File No. S360-50
Form C28-6557-1

IBM Systems Reference Library

IBM System/360 Basic Programming Support

Operating Guide for Basic Assembler and Utilities

360P-UT-017, -018, -019, -020

360P-AS-021

This publication describes the preparation and use of the Basic Assembler and Basic Utility Programs. The Basic Assembler converts programs written in the assembler language into machine language object code. The Basic Utility Programs are concerned with loading programs into storage, printing out the contents of storage, and using input/output devices.

PREFACE

This publication contains information on the preparation and use of the IBM System/360 Basic Programming Support Basic Assembler and Basic Utility Programs. The Basic Assembler is discussed first, followed by each of the Basic Utility Programs.

MAJOR REVISION (June 1965)

This revision, which supersedes the previous edition, represents Version 2 of the subject programs. The changes are marked by a vertical bar to the left of the revised text; revised illustrations are denoted by the symbol • to the left of the caption.

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 readers' comments appears at the back of this publication. It may be mailed directly to IBM. Address any additional comments concerning this publication to the IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

FIGURES

The IBM System/360 Basic Programming Support Operating Guide publication provides operating information and techniques for the System/360 Basic Assembler and Basic Utility Programs. The Basic Assembly Program accepts source programs written in System/360 Basic Assembler Language and produces object programs in System/360 machine language. The Basic Utility Programs enable the user to print out the contents of registers and/or storage, load assembled programs, and program the use of input/output devices.

This publication is concerned only with operating considerations, not with the internal logic of the programs. Specifically, the following topics are discussed:

1. Preparing and running an assembly job.
2. Using the IBM-supplied program loaders.
3. Preparing and generating self-loading program loaders.
4. Using the single- or two-phase dump program to print out the contents of storage.
5. Using the self-loading dump program.
6. Program waits and operator messages.

THE BASIC ASSEMBLER

The Basic Assembler is, essentially, a language translator. It translates source programs written in the Basic Assembler language into executable machine-language object programs. The assembler is divided into two parts, Phase 1 and Phase 2.

Input to Phase 1 consists of source program statements punched into cards or written on magnetic tape. Phase 1 partially translates the source program statements into machine-language object code. The partially translated statements are passed to Phase 2 (see Figure 1) where the translation process is completed. The output produced by Phase 1 (that is, the partially translated source statements) must be passed to Phase 2 via punched cards or magnetic tape.

Note: Certain character constants (C' ') that do not fall into the normal BCD configuration, when entered into System/360 by means of another computer, may lose bits during the card-to-tape phase.

Program Listings

The assembler provides a program or error listing for each assembly if a printer or printer-keyboard is attached to the system, and the assembler has been instructed to provide listings or error listings. This is described in detail in the section "Phase 1 Configuration Card."

Assembled Object Program Output

Assembled object programs produced by the assembler may be punched in cards or written on tape. The specification of the object program storage medium is described in detail in the section "Phase 1 Configuration Card."

Machine Configuration

The IBM System/360 Basic Programming Support Basic Assembler program requires the following minimum machine configuration:

- An IBM System/360 with 8,192 bytes of storage
- An IBM 2540 or 1442-2 Card Reader-Punch
- The Standard Instruction Set

If additional input/output devices are attached to the system, the assembler's operational capabilities are increased. The various input/output devices and their uses are listed below.

IBM 2400-Series Magnetic Tape Unit:

From one to five magnetic tape units can be used for the storage of any of the following:

