section 2. How to Use the 3800 Printing Subsystem" and "Section 4. The IEBIMAGE Utility Program" of this book.

## CHANNEL COMMANDS

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 Read | Load UCSB |
| Load UCS (without folding) | Fold |
| Load UCS (folding) | Unfold |
| Allow Buffer Loading | Read PLB |
| | Read UCSB |
| | Read FCB |
| | Check Read |
| | Diagnostic Write |
| | Diagnostic Gate |
| | Skip to Channel 0 |
| | Raise Cover |

## FORMS CONTROL

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 (12.7 mm) 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 of using the FCB statement in "Section 4. The IEBIMAGE Utility Program" shows how to make this modification.

# SECTION 2. HOW TO USE THE 3800 PRINTING SUBSYSTEM

## HOW TO SPECIFY PRINTER SETTINGS

The OS/VS1 and OS/VS2 operating systems support 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, and the operating system initializes the printer based on these parameter specifications. You can also initialize the 3800 or change its setup dynamically, in certain circumstances, using the SETPRT macro; or you can change the setup using the JFCBE exit. This chapter includes sections on using JCL, using SETPRT, and using the JFCBE exit, as well as a section on defaults. 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.
- SETPRT parameters for controlling the printer setup. For details, see the section "Using SETPRT" in this chapter.
- 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 Models 4 and 5, 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 and 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, in OS/VS2 only, 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 and printing:
  - Forms control buffer modules
  - Copy modification modules
  - Character arrangement table modules
  - Graphic character modification modules
  - Library character set modules

## USING JOB CONTROL LANGUAGE (JCL)

This section contains an overview of how JCL is used to specify 3800 printer settings. Specific examples of using JCL are included in the sections that describe how to do the various tasks related to the 3800. Additional details on the parameters discussed in this publication, and information on JCL parameters that are not described in this publication can be found, for VS1 systems, in OS/VS1 JCL Services and OS/VS1 JCL Reference, and for VS2 systems, in OS/VS2 JCL.

The job entry subsystem (JES) is a key link between the user and the 3800 for SYSOUT processing. In "Section 3. Supporting the 3800 Printing Subsystem," specific 3800-related considerations are discussed under "Job Entry Subsystems."

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."

The following JCL parameters can be used to define for the operating system the required printer settings for an output data set:

**BURST=** If the forms should be threaded into the Burster-Trimmer-Stacker for bursting.

**CHARS=** The names of the character arrangement tables that 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 to modify the data, and a table reference character for selecting the character arrangement table to be used to print the modifying data.

**OPTCD=J** That each line of data contains a table reference character for selecting a character arrangement table to be used to print that line. (This is a DCB subparameter.)

**OPTCD=U** 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.)

Sometimes it is desirable not to specify any of the keywords in either the JCL or the SETPRT macro, but, instead, to take the JES defaults or the 3800's machine defaults. However, some jobs 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 the 3800 is allocated 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.

If the 3800 Enhancements are installed on an OS/VS2 MVS system, you can use the JCL parameter FREE=CLOSE to schedule printer output for immediate processing. For details on how to use this, see the section entitled "Using SETPRT with Spooled Data Sets."

## USING SETPRT

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

The possible users of SETPRT are:

- JES, which uses SETPRT to set control information for each data set it prints on the 3800. The control information is obtained by JES from the JFCB Extension. (See "JFCB Extension and Exit" later in this section.)
- The Open routine, which uses SETPRT to reset the 3800 to the subsystem defaults and to initially set the printer with control information as presented in JCL.
- Problem programs, which, in the standard 3800 subsystem, can use SETPRT only when allocating the 3800 as a direct device (that is, specifying UNIT=3800 or device address). If the 3800 enhancements are installed, the problem program can use

SETPRT to address a spooled data set in OS/VS2 MVS. For details see "Using SETPRT with Spooled Data Sets".

The following explanation of SETPRT is divided into three parts: the SETPRT macro instruction, the SETPRT supervisor call (SVC 81), and using SETPRT with spooled data sets.

## THE SETPRT MACRO INSTRUCTION

In the standard 3800 subsystem, the SETPRT macro instruction allows specification of the following:

**BURST=** Whether forms are to be burst or stacked without bursting

**CHARS=** Up to four names or storage addresses 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 or storage address, 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 to initialize the printer to hardware defaults

**MODIFY=** The copy modification module name or storage address, and the table reference character for the copy modification data

**OPTCD=** Whether data checks should be blocked

**REXMIT=** Whether data is to be retransmitted

If the enhancements are installed, the SETPRT macro instruction has additional parameters that allow specification of the following:

**PRTMSG=** Whether to print error messages to the programmer for errors that occur during SETPRT processing

**MSGAREA=** An A-type or register-type area-address pointing to a message feedback area provided by the user

**DISP=** (This parameter is only applicable to OS/VS2 MVS systems.) How to schedule the previously created data set output for printing. (This parameter refers to the output that was created before this SETPRT instruction was issued. It is only available for SYSOUT data sets and only if the enhancements are installed. The parameter will be ignored if it is included in a SETPRT instruction issued by a direct user.) For a detailed explanation of this parameter, see the section entitled "Using SETPRT with Spooled Data Sets" in this chapter.

**LIBDCB=** A user-defined library that is to be substituted for SYS1.IMAGELIB—only for the user with direct control of the 3800. (See the section "Using SETPRT with a User-Defined Library.")

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 OS/VS1 Data Management Macro Instructions, or OS/VS2 MVS Data Management Macro Instructions.

## THE SETPRT SUPERVISOR CALL

The SETPRT supervisor call (SVC 81) accesses the following modules from SYS1.IMAGELIB or from a user-defined library (only with the 3800 enhancements), unless the user specifies a virtual storage address for an FCB, a copy modification module, or a character arrangement table. In the latter case, SETPRT uses the module(s) from the specified storage address. The SVC then builds channel programs to load data from these modules into the 3800's buffers.

- Forms control buffer modules
- Copy modification modules
- Character arrangement table modules
- Graphic character modification modules
- Library character set modules (only with 3800 enhancements)

In the standard 3800 printing subsystem, 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.

If the 3800 Enhancements are installed, SETPRT loads character sets with even numbered IDs from the 3800 flexible disk, and character sets with odd-numbered IDs from the library. If, in a character arrangement table, a character set ID of X'7F' or X'FF' is assigned to a WCGM, that WCGM is reserved, and SETPRT does not load any character set into it. With the 3800 Enhancements installed, SETPRT does not reload a requested character set when it is already in an appropriate WCGM, unless INIT=Y is specified.

If SETPRT loads a character set into a different WCGM from the one specified in the character arrangement table, SETPRT alters the two high-order bits, that is, the WCGM code, of the contents of each location of the translate table to correspond to the WCGM into which the character set is loaded. (If the contents of a table location are X'FF', they 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 is accompanied by its index into the translate table. The 3800 then modifies the appropriate location in the WCGM that is pointed to by the contents of the translate table location. (See Figure 6.)

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

## USING SETPRT WITH SPOOLED DATA

If the IBM 3800 Printing Subsystem is operating under OS/VS2 MVS and the 3800 Enhancements are installed, the problem program can use SETPRT during the process of creating SYSOUT data to change the 3800 printer setup for all or part of that data. At the same time, using a combination of SETPRT and JCL parameters, the program can specify how JES is to schedule the printing of the data. Only one DCB can be open for the SYSOUT data set when the SETPRT macro is issued. The following SETPRT keywords are valid for a SYSOUT data set:

```
INIT, BURST, CHARS, COPIES,
FCB, FLASH, MODIFY, DISP, PRTMSG, MSGAREA
```

All the parameters specified in the macro (except DISP) apply to the data produced by the program after the SETPRT macro is issued. Using the DISP parameter, the user can specify the disposition of the data that was created before the SETPRT macro was issued. No matter how the DISP parameter is coded, JES separates the data created prior to the SETPRT macro from the data created after.

The options for the DISP parameter are:

```
DISP=   SCHEDULE
        NOSCHEDULE
        EXTERNAL
```

The DISP parameter specification is combined with the JCL parameter FREE=CLOSE to determine the data disposition:

- If the DISP parameter is not specified or if DISP=EXTERNAL is specified, the JCL parameter FREE=CLOSE determines the scheduling of the printing of the data set.
    - If FREE=CLOSE is specified, when the problem program closes the data set, JES schedules the data set for immediate printing.
    - If FREE=CLOSE is not specified, when execution of the job is completed, JES schedules the data set for printing.
- If DISP=SCHEDULE is specified, it overrides any specification in the JCL, and JES schedules the data set for immediate printing.
- If DISP=NOSCHEDULE is used, it overrides any JCL specification, and, when execution of the job is completed, JES schedules the data for printing.

## USING SETPRT WITH A USER-DEFINED LIBRARY

This feature is only available if the 3800 Enhancements are installed and for the user who has direct control of the 3800. If the user wishes to supply a library for SETPRT to use in place of SYS1.IMAGELIB, the user can specify the address of the data control block for that library using the LIBDCB parameter in the SETPRT macro. The library must have the same format and naming conventions as SYS1.IMAGELIB, and it must be opened for input before the SETPRT macro is issued. Code the DCB macro as described for use by the LOAD macro in OS/VS1 Supervisor Services and Macro Instructions, and OS/VS2 Supervisor Services and Macro Instructions.

**Note**: In an OS/VS2 MVS system, a user-defined library must be authorized. The Authorized Program Facility (APF) is described in OS/VS2 SPL: Supervisor.

## DEFAULTS FOR JCL, SETPRT, AND THE 3800

When no parameters for controlling the 3800 printer setup are specified in the JCL, defaults are used as specified by JES, 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.

If the machine defaults are used, you get a Gothic 10-pitch character set and a folded translate 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. The only channel code that is defined is channel 1 for the first printable line. COPIES is equal to 1, BURST is equal to its last setting, MODIFY and FLASH are reset and Block Data Check is unchanged.

The JES1 defaults for CHARS, BURST, FLASH, and FCB are specified in the Writer procedure started, or in the actual Start Writer command. If the JES1 default is not specified, the machine defaults are used.

For JES2 and JES3, the defaults are set at initialization time. JES2 defaults are the same as the hardware defaults, except for BURST, which is set to BURST=N. Under JES3, the installation can specify defaults for all the parameters in Figure 7 except OPTCD=U and OPTCD=J.

| | **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 8 |
| count | note 6 | note 6 | note 7 | - |
| **MODIFY=** module name | Yes | Yes | Yes | note 8 |
| trc | note 9 | note 9 | note 9 | - |
| **FCB=** FCB id | Yes | Yes | Yes | note 10 |
| **VERIFY** | No | Yes | Yes | - |
| **BURST=Y\|N** | Yes | Yes | note 11 | note 12 |
| **OPTCD=B\|U** | note 13 | note 13 | note 14 | - |
| **OPTCD=J** (DCB subparameter) | Yes | Yes | - | - |

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

### Notes for Figure 7

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.

7. 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.

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

9. Yes, however, if the copy modification module name is specified, and the trc (table reference character) is not, a trc of 0 is assumed.

10. 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.

11. The default is BURST=N (no bursting).

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

13. Only the unblock data check can be specified as a DCB subparameter. The default is "block data check." JES ignores OPTCD for SYSOUT data sets, and forces "block data check."

14. Yes. However, if OPTCD is not specified in the SETPRT macro, the setting is determined by the DCB subparameter OPTCD and the default is "block data check."

## 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; you specify in the DCB the address of an exit list and in that list request a type X'15' exit (refer to OS/VS1 Data Management Services Guide or OS/VS2 MVS Data Management Services Guide 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).

If a program issues a SETPRT macro for a SYSOUT data set (only in OS/VS2 MVS, with the 3800 Enhancements installed), SETPRT updates the JFCBE with any new setup data specified. If no JFCBE exists, SETPRT creates one.

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 could 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.

## MULTIPLE OPENS OF A DATA SET

When a 3800 printer is directly allocated, each time the OPEN macro is issued, the Open routine initializes the printer settings without regard for any previously specified settings for the same device. Each OPEN macro causes a SETPRT macro with INIT=Y to be issued. The problem program's JCL and JFCBE exit routine determine the contents of the JFCBE and, therefore, the printer settings.

## DYNAMIC ALLOCATION IN OS/VS2

Dynamic allocation does not support the CHARS, MODIFY, BURST, and FLASH parameters, or the copy grouping subparameter of the COPIES parameter. However, there are other ways to specify them.

- As when data destined for an impact printer is being reassigned to a 3800, one value for a character arrangement table can be supplied in the UCS parameter for a call to dynamic allocation. See "Reassigning Printing to Alternate Printers" in "Section 3. Supporting the 3800 Printing Subsystem."
- When allocating the 3800 directly, an OPEN and a SETPRT macro can be issued.
- If the Enhancements are installed, an OPEN and a SETPRT macro can be issued after a SYSOUT data set has been dynamically allocated. The SETPRT routine will cause the creation of a JFCBE.

3800 character sets are used by means of character arrangement tables. A 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; writable character generation storage for two character sets is standard in the 3800, and storage for two more sets is an optional feature.

Note that the 64th position of any character set loaded into the fourth WCGM is unavailable for printing, because the system reserves the use of the 8-bit data code index into that position (X'FF') to designate an unprintable character.

Library character sets can be loaded into the WCGMs more rapidly than the character sets that reside on the flexible disk.

When two or more character arrangement tables specify the same character set, only one copy of that character set is loaded into a WCGM, unless one or more of the copies is to be modified by graphic character modification modules. In that case, each copy is considered a unique character set and is loaded into a WCGM. Library character sets can be modified in the image library while the character sets in the 3800 require graphic character modification. Therefore, it can be more efficient to use library character sets in character arrangement tables that share modified character sets.

The CHARS parameter allows you to specify up to four character arrangement table names for printing a data set.

The CHARS parameter can be specified either in JCL or in 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 34 at the beginning of "Appendix B: IBM-Supplied Character Arrangement Tables," gives the names of the 48 character arrangement tables in the six groups supplied by IBM. Since the various groups, other than the "Basic group" (listed first), are selected at system generation, you might not have all the listed character arrangement tables available on your system.

The tables in the "3211 group" and "1403 group" provide character arrangements that are identical to those 3211 and 1403 print trains that have the same names. Those in the "OCR group" and in the "Katakana group" also provide arrangements identical to the print trains with the same names.

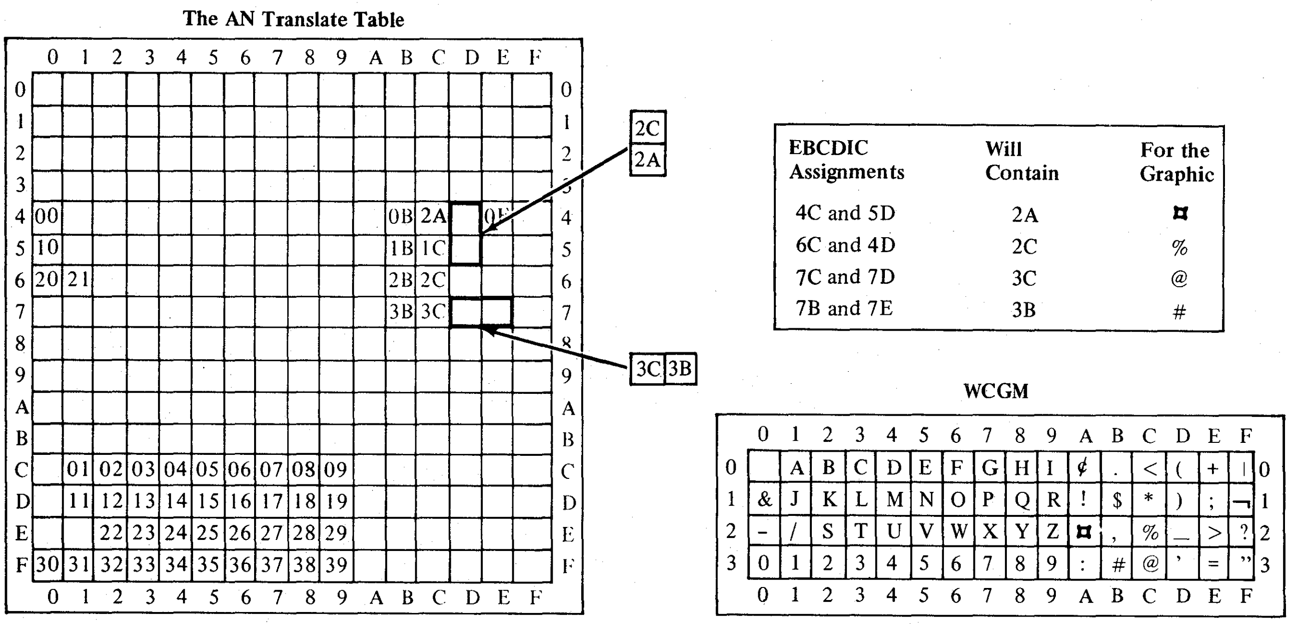
## MODIFYING CHARACTER ARRANGEMENT TABLES

You can modify the IBM-supplied character arrangement tables or create your own, as needed, by using the IEBIMAGE utility program (see "The IEBIMAGE Utility Program" for details). There are several instances when you might want or need to modify character arrangement tables.

