File No. S360-30
Form C27-6912-2

OS

Systems Reference Library

IBM System/360 Operating System Basic Graphic Programming Services for IBM 2260 Display Station (Local Attachment) Preliminary Specifications

This publication describes programming services for using the IBM 2260 Display Station with the System/360 Operating System. Included are descriptions of macro-instructions for input/output control and for Basic Attention Handling, along with some general requirements for using the display station effectively.

Although intended primarily for the programmer coding in System/360 assembler language, portions of this publication are also of interest to installation managers, system engineers, system programmers, and others seeking general information about programming the IBM 2260 Display Station.

PREFACE

This publication provides the reader with instructions and guidance in writing applications programs for the IBM 2260 Display Station (local attachment) under overall control of the IBM System/360 Operating System. The macro-instructions and control program routines supplied are described in detail. Coding examples with accompanying explanations are included to illustrate uses of the macro-instructions and other modules provided.

To take full advantage of the operating system and the programming services provided specifically for the IBM 2260, the reader should have completed a basic course in System/360 assembler language coding, or possess equivalent knowledge, and be familiar with the content of the following publications:

<u>IBM System/360 Component Description; IBM 2260 Display Station, IBM 2848 Display Control</u>, Form A27-2700
<u>IBM System/360 Operating System; Concepts and Facilities</u>, Form C28-6535
<u>IBM System/360 Operating System; Assembler Language</u>, Form C28-6514
<u>IBM System/360 Operating System; Control Program Services</u>, Form C28-6541
<u>IBM System/360 Operating System; Job Control Language</u>, Form C28-6539
<u>IBM System/360 Operating System; Linkage Editor</u>, Form C28-6538

<u>Third Edition</u> (September 1966)

This is a major revision of, and obsoletes, <u>IBM System/360 Operating System; Graphic Programming Services for IBM 2260 Display Station (Local Attachment); Preliminary Specifications</u>, Form C27-6912-1. Significant changes have been made throughout the present edition. Express attention handling is not discussed in this publication, but is described in <u>IBM System/360 Operating System; Express Graphic Programming Services for IBM 2260 Display Station (Local Attachment)</u>, Form C27-6925.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.

A form for reader's comments appears at the back of this publication. Address any additional comments concerning the contents of this publication to: IBM Corporation, Programming Publications, Department 637, Neighborhood Road, Kingston, New York 12401

The programming support described in this publication may be used with one or more IBM 2260 Display Stations connected to IBM System/360. Each display station is associated with an IBM 2848 Display Control, which is connected directly to the processing unit via either a multiplexor channel or a selector channel.

In this configuration, which is referred to as the "IBM 2260 Local," the display station may be located anywhere within a 2000-foot cable distance from the display control.

Through use of the IBM 2701 Data Adaptor Unit and appropriate common carrier facilities, display stations may be located any distance from the central processing unit. This configuration, which is referred to as the "IBM 2260 Remote," is supported under the Basic Telecommunications Access Method (BTAM).

CONFIGURATIONS AND FEATURES

The display station is available in three configurations, consisting of a cathode ray tube (CRT) display with or without an alphameric or numeric keyboard. The optional keyboard enables man-machine communication on a continuing basis at the display station. Other special features available with the IBM 2260 Local are Line Addressing and Nondestructive Cursor.

The IBM 2848 Display Control contains the interface control, a character generator, and buffer storage for the display stations. This buffer storage retains the video data to maintain display regeneration on the CRT. The data formats and number of characters that may be displayed on the face of the CRT, as well as the number of display stations that may be attached to a single display control, depend upon which of the three models of the display control is used, as shown in Table 1.

In addition, one IBM 1053 Printer Adapter Unit may be attached to each display control. This feature allows use of an IBM 1053 Printer at a central location to document and record periodic system transactions. Like the display stations, the 1053 Printer may be located up to 2000 feet from the display control.

The programming services described in this publication support all of the optional features of both the IBM 2260 and the IBM 2848 described in detail in the component description publication, Form A27-2700.

PROGRAMMING SUPPORT

Operating system programming support for the IBM 2260 Display Station includes macro-instructions and control routines. They are used for two key functions: input/output control and attention handling. The macro-instructions function as part of the problem program; the control routines function as part of the control program.

The following paragraphs contain brief descriptions of the graphic programming services that are explained in detail in later sections. Appendixes include a sample program and CCW formats.

Table 1. Display Control Characteristics

IBM 2848 Display Control			
Characteristic	Model 1	Model 2	Model 3
Max. No. of Display Stations	24	16	8
No. of Lines of Characters	6	12	12
No. of Characters Per Line	40	40	80
Total No. of Characters	240	480	960

INPUT/OUTPUT CONTROL

Input/output control macro-instructions are used for data transmission and control functions, for creating data control blocks (DCBs), and for establishing and terminating system communication between problem programs and display stations. Input/output control routines create channel command words (CCWs), issue supervisor calls to execute channel programs, and control data transmission between main storage and display station buffer storage.

BASIC ATTENTION HANDLING

Attentions are asynchronous (unscheduled) input/output interruptions of the CPU. They are caused by alphameric or numeric keyboard entries from the 2260 Display Station.

Basic attention handling facilities automatically detect the occurrence of attentions, and route them to appropriate user-written attention handling routines. They also provide for queuing attentions that occur within a short time interval, and for overlapping wait intervals for expected attentions with selective processing. Macro-instructions define the user's attention handling routines to the control program.

ERROR HANDLING

Synchronous errors (i.e., input/output errors that occur during channel operations) are handled by the IBM 2260 Standard Error routine. This module analyzes the status and sense bytes for synchronous error conditions and provides appropriate error-recovery procedures for specific error situations.

The IBM 2260 Standard Error routine provides a message to the operator whenever operator intervention is required, or whenever a Bus Out Check or Equipment Check occurs. These messages appear in the publication IBM System/360 Operating System; Operator's Guide, Form C28-6540.

SYSTEM GENERATION

The IBM 2260 user defines his graphic configuration to the operating system he is generating through use of appropriate SYSGEN macro-instructions. System generation requirements are described in the publication IBM System/360 Operating System; System Generation, Form C28-6554.

CONVENTIONS USED

An understanding of the following conventions is helpful in using the information supplied in this publication.

FORMAT ILLUSTRATION CONVENTIONS

Each macro-instruction discussed in this publication is illustrated by a coding format. The coding format illustrations use the following conventions:

- Upper case (capital) letters, numbers, and punctuation marks must be coded by the programmer exactly as shown. Exceptions to this convention are brackets, [ ], braces { }, and ellipses, ... These three exceptions are never coded.

- Lower case (small) letters and words represent variables for which specific information or specific values must be substituted by the programmer when coding.

- Items or groups of items within brackets, [ ], are optional. They may be omitted at the programmer's discretion.

- Braces, { }, indicate that one operand from the group must be chosen unless a default option is indicated.

- Underscoring indicates a default option; that is, if no operand is coded, the underscored operand is assumed.

- The vertical stroke (|) signifies exclusive "or." For example, A|B means that the programmer may write

A or B. Alternative operands are also indicated, in some cases, by vertical grouping within braces, { }.

- An ellipsis, ... , denotes that the preceding syntactical unit (which is enclosed in braces or brackets) can be written any number of times.
- Operands are separated by commas.
- All operands are written on one line of a coding sheet (except when a continuation card is necessary) even though they may appear on two or more lines in the format illustration.
- The last operand is always followed by a blank.

CODING CONVENTIONS

Positional operands must be coded in the order shown in the appropriate format illustration. Keyword operands may be written in any order, but must follow all positional operands. Commas must appear in place of omitted operands preceding the last-specified positional operand.

Where value mnemonics are used to describe macro-instruction operands, the mnemonics used are those defined in Section 1 of _IBM System/360 Operating System; Control Program Services_, Form C28-6541. Whenever the L-form of a macro-instruction is specified (MF=L), no indexing or register notation is permitted for its operands; only relocatable expressions, absolute expressions, or codes may be used.

Unless noted otherwise, where numeric values are indicated as macro-instruction operands, these are decimal integers and need not be preceded by leading zeros.

Unless otherwise specified, omission of the macro-form (MF) operand causes a macro-instruction to perform both L and E functions, i.e., to derive a parameter list from all the operands specified and, at execution time, to execute the macro-instruction using that list.

LINKAGE CONVENTIONS

Linkage conventions described in _IBM System/360 Operating System; Control Program Services_ apply to the programming support described here. Information about register assignments for specific linkage functions can be found in that publication.

TERMINOLOGY

In this publication, "display station buffer storage" refers to the buffer storage physically located in the display control unit. Throughout this publication, the unqualified term "register" means "general register."

A glossary is included at the end of this publication to define certain terms that are peculiar to graphic-device programming and that may be unfamiliar to the reader. Those terms that are found both in this and in other operating system publications are defined in the glossary contained in _IBM System/360 Operating System; Concepts and Facilities_.

INPUT/OUTPUT CONTROL

Input/output control is achieved by macro-instructions and associated control routines which together initiate input/output operations associated with data display. Input/output macro-instructions for the IBM 2260 Display Station are:

Macro-Instruction	Function
DCB	Establishes the format and reserves space for the data control block.
OPEN	Opens the DCBs for the display stations by setting up necessary control blocks and informing the operating system that the problem program will use the display stations.
GREAD	Reads manual input character data from display station buffer storage to main storage; or, Reads character data from display station buffer storage to main storage.
GWRITE	Writes character data into display station buffer storage; or, Erases the display station buffer storage and then writes; or, Writes into the display station buffer at the first display position of a specified line; or, Erases the display station buffer storage, then writes into the display station buffer at the first display position of a specified line.
GCNTRL	Erases display station buffer storage.
CLOSE	Resets the fields initialized by OPEN and informs the operating system that the problem program no longer needs the display stations.

Note: The address of the problem program save area must be loaded into register 13 before any input/output macro-instruction is issued.

FUNCTIONAL RELATIONSHIPS

To help the reader better understand subsequent discussion of input/output operations, the following paragraphs briefly describe the functional relationships among the various system control blocks associated with 2260 input/output operations under the operating system. Additional information may be found in the "Execute Channel Program" section of the IBM System/360 Operating System; System Programmer's Guide, Form C28-6550.

Each data control block (DCB) normally is established by a DCB macro-instruction and completed by execution of an OPEN macro-instruction. The DCB, which contains a reference to a DD statement (see below), may be associated with one display station or with a group of display stations connected to a single 2848 display control. More than one DCB may be completed by a single OPEN macro-instruction. The OPEN macro-instruction creates a data extent block (DEB) in protected core storage. Included in the DEB is a table of address pointers to specific unit control blocks (UCBs) associated with the data control block. One UCB is specified for each display station and for each 1053 Printer attached to a 2848 Display Control. The UCB is used to describe its associated device to the control program.