1. Source program
2. Basic Assembler object decks
3. Intermediate text

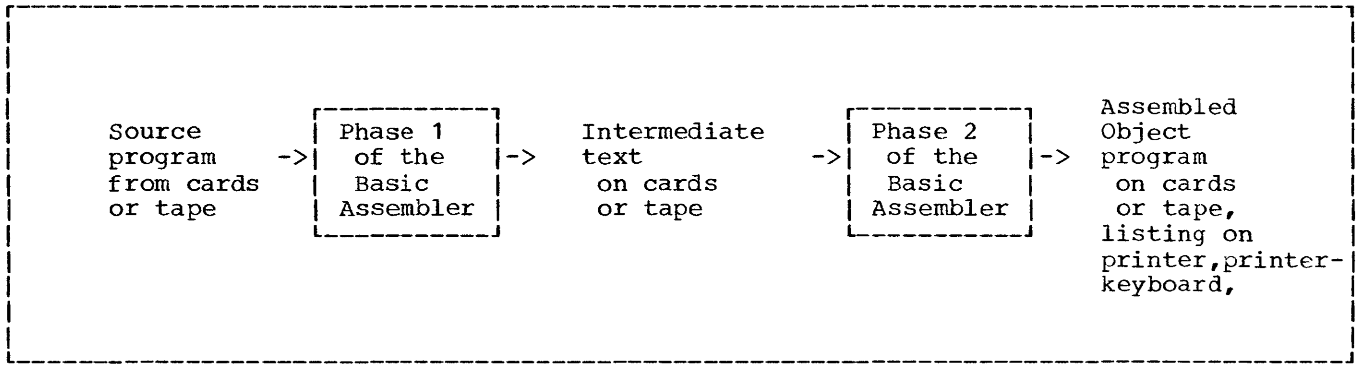

Source program from cards or tape
Phase 1 of the Basic Assembler
Intermediate text on cards or tape
Phase 2 of the Basic Assembler
Assembled Object program on cards or tape, listing on printer,printer-keyboard,

Figure 1. Basic Programming Support Basic Assembler

4. Program listing
5. Object program

One IBM 1403 or 1443-2 Printer: Used by the assembler to provide program listings, complete with operator and error messages, for each assembly.

One IBM 1052 Printer-Keyboard: Used by the assembler to provide program listings, complete with operator and error messages, for each assembly.

One IBM 1403 or 1443-2 Printer and One IBM 1052 Printer-Keyboard: The assembler uses the IBM 1403 or 1443-2 Printer to print program listings. The IBM 1052 Printer-Keyboard is used for operator and error messages.

ASSEMBLER INITIALIZATION

Since all installations do not have the same machine configuration, the Basic Assembler program must be tailored for operation at each installation. This tailoring consists of defining to the assembler:

1. The main-storage size of the system.

2. The input/output devices attached to the system and their addresses.

3. What use is to be made of cards and magnetic tape.

In addition to the initialization associated with the machine configuration, other initialization may instruct the assembler to print or suppress program listings or to print only error listings.

The Basic Assembler is initialized through the use of configuration cards. There are two configuration cards, one for each phase of the assembly program. The cards are called the Phase 1 and Phase 2 Configuration Cards.

Phase 1 Configuration Card

The Phase 1 Configuration Card is a Replace card which describes to Phase 1 of the assembler the machine configuration upon which it is to operate. The card is inserted in the Phase 1 deck just before the END card. The Phase 1 Configuration Card has the following format:

Cols.	Fld.	Description
1- 4	1	Contains a 12-2-9 punch followed by the characters REP. This identifies the card as a configuration card.
5- 6		Blank.
7-12	2	Contains 00008A. This is the hexadecimal starting address, in storage, where the data in columns 17-55 is to be placed.
13-14		Blank.
15-16	3	Contains 01. This is a constant.
17-20	4	Contains a 0, if the source input device is a card reader, followed by the three-digit hexadecimal address of the reader. If the source input device is a tape unit, the field contains a 1, followed by the three-digit hexadecimal address of the source device.
21	5	Comma.
22-25	6	If the intermediate text data is to be stored on cards, this field contains a 0, followed by the three-digit hexadecimal address of the card punch. If the intermediate text is to be stored on tape, the field contains a 1, followed by the three-digit hexadecimal address of the tape unit.
26	7	Comma.
27-30	8	If the assembler is to be copied on tape, this field contains a 1, followed by the three-digit hexadecimal address of the tape unit which will be used. If the assembler is not to be copied, the field contains 0's.
31	9	Comma.
32-35	10	Contains a 0, followed by the three-digit hexadecimal address of the system message device -- a printer or a printer-keyboard -- or 0's, if neither of these devices is attached to the system.
36	11	Comma.