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 8.

If the 3800 Enhancements are installed on your system, and if you plan to use library character sets, you must change existing character arrangement tables or create new character arrangement tables to point to the library character sets. Library character sets must have odd-numbered IDs, while character sets that are stored on the 3800's flexible disk have even-numbered IDs.



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 8. Dualing of Graphics, Where Two Data Codes Print the Same Character

## INTERMIXING CHARACTER STYLES AND PITCHES

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

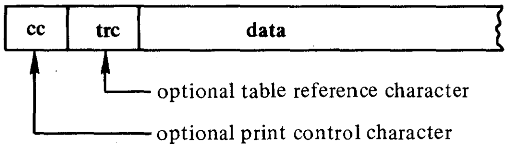
- More than one character arrangement table can be specified using the CHARS parameter, and with OPTCD=J coded as a DCB subparameter, you can select the table you want by coding a table reference character (trc) as the first byte of each output data line (following the optional print control character). 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 in other sections of this chapter, and also in the chapter "How to Change and Create Characters." The third method is explained below. The fourth method is described under "How to Modify Copies."

## USING TABLE REFERENCE CHARACTERS 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 method of selecting 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 such a 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. When JES2 or JES3 prints on a 3800, a trc that references a character arrangement table that has not been loaded is converted to reference table 0.
- 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.

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 9.

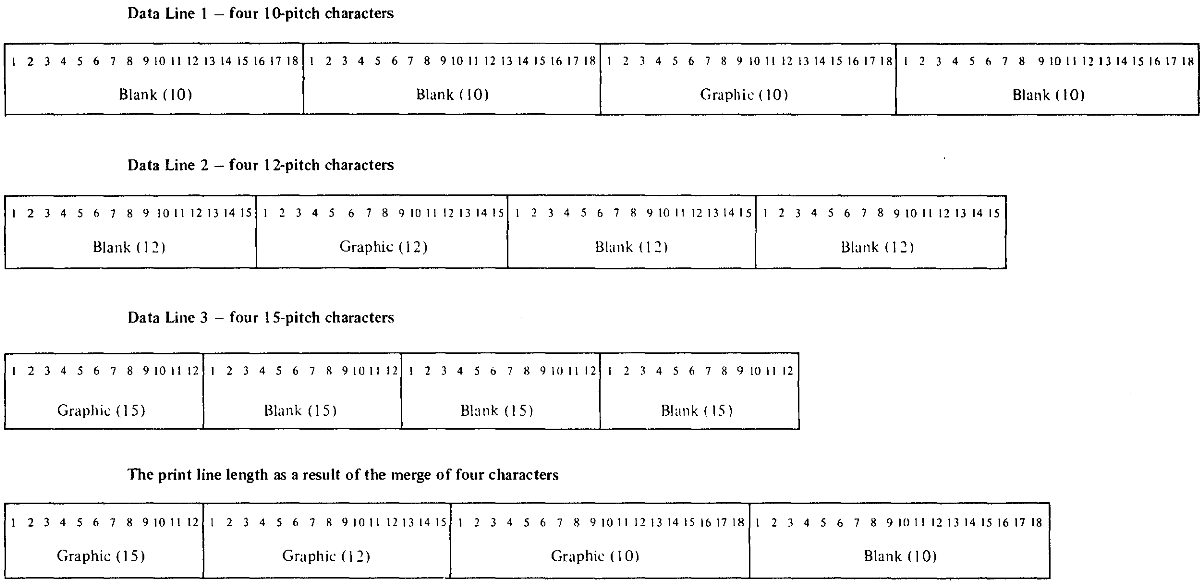


Figure 9. The Result of Merging Different Pitch Data Lines

## SPECIAL CONSIDERATIONS WHEN DESIGNING A CHARACTER SET

In designing a character set, give special consideration to the following:

- The scan pattern in the 00 position of WCGM 0 is sometimes used as a substitute for an unprintable character or as a pad character for a short print line. When the system encounters an unprintable character or needs a pad character, the system checks location '40' of the currently selected character arrangement table. If location '40' contains a code other than 'FF', the system uses the scan pattern from the WCGM location pointed to in location '40' (standardly, a blank). However, if location '40' of the character arrangement table contains 'FF', the system issues a Data Check and uses the scan pattern that is in the first position of WCGM 0. If this scan pattern is not a blank, it might be inappropriate as a fill character or as a substitute for an unprintable character.

- If you are using the built-in underscore for overprinting, the high-order bit of the ID of the character set you are using must be on. Also, you must specify '2D' as the WCGM location of the underscore character. When these conditions are satisfied, an underscore is generated by the 3800 without examining the '2D' location of the WCGM.

If the last (64th) character of a character set is to be used, that character set cannot be loaded into the fourth WCGM.

## HOW TO CHANGE AND CREATE CHARACTERS

There are two functions in the 3800 that you can use for creating and changing characters. In the standard 3800, graphic character modification can be used. However, with the 3800 Enhancements installed, library character sets are available in addition to graphic character modification.

Graphic character modification allows the substitution or extension of graphic characters in an already-defined character arrangement. A large number of IBM-supplied substitution characters are available, and you can design and use your own characters.

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.

Library character sets are duplicates of the character sets that reside on the 3800 disk. The library character sets can be stored in an image library, where they can be examined and modified, using IEBIMAGE control statements.

IBM supplies 20 library character sets; each set contains 64 segments. You can store as many of these character sets as you like in an image library for use in printing on the 3800.

You can use the IEBIMAGE utility program to build a graphic character modification module or a library character set module. You can include characters from existing modules or specify the design of newly-created characters, or both. Then, also using IEBIMAGE, place the identification of the new module into the character arrangement table you are using; so that, each time that character arrangement table is used, the graphic character modification or library character set is also used. (See the section entitled "The IEBIMAGE Utility Program" for details.)

## THE WORLD TRADE NATIONAL USE GRAPHICS

The standard substitution characters known collectively as the "World Trade National Use Graphics" are available in 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 1 of using graphic character modification in "Section 4. The IEBIMAGE Utility Program" shows how to list the contents of this module. Example 2 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 (size) 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 horizontal 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 which is to be 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, that is, only 1/8 inch high. 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.

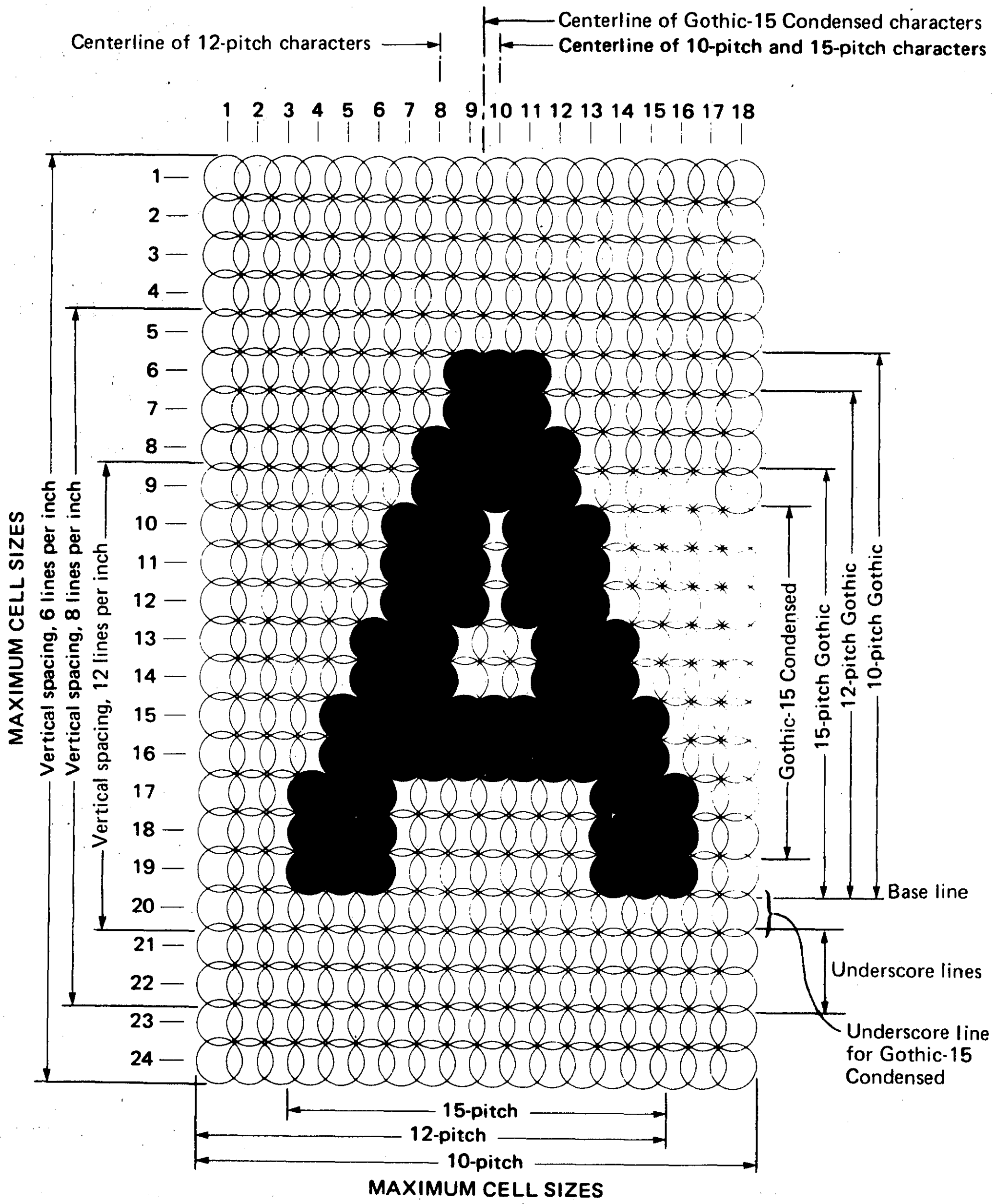

Centerline of Gothic-15 Condensed characters
Centerline of 12-pitch characters
Centerline of 10-pitch and 15-pitch characters
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
MAXIMUM CELL SIZES
Vertical spacing, 6 lines per inch
Vertical spacing,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

## 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).

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 1 of using the GRAPHIC statement in "Section 4. The IEBIMAGE Utility Program" shows how to do this.

The designs of the characters in the IBM-supplied character sets on the flexible disk cannot be printed out. However, if the 3800 Enhancements are installed, the same graphic character patterns can be stored in an image library, using the IBM-supplied library character sets. These library character sets can be listed using IEBIMAGE control statements.

**Note:** The IBM-supplied character definitions are also shown in the IBM 3800 Character Set Catalog.

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.

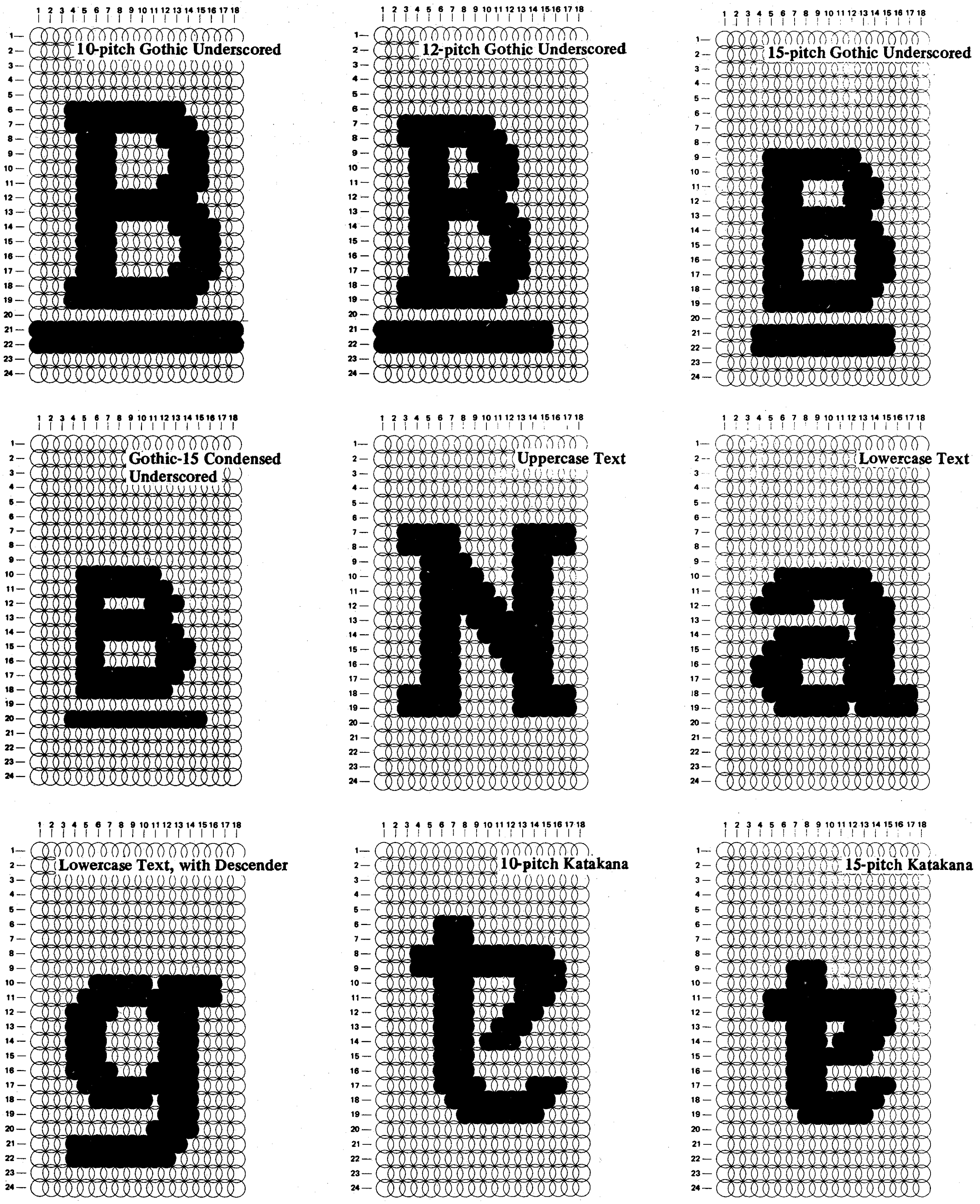


Figure 11. The Scan Patterns of Some Gothic Underscored, Text, and Katakana Characters

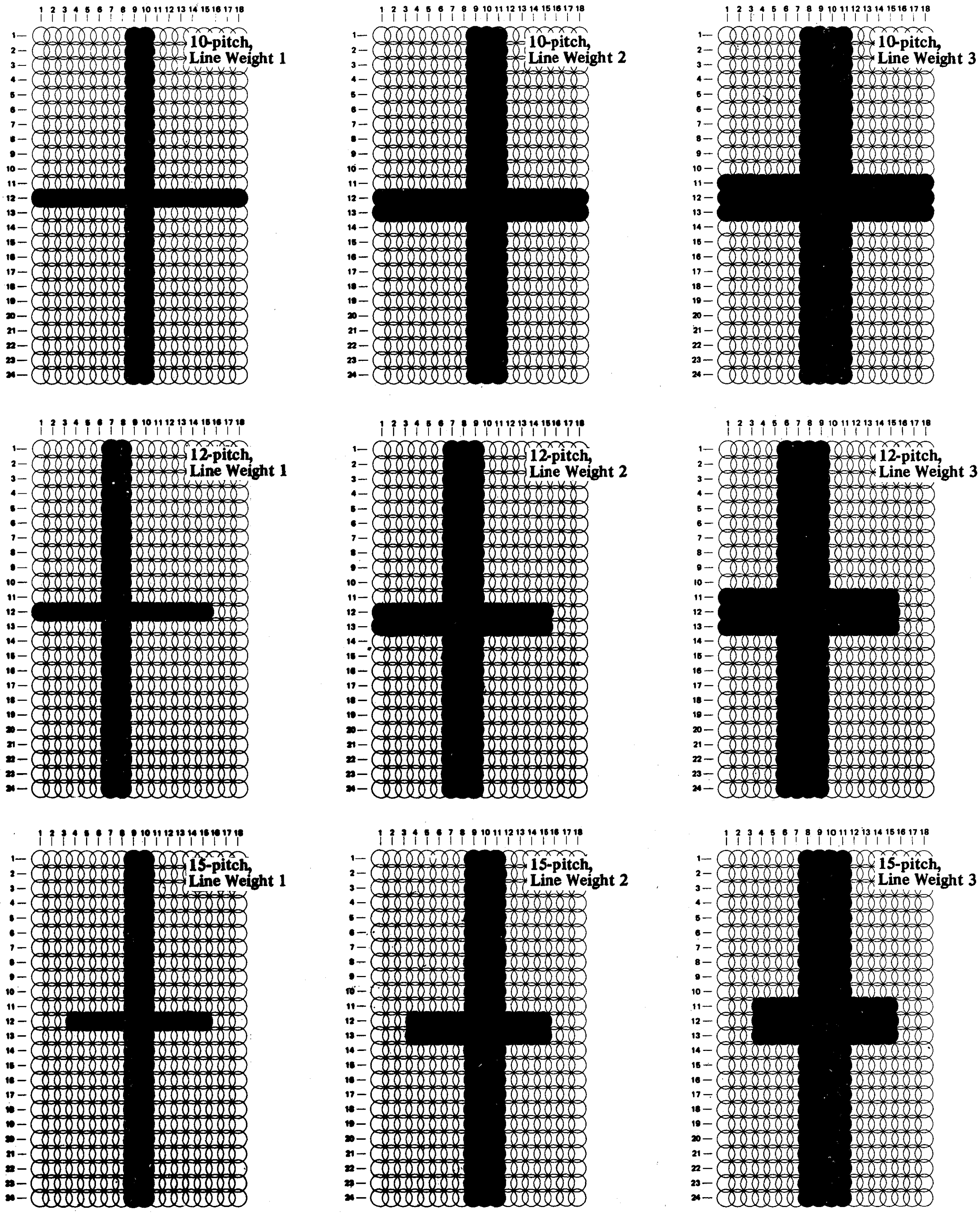


Figure 12. The Scan Patterns of the Format Intersection in Three Pitches and Line Weights