Some input/output macro-instructions contain a "unit" operand. The "unit" operand refers to a device within a group of devices which can be named at system generation time. This operand is used as an index to reference the desired element in the DEB list of UCB pointers. The value of the "unit" operand, expressed as a decimal integer, must not exceed the number of devices specified on the Data Definition (DD) card for the data set.

A DD statement can refer to any display station or 1053 Printer singularly, or to any group of display stations (up to 24) along with a 1053 Printer connected to the same display control. All units of a group need not

Table 2. Format of the DECB

Offset from DECB Name (bytes)	DECB Field	Associated I/O Macro-Instruction Operand
+00	Event control block (ECB) address	
+04	Type code	type
+08	Data control block (DCB) address	dcb
+12	Area address	area
+16	Code for incorrect length or permanent read error	
+17	Residual count from CSW if incorrect length	
+20	Length (byte count)	length
+24	(Reserved)	
+28	Index to reference an element in the DEB table of address pointers to UCBs associated with the DCB	unit
+29	Zeros	
+30	Zeros	
+31	Zeros	

be specified in one DD statement; subsequent DD statements can refer to the unspecified units in that group. Information about data definition specifications can be found in IBM System/360 Operating System; Job Control Language.

The value of the "unit" operand decremented by 1 is placed in the data event control block (DECB) when it is generated at macro-expansion time ("unit" - 1 = UCB index). At execution time, the Input/Output Control routine places this value into the input/output block (IOB). This completes the linkage between the input/output macro-instruction and the selected UCB, which in turn points to the device associated with the macro-instruction.

INPUT/OUTPUT MACRO-INSTRUCTIONS

The macro-instructions in this group implement 2260 input/output functions. They provide linkage to the Input/Output Control routine which creates the IOB, constructs channel programs, and issues calling sequences to execute the channel programs.

All input/output macro-instructions except the DCB macro-instruction may use the L- and E-forms of macro-definition. If all operands except the MF=L or MF=E keyword operand are written, the input/output function is executed using the specified operands. If any required operand is missing at execution time, the task is terminated. Detailed descriptions of these forms and their use are found in IBM System/360 Operating System; Control Program Services.

A 32-byte DECB is formed at macro-expansion time and the parameters derived from the macro-instruction operands are inserted into appropriate positions in the DECB. This data is utilized by the Input/Output Control routine to generate the CCWs. The format of the DECB for the IBM 2260 Local is shown in Table 2.

Note: If an input/output operation depends upon the completion of a previous input/output operation, a WAIT macro-instruction should be issued following the macro-instruction initiating that previous input/output operation.

DCB--Data Control Block Macro-Instruction

The DCB macro-instruction for the IBM 2260 Local reserves space for the data control block and establishes its format (see Table 3). In addition to the basic information required in the DCB, space is also reserved for addresses of the Input/Output Control routine and the input/output block, which are supplied at open time. One DCB services a maximum of 24 display stations and one 1053 Printer attached to the same display control. The format of the DCB macro-instruction is:

Name	Operation	Operand
[symbol]	DCB	DSORG=GS,MACRF=code ,GTYPE=BASIC [,DDNAME=symbol] [,EXLST=relexp] [,GNCP=absexp]

DSORG=GS
Specifies that a graphic data control block is defined. This operand is required for macro-expansion.

MACRF
Specifies the type or types of macro-instructions to be used, where R indicates GREAD, W indicates GWRITE, and C indicates GCNTRL. The following are the eight valid forms of the MACRF operand.

R
W
RC
WC
(R,W)
(RC,W)
(R,WC)
(RC,WC)

GTYPE=BASIC
Specifies that basic attention handling procedures are to be employed.

DDNAME
Specifies the name of the DD statement describing the data set being processed. DDNAME corresponds exactly to the contents of the Name field of the DD statement provided in the job control language. It can be defined by as many as eight characters, the first of which must be a letter. DDNAME furnishes a logical connection between the DD statement and the data control block of the user's problem program.

EXLST
Specifies the address of the user's exit list in the problem program as defined in IBM System/360 Operating System; Control Program Services. If EXLST is not specified, a value of zero is assumed.

GNCP
Specifies the maximum number of GREAD or GWRITE macro-instructions that can be issued before a WAIT macro-instruction (i.e., the number

Table 3. Data Control Block Elements

Offset from DCB name (bytes)	DCB Field
+0-13	Reserved
+14	GTYPE--Specifies basic attention handling procedures
+15-25	Reserved
+26-27	DSORG--Specifies a graphic data control block
+28-31	Reserved
+32	GNCP--Maximum number of I/O macro-instructions issued before a WAIT macro-instruction is issued
+33-35	Reserved
+36	Reserved
+37-39	EXLST--Address of user's exit list for exceptional conditions
+40-47	DDNAME--Corresponds to name of data definition (DD) statement
+48-49	Reserved
+50-51	MACRF--Types of macro-instructions used in problem program

of IOBs constructed at open time). The value of GNCP must be from 1 to 99 at execution time. If the value of GNCP is not specified, a value of 1 is assumed. The programmer may specify a value of 0 at assembly time, but must respecify the value within the above range at execution time. If the value 0 is present at open time, the job is abnormally terminated.

PROGRAMMING NOTES: The user may create a DCB at assembly time by use of the DCB macro-instruction. Data Definition (DD) card information may be supplied by the user at execution time. The GNCP operand may be specified at execution time by a DD card, provided GNCP is set to zero at assembly time.

The programmer can create his own DCB by using a constant area. In this case he may specify both the DSORG and GNCP operands in a DD card. By ensuring that this area contains the same information at the same locations as the DCB created by the DCB macro-expansion (see Table 3), the user can effectively create a DCB. The DDname of the DD card must be identical to the DDname set up in this area (offset 40 through 47). For information on coding of the DD card, refer to the publication IBM System/360 Operating System; Job Control Language.

OPEN--Open Data Control Block Macro-Instruction

The OPEN macro-instruction for the IBM 2260 Local initializes one or more data control blocks.

Name	Operation	Operand
[symbol]	OPEN	(dcb-addr [,,dcb-addr]...) [,MF={L (E,listname) (E,(1))}]

dcb
Specifies the addresses of the DCBs to be initialized for the desired graphic device(s). Any number of DCB addresses may be specified. This allows parallel opening of DCBs and their associated data sets. If more than one DCB address is specified, the "dcb" operands must be separated by two commas. The second comma indicates omission of optional operands not applicable to graphic programming services.

MF=L
Specifies that expansion of the macro-instruction is to consist of a parameter list only. The name assigned to the parameter list is the symbolic name specified in the Name field of this OPEN macro-instruction.

MF=(E,listname)
Specifies that the macro-instruction is to perform the open function using a parameter list created by another OPEN macro-instruction. The open function is executed for each "dcb" operand in the list. The "listname" operand must correspond exactly to the name specified in the Name field of the OPEN macro-instruction used to construct the parameter list.

MF=(E,(1))
Performs the same function as (E,listname). By writing the special register notation (1), the programmer specifies that he has loaded the address of the parameter list into register 1 before execution of the macro-instruction.

The OPEN macro-instruction fulfills the following functions:

- Indicates that the device is "open" by setting the appropriate bit in the DCB.
- Fills in and initializes appropriate fields in the DCB.
- Acquires main storage to initialize a chain of IOBs and associated channel programs. Linkage is provided in the IOB to its CCW list and to the next IOB in the chain. The number of IOBs constructed is one greater than the number specified by the GNCP operand. The additional IOB is constructed in order to always ensure the availability of an IOB, since IOBs are not constructed dynamically. A pointer to the top of the chain of IOBs is placed in the DCB.

- Loads the graphic I/O Control routine and places its address in the DCB.

- Performs initialization functions required by attention-handling facilities.

EXAMPLES: In the second and third examples, DCB1 and DCB2 are initialized using the parameter list constructed by the first example. In the third example, the location of the parameter list has been loaded into register 1.

```
EXAMP1    OPEN     (DCB1,,DCB2),MF=L
EXAMP2    OPEN     MF=(E,EXAMP1)
EXAMP3    OPEN     MF=(E,(1))
```

PROGRAMMING NOTES: If the operand specifies a DCB that is already open, no action results. If the operand does not specify the address of a valid DCB when opening, the job is abnormally terminated. Opening when a corresponding DD statement has not been provided causes no error indication; however, an attempt to use the data set will result in an abnormal termination of the task.

At execution time the OPEN routine checks the GNCP operand of the DCB macro-instruction. If the value of GNCP is zero or greater than 99 the task is abnormally terminated.

If the device specified on the DD card is not a graphic device, an invalid UCB results, and the task is abnormally terminated.

GREAD--Read Character Data Macro-Instruction

The GREAD macro-instruction performs one of three functions, depending upon the "type" operand specified. The commands generated and their functions are explained in the following paragraphs.

Name	Operation	Operand
[symbol]	GREAD	decb-symbol ,type-{DSM|SMI|DSB} ,dcb-addr [,length-integer] ,area-addr [,unit-integer] [,MF={L|E}]

decb

Is the name assigned to the data event control block generated in the macro-expansion. If this operand is missing, macro-expansion does not occur.

type

Specifies one of the following read operations:

DSM

Read Display Station Manual Input Message command for the 2260.

SMI

Read Display Station Short Manual Input Message command for the 2260.

DSB

Read Full Display Station Buffer Storage command for the 2260.

dcb

Is the address of the data control block associated with the graphic device.

length

Is the number of data bytes to be read.

area

Is the address of the main storage location into which the first byte of data will be read.

unit

Is a decimal integer; it is used to create an index to the desired element in the DEB table (address pointers to UCBs associated with the DCB). The indicated UCB contains the address of the desired device. The value of "unit" must be in the range 1 to N, where N is the number of devices specified on the Data Definition card for this data set. N cannot exceed 25 (the maximum configuration is 24 display stations and one 1053 Printer per display control).

Note: The device indicated by "unit" cannot be the 1053 Printer.

Omission of the "unit" operand results in the value 1 being assigned to "unit", indicating the first element in the DEB table.

MF=L

Specifies that expansion of the macro-instruction is to consist of a parameter list only. The name assigned to the parameter list is the name specified by the "decb" operand of this GREAD L-form macro-instruction.