(continued)

(continued)

Cols.	Fld.	Description
37-40	12	If the program listing is to be printed on the printer or printer-keyboard, this field contains a 0, followed by the three-digit hexadecimal address of one of these. If the listing is to be written on tape, the field contains a 1, followed by the address of the tape unit. If none of these devices is used, the field contains 0's.
41	13	Comma.
42-45	14	If the final object program is to be punched on cards, this field contains a 0, followed by the three-digit hexadecimal address of the card punch. If the object program is to be written on tape, the field contains a 1, followed by the address of the tape unit.
46	15	Comma.
47-50	16	Contains a four-digit hexadecimal number. The first two digits indicate the mode set code (see Figure 2) for the source input tape. The second two digits indicate the mode set code for the object program output.
51	17	Comma.
52-55	18	The first two digits of this field indicate the mode set code (Figure 2) for the device on which a listing is to be written. The second two digits are composed of the following bit formations: 1. Hexadecimal equivalent of a four-bit binary number generated in the following way: • The first two bits are 0's. • The next two bits depend on the machine's main storage size: 00 for 8K 01 for 16K 10 for 32K 11 for 64K 2. Hexadecimal equivalent of a four-bit binary number generated in the following way: • The first bit is a 0. • The second bit is a 0, if a 2540 Card Reader-Punch is attached to the system, or a 1, if the 1442-2 Card Reader-Punch is attached to the system. • The third and fourth bits are 0's.

(continued)

(continued)

Column 56 must be blank; columns 57-80 may include anything which the programmer wishes.

The code 03 must be used for all 9-track tapes and unit record devices. It is used for all 7-track tapes which will be used <u>only</u> on System/360.

The following mode set codes are used for 7-track tapes which will be used on System/360 and some other machine:

Code	Density	Parity
2B	200 BPI	Even
3B	200 BPI	Odd
6B	556 BPI	Even
7B	556 BPI	Odd
AB	800 BPI	Even
BB	800 BPI	Odd

●Figure 2. Mode Set Codes

Phase 2 Configuration Card

The Phase 2 Configuration Card is identical to the Phase 1 Configuration Card, except that columns 54 and 55 are punched in the following manner:

Column 54.
- bit 1 is a 0.
- bit 2 is a 1, if a program listing is to be generated for the assembly. It is a 0, if only an error listing is desired.
- bit 3 is a 0.
- bit 4 is a 0, if the printer is used to produce program listings. It is a 1, if the listings are produced on a printer-keyboard.

Column 55 is a 0.

The results of mispunching configuration cards are unpredictable.

A. ASSEMBLING ON A CARD SYSTEM USING THE 2540

Assembler: cards
Source deck: cards
Intermediate text: cards
Object program: cards
Listing: printer or printer-keyboard

1. Make the printer and printer-keyboard ready for use.

2. Clear the card reader of cards.

3. Insert the proper configuration cards immediately before the END card in both decks of the assembler. Detailed information concerning these cards is presented in the section "Assembler Initialization."

4. Place the Phase 1 deck of the assembler in the reader hopper; then place the source program deck in the hopper.

5. Place blank cards in the punch hopper. The number of blank cards must be equal to or greater than the number of cards contained in the source program deck.

6. Place additional blank cards for the symbol table in the punch hopper at the ratio of approximately one blank card for every twenty source program cards.

7. Initialize card reader-punch for use.

8. Press end-of-file key on the card reader-punch.

9. Select the card reader with the load-unit switches on the system control panel and press load key.

10. A program wait with the location counter containing 1EI occurs at the completion of Phase 1. If the system has provisions for typing messages, a message "1EI" is typed on the printer or printer-keyboard. The contents of the stackers at this point are shown in Figure 3.

11. Make printer or printer-keyboard ready for use if necessary.

12. Clear the card reader of cards.

13. Place the Phase 2 deck of the assembler in the card read hopper.

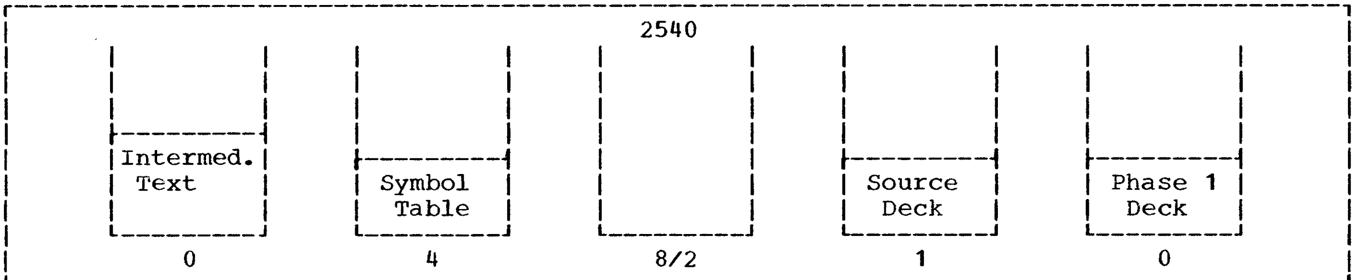
14. Place intermediate text deck on the assembler deck.