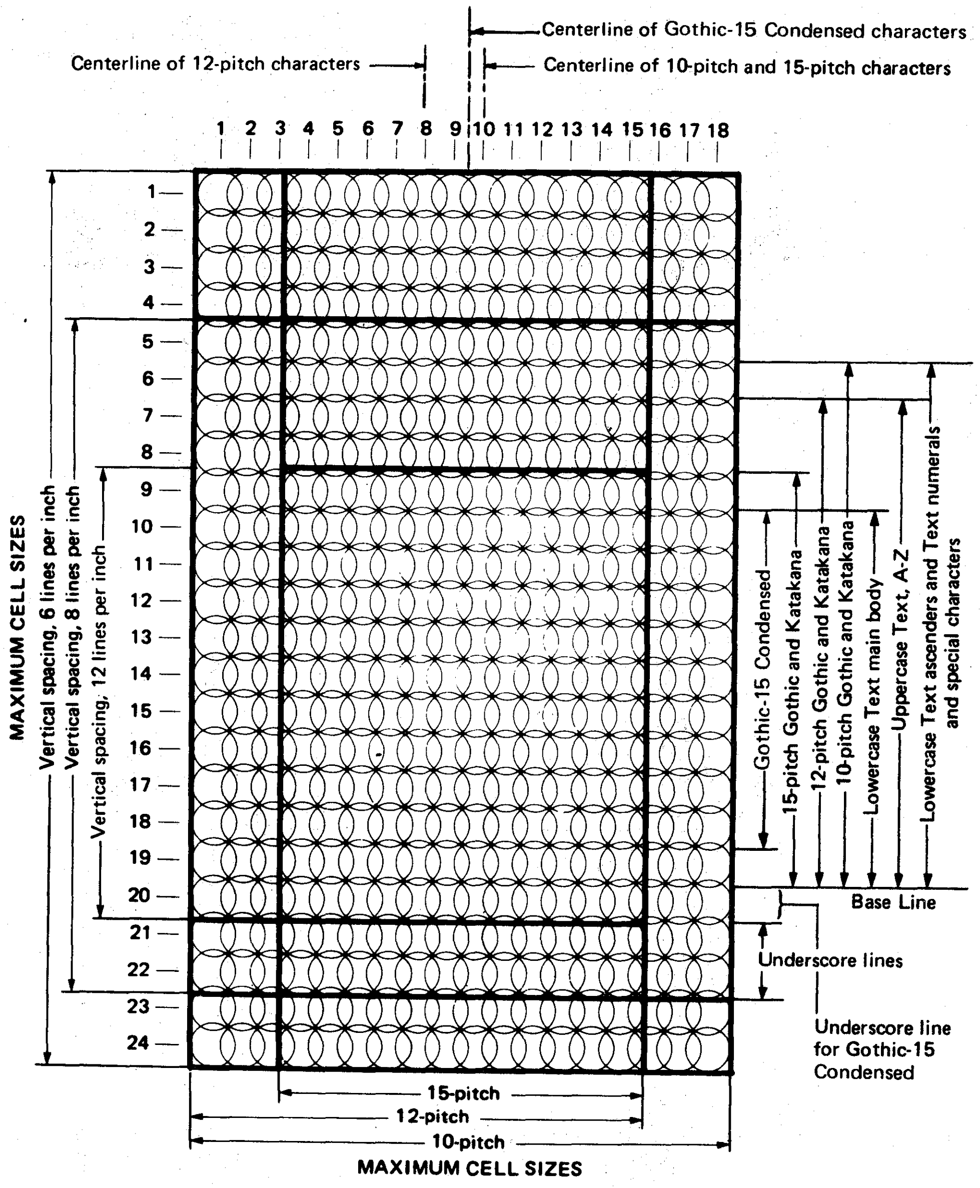

Centerline of Gothic-15 Condensed characters
Centerline of 12-pitch characters
Centerline of 10-pitch and 15-pitch characters
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
MAXIMUM CELL SIZES
Vertical spacing, 6 lines per inch
Vertical spacing,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

## 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 is already defined.

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 smoother.)

The next step is to transfer the dot pattern to the right side of the design 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, or the CHARSET ASSIGN statement. See "GRAPHIC Statement" and "CHARSET Statement" in "Section 4. The IEBIMAGE Utility Program."

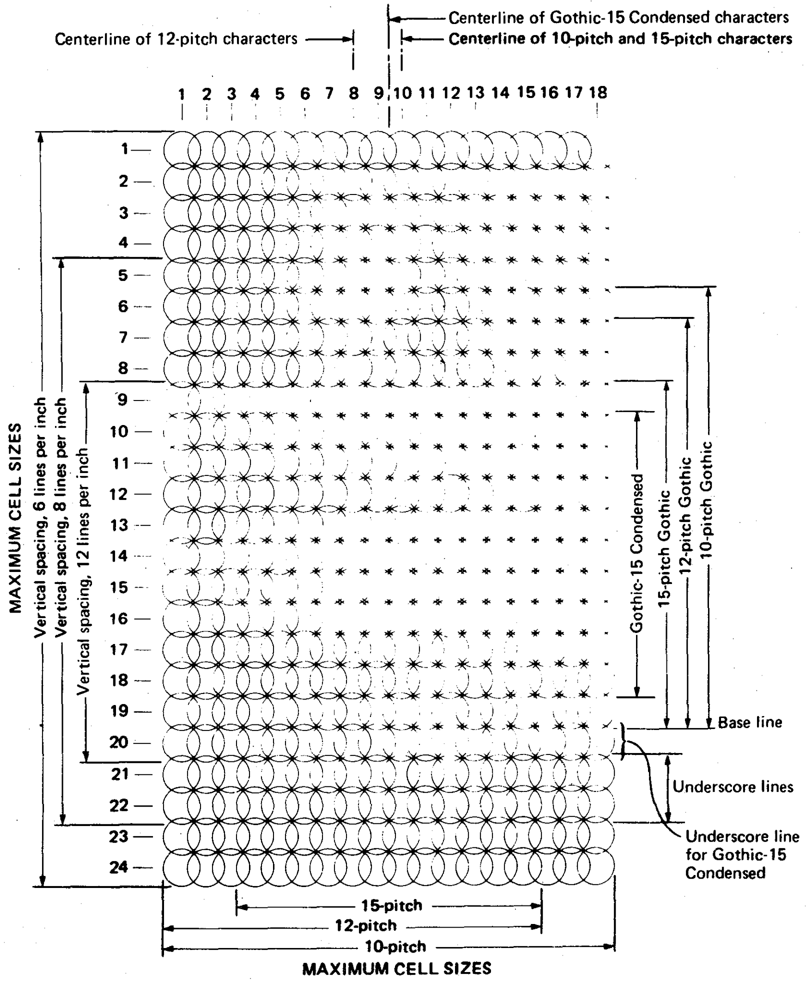
It is sometimes necessary to make other compromises in the design of a new character, some of which may only be apparent after the character has been printed. The desired effect may require modifying the design to fit the rectangular grid pattern better.

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 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

| | 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.

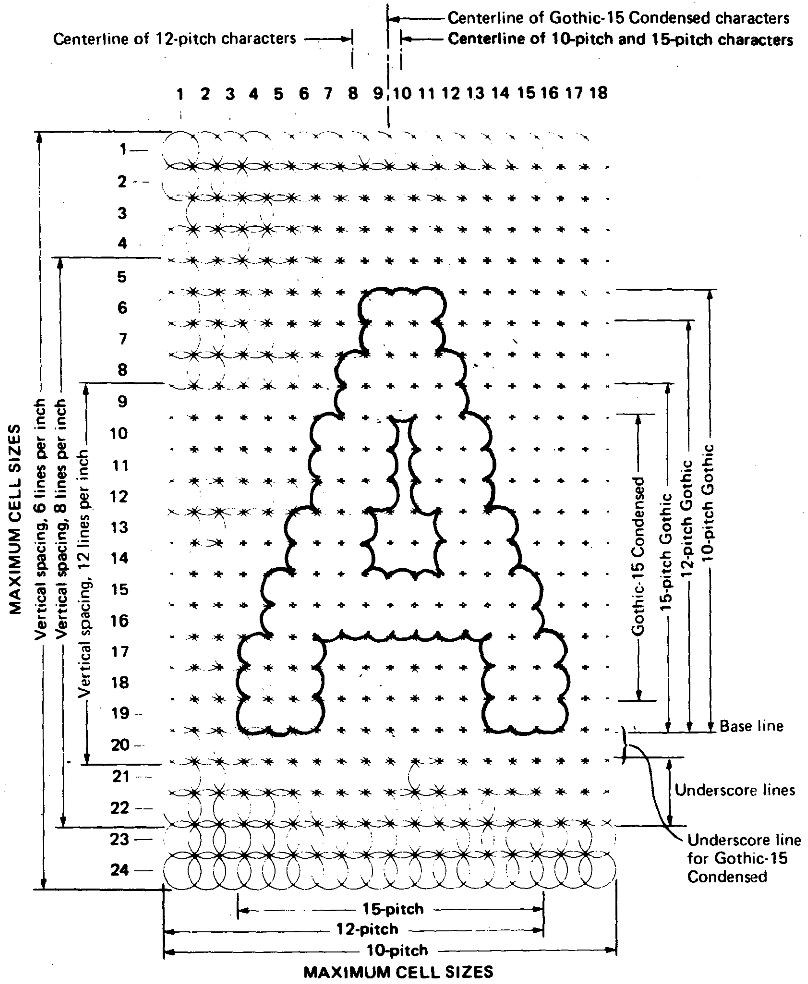
Figure 14. The IBM 3800 Printing Subsystem Character Design and Coding Form

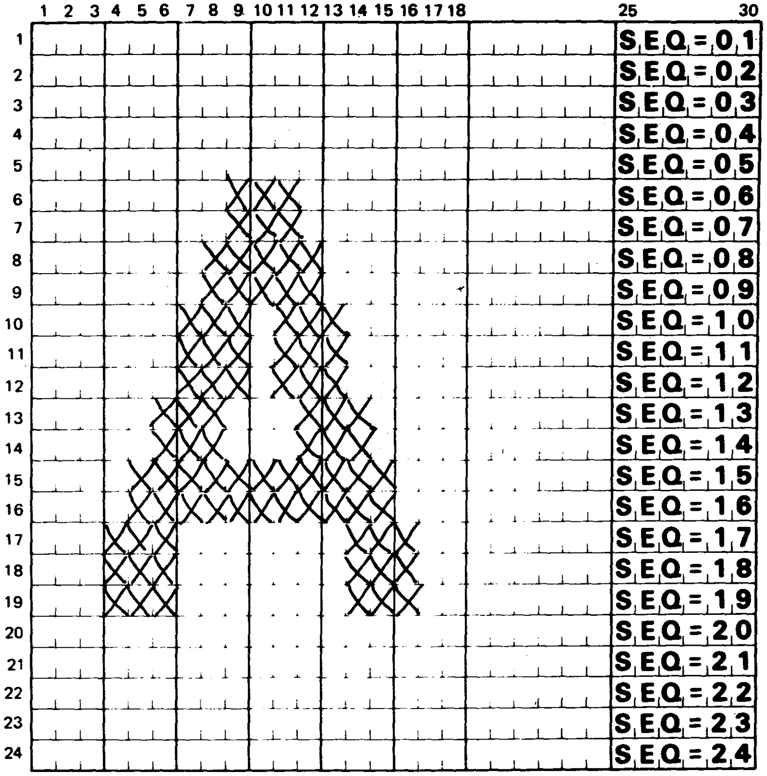
Figure 15. Transferring the Dot Pattern to the Right Side of the Design Form

**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 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



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.

## 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.

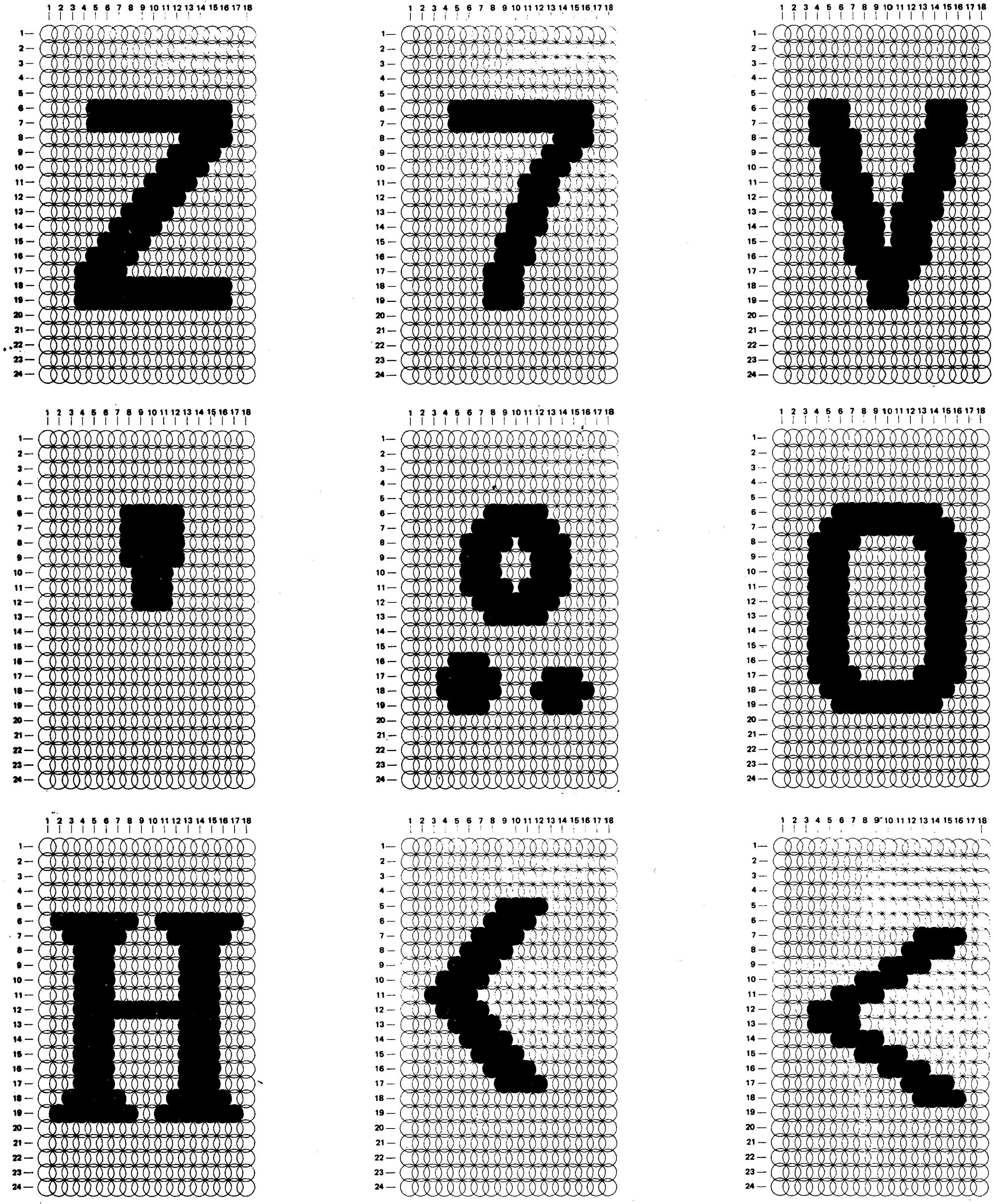
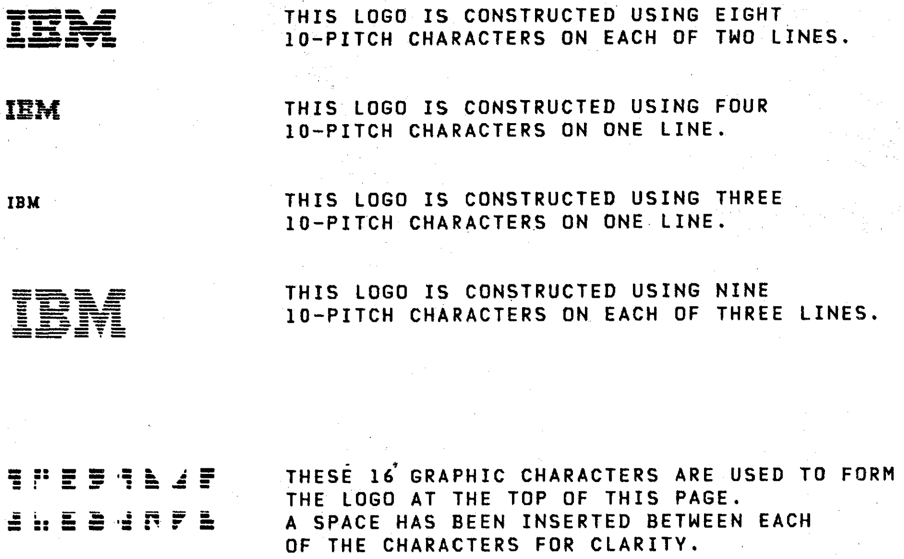