MF=E

Specifies that the macro-instruction is to perform the read function using a parameter list constructed by another GREAD macro-instruction. The first operand in the E-form macro-instruction must correspond exactly to the name of the parameter list. The parameter list may be updated at execution time by specifying new values in any of the operands preceding the MF=E keyword operand.

All operands required for execution that were not specified in an L-form macro-instruction should be specified in the corresponding E-form macro-instruction.

Note: A unique name must be assigned to each parameter list defined. Use of the same DECB name for more than one DECB results in a multiple definition error at assembly time.

If neither the MF=L nor MF=E keyword operand is written, the "decb" and "dcb" operands must be specified to accomplish macro-expansion. Omissions of other operands produce error messages on the assembly listing, but do not terminate macro-expansion.

TYPES OF READ OPERATIONS: The three types of read operations for the 2260 are described below.

DSM (Read Display Station Manual Input Message): This command causes character data in the selected display station buffer storage to be processed and transferred to main storage by the display control. DSM can be used only if either optional keyboard feature is available. Starting with the first display position following the Start symbol, character data will be sequentially retrieved and transferred until the end of message is detected or the channel byte count is reduced to zero. The Start symbol is then cleared. Data on the same line and to the right of a New Line symbol is not transferred. If the Check symbol is present, it is transferred. The end of message code is not transferred to main storage, and the CRT data display is not changed or destroyed by the read operation.

One channel command word is generated by GREAD, "type" DSM: the Read Display Station Manual Input Message command, which initiates the read operation.

SMI (Read Display Station Short Manual Input Message): This command is identical to the DSM command, except that the Start symbol is not erased when the end of message is detected or the channel byte count is reduced to zero. Consequently, keyboard lockout time is substantially less for the SMI command than for the DSM command. This is an important consideration for applications involving a high rate of data input and multiple 2260s attached to one 2848.

One channel command is generated by GREAD, "type" SMI: the Read Display Station Short Manual Input Message command, which initiates the read operation.

DSB (Read Full Display Station Buffer): This command transfers a maximum of 240, 480, or 960 data bytes, depending upon the size of the addressed display station buffer. The first symbol transferred is the first symbol in the upper left-hand corner of the CRT, and the last symbol transferred is the last symbol in the lower right-hand corner of the CRT. If no character is found in a displayable position, a space is transferred. If the addressed buffer has the optional non-destructive cursor feature, the cursor position is not indicated by the data transferred. If the addressed buffer has a standard destructive cursor feature, the cursor position is indicated by the location of the end of message code. The Check and Start symbols are transferred if they are present. The codes are:

bit	0	1	2	3	4	5	6	7
Destructive cursor	0	1	1	0	1	0	1	0
Check symbol	0	1	1	1	1	1	1	1
Start symbol	0	1	0	0	1	0	1	0

The CRT is erased after the operation, and the cursor is placed in the first cursor position (upper left-hand corner) of the CRT.

Multiple channel command words are generated by GREAD, "type", DSB. First, the Read Full Display Station Buffer command is issued, which initiates the read operation. Second, the Erase Display Station Buffer Storage command is issued, which deletes all character data from the display station buffer storage and resets the cursor to the first display position of the CRT.

PROGRAMMING NOTES: If the channel byte count in the CCW reaches zero, a Stop sequence is generated by the channel. During a read operation, receipt of the Stop sequence terminates transmission of data from the display control to main storage. The display control continues reading data from the display station buffer storage but does not transfer data to main storage. Detection of the end of message when using DSM causes the display control to restore the keyboard of the selected display station and delete the Start symbol. When SMI is used, the keyboard is restored and the Start symbol is not erased. When DSB is used, the display is erased at completion of the command, and the cursor is reset. If an error is detected during execution of the DSB command, the erase operation is not performed. The problem programmer, therefore, should ensure that the correct length is specified if he wants the erase operation to be performed.

An unspecified "length" operand or zero entry results in the insertion of a byte count of 960 in the generated CCW. This assures that the Stop sequence will not be encountered on read operations. If the "length" operand is incorrectly specified, an error message is printed on the assembly listing.

If incorrect length occurs during execution, the count from the CSW is placed in its appropriate location in the DECB (DECB+17), and a hexadecimal code of 43 is posted in the DECB (DECB+16). Since this case is not considered an error, a hexadecimal code of 7F is also posted in the high-order byte of the ECB. If a permanent read error occurs, a hexadecimal code of 45 is posted in the DECB (DECB+16). If the read error occurs during a Read Display Station Manual Input Message (DSM) operation, the screen is erased and the message ERR, RE-ENTER MSG is displayed. The 2260 operator may retype and reenter the message. Should the error occur a second time, a message is typed for the system operator, and the display station should be considered inoperative.

If the data control block address is not the address of a valid DCB, or if a required operand is missing, the task is abnormally terminated.

EXAMPLES: In the following example, a DECB named DECB1 is to be generated as part of the macro-expansion. The "type" operand specifies that a Read Display Station Manual Input Message command for the data set associated with the data control block INDCB will be performed. A length of 100 bytes will be read into an area in main storage called INAREA. The value of the "unit" operand is 1. In example 1 L-form is specified, which generates the parameter list. Example 2 shows coding when positional operands are omitted in an L-form GREAD. Example 3 shows the E-form GREAD with the length changed to 150 bytes.

```
EXAMP1 GREAD      DECB1,DSM,INDCB,100,
                  INAREA,1,MF=L
EXAMP2 GREAD      DECB1,MF=L
EXAMP3 GREAD      DECB1,,,150,MF=E
```

Register Notation: The programmer may specify register notation, in which case the address of the DECB must be loaded into register 1 before execution of the macro-instruction. Other positional operands may be loaded into other registers (2-12). For example:

```
EXAMP1 LA         1,DECB
       GREAD      (1),MF=E
or

EXAMP2 LA         1,DECB1
       LA         2,INDCB
       LA         3,INAREA
       GREAD      (1),,(2),,(3),MF=E
```

GWRITE--Graphic Write Macro-Instruction

The GWRITE macro-instruction is used to perform any of four functions depending upon the "type" operand specified. The commands generated and their functions are explained in the following paragraphs.

Name	Operation	Operand
[symbol]	GWRITE	decb-symbol ,type-{DSB|EBW| LNE|EWL} ,dcb-addr ,[length-integer] ,area-addr [,unit-integer] [,MF={L|E}]

decb

Is the name assigned to the data event control block generated in the macro-expansion. This operand is required for macro-expansion.

type

Specifies one of the following write operations:

DSB

Write Display Station Buffer Storage command for the 2260, or Write 1053 Printer Buffer Storage command for the 1053.

EBW

Erase and Write Display Station Storage command for the 2260.

LNE

Write Display Station Line Address command for the 2260.

EWL

Erase and Write Display Station Line Address command for the 2260.

dcb

Is the address of the data control block associated with a graphic device.

length

Is the number of bytes of display data to be written. This includes the New Line character and the Line Address control byte when used.

area

Is the address of the main storage location from which the first byte of display data is to be written.

unit

Is a decimal integer; it is used to create an index to the desired element in the DEB table (address pointers to UCBs associated with the DCB). The indicated UCB contains the address of the desired device. The value of "unit" must be in the range 1 to N, where N is the number of devices specified on the Data Definition card for this data set. N cannot exceed 25 (the maximum configuration is 24 display stations and one 1053 Printer per display control).

Note: The 1053 Printer can be indicated by "unit" only when the "type" operand is written as DSB.

Omission of the "unit" operand results in the value 1 being assigned to "unit", indicating the first element in the DEB table.

MF=L

Specifies that expansion of the macro-instruction is to consist of a parameter list only. The name assigned to the parameter list is specified by the "decb" operand of this GWRITE L-form macro-instruction.

MF=E

Specifies that the macro-instruction is to perform the write function using a parameter list constructed by another GWRITE macro-instruction. The first operand in the E-form macro-instruction must correspond exactly to the name of the parameter list. The parameter list may be updated at execution time by specifying new values in any of the operands preceding the MF=E operand.

All operands required for execution that were not specified in an L-form macro-instruction should be specified in the corresponding E-form macro-instruction.

Note: A unique name must be assigned to each parameter list defined. Use of the same DECB name for more than one DECB results in a multiple definition error at assembly time.

If neither the MF=L nor MF=E keyword operand is written, the "decb" and "dcb" operands must be specified to accomplish macro-expansion. Omissions of other operands result in error messages on the assembly listing, but do not terminate macro-expansion.

TYPES OF WRITE OPERATIONS: The four types of write operations for the 2260 are described below.

DSB (Write Buffer Storage): This command causes sequential bytes of character data to be transmitted to the display control in either of two modes depending upon whether the address is specified as display station or 1053 Printer.

Upon recognition of the display station address by the display control, character data is placed into the display station buffer storage starting at the cursor symbol position. The cursor is automatically advanced to the next cursor position. The cursor is placed in the first display position (upper left-hand corner) of the CRT when a displayable character is inserted into the last display position of the CRT.

When the 1053 Printer address is recognized by the display control, incoming character data is placed into contiguous print positions of 1053 buffer storage and the 1053 print operation is initiated. A New Line character should be inserted at least after every 130th character; otherwise, overprinting results.

If a write error occurs during execution of a GWRITE, "type" DSB, operation, the system error recovery procedure erases the screen completely before initiating a retry.

One channel command word is generated by GWRITE, "type" DSB: the Write Buffer Storage command, which initiates the write operation.

EBW (Erase and Write Display Station Buffer Storage): This command causes the display control to delete all character data from the display station buffer storage, and to reset the cursor to the first display position (upper left-hand corner) of the CRT. The display data is then placed into display station buffer storage, starting with the position at which the cursor is reset.

Two channel command words are generated by GWRITE, "type" EBW. First, the Erase Display Station Buffer Storage command is issued to delete all character data from the display station buffer storage and reset the cursor to the first display position of the CRT. Second, the Write Display Station Buffer Storage command is issued to initiate the write operation.

LNE (Write Display Station Line Address): This command, which can only be used if the Line Addressing special feature is available, causes the display control to transmit sequential bytes of character data into the display station buffer storage under channel byte count control. The starting location in the display station buffer storage, as indicated in the first data byte, is interpreted as a control byte. This byte specifies the line to which the display data will be transmitted (see Table 4). The cursor is moved to the first display position of the line indicated and display data is placed into display station buffer storage starting at the position of the cursor symbol. The cursor is automatically advanced to the next display position.

LNE executes essentially the same command function as DSB. The basic difference is that information in the first data byte (a hexadecimal value in the range F0 through FB), which must be supplied by the user, specifies the line to which the display data will be transmitted.

One channel command word is generated by GWRITE, "type" LNE: the Write Display Station Line Address command, which initiates the write operation.