15. Place blank cards in the punch hopper at the ratio of one blank card for every ten source program cards.

16. Press end-of-file key on card reader-punch.

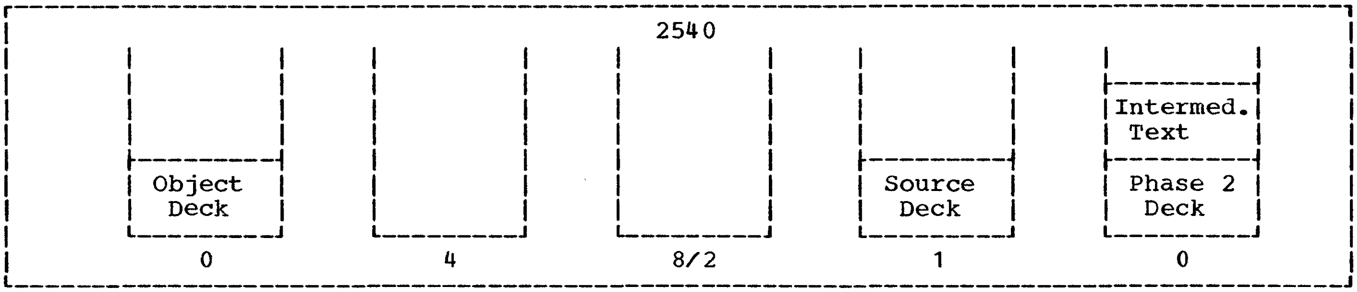
17. Select the card reader with the load-unit switches on the system control panel and press the load key.

18. A "2EI" message or a program wait with the location counter containing 2EI signals the end of the second phase of assembling. The contents of the stackers at the completion of the assembly job are shown in Figure 4.



The intermediate text deck consists of data produced by Phase 1 for use as input to Phase 2. The symbol table deck is produced for use in a special application described in the section "Special Procedures."

•Figure 3. Stacker Contents at End of Phase 1



●Figure 4. Stacker Contents at End of Assembly

B. ASSEMBLING ON A CARD SYSTEM USING THE 1442

Assembler: cards
Source deck: cards
Intermediate text: cards
Object program: cards
Listing: printer or printer-keyboard

1. Make the printer and printer-keyboard ready for use.

2. Clear the card reader of cards.

3. Insert the proper configuration cards immediately before the END card in both decks of the assembler. Detailed information concerning configuration cards is presented in the section "Assembler Initialization."

4. Place the Phase 1 deck of the assembler in the reader hopper. Then place the source program deck in the hopper.

5. Place blank cards for the symbol table in the hopper at the ratio of approximately one blank card for every twenty source program cards.

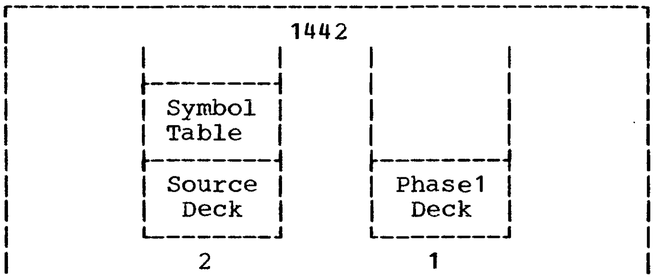
6. Initialize card reader-punch for use. (Do not press end-of-file key.)

7. Select the card reader with the load-unit switches on the system control panel and press the load key.

8. A program wait with the location counter containing 1EI occurs at the completion of Phase 1. If the system has provisions for typing messages, a message "1EI" is typed on the printer or printer-keyboard. The contents of the stackers at this point are shown in Figure 5.

9. Make the printer and printer-keyboard ready for use if necessary.

10. Clear the card reader of cards.



The intermediate text deck consists of data produced by Phase 1 for use as input to Phase 2. The symbol table deck is produced for use in a special application described in the section "Special Procedures." Note that the intermediate text is contained in columns 1-24 of the source deck on the 1442.