Figure 16. Some Scan Patterns to Illustrate the Design Guidelines

## 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 17. 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.

If you are using a SYSOUT data set, the number of copies you want and, optionally, whether copies are to be arranged in copy groups, that is, with multiple copies of each page grouped together, are specified using the COPIES parameter in your JCL:

**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 to be printed of each page before starting the printing of the next page. That is, it specifies the number of copies in a copy group. 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).

For information on defaults and rules for coding, see "The COPIES Parameter" in either OS/VS1 JCL Reference or OS/VS2 MVS JCL.

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

**COPIES=14**
: In this example, 14 copies of 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.

**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.

If your data set is directly allocated rather than a SYSOUT data set, the results will be different when you use the COPIES parameter in your JCL DD statement.

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 job entry subsystem.

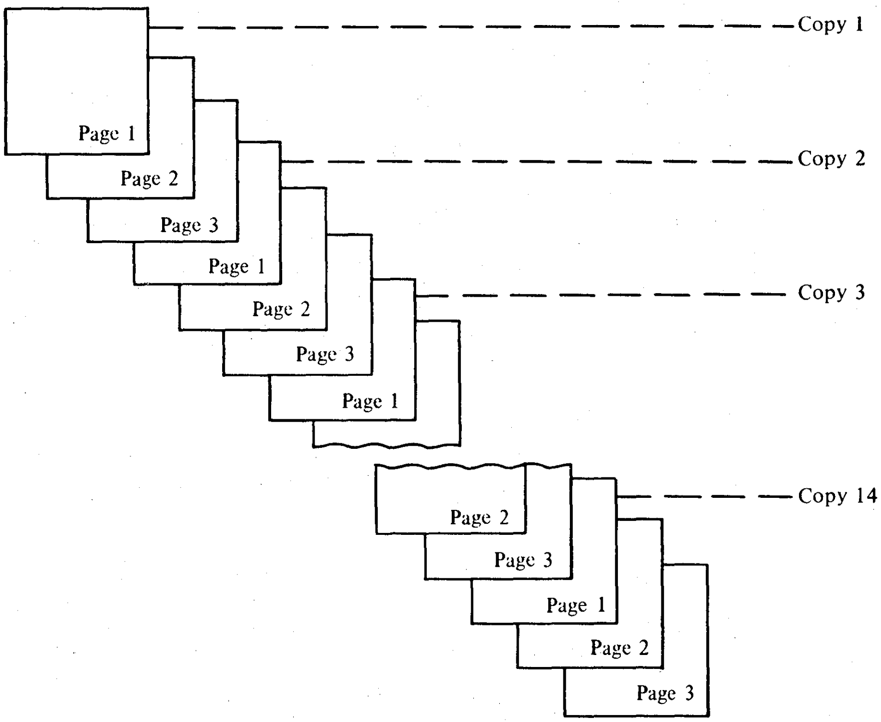


Figure 18. Output in Response to COPIES=14

**COPIES=14**
In this example, when your data set is not a SYSOUT data set, 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))**
If your data set is directly allocated, and you are using this format, the total number of copies (in this case 6) is ignored, and 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.

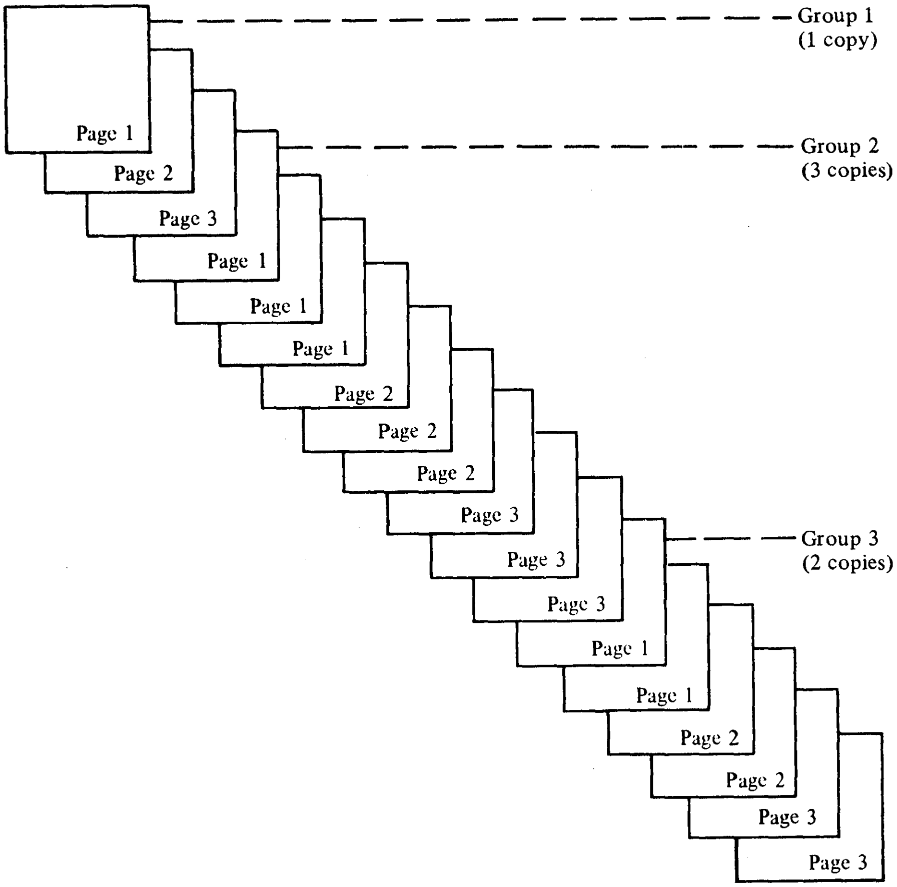


Figure 19. Output in Response to COPIES=(6,(1,3,2))

## HOW TO MODIFY COPIES

The copy modification feature allows printing of predefined data on a specified copy or copies of pages of a data set. Examples of such predefined 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 and is placed in an image library using the IEBIMAGE utility program (see "Section 4. The IEBIMAGE Utility Program" 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 image library.

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 either OS/VS1 JCL Reference or OS/VS2 MVS JCL.

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 preprinted for specific requirements, and reduces job setup 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 requesting the insertion of that particular forms overlay frame into the 3800. If the 3800 Enhancements are installed, a code is loaded into the display on the 3800, and the printer is made not-ready.

**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 effective 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, or, if it is larger than the total number of copies, all copies have the overlay printed.

For coding rules and examples, see the "FLASH Parameter" section in either OS/VS1 JCL Reference or OS/VS2 MVS JCL.

## 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 in "How to Modify Copies")

- Intermix and merge multiple data lines using table reference characters with OPTCD=J in your DCB (as described in "Using Table Reference Characters 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 alphamerics 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 using the IEBIMAGE utility program and is stored in a library. (See "Section 4. The IEBIMAGE Utility Program" for details.) To use it, specify its name with the FCB parameter in the DD statement for your output data set, or, with OS/VS2 3800 Enhancements, in the SETPRT macro. For details on use of the FCB parameter, see "FCB Parameter" in either OS/VS1 JCL Reference or OS/VS2 MVS JCL.

If you do not specify an FCB in your JCL for a SYSOUT data set, JES uses its own default. If JES does not specify an FCB, 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:

| | Machine Code Control Characters | | ANS Control Characters |
|---|---|---|---|
| Action | Action After Printing | Action Only (Immediate) | Action Before Printing |
| Print only (no space) | 01 | - | 4E (+) |
| Space 1 line | 09 | 0B | 40 (b̸) |
| 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. The PRTOV macro can be used to test for channels 9 or 12. (See OS/VS1 Data Management Macro Instructions and OS/VS2 MVS Data Management Macro Instructions.)

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 with the OPTCD=B parameter of the SETPRT macro.

## HOW TO BURST AND OFFSET STACK OUTPUT

The BURST parameter can be specified in the JCL or in the SETPRT macro. It is used to specify whether the paper output is to go to the optional Burster-Trimmer-Stacker or to the continuous forms stacker. This parameter 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 the paper is already threaded to the desired stacker. If the 3800 Enhancements are installed, a code is loaded into the display on the 3800, and the printer is made not-ready.

**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.

If you direct your output to the Burster-Trimmer-Stacker and you are using the 3800 as a system printer, JES causes the stacked sheets of each group of copies of a data set to be offset from the preceding group of copies in the stacker. If you are using the 3800 as a directly attached printer, you can get offset stacking by issuing an End of Transmission channel command.

# SECTION 3. SUPPORTING THE 3800 PRINTING SUBSYSTEM

# JOB ENTRY SUBSYSTEMS

JES1, JES2, JES2 NJE, and JES3 contain support for the 3800 Printing Subsystem and for the 3800 Enhancements. Further information on JES1, the job entry subsystem for OS/VS1, is in the OS/VS1 Planning and Use Guide. For OS/VS2, information on JES2 is in OS/VS2 MVS System Programming Library: JES2,, information on JES2 NJE is in System Programming Library: Network Job Entry Facility for JES2, and information on JES3 is in OS/VS2 MVS System Programming Library: JES3.

## CONTROLLING THE OUTPUT

With the 3800 Enhancements installed, the job entry subsystems monitor a 3800 print job from the time that JES starts to print the job output until the printed output is completely stacked. The following subsections discuss some ways to optimize the use of the 3800 with or without the Enhancements installed.

### SYSOUT CLASSES

Any interruption for operator action (such as changing forms) reduces the overall throughput. 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.

### PAGE DEFINITION

Because the 3800 cannot print on the top and bottom half inch of a page, care must be exercised in the definition of a page (number of lines) to be used by JES. The printer default for printing is 6 lines per inch, which means that on a standard form of 11 inches (12 inches for a 3800 using ISO paper sizes), only 60 lines (66 for an ISO-paper-size machine) can be printed. Unless a special FCB is used, the number of lines on a page defined for JES should not exceed this maximum. To print more than 60 lines per page (66 for an ISO-paper-size machine), an FCB that defines 8 lines per inch or an intermix of 6 and 8 lines per inch must be used.

### 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 JES after each group of copies of a data set, this marking is changed from one vertical bar to two, or vice versa. If the output is going to the Burster-Trimmer-Stacker, each group of copies of a data set 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. JES 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 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 chapter "How to Change and Create Characters" for more information on this.

## SPOOL SPACE

The proper spool block size is dependent on the work load and the type of printing. Because of the speed of the 3800 printer, it is important to calculate a large enough block size when the 3800 is used in an installation. However, since the spool volume is used to store system and data set control information, as well as the output data, selection of too large a spool block size may create a large amount of unused space. This may eventually lead to spool volume saturation, not leaving enough space to store additional data. This results in the suspension of reader and problem program tasks until enough space has been freed up.

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 multiple-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. It is recommended that, in JES2, track-call despooling be used for the 3800, and that, in JES3 initialization, the RECORDS parameter be used.

## WRITER PROCEDURES IN OS/VS1

For output operations under JES1 (the job entry subsystem for OS/VS1), there are two types of buffers involved. The first is the central pool buffers used to handle data to and from the spool volume(s). The second is the JES access method (JAM) output buffers used to send data to unit record output devices. The size and quantity of the central pool buffers are defined during system generation with the JES macro. The JAM buffer sizes are determined by multiplying the value of the "Z" subparameter of the JES macro by the LRECL specification in the Writer procedure. The "Z" value defaults to 6 during system generation and LRECL in the standard IBM Writer procedure is 133. Both these values may be inappropriate for the high speed of the 3800, although they do meet the requirements of slower-speed, shorter line-length impact printers.

It is recommended that you override these values when starting a Writer procedure to a 3800 Printing Subsystem by including in the START command values for the DCB subparameters LRECL and BLKSIZE. The BLKSIZE value will be used to determine the size of each of the three JAM buffers and hence will override the multiplication calculation mentioned above. To increase efficiency and minimize the number of EXCPs issued, BLKSIZE shouŀd have a value that is a multiple of the page boundary size. LRECL should be 206. This value is used by the Writer only as the maximum value that a SYSOUT data set can have; the 206 is the sum of 204 (the maximum printable line length), and one byte each for possible print control and table reference characters.

An alternative method to overriding the standard Writer procedure is for the system programmer to create an installation Writer procedure for the 3800. Once the installation's procedure is in operation, the only thing necessary on the Start.Writer command is the name of the new procedure. For information on creating Writer procedures, see OS/VS1 Planning and Use Guide.

## REASSIGNING PRINTING TO ALTERNATE PRINTERS

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

- If the UCS parameter is not specified for a data set that was originally directed to the 3800, JES 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. For a 1403 in OS/VS1, the UCS and CHARS parameters are ignored.

- If an output data set was originally directed to an impact printer and, therefore, no CHARS parameter is specified, JES recognizes a valid name specified in the UCS parameter as being the character arrangement table name for a 3800.

- 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 on an impact printer.

- Specification of features unique to the 3800 (such as FLASH for forms overlay) is ignored on an impact printer.

- Any group value subparameters of the COPIES parameter are ignored for an impact printer.

- If you are using JES1 with an impact printer and table reference characters in your output data set, the table reference characters will print along with the data. (If the trcs are entered as X'0n' rather than decimal n, a blank might be printed). In OS/VS2, if OPTCD=J is specified, the table reference characters are not printed.

- 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 length of output lines is truncated to the maximum line length of that printer.

- For jobs with specific forms control requirements, the alternate impact 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 or carriage control tape that is equivalent to a 3800 FCB if that 3800 FCB intermixes different vertical line spacings on a page.

## RESPONDING TO PAPER JAM OR CANCEL KEY

A problem program with direct control of the 3800 can detect a "system restart required" type of paper jam and recover from it, and can detect when the Cancel key is pressed and respond to it. When a paper jam occurs or the Cancel key is pressed, the system turns on the DCBIFLDT bit in the DCBIFLGS field. The UCBPGID and the UCBLDATA fields of the 3800 UCB extension contain information about the pages being processed at the time.

UCBPGID contains the page ID of the page that was at the fuser when the paper jam occurred, or of the page that was at the transfer station when the Cancel key was pressed.

UCBLDATA contains the number of pages that were lost or purged. For a paper jam, this number reflects the number of pages in the page buffer, plus the number of pages on the drum, plus the number of pages between the transfer station and the fuser at the time of the failure. For the Cancel key, the number in UCBLDATA reflects the number of pages in the page buffer at the time of the cancel.

You are responsible for resetting the DCBIFLDT bit before resuming processing.

If you are using BSAM or QSAM in your problem program, you must supply an I/O error routine (SYNAD). See the OS/VS1 Data Management Services Guide or the OS/VS2 MVS Data Management Services Guide for details on how to use SYNAD.