Table 4. Format of First Data Byte for LNE or EWL

Bit Positions	Hex.	Selected
0 1 2 3 4 5 6 7	Equiv.	Line
1 1 1 1 0 0 0 0	F0	one
1 1 1 1 0 0 0 1	F1	two
1 1 1 1 0 0 1 0	F2	three
1 1 1 1 0 0 1 1	F3	four
1 1 1 1 0 1 0 0	F4	five
1 1 1 1 0 1 0 1	F5	six
1 1 1 1 0 1 1 0	F6	seven
1 1 1 1 0 1 1 1	F7	eight
1 1 1 1 1 0 0 0	F8	nine
1 1 1 1 1 0 0 1	F9	ten
1 1 1 1 1 0 1 0	FA	eleven
1 1 1 1 1 0 1 1	FB	twelve

EWL (Erase and Write Display Station Line Address): This command, which can be used only if the Line Addressing special feature is available, causes the display control to delete all character data from the display station buffer storage, and resets the cursor to the first display position (upper left-hand corner) of the CRT. The cursor is moved to the first display position of the specified line and the display data is placed into the display station buffer storage starting at the position of the cursor symbol. The cursor is automatically advanced to the next display position.

EWL executes the same command function as LNE, except that the Erase command is performed first.

Two channel command words are generated by GWRITE, "type" EWL. First, the Erase Display Station Buffer Storage command is issued to delete all character data from the display station buffer storage and reset the cursor symbol. Second, the Write Display Station Line Address command is issued to initiate the write operation.

PROGRAMMING NOTES: If the data control block address is not the address of a valid DCB, or if a required operand is missing, the job is abnormally terminated.

EXAMPLE: In the following example, a DECB named DECB1 is to be generated as part of the macro-expansion. The "type" operand specifies a Write Display Station Buffer Storage command for the data set associated with the data control block OUTDCB. One hundred bytes of data will be written from an area in main storage called OUTAREA. The value of the "unit" operand is 1. In Example 1, L-form is specified to generate the parameter list. Example 2 shows coding for an L-form GWRITE macro-instruction when positional operands are omitted. Example 3 shows coding for an E-form GWRITE macro-instruction with the length changed to 150 bytes.

```
EXAMP1 GWRITE    DECB1,DSB,OUTDCB,100,
                 OUTAREA,1,MF=L

EXAMP2 GWRITE    DECB1,MF=L

EXAMP3 GWRITE    DECB1,,,150,MF=E
```

Register Notation: The programmer may specify register notation. The address of the DECB must be loaded into register 1 before execution of the macro-instruction. Other positional operands may be loaded into other registers (2-12). For example:

```
EXAMP1 LA        1,DECB1
       GWRITE    (1),MF=E
```

or

```
EXAMP2 LA        1,DECB1
       LA        2,OUTDCB
       LA        3,OUTAREA
       GWRITE    (1),,(2),,(3),MF=E
```

GCNTRL--Erase Display Station Buffer Storage Macro-Instruction

The GCNTRL macro-instruction causes the display control to delete all character data from the display station buffer storage.

Name	Operation	Operand
[symbol]	GCNTRL	decb-symbol ,type-ERS ,dcb-addr [,unit-integer] [,MF={L|E}]

decb

Is the name assigned to the data event control block. This operand is required for macro-expansion.

ERS

Specifies an Erase Display Station Buffer Storage command for the 2260.

dcb

Is the address of the data control block associated with the graphic device.

unit

Is a decimal integer; it is used to create an index to the desired element in the DEB table (address pointers to UCBs associated with the DCB). The indicated UCB contains the address of the desired device. The value of "unit" must be in the range 1 to N, where N is the number of devices specified on the Data Definition card for this data set. N cannot exceed 25 (the maximum configuration is 24 display stations and one 1053 Printer per display control).

Note: The device indicated by the "unit" operand cannot be the 1053 Printer, since an erase operation is invalid for the 1053 Printer.

Omission of the "unit" operand results in a value of 1 being assigned to "unit", indicating the first element in the DEB table. If "unit" is incorrectly specified, an error message is printed on the assembly listing.

MF=L

Specifies that expansion of the macro-instruction is to consist of a parameter list only. The name assigned to the parameter list is the name specified by the "decb" operand of this GCNTRL L-form macro-instruction.

MF=E
Specifies the macro-instruction is to perform the erase function using a parameter list constructed by another GCNTRL macro-instruction. The first operand in the E-form macro-instruction must correspond exactly to the name of the parameter list. The parameter list may be updated at execution time by specifying new values in any of the operands preceding the keyword operand.

All operands required for execution that were not specified in an L-form macro-instruction should be specified in the corresponding E-form macro-instruction.

Note: If the MF operand is omitted, the "decb" and "dcb" operands are required for macro-expansion. Omissions of other operands result in error messages on the assembly listing, but do not terminate macro-expansion.

TYPE OF CONTROL OPERATION: Only one type of control operation is valid for the 2260, the ERS (Erase Display Station Buffer Storage). It causes the display station to delete all character data from the display station buffer storage. The cursor symbol is placed in the first display position of the CRT (upper left-hand corner), and the remainder of the display is blank.

GCNTRL, "type" ERS, generates one channel command word, the Erase Display Station Buffer Storage Control command, which initiates the erase operation.

PROGRAMMING NOTES: If the data control block address is not the address of a valid DCB, or if a required operand is missing, the job is abnormally terminated.

EXAMPLES: In the coding below, a DECB named DECB1 is to be addressed. The "type" operand specifies that an Erase Display Station Buffer Storage command for the data set associated with the data control block OUTDCB will be performed. The value of the "unit" operand is 1. L-form is specified in Example 1. Example 2 shows coding for an L-form GCNTRL macro-instruction when positional operands are omitted. Example 3 shows coding for an E-form GCNTRL macro-instruction when a positional operand is changed.

```
EXAMP1 GCNTRL    DECB1,ERS,OUTDCB,1,
                 MF=L

EXAMP2 GCNTRL    DECB1,MF=L

EXAMP3 GCNTRL    DECB1,,,4,MF=E
```

Register Notation: The programmer may specify register notation. In such cases he must load the address of the DECB into register 1 before execution of the macro-instruction. Other positional operands may be loaded into other registers (2-12). For example:

```
EXAMP1 LA        1,DECB1
       GCNTRL    (1),MF=E
or

EXAMP2 LA        1,DECB1
       LA        2,4
       SLL       2,24
       GCNTRL    (1),,,(2),MF=E
```

When expressing the "unit" operand in register notation, as in EXAMP2, the value loaded into the register is the same as the one that would be used to express "unit" as a decimal integer. Note that the value for "unit" (4 in EXAMP2) is placed in the high-order byte of the register.

CLOSE--Close Graphic Device Macro-Instruction

The CLOSE macro-instruction disassociates one or more data control blocks from the user's program.

Name	Operation	Operand
[symbol]	CLOSE	(dcb-addr [,,dcb-addr]...) [,MF={L (E,listname) (E,(1))}]

dcb
Are the symbolic addresses of the DCBs for the graphic device(s) being closed. Any number of DCB addresses may be specified. If more than one DCB address is specified, the "dcb" operands must be separated by two commas. The second comma indicates omission of an optional operand pertaining to volume disposition, as explained in

the Control Program Services publication.

MF=L
: Specifies that expansion of the macro-instruction is to consist of a parameter list only. The name assigned to the parameter list is the name specified in the Name field of this CLOSE macro-instruction.

MF=(E,listname)
: Specifies that the macro-instruction is to perform the close functions using a parameter list constructed by another CLOSE macro-instruction. The close function is executed for each "dcb" operand in the list. "Listname" must correspond exactly to the name specified in the Name field of the CLOSE macro-instruction used to construct the parameter list.

MF=(E,(1))
: Performs the same function as (E,listname). By writing the special register notation (1), the programmer signifies that he will load the address of the parameter list into register 1 before macro-instruction execution.

The CLOSE macro-instruction fulfills the following functions:

- Indicates that the DCB for the device is "closed" by setting the appropriate bit in the DCB.
- Releases main storage acquired at open time, including the chain of IOBs and associated channel programs, as well as storage acquired for attention handling functions.
- Clears the fields in the DCB initialized at open time.
- Deletes attention routines associated with the DCB being "closed."

PROGRAMMING NOTES: If the operand specifies a DCB that is already closed, no action results. If the operand does not specify the address of a DCB, the task is abnormally terminated.

Modifying the DECB at Execution Time

Required parameters in the DECB may be modified at execution time by using an E-form macro-instruction. The following programming considerations apply:

1. Register notation may be specified. The register is loaded with the appropriate parameter before execution of the macro-instruction.
2. In using register notation, registers 2 through 12 should be used except for the "decb" operand. Register 1 must be used for it.
3. If the "type" operand is changed, it should only be specified by using an E-form macro-instruction containing the new "type" code. Register notation is not allowed for the "type" operand.

Upon completion of an input/output operation, the completion flag in the event control block is set to 1. If an ECB is used for more than one input/output operation, it is the problem programmer's responsibility to set the completion flag to 0 before execution of the second input/output operation and all subsequent input/output operations that refer to that ECB. The following illustration shows the position of the completion flag in the ECB.

```
     ┌───┬───┬───┬─────────────────────┐
ECB  │   │ 1 │   │                     │
     └───┴───┴───┴─────────────────────┘
bits  0   1   2                       31
```

Assume, for example, that a data event control block named DECB1 is to be used for two read operations. The second GREAD macro-instruction can be preceded by the following line of code to clear the completion flag in the ECB.

```
     NI   DECB1,X'BF'
```

BASIC ATTENTION HANDLING

The operator at the 2260 Display Station can communicate information to the system through use of the alphameric keyboard or the numeric keyboard. Input from the keyboard results in an attention (an unscheduled input/output interruption). In order for such input to be meaningful, there must exist a routine, or routines, to be entered upon receipt of an attention. The programming support described in this section includes macro-instructions to help the user notify the system of the routines available for handling attentions.

The following presentation consists of (1) a general discussion of basic attention handling, (2) detailed descriptions of macro-instructions, and (3) suggestions for a user's attention handling routine.

BACKGROUND INFORMATION

Basic attention handling provides the facilities to

- Detect the occurrence of attentions automatically, by means of the asynchronous exit form of linkage (see IBM Operating System/360; Control Program Services).

- Route these attentions to the user's attention handling routines.

- Queue succeeding attentions, if necessary, until their routine is ready to process them.

- Wait for expected attentions, optionally continuing selective processing in the meanwhile.