Figure 5. Stacker Contents at End of Phase 1

11. Place Phase 2 deck of the assembler in the card read hopper.

12. Place the source program deck in the card hopper.

13. Place blank cards in the hopper behind the source program deck.

14. Select the card reader with the load-unit switches on the system control panel and press the load key.

15. A program wait with the location counter containing 2HA or a message "2HA" indicates that blank cards must be placed in the 1442 card hopper. Remove any cards in the hopper, insert blanks, and replace the cards just removed. The number of blank cards is governed by the machine's storage size: 15 blanks for an 8K machine, 80 blanks for a 16K machine, 200 blanks for a 32K machine, and 800 blanks for a 64K machine. After inserting the blanks, press the interrupt key.

16. A "2EI" message or a program wait with the location counter containing 2EI signals the end of the second phase of assembling.

17. The contents of the stackers at the completion of assembling are shown in Figure 6.

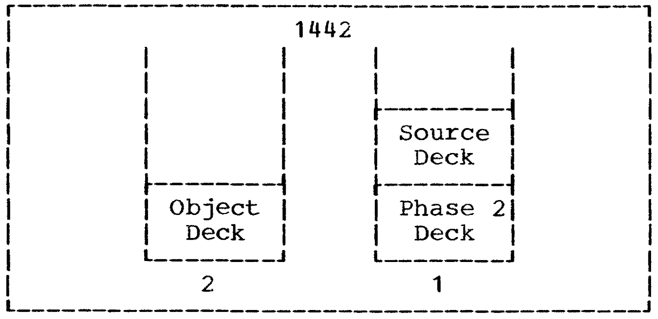


Figure 6. Stacker Contents at End of Assembly

C. COPYING THE ASSEMBLER ON TAPE

1. Punch the correct configuration cards (described in detail in "Assembler Initialization") and place them before the END cards in Phases 1 and 2.

2. Ready the card reader.

3. Place Phases 1 and 2 of the assembler in the read hopper.

4. Load tape on tape unit whose address was specified in field 8 of configuration cards.

5. Ready tape unit.

6. Press end-of-file on card reader.

7. Select the card reader with the load-unit switches on the system control panel and press load key.

8. When the "1EI" message appears in the location counter, press load key a second time to write Phase 2 on tape.

9. When the "2EI" message is printed or appears in the location counter, the assembler has been written entirely on the selected tape. Rewind before using. (Note that the assembler may be written on one tape unit and run on a different tape unit, provided they both have the same number of tracks.)

D. ASSEMBLING WITH CARD AND TAPE CONFIGURATION

Assembler: cards or tape
Source deck: cards or tape
Intermediate text: cards or tape
Object program: cards or tape
Listing: printer, printer-keyboard, or tape

1. Ready assembler input device.

2. Ready source program input device.

3. Ready intermediate text device.

4. Ready object program device.

5. Ready listing device.

6. Select assembler input device with load-unit switches on the system control panel and press load key.

7. If intermediate text medium is punched cards, see steps A 10-18 or B 8-16 in the preceding sets of instructions.

8. If intermediate text is on tape and the assembler is on cards, a "1EI" message or a program wait with the location counter containing 1EI signals the end of Phase 1. Load Phase 2 of the assembler. If the assembler is on tape, the message will not appear and control will automatically pass to Phase 2. Note that when the intermediate text is on tape, Phase 2 must always follow immediately after Phase 1, since no symbol table is punched.

9. A "2EI" message or a program wait with the location counter containing 2EI signals the end of the second phase of assembling.

SPECIAL PROCEDURES

There are three special procedures available for use with card systems. They are:

1. A procedure for saving time when reassembling a previously assembled program on a 1442 card system.

2. A procedure for running an assembly job on a card system when Phase 2 is not executed immediately after Phase 1.

3. A procedure for saving time during Phase 2 when using a 1442 card system that punches the assembled object programs into cards.

1442 Card System Reassembly Procedure

A special reassembly procedure is provided for card systems using the IBM 1442-2 Card Reader-Punch. This procedure enables a previously assembled program to be reassembled in less time than that required for a new assembly.

To use this procedure, one must have the symbol table deck produced by Phase 1 of the previous assembly (see Figure 3). Input to Phase 1 during a reassembly consists of the Phase 1 assembler deck followed, in order, by the previously punched symbol table, the source program, and blank cards (into which the new symbol table will be punched). The number of blank cards should be approximately equal to the number of cards in the previously punched symbol table. Note that the only difference between the Phase 1 input required for a new assembly and the Phase 1 input required for a reassembly is the inclusion of the symbol table deck in the latter case. Other than preparing the Phase 1 input, the actions required to run a reassembly job are exactly the same as those required for a new assembly.