## 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.

You should let your users know that they can request high-density dumps on the 3800, and where to find information on requesting and reading high-density dumps. For ABEND dumps for VS1, this information is in the OS/VS1 Debugging Guide and, for VS2, is in the OS/VS2 System Programming Library: Debugging Handbook. High-density dumps are supported for SNAP dumps in OS/VS2 only; information on SNAP dumps is in OS/VS2 Supervisor Services and Macro Instructions. The SVC and stand-alone dumps are supported through the PRDMP facility; information on this is, for VS1, in OS/VS1 Service Aids, and, for VS2, in OS/VS2 System Programming Library: Service Aids.

When CHARS=DUMP or any 15-pitch character set 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 and a print control character.

## PERFORMANCE

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.

To estimate the maximum data content that can be allowed per printed page without causing performance degradation, use the formula PB/N, where PB is the number of bytes of available page buffer storage and N is as given in the following tables.

| **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 nonrepetitive 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 required 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 might not provide for the enhanced page recovery available with SYSOUT in the case of a "system restart required" paper jam. Data to be printed by the 3800 should be assigned to SYSOUT data sets.

- To estimate the amount of 3800 page buffer 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 JES 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 might 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 .346 or about 8960 bytes per second.

- When much of the printing on a system is at relatively high print densities (for instance, when storage dumps are frequently printed), a multivolume spool is recommended.

- When using JES2, the track cell option for despooling data to the 3800 should be used. For information on this, see OS/VS2 MVS System Programming Library: JES2.

- For initializing JES3, the RECORDS parameter in the DEVICE statement should be carefully considered. This parameter determines the amount of spool read-ahead to be performed for a specific output device. The CB parameter in the DEVICE or OUTSERV statement should not specify D or J. See OS/VS2

_MVS System Programming Library: JES3_ for details.

- To avoid I/O interrupts caused by including 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 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** | | **ISO paper sizes** | |
|---|---|---|---|
| **Page length in inches** | **Maximum characters per page** | **Page length in inches** | **Maximum characters per page** |
| 3-1/2 | 5190 | 3 | 4530 |
| 5-1/2 | 7770 | 4 | 5830 |
| 7 | 9720 | 6 | 8400 |
| 8-1/2 | 11700 | 8 | 11000 |
| 11 | 14880 | 10 | 13600 |
| | | 12 | 16230 |

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

## SYSTEM GENERATION

The 3800 Printing Subsystem can be added to your system through a full system generation or an I/O device generation. 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 34 in "Appendix B: IBM-supplied Character Arrangement Tables") can be selected at system generation.

If you are generating an OS/VS1 system using a starter system, rather than an existing system, the SYS1.IMAGELIB provided with the starter system includes only the "Basic group" and the "3211 group" of character arrangement tables and the IBM-supplied FCB module named STD3. It does not include other groups of character arrangement tables or the IBM-supplied "World Trade National Use Graphics" graphic character modification module.

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 either _OS/VS1 System Generation Reference_ or _OS/VS2 System Programming Library: System Generation Reference_.

# SECTION 4. THE IEBIMAGE UTILITY PROGRAM

## WHAT IS IEBIMAGE?

IEBIMAGE is a utility program that creates and maintains the following types of 3800 modules and stores them in a library:

- 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.

- Library character set modules, which contain the scan patterns of IBM-supplied character sets and/or user-defined character sets. These modules and functions are only available if the 3800 Enhancements are installed.

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 the "IEBIMAGE Program Messages" section of OS/VS Message Library: Utilities Messages.

## INPUT FOR IEBIMAGE

The IEBIMAGE utility program uses as input the five function control statements, FCB, COPYMOD, TABLE, GRAPHIC, and CHARSET, combined with one or more of the auxiliary control statements, INCLUDE, OPTION, and NAME.

### OPERATION GROUPS

IEBIMAGE utility control statements are grouped together to create or print a library module. Each group of statements is called an operation group. Your job's input stream can include many operation groups. The operation groups (shown below without operands) 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 create or print a library character set module (only with 3800 Enhancements):

  ```
  [INCLUDE]
  {CHARSET|CHARSET, followed immediately by
  data statements}
  [additional CHARSET statements]
  NAME
  ```

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

## NAMING THE MODULE

You can assign a 1- to 4-character identifier (name) to the module you create by using the NAME control statement in the operation group you use to build the module. If the module is a library character set, the ID assigned to it must be exactly two characters in length. Each of those characters must be within the range 0 through 9, and A through F; and the second character must represent an odd hexadecimal digit. However, the combinations X'7F' and X'FF' are not allowed. This identifier is used in the JCL, the SETPRT parameter, or the character arrangement table, to identify the module to be loaded.

## 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, CHARSET, 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 quotation marks.

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 are not allowed on a statement that specifies an operation and no operands. If the 3800 Enhancements are installed, an asterisk can be coded in position one to indicate that the entire line is a comment.

**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. When a card is included for the sole purpose of continuing a comment, the continuation must begin in column 16.

## OUTPUT FROM IEBIMAGE

When IEBIMAGE builds a module to be stored in an image library, the program also produces an output data set (a listing) that includes the module's identification, the contents of the utility control statements that were used, the module contents, and messages and return codes.

Messages produced by the IEBIMAGE program are described in OS/VS Message Library: Utilities Messages. Note that a single input statement could cause multiple error messages.

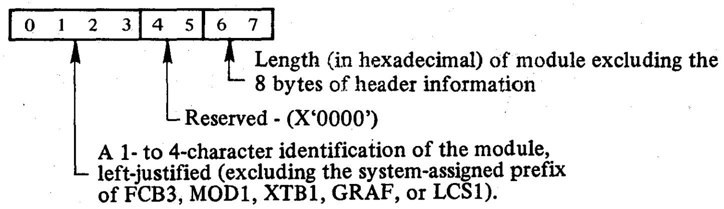
### RETURN CODES

IEBIMAGE produces a return code which represents the most severe 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 value 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 |

### MODULE STRUCTURE

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 8 bytes of header information
Reserved - (X'0000')
A 1- to 4-character identification of the module, left-justified (excluding the system-assigned prefix of FCB3, MOD1, XTB1, GRAF, or LCS1).

The SETPRT SVC uses the name to:

- Identify the module in the image library
- Store the name in the UCB extension

The SETPRT SVC uses the length to:

- Obtain sufficient storage for the module
- Build channel programs to load the data into the printer

## NAMING CONVENTIONS FOR MODULES

Each module placed in a library 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

LCS1 for library character set modules

While the 3800's programming support refers only to the 1- to 4-character name, or the 2-character ID (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 in alphameric and national characters ($, @, and #) in any order. Each character of the 2-character ID for library character sets must be within the range 0 through 9, and A through F; and the second character must represent an odd hexadecimal digit. Note, however, that 7F and FF are not used.

## JOB CONTROL FOR IEBIMAGE

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/VS1 Utilities or OS/VS2 MVS Utilities for details about the IEBUPDTE utility program.)

The modules that the IEBIMAGE utility program builds are stored in an image library that is a partitioned data set specified in the SYSUT1 DD statement. 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=library name,DISP=OLD
//SYSIN       DD   *
     utility control statements
/*
```

**Note:** DISP=OLD is specified to ensure that the library data set, which can be SYS1.IMAGELIB or a user-defined library, is not updated by other jobs while this job is running.

**Restrictions:**

The following restrictions should be noted.

- 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 0-1 | specify number of bytes following in ddname list |
| Bytes 2-9 | are binary zero, or ddname entry |
| Bytes 10-17 | are binary zero, or ddname entry |
| Bytes 18-25 | are binary zero, or ddname entry |
| Bytes 26-33 | are binary zero, or ddname entry |
| Bytes 34-41 | are system input ddname, normally SYSIN |
| Bytes 42-49 | are system output ddname, normally SYSPRINT |
| Bytes 50-57 | are binary zero |
| Bytes 58-65 | are library input ddname, normally SYSUT1 |
| Bytes 66-73 | 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 6-byte list that contains an EBCDIC initial page number for the output device. The 6-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 the appropriate OS/VS Utilities publication 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/VS1 Utilities* or *OS/VS2 MVS Utilities*) and HMASPZAP or AMASPZAP (as described in *OS/VS1 Service Aids* or *OS/VS2 System Programming Library: Service Aids*, respectively) should be used to help maintain SYS1.IMAGELIB.

If you use utility programs other than IEBIMAGE for maintenance, you must specify the full module name as it was specified in the NAME statement when the module was built. 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:

FCB3xxxx, which identifies an FCB module

MOD1xxxx, which identifies a copy modification module

XTB1xxxx, which identifies a character arrangement table module

GRAFxxxx, which identifies a graphic character modification module

LCS1nn, which identifies a library character set module

where:

xxxx
is the 1- to 4-character user-assigned name of the module.

nn
is the 2-character user-assigned ID of the module.

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.

## AUXILIARY CONTROL STATEMENTS

### INCLUDE STATEMENT

When an IEBIMAGE operation group is used to create a new module, the INCLUDE statement can identify an existing image library module to be copied and used as a basis for the new module. When the operation group is used to update an image library 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, or, in the case of a library character set module, its 2-character ID. 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. In this parameter, segment 1 is used to refer to the first segment of the module. 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, a graphic character modification module, or a library character set module (3800 Enhancements).

**Restrictions:**

- When the INCLUDE statement is coded in an operation group, it must precede any COPYMOD, TABLE, GRAPHIC, or CHARSET statements.
- Only one INCLUDE statement should be coded for each 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, a graphic character modification module, or a library character set 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)
  ```

## NAME STATEMENT

The NAME statement can name a new library module to be built by the IEBIMAGE utility program. The NAME statement can also specify the name of an existing library 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 library module. The module name is 1 to 4 alphameric and national ($, #, and @) characters, in any order, or, for a library character set module, a 2-character ID that represents two hexadecimal digits (0-9, A-F), the second digit being odd. Note that 7F and FF cannot be used.

**(R)**
indicates that the name and reference to the original copy, if any, of the named module are 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 I040:

  ```
  FORM1 NAME I040
  ```

- 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 *
  ```

## FORMS CONTROL BUFFER MODULES: FCB

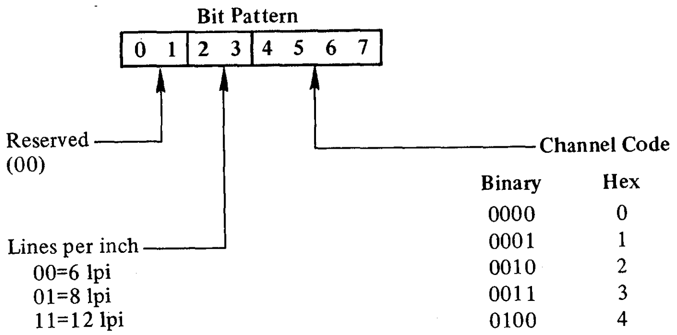
The FCB module is of variable length and contains vertical line spacing information (6, 8, or 12 lines per inch). The FCB module can also identify one of 12 carriage-control channel codes for each line.

The FCB module is created and stored in an image library, using the FCB and NAME utility control statements of the IEBIMAGE program. 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.

## 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.

## FCB MODULE LISTING

Figure 20 shows the IEBIMAGE listing of an FCB module. The notes that follow the figure describe the items in the figure that are marked with circled numbers.

```
                ①   ②
PRINT LINE   1 AT  6 LINES PER INCH - HAS CHANNEL  1 CODE.
PRINT LINE   2 AT  8 LINES PER INCH
PRINT LINE   3 AT  8 LINES PER INCH
PRINT LINE   4 AT 12 LINES PER INCH
PRINT LINE   5 AT 12 LINES PER INCH
PRINT LINE   6 AT 12 LINES PER INCH
PRINT LINE   7 AT 12 LINES PER INCH
PRINT LINE   8 AT 12 LINES PER INCH
PRINT LINE   9 AT 12 LINES PER INCH
PRINT LINE  10 AT 12 LINES PER INCH
PRINT LINE  11 AT 12 LINES PER INCH
PRINT LINE  12 AT 12 LINES PER INCH
PRINT LINE  13 AT 12 LINES PER INCH
PRINT LINE  14 AT 12 LINES PER INCH
PRINT LINE  15 AT 12 LINES PER INCH
PRINT LINE  16 AT 12 LINES PER INCH
PRINT LINE  17 AT 12 LINES PER INCH
PRINT LINE  18 AT 12 LINES PER INCH
PRINT LINE  19 AT 12 LINES PER INCH
PRINT LINE  20 AT 12 LINES PER INCH
PRINT LINE  21 AT 12 LINES PER INCH            ③
PRINT LINE  22 AT 12 LINES PER INCH
PRINT LINE  23 AT 12 LINES PER INCH
PRINT LINE  24 AT 12 LINES P
PRINT LINE  25 AT 12 L
PRINT LINE  26 AT
PRINT LINE  27
 INT LINE
                  ●

                  ●

                  ●

                         ER INCH
                     NES PER INCH
                   2 LINES PER INCH
                AT 12 LINES PER INCH
P           98 AT 12 LINES PER INCH
PRIN  LINE  99 AT 12 LINES PER INCH
PRINT LINE 100 AT 12 LINES PER INCH - HAS CHANNEL 12 CODE.
PRINT LINE 101 AT 12 LINES PER INCH
PRINT LINE 102 AT 12 LINES PER INCH
PRINT LINE 103 AT 12 LINES PER INCH
PRINT LINE 104 AT 12 LINES PER INCH
PRINT LINE 105 AT 12 LINES PER INCH
PRINT LINE 106 AT 12 LINES PER INCH
PRINT LINE 107 AT 12 LINES PER INCH
PRINT LINE 108 AT 12 LINES PER INCH
PRINT LINE 109 AT 12 LINES PER INCH
PRINT LINE 110 AT 12 LINES PER INCH
PRINT LINE 111 AT 12 LINES PER INCH
PRINT LINE 112 AT 12 LINES PER INCH
PRINT LINE 113 AT 12 LINES PER INCH
PRINT LINE 114 AT 12 LINES PER INCH
PRINT LINE 115 AT 12 LINES PER INCH
PRINT LINE 116 AT 12 LINES PER INCH
PRINT LINE 117 AT 12 LINES PER INCH
PRINT LINE 118 AT 12 LINES PER INCH
```

Figure 20. IEBIMAGE Listing of a Forms Control Buffer Module

**Notes to Figure 20:**

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.

## 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 image library, 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 shown in the chapter "FCB Module Listing."

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:**

- It is the user's responsibility to ensure that the total number of lines specified results 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.

**CHx**=(line[,line...])],**CHx**=(line...)]

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

**CHx**

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 job entry subsystem, and the Close routines for direct allocation to a 3800 with BSAM or QSAM, require a channel 1 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 USING THE FCB STATEMENT

Each of the following examples illustrates 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 5 and 35 at channel 2 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  CH2=(5,35),CH3=15,SIZE=85
  ```

### 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; 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,                                     X
               LPI=(8,88),                                  X
               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.

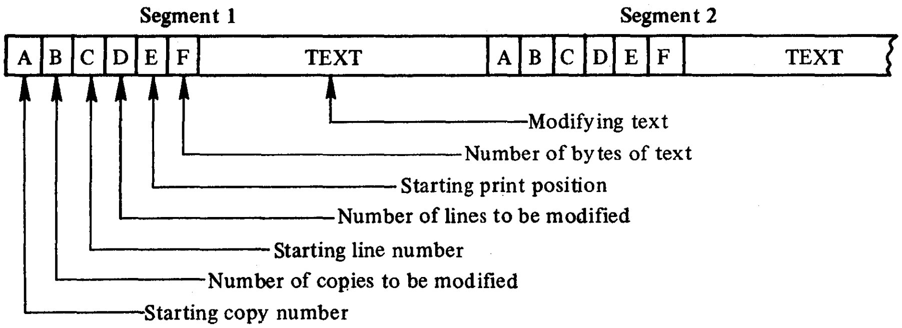
## COPY MODIFICATION MODULES: COPYMOD

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 created and stored in an image library using the INCLUDE, OPTION, COPYMOD, and NAME utility control statements of IEBIMAGE. 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.