Particular advantages of basic attention handling are:

- Attentions are not lost if the routine designed to handle them is responding to an earlier attention at the time they are received.

- Processing may be overlapped with attention input and/or operator response time.

- System overhead is reduced, since there is no need to continually check for attentions.

When the control program detects an attention from a 2260, it interrupts the currently processing routine (which may be an attention routine) and places the attention data on the queue of the highest priority attention routine for the device originating the attention that is either active or available. (To be available, a routine must be designated by a SAEC macro-instruction for entry on receipt of a 2260 attention, and the associated GACB must have been specified by a SPAR macro-instruction and not subsequently deleted by a DAR macro-instruction; see below. An active routine is one that is in control or has been interrupted.) If no attention routine for the device is available (or active), the attention is lost. In either case, control then returns to the user's program. If the routine obtaining the queued attention had previously been inactive, it receives control; otherwise, control reverts to the routine just interrupted.

When an attention routine is in control, it processes the attentions on its queue in the order in which they occurred, unless an ATTNINQ macro-instruction within the routine instructs it to examine the queue for attentions from a particular display station.

When an attention routine relinquishes control (by means of the R mode of the ATTNINQ macro-instruction, or by executing a RETURN), any attention remaining on its queue is serviced before the routine becomes inactive. Attentions may, however, be released from the queue without being serviced; this is done by the CLEAR mode of the ATTNINQ macro-instruction.

The user can designate one attention handling routine to service attentions from all the graphic devices engaged in a task, or he can provide separate routines for individual devices.

Whatever his decision as to how to handle attentions, the user must define his attention handling capability to the system. He can do this with the following three macro-instructions:

1. SAEC - This macro-instruction creates a parameter list called a graphic attention control block (GACB), which defines an attention handling routine and relates it to a data control block.

2. SPAR - This macro-instruction activates a previously existing GACB, i.e., makes the information it contains available to the control program. It also assigns priority, in the case of multiple attention routines.

3. DAR - This macro-instruction is the inverse of SPAR, deleting control program references established by SPAR.

One more macro-instruction, already mentioned, completes the set furnished by basic attention handling:

4. ATTNINQ - This macro-instruction facilitates establishing and maintaining a communication sequence between the display station operator and an attention handling routine.

SAEC--Specify Asynchronous Entry Conditions Macro-Instruction

The SAEC macro-instruction establishes a graphic attention control block (GACB) which defines an attention routine. The GACB can subsequently be referred to by the symbolic name used for the SAEC macro-instruction.

<table>
<tr><th>Name</th><th>Operation</th><th>Operand</th></tr>
<tr><td>[symbol]</td><td>SAEC</td><td>[EP=addrx][,DCB=addrx][,COMAREA=addrx]
[,ATTNTYP=({A|S|R},2260)|SAVE|RESTORE]
[,MF={L|(E,gacb-addrx)}]</td></tr>
</table>

EP

Specifies the entry point of an attention routine.

DCB

Specifies the address of a DCB associated with the display station, or group of display stations, from which attentions are to be serviced.

COMAREA

Specifies the address of a programmer-defined area in main storage, aligned on a full-word boundary and at least 16 bytes long, which is to be used by the control program to pass attention information to the attention routine.

ATTNTYP

Specifies the type of attention that will cause entry to the attention routine.

A, S, or R

Specify, respectively, whether the information that follows is to be added to or subtracted from, or is to replace, the existing ATTNTYP field in the GACB. If this operand is omitted from an E-form of the macro-instruction, R is assumed.

2260

Specifies that the routine is to service attentions originating from a 2260.

SAVE, RESTORE

Specify, respectively, that the contents of the ATTNTYP field of the GACB are to be stored, or are to be replaced with the mask saved by a previous SAEC macro-instruction with an ATTNTYP=SAVE operand.

Note: The codes A, S, R, SAVE, and RESTORE are provided for compatibility with other graphic devices, which may allow for more than one type of attention.

MF=L

Specifies that the macro-instruction is to construct a GACB utilizing the specified operands. Operands omitted from the L-form cause the associated fields to be set to zeros. If neither the L- nor E-form is specified, L is assumed. A, S, R, SAVE, and RESTORE are not relevant when the L-form of this macro-instruction is used. Omission of the A, S, or R operand need not be indicated with a comma.

MF=E

Causes the GACB established by an L-form of this macro-instruction to be modified, as indicated by the other operands. If an operand is omitted, the corresponding field of the GACB is not affected. A comma must show the omission of the A, S, or R operand.

PROGRAMMING NOTES: The values specified for the COMAREA, DCB, ATTNTYP, and EP operands of the SAEC macro-instruction are inserted in the first, second, fourth, and fifth words of the GACB, respectively. All should be defined before a SPAR macro-instruction referring to this GACB is issued.

The first four words of the COMAREA have the following format:

COMAREA +0	IX	RESERVED	TYPE
+4	RESERVED		
+8	RESERVED		
+12	RESERVED		

Field Contents

RESERVED These fields are used by the control program.

IX If multiple 2260s are assigned to a single DCB, this byte is set to an integer from 1 through 25 to indicate which device originated the attention that the routine has currently available for processing. The value of IX corresponds to the position of the device in the device list supplied by the DD card and placed in the DEB by an OPEN macro-instruction. (IX functions in the same manner as the "unit" operand of the GREAD and GWRITE macro-instructions.)

TYPE This is always a hexadecimal 07, signifying a 2260 attention.

The user may add other fields to the COMAREA; see the suggestion under "User's Attention Handling Routine," below.

If an attention routine is to serve more than one device, and these devices have been specified in separate SAEC macro-instructions, its executable code must be reenterable.

The DCB address in a GACB should not be changed without first deleting the attention routine with a DAR macro-instruction. After changing the DCB address, the routine must be respecified with a SPAR macro-instruction.

EXAMPLES:

```
GACBX1   SAEC   EP=AR1,DCB=GRAPHD1,
                ATTNTYP=2260,
                COMAREA=AREA1
```

The GACB established by this macro-instruction can be referred to by the symbol GACBX1. The GACB defines an attention routine whose entry point is AR1. This routine is able to service attentions from the 2260 display station or stations associated with the identified DCB, and will receive attention information in the COMAREA identified by the symbol AREA1. Because the macro-form (MF) is not specified, L-form is assumed.

```
EX2      SAEC   DCB=GRPHDCB2,EP=(3),
                MF=(E,GACBX1)
```

This macro-instruction changes the DCB address and the attention routine entry point in the GACB defined in the first example.

Note: If this routine has been previously specified (with a SPAR macro-instruction) it must be deleted (with a DAR macro-instruction) before changing the DCB address. After issuing the SAEC macro-instruction to change the DCB address, the attention routine must then be respecified before it is available to the system.

SPAR--Specify Attention Routine Macro-Instruction

The SPAR macro-instruction establishes control program references to attention routines, making them available for system use.

Name	Operation	Operand
[symbol]	SPAR	(gacb-addr[,gacb-addr]...) [,PRTY=integer] [,MF=L|(E,listname-addrx)]

gacb
: Specifies the address of a graphic attention control block (GACB), defined by a SAEC macro-instruction, which is to be made available to the system.

PRTY
: Is a decimal integer from 0 through 127, used to determine priority among attention routines. Priority increases with numeric value (127 gives highest priority). No attention routine may have control while a higher-priority routine for the same device has any processing to do.

MF=L
: Specifies the construction of a list of GACB addresses which can be used by an E-form of either a SPAR or DAR macro-instruction.

MF=E
: Specifies that the list of GACBs referred to by the "listname" operand is to be modified as indicated by the "gacb" operands, and that the GACBs in the list are then to be specified, i.e., made available for system use. The list cannot be lengthened or shortened by the E-form of this macro-instruction.

listname
: Specifies the address of a list of GACBs established by an L-form SPAR macro-instruction.

PROGRAMMING NOTES: The PRTY operand is provided primarily for compatibility with certain other graphic devices, which are capable of originating more than one type of attention. For such devices, the PRTY operand permits a routine designed to handle attentions of a particular type to interrupt another, less important, attention routine that handles attentions of another type from the same device. The programmer establishes the relative "importance" of routines for this purpose by use of the PRTY operand. This usage is not possible on the 2260, which can originate only a single type of attention. Therefore, the PRTY operand can normally be omitted.

When the PRTY operand is omitted, a priority of 0 is assigned.

If more than one attention routine is available for any one 2260 at any time, and if these routines have been assigned the same numeric priority (as, for example, by omission of the PRTY operand), the routine referred to last in the last SPAR macro-instruction has the highest priority.

All attention handling routines take precedence over any background routine.

An attention routine may be interrupted and immediately reentered if the same routine is defined in separate SPAR macro-instructions for two or more display stations or if the routine is used by more than one task. Such a routine must be reenterable (it must not modify itself in any way).

If the same GACB is referred to more than once in a SPAR macro-instruction, references after the first are treated as no-operations. A GACB address of 0, in a list of GACB addresses, will also be treated as a no-operation.

ERROR CONDITIONS: Certain error conditions will stop the processing of a list of GACBs and cause an immediate return. Upon return to the program, register 1 will contain the address of the GACB associated with the error condition, and register 15 will contain one of the following return codes to indicate the type of error:

Return Code	Error Condition
04	GACB contains address of invalid DCB
08	GACB already specified by a previous SPAR macro-instruction
0C	Request for main storage needed for control program references was not satisfied

EXAMPLES:

```
EX1   SPAR   (GACBX2)
```

This macro-instruction establishes control references to the routine defined in a GACB identified by GACBX2.

```
EX2   SPAR   (GACBX1,NAME3,TAG6),MF=L
```

This macro-instruction constructs a

list containing the three GACB addresses specified.

```
EX3   SPAR   (,,GACBX3),MF=(E,EX2)
```

This macro-instruction causes the parameter list constructed by EX2 to be modified, replacing the third GACB address (TAG6) with GACBX3. Control references are then established for the three GACBs referred to by the list.

DAR--Delete Attention Routine Macro-Instruction

The DAR macro-instruction is the inverse of the SPAR macro-instruction, deleting control program references and thus making an attention handling routine unavailable for system use.

Name	Operation	Operand
[symbol]	DAR	(gacb-addr[,gacb-addr]...) [,MF=L|(E,listname-addrx)]

gacb
: Specifies the address of a GACB defining the attention routine to be deleted.

MF=L
: Specifies formation of a list of the GACB addresses written in the macro-instruction.

MF=E
: Specifies that the list of GACBs referred to by the "listname" operand is to be modified by the "gacb" operands, and that all attention routines referred to in the list are then to be deleted, i.e., made unavailable for system use.

listname
: Specifies the address of a list of GACB addresses constructed by an L-form SPAR or DAR macro-instruction.