Interrupted Assemblies on Card Systems

If a card system assembly job is interrupted after the completion of Phase 1, but before the conclusion of Phase 2, a special procedure is provided to complete the assembly job without re-executing Phase 1. Tape assembly jobs may not be interrupted.

When this procedure is used, it is only necessary to run Phase 2 of the assembler (Phase 1 was run before the interruption). Input to Phase 2, in this case, is the same as that required for a new assembly, with one exception. That exception consists of placing the symbol table deck (produced by Phase 1 before the interruption) on top of the Phase 2 assembler deck in the input card hopper. The rest of the Phase 2 input is then placed on top of the symbol table deck. Other than setting up the Phase 2 input, the actions required to run Phase 2 are the same as those required to run Phase 2 of a new assembly.

1442 Card Systems with Card Output

Occasionally, when running an assembly job on a 1442 card system with card output, a program wait occurs during Phase 2 with the location counter containing 2HA, indicating the need for more blank cards. If the system has provisions for typing messages, a message "2HA" is typed out. In either case, blank cards must be placed in the 1442 card hopper and the interrupt key must be pressed (see Figure 4).

This intervention may be avoided by interleaving blank cards with the source program deck before starting Phase 2 of the assembler. The size of the system's storage governs the manner in which the blank cards are interleaved with the source program, as shown in the following:

Main-Storage Size	Action
8K	Insert approximately 15 blank cards after each 150 source program cards.
16K	Insert approximately 80 blank cards after each 800 source program cards.
32K	Insert approximately 200 blank cards after each 2,000 source program cards.
64K	Insert approximately 800 blank cards after each 8,000 source program cards.

The single-phase dump program is designed to produce listings of the contents of registers and/or storage.

When used, the single-phase dump program resides in storage along with the user's program. Figure 7 shows the relationship between the single-phase dump program and the user's program.

INITIALIZATION OF THE SINGLE-PHASE DUMP PROGRAM

The single-phase dump program is available from IBM in symbolic form on punched cards (that is, a source deck) and as an assembled relocatable object deck. Before the program can be used, it will require modification for operation on the installation's machine. This modification consists of altering three constants near the end of the IBM-supplied program. These constants are shown in Figure 8. They are identified in the figure by the number 1 in column one on the left-hand side. One must ensure that these locations properly describe the installation's machine configuration before using the single-phase dump program. Note that a card's position in the deck should not be altered during the modification process.

USING THE SINGLE-PHASE DUMP PROGRAM

The single-phase dump program is essentially a subprogram designed for use by the programmer; its use, therefore, is primarily his concern. He must define, in his program, the registers and/or storage areas whose contents are to be listed. In addition, he must define the format of the listing. Finally, he must transfer control to the single-phase dump program in order to have the listing produced.

In order to execute a program that uses the single-phase dump program, both programs must reside in storage, and the proper linkage must exist between them. These requirements can be fulfilled by either of two methods.

One method consists of assembling the single-phase dump and user's programs together. The resulting assembled object program contains both the single-phase dump and problem programs with the appropriate linkages. It can be loaded into storage for execution by the Absolute or Relocating Loaders. (The Relocating Loader cannot be used to accomplish this on an 8K configuration.)

The other method consists of assembling the single-phase dump and user programs separately. In this case, the respective assembled object programs must be loaded into storage for execution by the Relocating Loader. Note that during the load process, the dump program should precede the user's program into storage so that the loader can establish the proper linkages.