## 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:

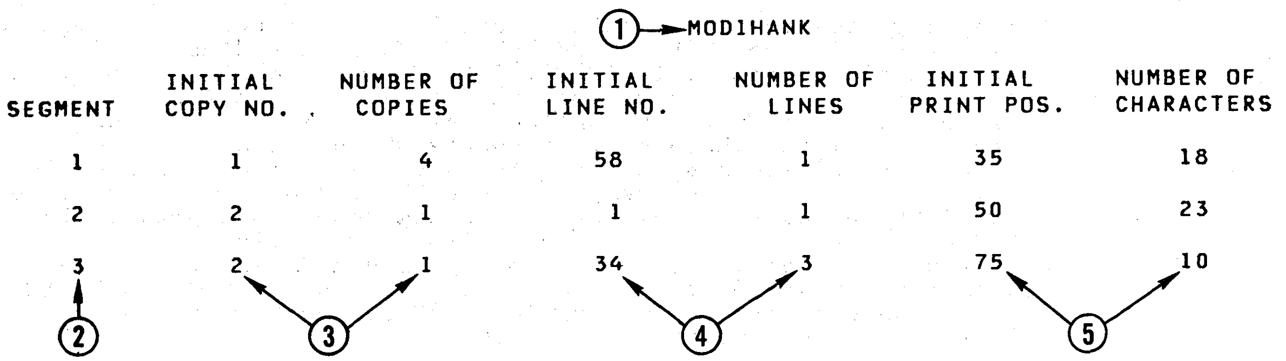


A, B, C, D, E, and F are each 1-byte fields.

- 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 8192 bytes of data and 8 bytes of header information.

## COPYMOD MODULE LISTING

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. The numbered notes that follow the figure describe the items marked with the circled numbers.



```
                                          MODIHANK

          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
```

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

### Notes to Figure 21:

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.

## COPYMOD STATEMENT

A copy modification module consists of header information followed by one or more modification segments. The header information contains 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 modifications per line

- 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 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.

For information on using your copy modification module, see the chapter "How to Modify Copies" in "Section 2. How to Use the 3800 Printing Subsystem." 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.

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 shown under "COPYMOD Module Listing" in this 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.

**Restriction:**

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 for each 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 for each 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 quotation marks.

If the text type is **C**, you can specify any valid character. Blanks are valid characters. A single quotation mark is coded as two single quotation marks. 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. This error indicator can be blocked.

## OPTION STATEMENT

The OPTION statement with the OVERRUN parameter is used in a COPYMOD operation group. When the IEBIMAGE utility program builds a copy modification module from the user's specifications, the program calculates an estimate of the time the modification will require during the planned printing. If the modification can be done in the time available for printing a line at 12 lpi, it can also be done at 6 or 8 lpi. (Note that 6, 8, and 12 lpi are the only print densities available on the 3800 printer.) On the other hand, if the copy modification module being built is too complex to be done in the time available for printing a line at 6 lpi, it certainly cannot be done at 8 or 12 lpi. (Note that at 12 lpi there is much less time available for printing a line than at 6 lpi.)

When the IEBIMAGE utility program determines that a copy modification module is likely to cause an overrun if it is used when printing at a specified number of lines per inch, the program produces a warning message to that effect. If the warning applies to 6 lpi, the message will also be produced for 8 and 12 lpi. If the warning applies to 8 lpi, the message will also be produced for 12 lpi.

If you are planning to use a particular copy modification module only while printing at 6 lpi, you can request suppression of the unwanted warning messages for 8 and 12 lpi by specifying the OPTION statement with 6 as the value of the OVERRUN parameter. If you are planning to print only at 8 lpi, you can use the OPTION statement with OVERRUN = 8 to request suppression of the unwanted warning messages for 12 lpi.

An effective use of the OPTION statement would be to determine the greatest print-line-density (6, 8, 12) at which the copy modification module will be used, then specify that density in the OVERRUN parameter to eliminate the warning messages for higher line densities.

If you specify OVERRUN=0, all overrun warning messages will be suppressed; if you specify OVERRUN=12, none will be suppressed.

The OPTION statement with the OVERRUN parameter is used only in a COPYMOD operation group, and can be placed before or after any INCLUDE statement for the group. The value in the OVERRUN parameter specifies the greatest line density for which the user wants the overrun warning message IEBA33I to be printed. The warning message is suppressed for a greater lpi density. 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 also 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
```

## COPYMOD LISTING WITH OVERRUN

Figure 22 shows the listing of segments of a copy modification module, where an overrun warning was in order. Even if the OPTION statement specifies OVERRUN=0 and the overrun warning message is not printed, a 'Note' is printed to the left of each segment description for which an overrun is possible. The numbered notes that follow the figure describe the items marked with the circled numbers.

```
                                              MOD1MH

                      INITIAL    NUMBER OF    INITIAL     NUMBER OF   INITIAL     NUMBER OF
           SEGMENT    COPY NO.    COPIES      LINE NO.      LINES     PRINT POS.  CHARACTERS
      ❶
              1          1          200          10           96          10          180

NOTE(1)       2          2          200          10           96          11          180

NOTE(1)       3          3          200          10           96          12          180
        ❷
NOTE(2)       4          4          200          10           96          10          180

NOTE(2)       5          5          200          10           96          11          180

NOTE(3)       6          6          200          10           96          12          180

NOTE(3)       7          7          200          10           96          10          180
        ❸
NOTE(3)       8          8          200          10           96          11          180

NOTE(3)       9          9          200          10           96          12          180
```

Figure 22. IEBIMAGE Listing of a Copy Modification Module with Overrun Notes

**Notes to Figure 22:**

1. Note 1 indicates that, for segments 2 and 3, you might have a copy modification overrun if you are printing at 12 LPI.

2. Note 2 indicates that, for segments 4 and 5, you might have a copy modification overrun if you are printing at 8 or 12 LPI.

3. Note 3 indicates that, for segments 6, 7, 8, and 9, you might have a copy modification overrun if you are printing at 6, 8, or 12 LPI. In other words, you might have an overrun at any LPI.

## EXAMPLES OF USING 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,' ')
```

### EXAMPLE 1:  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  RT01
/*
```

**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 "Controller'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 quotation marks 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 RT01, and it is stored as a member of the SYS1.IMAGELIB data set.

## EXAMPLE 2: BUILDING A COPY MODIFICATION MODULE

In this example, a copy of an existing copy modification module, RT01, 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, RT01, remains unchanged and available for use.

```
//          JOB  ...
//          EXEC PGM=IEBIMAGE
//SYSUT1    DD   DSNAME=SYS1.IMAGELIB,DISP=OLD
//SYSPRINT DD    SYSOUT=A
//SYSIN     DD   *
     INCLUDE  RT01,                                               X
                DELSEG=1
     OPTION   OVERRUN=8
     COPYMOD  COPIES=(2,3),                                       X
                LINES=(52,6),POS=100,                             X
                TEXT=(X,'404040404040405C5C')
     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 RT01 is used as a basis for the new module, and that the first modification segment of RT01 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 also 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.

## CHARACTER ARRANGEMENT TABLES: TABLE

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 (WCGMs), except the last position in the fourth WCGM (WCGM 3), which would be addressed by X'FF'. 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 on (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. If a character set identifier is even, the character set is accessed from the printer's flexible disk. If the ID is odd, the character set is retreived from the image library.

The character arrangement table is created 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.

"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 referred to by a translate table can be printed.

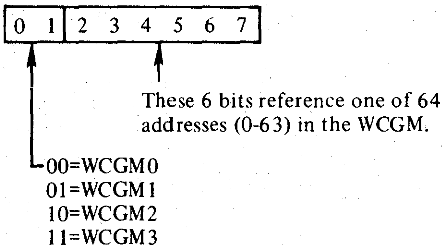
## THE CHARACTER ARRANGEMENT TABLE MODULE STRUCTURE

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 WCGM sequence numbers
- Four 4-byte fields for graphic character modification module names

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

- Bits 0 and 1 of each translate table byte refer to 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, using a copy of the translate table, alters bits 0 and 1 of each non-X'FF' byte of 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 fields containing character set IDs with their corresponding WCGM sequence numbers, followed by four 4-character names of graphic character modification modules. The format is as follows:

- Each CGMID is a 1-byte character set ID containing two hexadecimal digits (as listed in Figure 26 in "Appendix A. IBM-Supplied Character Sets"). If the second (low-order) digit is odd, and the 3800 Enhancements are installed, the ID refers to a library character set; if it is even, the ID refers to a character set on the flexible disk. 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.

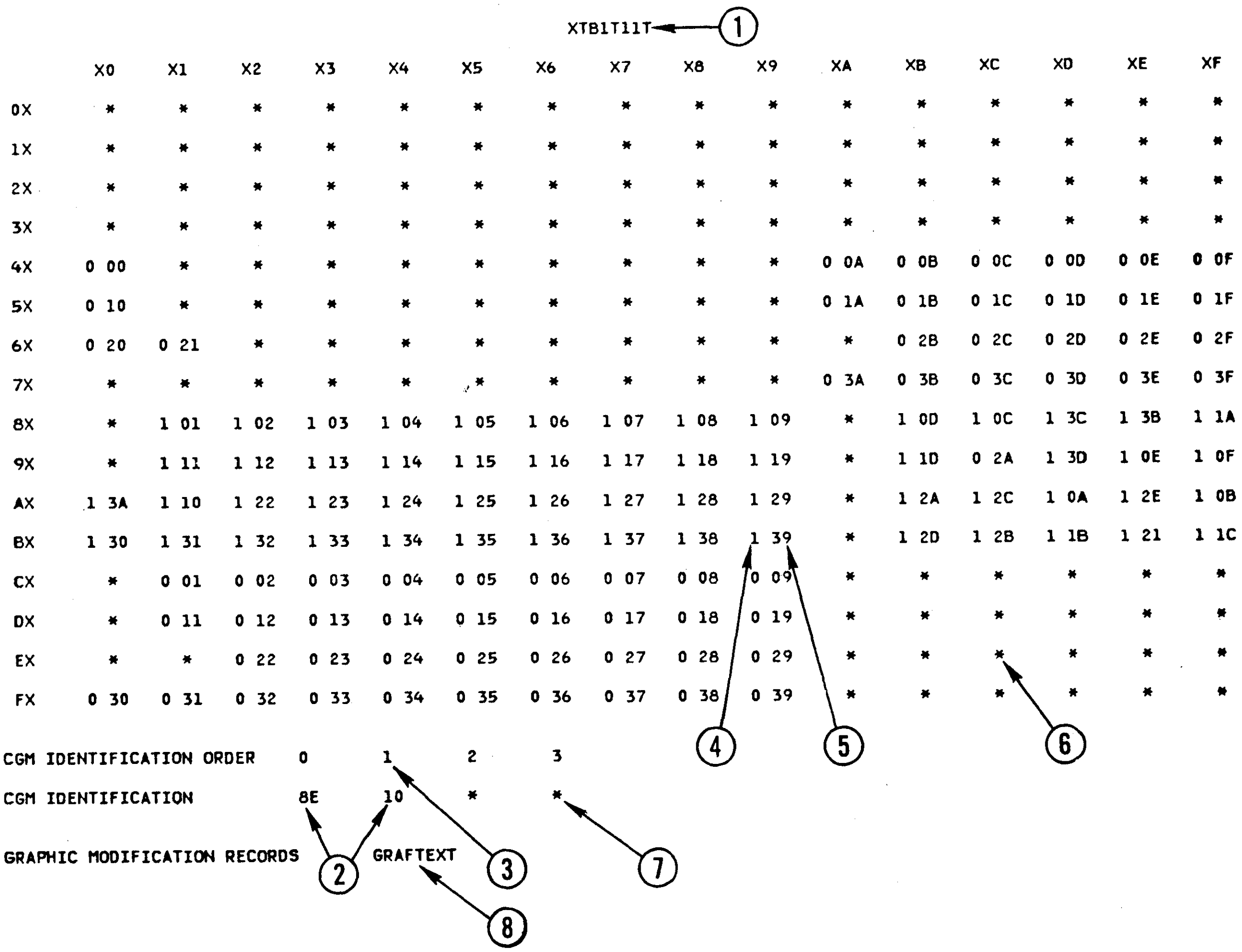
| | | | |
|---|---|---|---|
| CGMID0 | WCGMNO0 | CGMID1 | WCGMNO1 |
| CGMID2 | WCGMNO2 | CGMID3 | WCGMNO3 |
| Name1 | | | |
| Name2 | | | |
| Name3 | | | |
| Name4 | | | |

- 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 CGMIDx and the WCGMNOx are both X'00' when there are no character sets referred to after the first one.

## TABLE MODULE LISTING

Figure 23 shows the listing of a character arrangement table module. Each of the notes following the figure describes the item in the figure that is marked with the circled number.



```
                                                                 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
```

Figure 23. IEBIMAGE Listing of a Character Arrangement Table Module

**Notes to Figure 23:**

1. The name of the character arrangement table module, as it exists in the image library'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. The 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. The 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 list of character set identifiers indicates that no character set is specified to use the corresponding WCGM. If you specify 7F or FF as a character set identifier (to allow accessing a WCGM without loading it), a 7F or FF prints here.

8. The name of a graphic character modification module, as the name exists in the library's directory (including the system-assigned prefix).

## 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 on the flexible disk). 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 can be specified for each operation group. 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 library, causes the module to be formatted and printed. The format of the printed character arrangement table module is shown under "Table Module Listing" in this 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 character sets that are to be used with the character arrangement table. (The IBM-supplied character sets are 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 a character set. If the character set is on the flexible disk, the ID is even; if the set is in a library, the ID is odd. 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 7F or FF as the character set identifier eliminates the unnecessary step of loading a character set from the flexible disk. The difference between 7F and FF is related to overprinting and underscoring. (See "Special Considerations When Designing a Character Set" in "Section 2. How to Use the 3800 Printing Subsystem.")