PROGRAMMING NOTES: The CLOSE function will delete all attention routines defined for the associated DCB. However, an attempt to issue a CLOSE macro-instruction from an attention routine will result in abnormal termination of the task. Similarly, an attempt to delete an active attention routine will result in an error condition.

ERROR CONDITIONS: Certain error conditions will stop the processing of a list of GACBs (or of one GACB) and cause an immediate return. Upon return to the program, register 1 will contain the address of the GACB associated with the error condition, and register 15 will contain a return code to identify the error condition as follows:

Return Code	Error Condition
04	GACB contains address of invalid DCB
08	Routine not found (not specified by a SPAR macro-instruction)
0C	Routine active

ATTNINQ--Attention Inquiry Macro-Instruction

The ATTNINQ macro-instruction permits the selection of an attention from a particular display station from among all attentions on a queue, or entry into a wait state or temporary relinquishment of control pending a desired attention, or clearing of a queue.

Name	Operation	Operand
symbol	ATTNINQ	gacb-addrx, MODE={R|W|(C,branch-addr)|CLEAR} [,UNIT=integer|ALL]

gacb
: Specifies the address of the associated GACB.

MODE
: Specifies one of the four modes of inquiry, as follows:

R
Relinquish mode. The attention routine is to relinquish control until attention information from any display station the routine is servicing is available for that routine. When this information is available, processing continues with the next sequential instruction. This mode is treated by the

system as a W (wait) mode if the ATTNINQ macro-instruction is issued from a routine entered by means of a LINK macro-instruction.

W
Wait mode. The task is to be placed in a wait state pending availability to it of attention information from the display station specified by the UNIT operand. This wait state does not prevent interruption by a routine that has been inactive, but prevents control from returning to an interrupted routine or one that has been in a wait state. When the wait condition is satisfied, processing continues with the next sequential instruction.

C, branch
Condition mode. If attention information from the device specified by the UNIT operand is presently queued for this routine, a branch is taken to the location specified by the "branch" operand; otherwise, processing continues with the next sequential instruction.

CLEAR
Clear mode. All attentions queued for this routine are to be released, and control returned to the interrupted routine.

UNIT
Is a decimal integer from 1 through 25, and corresponds to the position of the device in the device list supplied by a DD statement, where multiple devices are assigned to a single DCB. It is used as an index to select one of the several UCB addresses appended to the data extent block (DEB). The index value corresponding to the 1053 printer may not be specified. (This operand functions in the same manner as the "unit" operand of the GREAD and GWRITE macro-instruc- tions.)

ALL
Specifies that an attention will be accepted from any 2260 assigned to the DCB.

PROGRAMMING NOTES: The ATTNINQ macro-instruction may be issued only from within the scope of an attention routine. Standard register conventions must be observed (see the Control Program Services publication).

The UNIT operand must be used in conjunction with the W or C mode if there are multiple devices for the associated DCB. It is meaningless in conjunction with the CLEAR or R mode.

Under R, W, or C mode, information pertaining to the attention that satisfies the inquiry is made available in the COMAREA.

USER'S ATTENTION-HANDLING ROUTINE

The following notes may be helpful to a user writing a program designed for operator-machine communication.

ESTABLISHING THE ATTENTION ROUTINE

An attention routine is defined by means of a SAEC macro-instruction, which establishes a graphic attention control block (GACB) containing the information provided in the SAEC macro-instruction. In the GACB are:

1. The address of the entry point of the associated attention routine.

2. The address of a DCB associated with the device to be serviced.

3. The address of a communication area (COMAREA) to be used to pass attention information to the attention routine.

The GACB thus provides a link between a display station, or group of display stations, and an attention routine.

Although defined, an attention routine is not available for use until it has been "specified" by means of a SPAR macro-instruction. This establishes control program references to it and, by means of a priority designation, establishes its processing precedence in the hierarchy of attention routines.

The DAR macro-instruction can be used to disable an attention routine which is no longer desired active although the associated DCB remains open. The SPAR and DAR macro-instructions may be thought of as analogous to OPEN and CLOSE.

PROCESSING THE ATTENTION

Upon entry to an attention routine, register 1 contains the address of the GACB. In the first word of the GACB is the address of the communication area, or COMAREA. The first byte of the COMAREA points to the device that originated the attention, if more than one device is assigned to the DCB.

Use of the communication area allows the programmer to pass information between the housekeeping/background routine and the attention routine(s). Each display station may be associated with its own communication area, and the attention data for each display station may be recorded independently of that for other display units. One of the fields in the communication area can be an ECB where completion of attention processing can be posted. The attention routine can post the appropriate ECB when it completes attention processing for that display station.

Using the information available in the COMAREA and the DCB, the attention routine can perform necessary calculations, issue appropriate input/output commands, and do whatever else is required to respond to the attention. If further communication from the display station operator is required, an ATTNINQ macro-instruction can be used to relinquish control or enter a wait state until the desired attention occurs, or to set up a conditional branch based on its availability.

When processing of the last attention of a communication sequence has been completed, the queue (for this routine) can be cleared by means of an ATTNINQ, mode CLEAR, macro-instruction before passing control to the interrupted routine by means of the RETURN macro-instruction.

Following is a brief outline of functions that might be performed by a housekeeping/background routine and by an attention handling routine.

Housekeeping/Background Routine

1. Open the DCB for the display station or stations.

2. Establish the attention routine by means of the SAEC and SPAR macro-instructions.

3. Issue input/output commands.

4. Wait for posting of completion in the event control block (ECB) or perform background processing, which may or may not be related to the attention routine.

5. Close the DCB.

6. Indicate task completion by issuing the RETURN macro-instruction.

Attention Routine

1. Perform standard entry functions: save registers, establish addressability, etc.

2. Perform operations based on the attention information.

3. Issue input/output commands.

4. Wait for the next attention by means of the ATTNINQ macro-instruction.

5. After the last attention is serviced, post completion in the ECB and return control to the background routine by means of the POST and RETURN system macro-instructions.

SYNCHRONIZING THE USER'S ROUTINES

Synchronization of the housekeeping/background routine and the attention routine is the programmer's responsibility. He may use the WAIT system macro-instruction to delay execution of the housekeeping/background routine until completion has been posted in an ECB by the user's attention routine. The attention routine may either explicitly post the ECB by means of the POST system macro-instruction or, by means of a user convention, locate the appropriate ECB via the graphic attention control block and then post completion. The attention routine passes control back to the housekeeping/background routine by means of the RETURN macro-instruction. Since completion has been posted, the housekeeping routine is taken out of its wait state and execution continues. This cycle, the passing of control to the attention routine when an attention occurs and the returning of control to the interrupted (housekeeping/background) routine when completion is posted, continues as long as the attention routine is defined and as long as attentions occur.

APPENDIX A: SAMPLE PROGRAM

Coding shown in Figure 3 is intended for use with one IBM 2260 Display Station (Local Attachment), equipped with the alphameric keyboard feature and without a 1053 Printer.

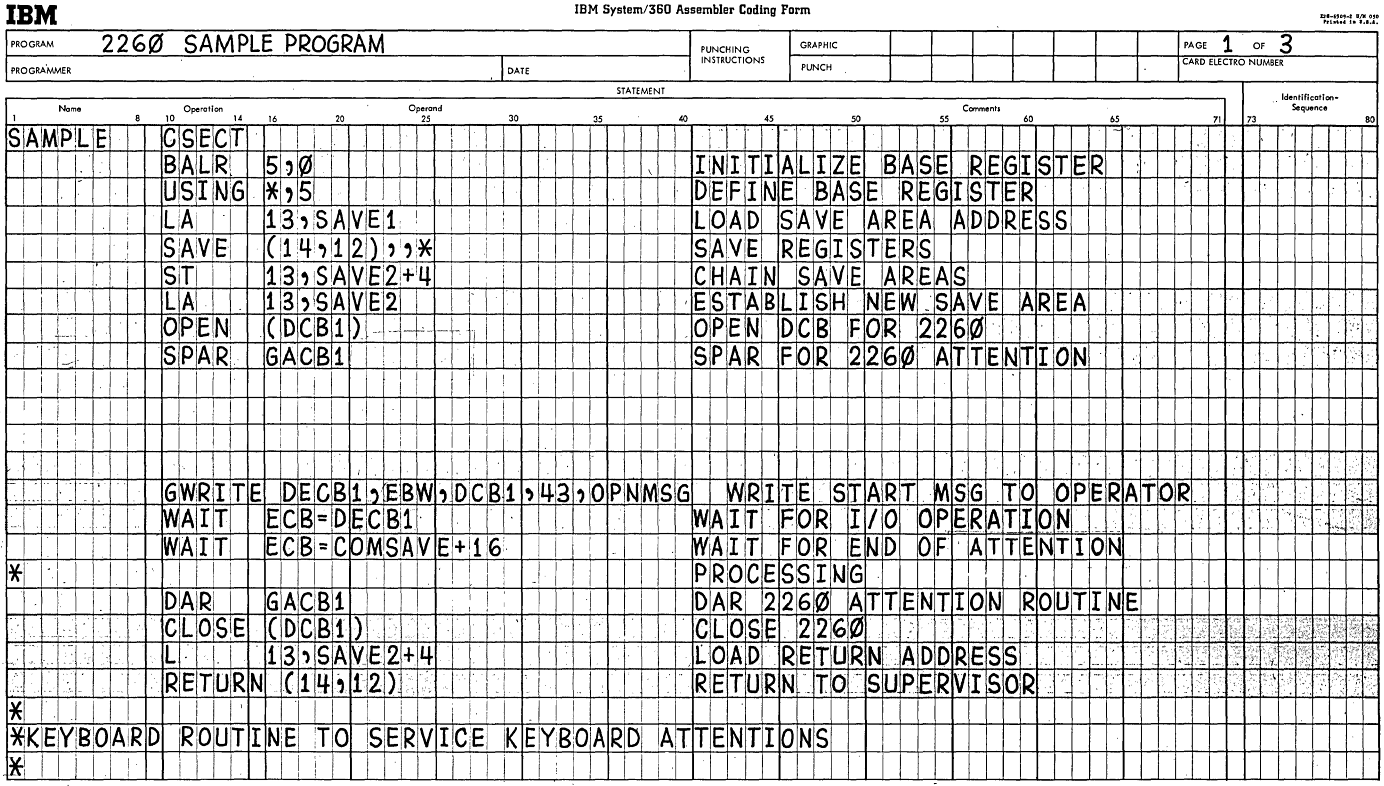
For the purpose of this sample program, it is assumed that the first four characters of the manual input message contain a code which determines whether the routine processes the message and stores the results, or ends the job and displays results derived from the processing of each individual request.

The purpose of each element of the program is explained by the comments in the illustration. The following paragraphs point up applicable graphic programming considerations.

Opening the DCB: The DCB for the 2260 must be opened before any references are made to it. Up to 24 display stations and one 1053 Printer associated with a single DCB can be opened with one OPEN macro-instruction.

Use of the GREAD Macro-Instruction: The GREAD macro-instruction shown in the coding generates a Read Display Station Manual Input (DSM) command.

Use of the GWRITE Macro-Instruction: The GWRITE macro-instruction shown issues an Erase and Write Display Station Buffer Storage (EBW) command to clear the storage and write a message to the operator. A Write Display Station Buffer storage (DSB) command is used when processing of a manual input request is completed and the operator is notified to submit the next request.



IBM — IBM System/360 Assembler Coding Form

PROGRAM 226Ø SAMPLE PROGRAM — PUNCHING INSTRUCTIONS — GRAPHIC — PUNCH — PAGE 1 OF 3

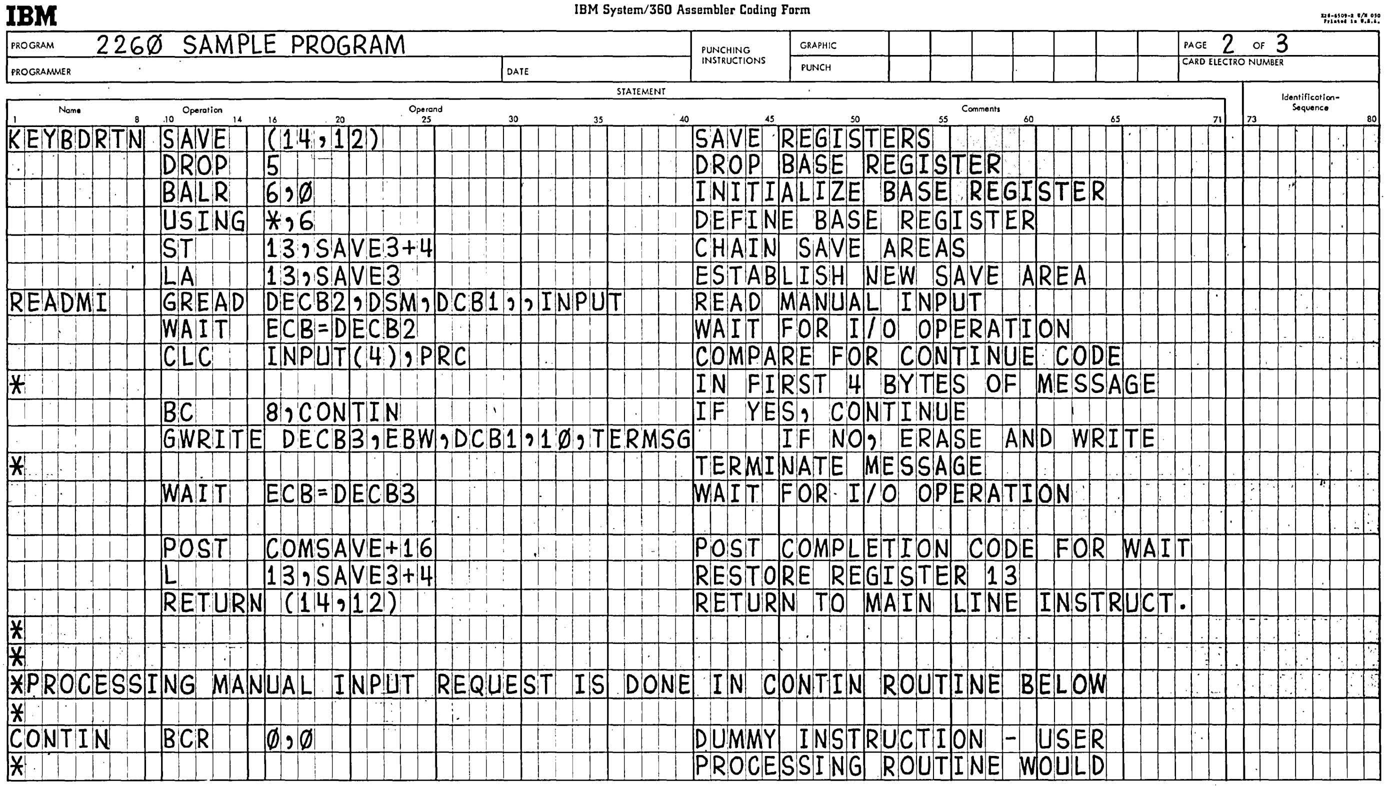
PROGRAMMER — DATE — CARD ELECTRO NUMBER

```
SAMPLE   CSECT
         BALR  5,Ø                    INITIALIZE BASE REGISTER
         USING *,5                    DEFINE BASE REGISTER
         LA    13,SAVE1               LOAD SAVE AREA ADDRESS
         SAVE  (14,12),,*             SAVE REGISTERS
         ST    13,SAVE2+4             CHAIN SAVE AREAS
         LA    13,SAVE2               ESTABLISH NEW SAVE AREA
         OPEN  (DCB1)                 OPEN DCB FOR 226Ø
         SPAR  GACB1                  SPAR FOR 226Ø ATTENTION




         GWRITE DECB1,EBW,DCB1,43,OPNMSG  WRITE START MSG TO OPERATOR
         WAIT  ECB=DECB1              WAIT FOR I/O OPERATION
         WAIT  ECB=COMSAVE+16         WAIT FOR END OF ATTENTION
*                                     PROCESSING
         DAR   GACB1                  DAR 226Ø ATTENTION ROUTINE
         CLOSE (DCB1)                 CLOSE 226Ø
         L     13,SAVE2+4             LOAD RETURN ADDRESS
         RETURN (14,12)               RETURN TO SUPERVISOR
*
*KEYBOARD ROUTINE TO SERVICE KEYBOARD ATTENTIONS
*
```

Figure 3. Coding for Sample Program (Sheet 1 of 3)



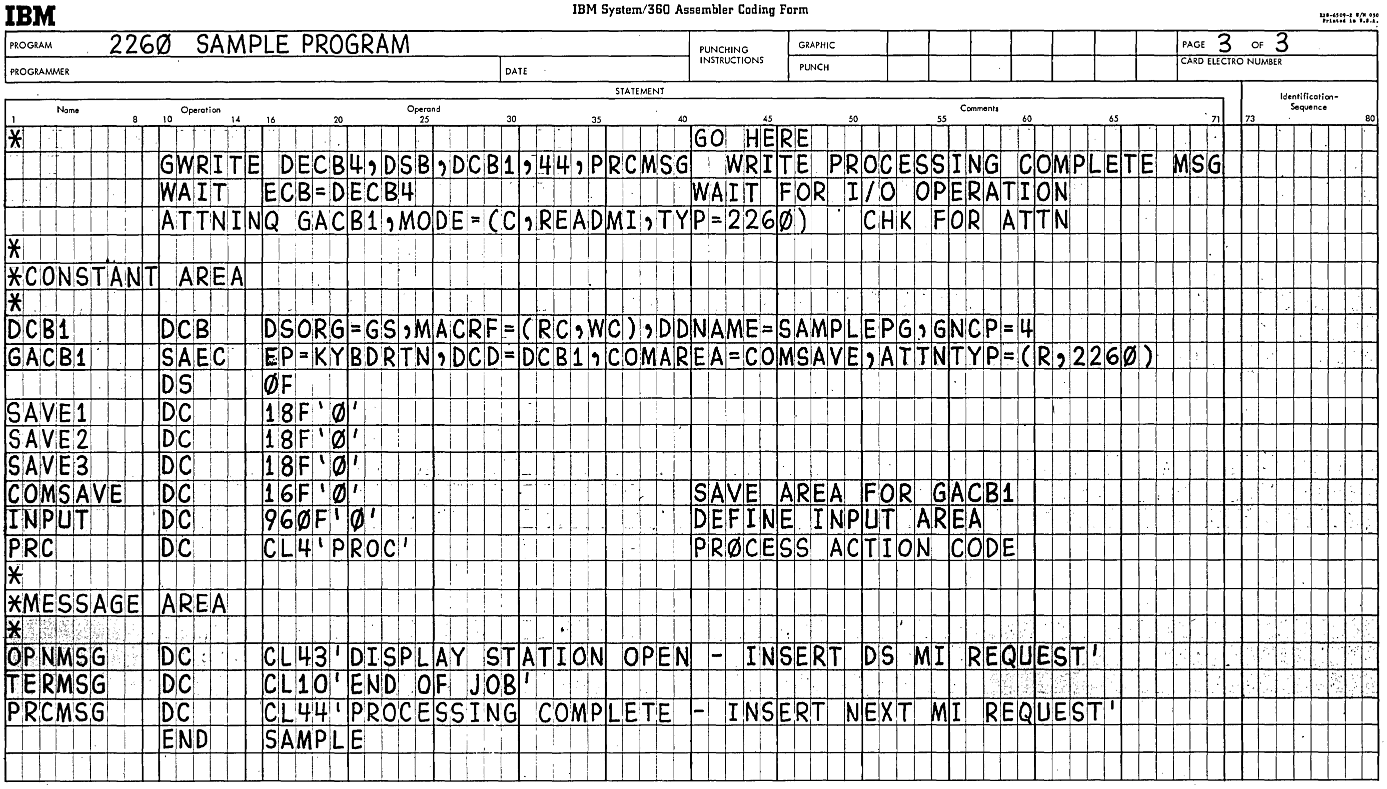
IBM — IBM System/360 Assembler Coding Form

PROGRAM 226Ø SAMPLE PROGRAM — PAGE 2 OF 3

```
KEYBDRTN SAVE  (14,12)                          SAVE REGISTERS
         DROP  5                                DROP BASE REGISTER
         BALR  6,Ø                              INITIALIZE BASE REGISTER
         USING *,6                              DEFINE BASE REGISTER
         ST    13,SAVE3+4                       CHAIN SAVE AREAS
         LA    13,SAVE3                         ESTABLISH NEW SAVE AREA
READMI   GREAD DECB2,DSM,DCB1,,INPUT            READ MANUAL INPUT
         WAIT  ECB=DECB2                        WAIT FOR I/O OPERATION
         CLC   INPUT(4),PRC                     COMPARE FOR CONTINUE CODE
*                                               IN FIRST 4 BYTES OF MESSAGE
         BC    8,CONTIN                         IF YES, CONTINUE
         GWRITE DECB3,EBW,DCB1,1Ø,TERMSG             IF NO, ERASE AND WRITE
*                                               TERMINATE MESSAGE
         WAIT  ECB=DECB3                        WAIT FOR I/O OPERATION

         POST  COMSAVE+16                       POST COMPLETION CODE FOR WAIT
         L     13,SAVE3+4                       RESTORE REGISTER 13
         RETURN (14,12)                         RETURN TO MAIN LINE INSTRUCT.
*
*
*PROCESSING MANUAL INPUT REQUEST IS DONE IN CONTIN ROUTINE BELOW
*
CONTIN   BCR   Ø,Ø                              DUMMY INSTRUCTION - USER
*                                               PROCESSING ROUTINE WOULD
```