**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 an image library.

**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 USING 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
               LOC=((FA,3D,1),(FB,3E,1),(FC,3F,1))
  ```

  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.

### EXAMPLE 1: 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 IBM-supplied 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 2: 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 3: 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 4: 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.

## GRAPHIC CHARACTER MODIFICATION MODULES: GRAPHIC

The graphic character modification module is variable length and contains up to 64 segments. Each segment contains the 8-bit data code and the 72-byte scan pattern of a graphic character.

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 that is to be copied and used as a basis for the new module.

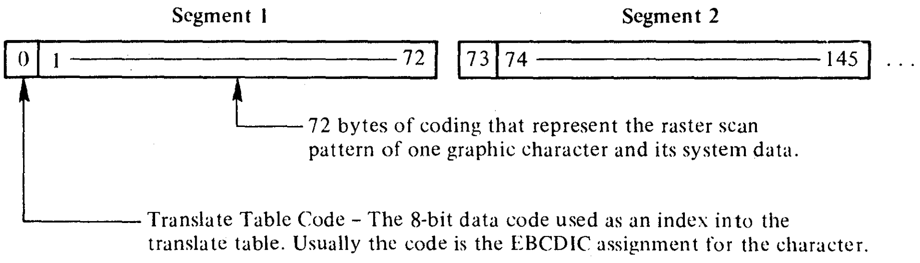
A GRAPHIC statement, when followed by one or more data statements, defines a user-designed character. A GRAPHIC statement can also select a character segment from another graphic character modification module. Each GRAPHIC statement causes a segment to be created for inclusion in the new module.

The NAME statement identifies the new module, and indicates that the module is to be added to the library or is to replace an existing module of the same name. More than one GRAPHIC statement can be coded between the INCLUDE and NAME statements; and all such GRAPHIC statements apply to the same graphic character modification module.

## THE GRAPHIC CHARACTER MODIFICATION MODULE STRUCTURE

The module contains eight bytes of header information. For details of this header information, see "Module Structure" under "Output from IEBIMAGE" earlier in the first chapter of this section.

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



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. .The contents found at that location (a 1-byte WCGM code) determine the WCGM location into which the 72 bytes of scan pattern and system data are to be placed.

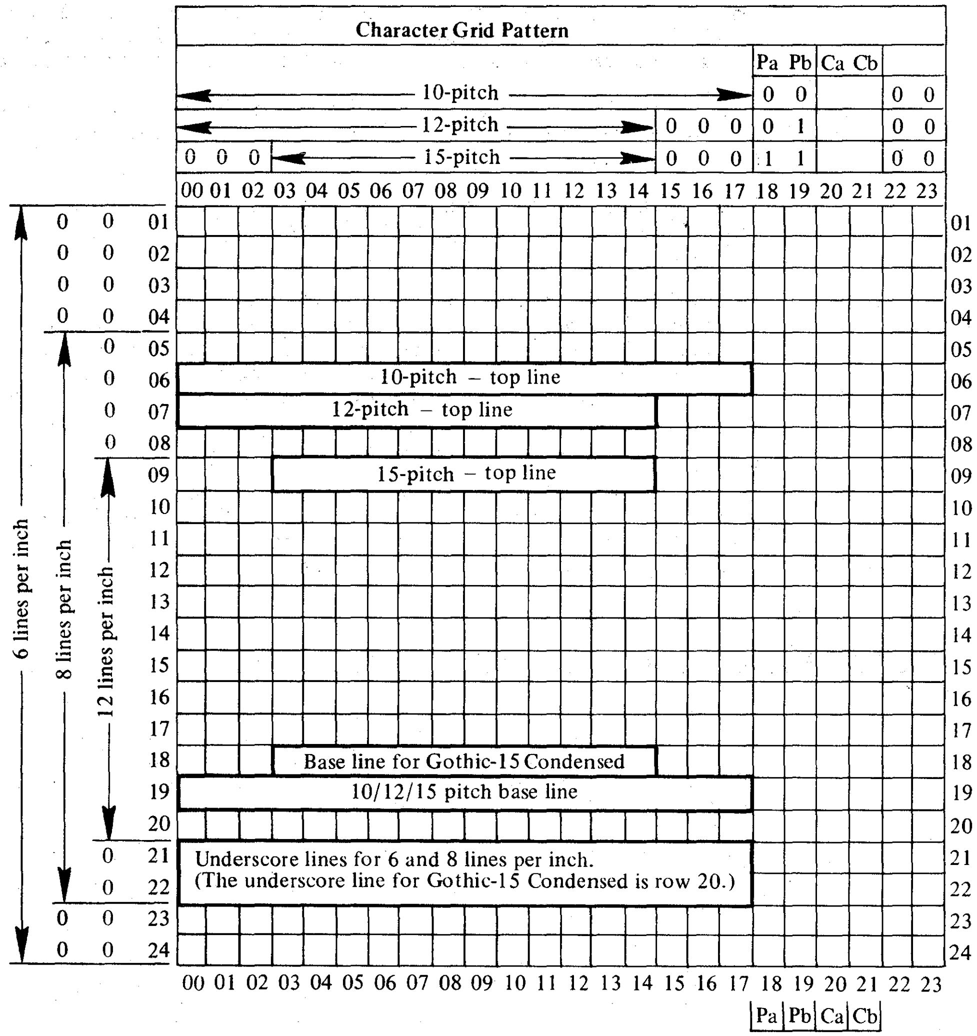
The 72-byte graphic definition that makes up the scan pattern and system data for one character is divided into twenty-four 3-byte groups. Each 3-byte group is expressed below as a horizontal row of twenty-four 1-bit elements:

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.



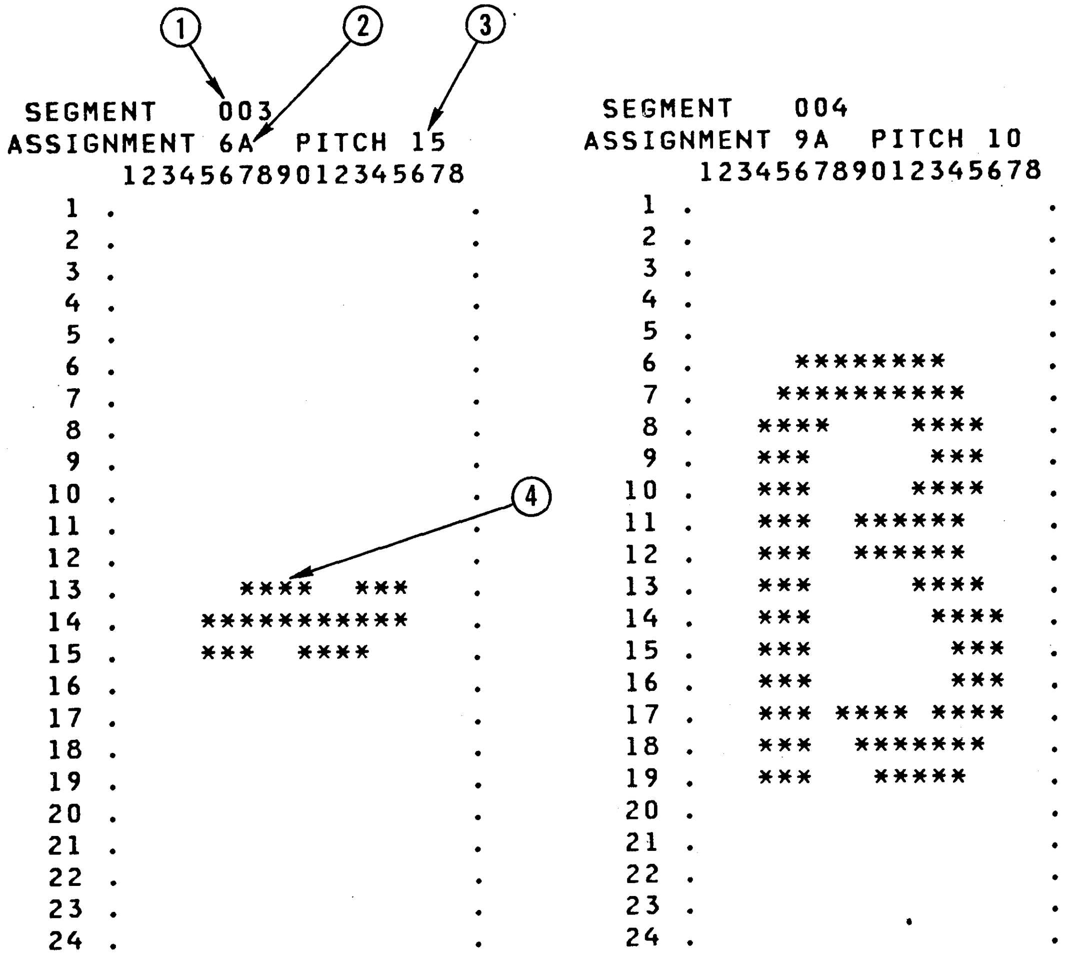
Pa & Pb = pitch (which must be constant within a character)

00=10-pitch<br>
01=12-pitch<br>
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

## GRAPHIC MODULE LISTING

Figure 24 shows an extract from a listing of a graphic character modification module. This extract contains the listing of two segments of the module. Each of the notes following the figure describes the item in the figure that is marked with the circled number.



```
 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 .                  .
```

Figure 24. IEBIMAGE Listing of Two Segments of a Graphic Character Modification Module

**Notes to Figure 24:**

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.

## GRAPHIC STATEMENT

The GRAPHIC statement specifies the contents of one or more 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 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. Whan 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.

**Restrictions:**

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 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 of this book. 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 through 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 USING 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
  ```

### EXAMPLE 1: 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=SHR
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
    GRAPHIC
        NAME  *
/*
```

**Notes:**

- DISP=SHR is coded because the library is not being updated.
- 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 2: 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 3: 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 4: 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 <u>data statements</u> 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 5: 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:

```
┌──┐
│38│
│00│
└──┘
```

## LIBRARY CHARACTER SET MODULES: CHARSET

The 3800 Enhancements must be installed before library character sets can be created or used. The library character set module is a fixed-length module made up of 64 segments. Each segment contains the 72-byte scan pattern of a graphic character, and a code (00-3F) that identifies the WCGM location into which the scan pattern is to be loaded.

IBM supplies 20 library character sets, each of which is a copy of one of the character sets that is resident on the 3800 disk. Each library character set ID is one greater than the ID of the comparable character set on the 3800 flexible disk. The 3800 disk copies have even-numbered IDs, while the copies for the image library have odd-numbered IDs (two characters representing hexadecimal digits, except 7F and FF).

The library character set module is created using the INCLUDE, CHARSET, and NAME control statements.

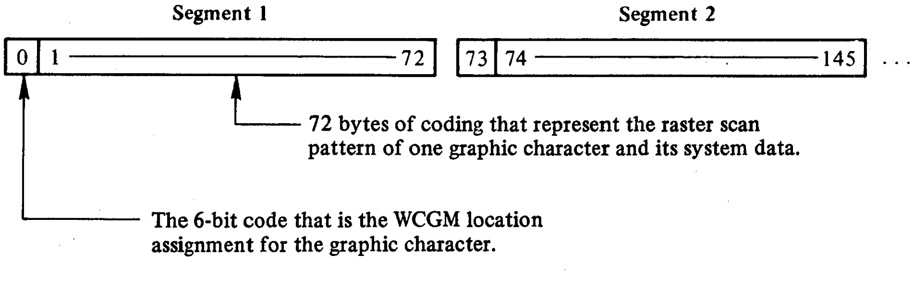
The INCLUDE statement identifies an existing module.

A CHARSET statement, when followed by one or more data statements, defines a user-designed character. A CHARSET statement can also select a character segment from another library character set or from a graphic character modification module.

The NAME statement specifies the ID of the character set being created, and indicates if it is to replace an existing module. More than one CHARSET statement can be coded between the INCLUDE and NAME statements; all such CHARSET statements apply to the same library character set module.

## THE LIBRARY CHARACTER SET MODULE STRUCTURE

The library character set data following the header information is a series of 73-byte segments. Each module contains 64 segments. For each segment left undefined in a library character set module, IEBIMAGE inserts the graphic symbol for an undefined character as described in the note in Figure 33 in Appendix A.


Segment 1
Segment 2
0
1
72
73
74
145
...
72 bytes of coding that represent the raster scan pattern of one graphic character and its system data.
The 6-bit code that is the WCGM location assignment for the graphic character.

A library character set is loaded directly into a WCGM. SETPRT uses the 6-bit code, contained in the first byte of each 73-byte segment, as the address of the WCGM location into which the remaining 72 bytes are loaded.

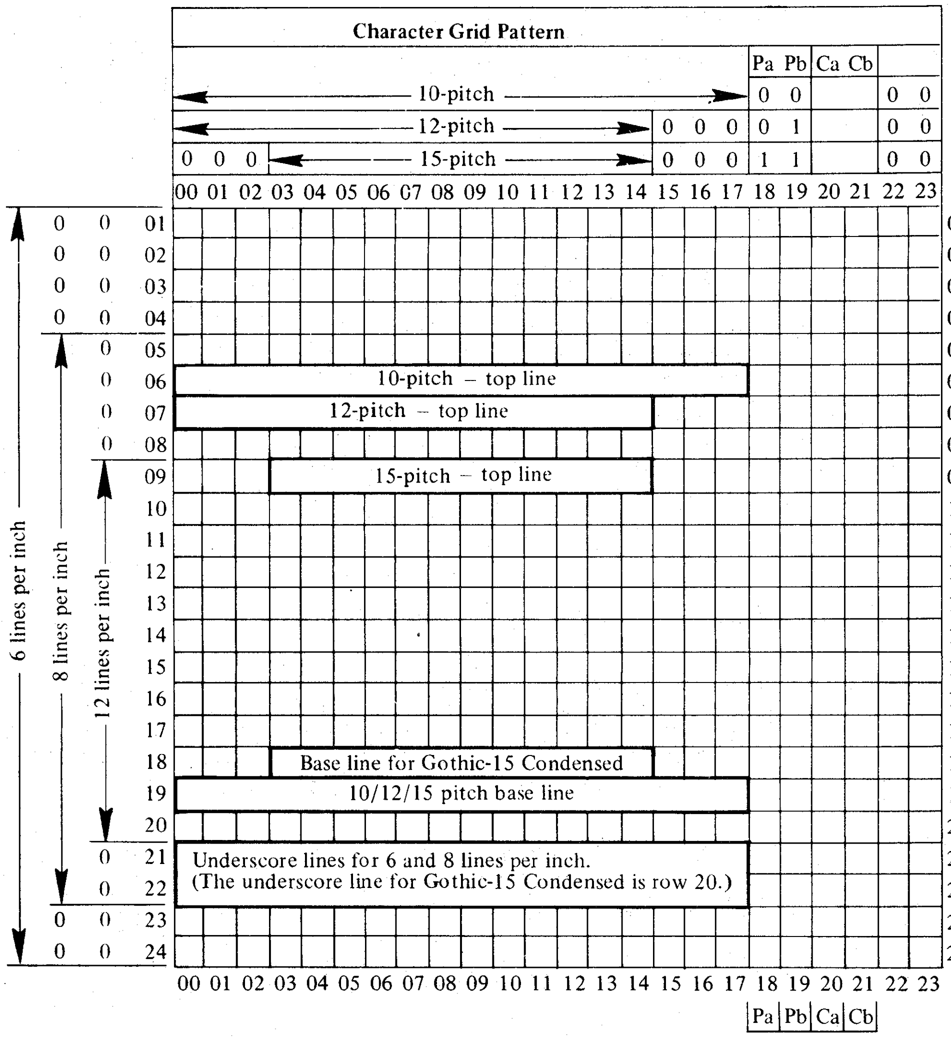
The 72-byte graphic definition that makes up the scan pattern and system data for one character is divided into twenty-four 3-byte groups. Each 3-byte group is expressed below as a horizontal row of twenty-four 1-bit elements:

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.



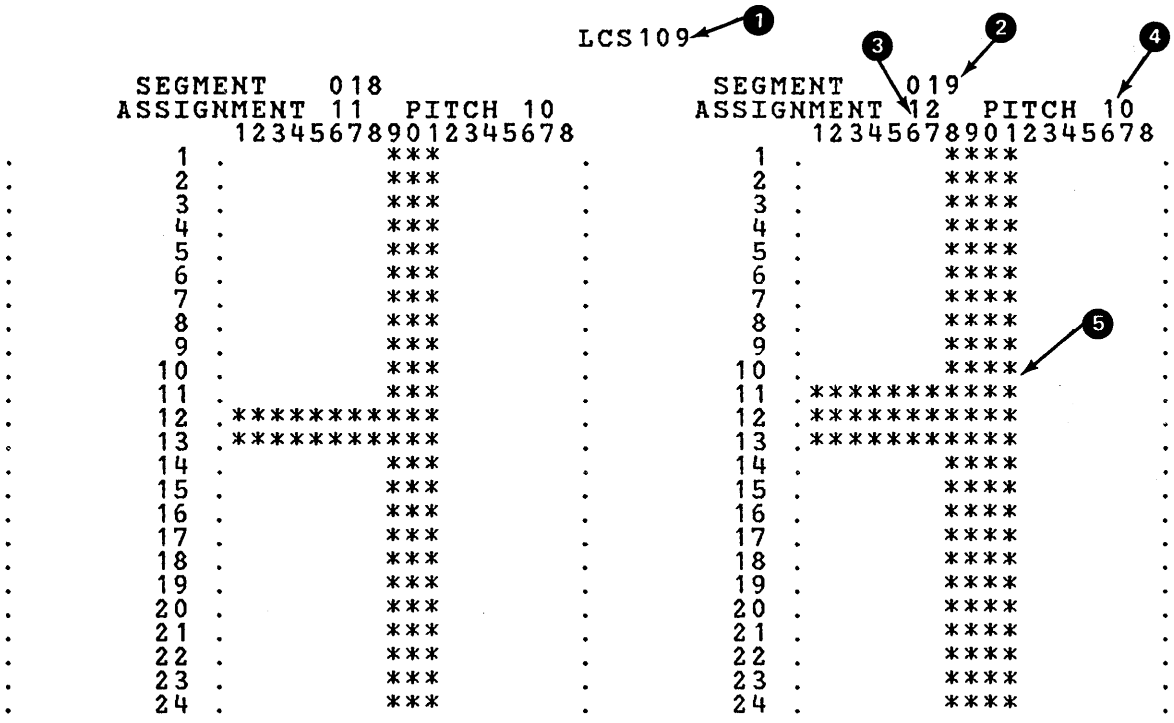
Pa & Pb = pitch (which must be constant within a character)

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

## CHARSET MODULE LISTING

Figure 25 shows an extract from a listing of a library character set module. This extract contains the listing of two segments of the library character set. The numbered notes that follow the figure describe the items marked with the circled numbers.



```
                              LCS109
       SEGMENT   018                 SEGMENT   019
      ASSIGNMENT 11  PITCH 10        ASSIGNMENT 12  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 .       ****       .
```

Figure 25.' IEBIMAGE Listing of Two Segments of a Library Character Set

**Notes to Figure 25:**

1. The name of the library character set module, including the four-byte system-assigned prefix.
2. The segment number of the character segment within the module.
3. The 6-bit code for the WCGM location.
4. The pitch of the character.
5. 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.

## CHARSET STATEMENT

The CHARSET statement specifies the contents of one or more of the character segments of a library character set module. A library character set module consists of header information followed by 64 character segments. Each character segment contains the character's 6-bit code for a WCGM location, 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, using the CHARSET statement, character segments can be selected from any module named with a library character set ID or the GCM keyword. The CHARSET statement can also specify the scan pattern and characteristics for a new character.

The CHARSET statement must always be followed by a NAME statement, another CHARSET statement, or one or more data statements. The CHARSET statement can be preceded by an INCLUDE statement. More than one CHARSET statement can be coded in the operation group. The operation group can include CHARSET statements that select characters from existing modules and CHARSET statements that create new characters. The CHARSET 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 CHARSET statement with no operands specified, followed by a NAME statement that identifies a library character set module, is used to format and print the module. The format of the CHARSET statement, when it is used to select a character segment from another module, is:

```
[label] CHARSET [REF=((segno,cloc)[,(segno,cloc)...])
                [,GCM=name|ID=xx]]
```

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

```
[label] CHARSET  ASSIGN=(cloc[,pitch|10])
        data statements
```

where

**REF=((segno,cloc)[,(segno,cloc)...])**
identifies one or more character segments within an existing graphic character modification module or library character set module. If the reference is to a GCM, the scan pattern and pitch of the referenced character are used, and a 6-bit WCGM location code is assigned. If the reference is to a character in a library character set, the entire segment, including the 6-bit WCGM location code, is used, unless the 'cloc' subparameter is specified for that segment. 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 or library character set module built with the IEBIMAGE program, _segno_ is a number from 1 to 64.

cloc
specifies a 6-bit code that points to a WCGM location, and can be any value between X'00' and X'3F'. When a library character set segment is referenced, if _cloc_

is not specified, the character's 6-bit code remains unchanged when the segment is copied. If a graphic character modification segment is referenced, cloc must be specified.

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

**GCM**=name
can be coded when the REF parameter is coded and identifies a graphic character modification module that contains the character segments referenced by the REF parameter.

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

**ID**=xx
can be coded when the REF parameter is coded and identifies a library character set that contains the character segments referenced by the REF parameter.

xx
specifies the two hexadecimal-digit ID of the library character set module.The second digit must be odd, and '7F' and 'FF' are not allowed.

**Default:**

When GCM and ID are 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 and ID must not be coded.

**ASSIGN**=(cloc[,pitch|**10**])
identifies a newly-created character and its characteristics. The ASSIGN parameter specifies the new character's 6-bit code and its pitch. The data statements that follow the CHARSET 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.

cloc
specifies the character's 6-bit code for a WCGM location, and can be any value between X'00' and X'3F'. The cloc 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 CHARSET 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 in the chapter entitled "How to Change and Create Characters." 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 through 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 USING THE CHARSET STATEMENT

### EXAMPLE 1: LISTING A LIBRARY CHARACTER SET MODULE

In this example, each segment of a library character set is printed. The scan pattern of each of the characters in the module is printed.

```
//          JOB  ...
//          EXEC PGM=IEBIMAGE
//SYSUT1    DD   DSNAME=SYS1.IMAGELIB,DISP=SHR
//SYSPRINT  DD   SYSOUT=A
//SYSIN     DD   *
    CHARSET
        NAME  83
/*
```

### EXAMPLE 2: BUILDING A LIBRARY CHARACTER SET MODULE

In this example, a library character set 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. The EBCDIC assignments for the characters are replaced by WCGM-location codes.) 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=((24,01),(25,02),(26,03),(27,04),(28,05), X
                 (31,06),(33,07),(35,08),(38,09),(40,0A))
    NAME     73
/*
```

**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 or a library character set ID, the CHARSET 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. For example, the 24th segment is to be copied and assigned the WCGM location 01. See the REF parameter (24,01).
- The name of the library character set module is 73, and it is stored as a new module in the SYS1.IMAGELIB data set.

### EXAMPLE 3: BUILDING A LIBRARY CHARACTER SET MODULE AND MODIFYING A CHARACTER ARRANGEMENT TABLE TO USE IT

In this example, a library character set module is built. The module contains one user-designed character, a reverse 'E', whose 6-bit WCGM-location code is designated as X'03', 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   *
        CHARSET  ASSIGN=(03,10)
   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  73
    INCLUDE  GS10
    TABLE    CGMID=(82,73),                             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 CHARSET statement's ASSIGN parameter establishes the 6-bit WCGM-location code, X'03', and the width, 10-pitch, for the user-designed character. The data statements that follow the CHARSET statement describe the character's scan pattern.

- The name of the library character set module is 73, 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 library character set containing the reverse E to that copy of the GS10 table.

  CGMID=(82,73) 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'73' as a character set identifier to allow loading of the second WCGM with the library character set 73.

  LOC=(E0,03,1) specifies that the reverse E, which has been assigned the WCGM location 03 in the second WCGM, is to be referenced by the EBCDIC code X'E0'.

- The new character arrangement table is named RE10 and stored as a new module in SYS1.IMAGELIB.

## EXAMPLE 4: BUILDING A LIBRARY CHARACTER SET MODULE

In this example, a library character set module is created. Its contents come from three different sources: 62 segments are copied from an existing module with the INCLUDE statement; the CHARSET statement is used to select another segment to be copied; a second CHARSET statement is used to establish characteristics for a user-designed character. The new library character set 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  33,DELSEG=(3,4)
    CHARSET  REF=(1,02),GCM=BODE,ASSIGN=03
     ********           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  53
/*
```

**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 library character set module named 33 is to be included with the new module. All segments of 33, except the third and fourth segments (as a result of DELSEG=3,4), are to be copied into the new module, and become the basis for the new module.

- The CHARSET statement specifies the module's third and fourth segments:

  REF=(1,02) and GCM=BODE specify that the third segment of the new module is to be obtained by copying the first segment from the graphic character modification module named BODE. The segment's 6-bit WCGM-location code is to be set so that its character is identified with the code X'02'.

  ASSIGN=03 specifies that the new module's fourth segment is a user-designed character whose 6-bit WCGM-location code is X'03' and whose width is 10-pitch (the default when no pitch value is specified). The CHARSET statement is followed by data statements that specify the character's scan pattern.

- The name of the library character set module is 53, and it is stored as a new module in the SYS1.IMAGELIB data set.

# APPENDIX A. IBM-SUPPLIED CHARACTER SETS

Twenty character sets are supplied with the 3800. Figure 26 lists the supplied sets. The WCGM assignments of the characters in each of the sets are shown in Figure 27 through Figure 33.

| Character set | Characters per inch (pitch) | Number of nonblank characters in the set | Character set identifier, CGMID[3] | IBM-supplied Library Character Set IDs[3,4] |
|---|---|---|---|---|
| Gothic-10 | 10 | 63 | 82 | 83 |
| Gothic-12 | 12 | 63 | 84 | 85 |
| Gothic- 15 | 15 | 63 | 86 | 87 |
| Gothic-15 Condensed[1] | 15 | 63 | 92 | 93 |
| Gothic-10 underscored | 10 | 63 | 38 | 39 |
| Gothic-12 underscored | 12 | 63 | 3A | 3B |
| Gothic-15 underscored | 15 | 63 | 3C | 3D |
| Gothic-15 Condensed underscored[1] | 15 | 63 | 36 | 37 |
| Katakana-10[2] | 10 | 64 | 1A | 1B |
| Katakana-12[2] | 12 | 64 | 1C | 1D |
| Katakana-15[2] | 15 | 64 | 1E | 1F |
| OCR-A | 10 | 52 | 16 | 17 |
| OCR-B | 10 | 54 | 18 | 19 |
| Text 1 | 10 | 63 | 8E | 8F |
| Text 2 | 10 | 63 | 10 | 11 |
| Text 1 underscored | 10 | 63 | 3E | 3F |
| Text 2 underscored | 10 | 63 | 40 | 41 |
| Format-10 | 10 | 36 | 08 | 09 |
| Format-12 | 12 | 36 | 0A | 0B |
| Format-15 | 15 | 36 | 0C | 0D |

[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 not print.

[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 the TABLE statement for an explanation of the use of 7F or FF as a character set identifier (CGMID).

[4] These IDs can be changed at the user's discretion.

Figure 26. 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 27. WCGM Assignments for Gothic and Gothic Underscored (in All Three Pitches) and Gothic-15 Condensed Characters

| WCGM<br>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 28. 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] | | |

Figure 29. WCGM Assignments for OCR-A Characters

### Notes to Figure 29:

[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.

| 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 | | |
| 3C | \| | LONG VERTICAL MARK[1] |
| 3D | ' | PRIME OR APOSTROPHE |
| 3E | = | EQUAL SIGN |
| 3F[2] | | |

Figure 30. WCGM Assignments for OCR-B Characters

**Notes to Figure 30:**

[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.

| 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 31. 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] |

Figure 32. WCGM Assignments for Text 2 and Text 2 Underscored Characters

**Note to Figure 32:**

[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 |

| 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 33. WCGM Assignments for Format Characters (in All Three Pitches)

## APPENDIX B. IBM-SUPPLIED CHARACTER ARRANGEMENT TABLES

Figure 34 lists the character arrangement tables supplied with the 3800 programming support. Figure 35 through Figure 42 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-12 | 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 34. 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 35. 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 |

Figure 36. The GF10, GF12, GF15, and GFC Folded Character Arrangement Tables

**Note to Figure 36:**

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

| 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 37. The 3211 Group and 1403 Group of Gothic Character Arrangement Tables

| | | | Character Arrangement Table | | |
|---|---|---|---|---|---|
| EBCDIC Assignment | WCGM Number[1] | WCGM Location | TN<br>T11<br>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 38. 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 39. The DUMP Character Arrangement Table

Character Arrangement Tables
Gothic and OCR-A

| EBCDIC Assignment | WCGM Location | AOA<br>OAA |  | AON<br>ONA |  | ODA |  | AOD |  | Description |
|---|---|---|---|---|---|---|---|---|---|---|
|  |  | Gothic | OCR-A | Gothic | OCR-A | Gothic | OCR-A | Gothic | OCR-A |  |
| 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 OCR-B

| EBCDIC Assignment | WCGM Location | BOA |  | BON<br>ONB |  | OAB | Description |
|---|---|---|---|---|---|---|---|
|  |  | Gothic | OCR-B | Gothic | OCR-B | OCR-B |  |
| 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 40. 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 41 (Part 1 of 2). The Katakana Group of Character Arrangement Tables

| EBCDIC | WCGM | WCGM | Character Arrangement Table | | | |
|---|---|---|---|---|---|---|
| Assignment | Number[1] | Location | 2773 | 2774 | KN1 | Description |
| 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 | ゛ | ゛ | ゛ | Voiced |
| BF | 1 | 3F | ゜ | ゜ | ゜ | 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 41 (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 42. The Format Group of Character Arrangement Tables

## APPENDIX C. WORLD TRADE NATIONAL USE GRAPHICS

The IBM-supplied graphic character modification module named 'GRAFȸȸȸȸ' 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 'GRAFȸȸȸȸ', with their EBCDIC assignments and segment numbers, are shown in Figure 43. 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 | | | EBCDIC | | |
| 10-pitch | 12-pitch | 15-pitch | 10-pitch | 12-pitch | 15-pitch | 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 43 (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 43 (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 43 (Part 3 of 4). The World Trade National Use Graphics

| Gothic-15<br>Condensed<br>Characters<br>Segment<br>Number | Gothic-15<br>Condensed<br>Under-scored<br>Characters<br>Segment<br>Number | EBCDIC<br>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 43 (Part 4 of 4). The World Trade National Use Graphics

# APPENDIX D. PAPER SIZES, WEIGHTS, AND MAXIMUM CHARACTERS PER LINE

Figure 44 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 45 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 44. Common-Use Paper Sizes Usable in the 3800

| Widths in millimeters (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 45. ISO Paper Sizes Usable in the 3800 (Not Available in the United States and Canada)

Figure 46 and Figure 47, 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 46. Maximum Characters per Line on Common-Use Paper Sizes

| Paper width in millimeters | 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 47. 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 Immdediately | 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 |

CHANNEL COMMANDS, Continued

| Type Command | Command Name | Hex Code |
|---|---|---|
| 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 | 17 |
| | Execute Order | 33 |

## 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** | **Interven-tion 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) | |

## 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**[1]**:** 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**[1]**:** 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:** In the 3800 Printing Subsystem, a module 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 a set of a maximum of 64 graphic characters, all of one size and style.

**character set identifier:** A 1-byte code identifying a particular character set within the 3800 Printing Subsystem. Same as character generation module identifier (CGMID).

**condensed:** Character sets having a face that is smaller than that of a set not so characterized.

**control character**[1]**:** 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 group:** If multiple copies of a printed data set are produced by the 3800 printer so that a printed page and its copies are contiguous, the page and its copies are called a copy group.

**copy modification:** A feature available in the 3800 Printing Subsystem that allows printing of predefined data on each page of specified copies of a data set.

**copy modification module:** A program module that can be used for copy modification by specifying the name of the module with the MODIFY keyword.

**data code:** A byte of data that represents a graphic character. This data code can be a member of a coded character set (usually EBCDIC) or a user-assigned code.

**EBCDIC:** Extended binary-coded decimal interchange code.

**EXCP:** Execute channel program.

**Execute Order CCW:** A 3800 CCW that has order codes for displaying status codes, for purging the page buffer, and for requesting information about the status of the printer.

**FCB:** Forms control buffer.

**folding:** A technique used with the Universal Character Set (UCS) feature on an impact printer that allows two or more of the 256 possible 8-bit character codes to represent the same graphic character on a chain or train. The first 2 bits of each 8-bit code are ignored. For example, it can be used to allow uppercase graphic characters to be printed if the 8-bit codes for lowercase characters are specified when lowercase graphics are not available in the character array on the chain or train.

[1] American National Standard definition

**Format character sets:** 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 that is used to store information for controlling the vertical format of printed output; it is analogous to the punched paper carriage control tape used on IBM 1403 Printers.

**forms control buffer module:** A program module that is loaded into the forms control buffer when specified with the FCB keyword.

**forms overlay:** A feature of the 3800 printer that allows the printing of a form, grid, design, or other constant data from an overlay negative at the same time as the variable data is being printed. Synonymous with forms flash.

**fuser:** The unit that fuses the toned image into the paper.

**Gothic character sets:** Character sets (available in 10-, 12-, and 15-pitch) with 63 sans serif graphic characters.

**graphic**[1]**:** A symbol produced by a process such as handwriting, drawing, or printing.

**graphic character**[1]**:** A character, other than a control character, that is normally represented by a graphic.

**graphic character modification:** A feature available in the 3800 Printing Subsystem that allows the substitution or addition of graphic characters into an already-defined character arrangement.

**graphic character modification module:** A program module that can be used for graphic character modification when it is named in a character arrangement table.

**impact printer:** A printer in which printing is the result of mechanical impact.

**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.

**JES:** Job entry subsystem.

**job entry subsystem (JES):** A system facility for spooling, job queuing, and managing I/O.

**Katakana character sets:** Sets of symbols used in one of the two common Japanese phonetic alphabets.

**library character set:** A 3800 character set that is stored in SYS1.IMAGELIB or a user-defined library, rather than on the flexible disk.

**line overrun:** An indication that copy modification was not completed in time for printing the line.

**OCR**[1]**:** Optical character recognition.

**OCR-A, OCR-B:** Character sets that are designed for optical character recognition use.

**optical character recognition (OCR)**[1]**:** The machine identification of printed characters through use of light-sensitive devices.

**pitch:** A unit of type width that is based on the number of copies of a printed character that can fit into one linear inch. For example, 10-pitch type can have 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**[1]**:** A control character for print operations such as line spacing, page ejection, or carriage return.

**table reference character (trc):** An 8-bit byte that specifies which character arrangement table is to be used for printing the current line.

**Text character sets:** Character sets of upper- and lowercase graphic characters with serifs.

**toner:** A thermoplastic material impregnated with lampblack. Toner adheres to the exposed areas of the photoconductor on the drum during the developing process and is then transferred to the paper to form the developed image on the paper.

**transfer station:** The location at which the toned image on the photoconductor, which is on the drum, is transferred to the paper.

**translate table:** That 256-byte portion of the character arrangement table that translates the data code for a character into the code required by the 3800 printer.

**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 printer's character generation storage that holds the scan patterns of a maximum of one set of characters to be used for printing. There are two WCGMs in the basic 3800, and optional additional storage provides two more.

# INDEX

# Technical Newsletter

| | |
|---|---|
| This Newsletter No. | GN26-0964 |
| Date | 25 January 1980 |
| Base Publication No. | GC26-3846-3 |
| File No. | S370-30 |
| Prerequisite Newsletters | None |

**IBM 3800 Printing Subsystem Programmer's Guide**

This technical newsletter, a part of Release 3.8 of OS/VS2 MVS and Release 7 of OS/VS1, provides replacement pages for the subject publication. These replacement pages remain in effect for any subsequent releases unless specifically altered. Pages to be inserted and/or removed are:

Title page, edition notice
29, 30
189–191

## Summary of Amendments

This technical newsletter incorporates a technical addition and editorial changes.

**Note**: Please file this cover letter at the back of the publication to provide a record of changes.

IBM Corporation, Dept. 61C, General Products Division, P. O. Box 27155, Tucson, Arizona 85726

Printed in U.S.A.

IBM 3800 Printing Subsystem
Programmer's Guide
GC26-3846-3

**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 may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

**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.

## Reader's Comment Form

Fold and tape Please do not staple 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
General Products Division
P. O. Box 27155
Tucson, Arizona 85726
Attn: Dept. 61C

Fold and tape Please do not staple 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

GC26-3846-3

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