Figure 3. Coding for Sample Program (Sheet 2 of 3)



IBM — IBM System/360 Assembler Coding Form

PROGRAM 226Ø SAMPLE PROGRAM — PAGE 3 OF 3

```
*                                               GO HERE
         GWRITE DECB4,DSB,DCB1,44,PRCMSG          WRITE PROCESSING COMPLETE MSG
         WAIT  ECB=DECB4                        WAIT FOR I/O OPERATION
         ATTNINQ GACB1,MODE=(C,READMI,TYP=226Ø)     CHK FOR ATTN
*
*CONSTANT AREA
*
DCB1     DCB   DSORG=GS,MACRF=(RC,WC),DDNAME=SAMPLEPG,GNCP=4
GACB1    SAEC  EP=KYBDRTN,DCD=DCB1,COMAREA=COMSAVE,ATTNTYP=(R,226Ø)
         DS    ØF
SAVE1    DC    18F'Ø'
SAVE2    DC    18F'Ø'
SAVE3    DC    18F'Ø'
COMSAVE  DC    16F'Ø'                           SAVE AREA FOR GACB1
INPUT    DC    96ØF'Ø'                          DEFINE INPUT AREA
PRC      DC    CL4'PROC'                        PRØCESS ACTION CODE
*
*MESSAGE AREA
*
OPNMSG   DC    CL43'DISPLAY STATION OPEN - INSERT DS MI REQUEST'
TERMSG   DC    CL10'END OF JOB'
PRCMSG   DC    CL44'PROCESSING COMPLETE - INSERT NEXT MI REQUEST'
         END   SAMPLE
```

Figure 3. Coding for Sample Program (Sheet 3 of 3)

APPENDIX B: CCW'S ASSOCIATED WITH INPUT/OUTPUT CONTROL

The following are formats of CCWs generated when GREAD, GWRITE, and GCNTRL macro-instructions are used.

GREAD DSM CCW FORMAT

Command Code	Data Address	Flags*						Byte Count
02								
0 7	8 31	32	33	34	35	36		48 63

Command Name	Function
Read Display Station Manual Input Message	Causes character data in the display station buffer storage to be processed and transferred to main storage by the display control. The start symbol is erased.

GREAD SMI CCW FORMAT

Command Code	Data Address	Flags*						Byte Count
0A								
0 7	8 31	32	33	34	35	36		48 63

Command Name	Function
Read Display Station Short Manual Input Message	Causes character data in the display station buffer storage to be processed and transferred to main storage by the display control. The start symbol is not erased.

GREAD DSB CCW FORMAT

Command Code	Data Address	Flags*						Byte Count
06			1	1				02
0 7	8 31	32	33	34	35	36		48 63

Command Code	Data Address	Flags*						Byte Count
07				1				02
0 7	8 31	32	33	34	35	36		48 63

Command Name	Function
Read Full Display Station Buffer	Two channel command words are generated. The Read Full Display Station Buffer command causes data starting at the first symbol in the upper left-hand corner of the CRT to be transferred to main storage. The Erase Display Station Buffer Storage command deletes all character data from the display station buffer storage and resets the cursor to the first display position of the CRT.

* See key at end of table

(continued)

GWRITE DSB CCW FORMAT

```
Command    Data                                                Byte
  Code    Address           Flags*                             Count
+-------+---------------+---+---+---+---+---+-----------------+-----------+
|  01   |               |   |   | 1 |   |   |                 |           |
+-------+---------------+---+---+---+---+---+-----------------+-----------+
0      7 8            31  32  33  34  35  36                48          63
```

Command Name	Function
Write Buffer Storage	Transmits sequential bytes of character data to the display control for output on the display station or 1053 Printer.

GWRITE EBW CCW FORMATS

```
Command    Data                                                Byte
  Code    Address           Flags*                             Count
+-------+---------------+---+---+---+---+---+-----------------+-----------+
|  07   |               |   | 1 | 1 |   |   |                 |    02     |
+-------+---------------+---+---+---+---+---+-----------------+-----------+
0      7 8            31  32  33  34  35  36                48          63

Command    Data                                                Byte
  Code    Address           Flags*                             Count
+-------+---------------+---+---+---+---+---+-----------------+-----------+
|  01   |               |   |   | 1 |   |   |                 |           |
+-------+---------------+---+---+---+---+---+-----------------+-----------+
0      7 8            31  32  33  34  35  36                48          63
```

Command Name	Function
Erase and Write Display Station Buffer Storage	Two channel command words are generated. The Erase Display Station Buffer Storage command deletes all character data from the display station buffer storage and resets the cursor to the first display position of the CRT. The Write Display Station Buffer Storage command initiates the write operation.

GWRITE LNE CCW FORMAT

```
Command    Data                                                Byte
  Code    Address           Flags*                             Count
+-------+---------------+---+---+---+---+---+-----------------+-----------+
|  05   |               |   |   | 1 |   |   |                 |           |
+-------+---------------+---+---+---+---+---+-----------------+-----------+
0      7 8            31  32  33  34  35  36                48          63
```

Command Name	Function
Write Display Station Line Address	Causes the display control to transmit sequential bytes of character data into the display station buffer storage under channel byte count control. The first byte specifies the row to which the display data will be transmitted. The cursor is moved to the first display position of the row indicated and display data is placed into the display station buffer storage starting at the cursor symbol position.

* See key at end of table

(continued)

GWRITE EWL CCW FORMATS

```
Command    Data                                                  Byte
 Code     Address           Flags*                               Count
┌───────┬────────────┬───┬───┬───┬───┬───┬───────────────┬────────────┐
│  07   │            │   │ 1 │ 1 │   │   │               │     02     │
└───────┴────────────┴───┴───┴───┴───┴───┴───────────────┴────────────┘
0      7 8         31  32  33  34  35  36              48           63
```

```
Command    Data                                                  Byte
 Code     Address           Flags*                               Count
┌───────┬────────────┬───┬───┬───┬───┬───┬───────────────┬────────────┐
│  05   │            │   │   │ 1 │   │   │               │            │
└───────┴────────────┴───┴───┴───┴───┴───┴───────────────┴────────────┘
0      7 8         31  32  33  34  35  36              48           63
```

Command Name	Function
Erase and Write Display Station Line Address	Two channel command words are generated. The Erase Display Station Buffer Storage command deletes all character data from the display station buffer storage and resets the cursor symbol to the first display position of the CRT. The Write Display Station Line Address command moves the cursor to the first position of the required line, and places display data into the display station buffer storage starting at the position of the cursor symbol.

GCNTRL ERS CCW FORMAT

```
Command    Data                                                  Byte
 Code     Address           Flags*                               Count
┌───────┬────────────┬───┬───┬───┬───┬───┬───────────────┬────────────┐
│  07   │            │   │   │ 1 │   │   │               │     02     │
└───────┴────────────┴───┴───┴───┴───┴───┴───────────────┴────────────┘
0      7 8         31  32  33  34  35  36              48           63
```

Command Name	Function
Erase Display Station Buffer Storage	Causes the display control to delete all character data from the display station buffer storage. The cursor symbol is placed in the first display position of the CRT and the remainder of the display is blank.

*KEY

Bit 33 -- chain command flag
Bit 34 -- suppress length indication flag

data control block (DCB): Is the means by which the user transmits data-set-oriented information to the Input/Output Supervisor.

data event control block (DECB): Is the name assigned to a block which contains the event control block and a parameter list which is constructed during the expansion of a GREAD, GWRITE, or GCNTRL macro-instruction.

input/output block (IOB): Contains information required by the Input/Output Supervisor to start an input/output operation. This block points to the channel program that is to be executed. When the operations for that channel program terminate, the CSW is stored in the input/output block associated with the channel program.

synchronous error: For the IBM 2260 Local, an error that occurs during execution of input/output operations before channel end.

UCB index: Is an indexing factor used by the Input/Output Supervisor to refer to the appropriate device from a table of UCB addresses in the data extent block. It is derived by subtracting 1 from the value of the "unit" operand in input/output macro-instructions. The resulting value is placed in the data event control block, where it is available to the Input/Output Supervisor. The UCB index for unit 1 is 0, for unit 2 is 1, and so on, up to a maximum of 25 devices (24 display stations and 1 printer) per display control.

unit control block (UCB): A block that contains information describing the input/output unit to which it is related.

INDEX

Where more than one reference is given, the first page number indicates the major reference.

Printed in U.S.A. C27-6912-2

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]

READER'S COMMENTS

IBM System/360 Operating System; Basic Graphic Programming Services for IBM 2260 Display Station (Local Attachment); Preliminary Specifications

Form C27-6912-2

Your comments will help us to produce better publications for your use. Please check or fill in the items below and add explanations and other comments in the space provided.

Which of the following terms best describes your job?

□ Programmer
□ Manager
□ Operator
□ Instructor
□ Systems Analyst
□ Engineer
□ Mathematician
□ Student/Trainee
□ Customer Engineer
□ Systems Engineer
□ Sales Representative
□ Other (explain) ______________

Does your installation subscribe to the SRL Revision Service? □ Yes □ No

How did you use this publication? ______________________________________

□ As an introduction
□ As a reference manual
□ As a text (student)
□ As a text (instructor)
□ For another purpose (explain) ______________________________________

Did you find the material easy to read and understand? □ Yes □ No (explain below)

Did you find the material organized for convenient use? □ Yes □ No (explain below)

Specific Criticisms (explain below)

Clarifications on pages
Additions on pages
Deletions on pages
Errors on pages

Explanations and Other Comments

No postage necessary if mailed in U.S.A.

FOLD FOLD

FIRST CLASS
PERMIT NO. 116
KINGSTON, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
NEIGHBORHOOD ROAD
KINGSTON, N. Y. 12401

ATTN: PROGRAMMING PUBLICATIONS
DEPARTMENT 637

FOLD FOLD

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]