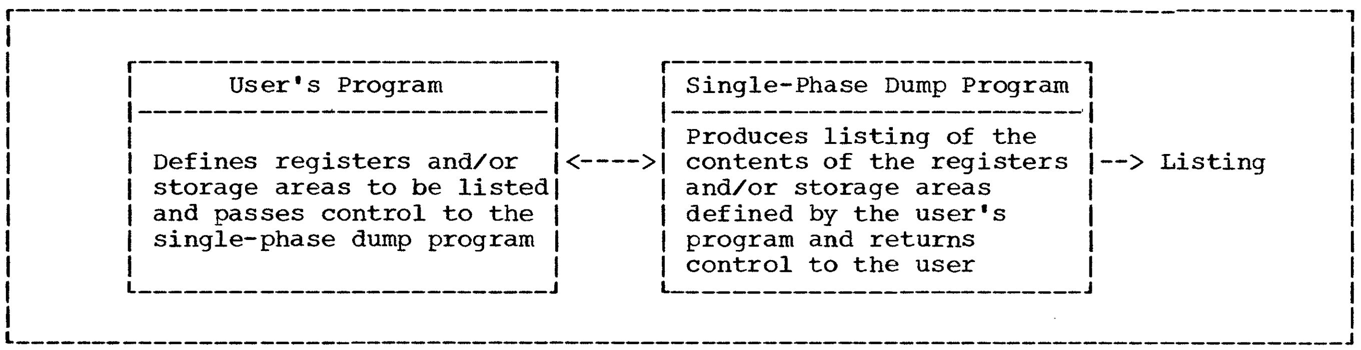

User's Program
Defines registers and/or storage areas to be listed and passes control to the single-phase dump program
<---->
Single-Phase Dump Program
Produces listing of the contents of the registers and/or storage areas defined by the user's program and returns control to the user
--> Listing

Figure 7. The Single-Phase Dump Program

1. Single-Phase Dump Program
2. Phase 1 of the Two-Phase Dump Program
3. Phase 2 of the Two-Phase Dump Program

1	2	3	Name	Operation	Operand	Description
1	2		DSTOPL	DC	AL3(zzzzz)	zzzzz represents the machine's storage size in bytes. Valid operands may be: AL3(8192) AL3(16384) AL3(32768) AL3(65536)
		3	INDEV	DC	X'zzzzzzzz'	zzzzzzzz represents eight hexadecimal digits generated in the following way: Digits Description 1-3 Always 0. 4 Specifies the Phase 2 input device: 0 = IBM 2400 Series Magnetic Tape Unit 1 = IBM 2540 or 1442-2 Card Reader-Punch 5 Always 0. 6-8 The three-digit hexadecimal address of the Phase 2 input device.
1	2	3	OUTDEV	DC	X'zzzzzzzz'	zzzzzzzz represents eight hexadecimal digits generated in the following way: Digits Description 1-3 Always 0. 4 For the Single-Phase Dump Program and Phase 2 of the Two-Phase Dump Program, this digit specifies the type of output device used to produce listings. 0 = IBM 1403 or 1443-2 Printer 1 = IBM 1052 Printer-Keyboard For Phase 1 of the Two-Phase Dump Program, this digit specifies the type of output device used by Phase 1. 0 = IBM 2400 Series Magnetic Tape Unit 1 = IBM 2540 Card Reader-Punch 2 = IBM 1442-2 Card Reader-Punch 5 Always 0. 6-8 The three-digit hexadecimal address of the output device.
1	2	3	TYPWTR	DC	X'zzzz'	zzzz represents a 0 followed by the three-digit hexadecimal address of the printer or printer-keyboard used to produce operator messages. If neither device is available, the operand must be specified as X'FFFF'. Placing hexadecimal F's in TWPWTR only informs the dump program that no IBM 1052 Printer-Keyboard is available and does not disable the Input/Output Routine from trying to print error messages. There are two ways to disable printing by the I/O routines. 1. Prior to assembly time, remove the Write Error Message Base Routine module from the I/O routines. 2. At object time, insert a Replace card to put the LPSW instruction at SAGINW+4 (in the I/O Base Routine - Group 1, Interrogate I/O Interrupt or Condition Code 1 module) back in the same form as on the assembly listing after it has been overlaid by the branch instruction in the Write Error Message Base Routine module.

●Figure 8. Dump Program Initialization Cards

The Two-Phase Dump Program produces listings of the contents of registers and/or main storage. The program consists of two phases, Phase 1 and Phase 2.

Phase 1 is designed to produce card image records (on punched cards or magnetic tape) of the contents of registers and/or storage. When used, it resides in storage along with the user's program. Figure 9 shows the relationship between Phase 1 and the user's program.

INITIALIZATION OF THE TWO-PHASE DUMP PROGRAM

The Two-Phase Dump Program is available from IBM in symbolic form on punched cards (that is, a source deck) for each phase as an assembled relocatable object deck for Phase 1 and as an assembled nonrelocatable deck (self-loading deck) for Phase 2. Before the program can be used, each phase may require modification for operation on the installation's machine. This modification consists of altering three constants near the end of each phase in the IBM-supplied programs or punching information in the END card in the case of Phase 2 assembled nonrelocatable deck.

The constants in question are shown in Figure 8. Constants to be modified in the Phase 1 program are identified by the number 2 in column two on the left-hand side of the figure. Constants to be modified in the Phase 2 program are identified by the number 3 in column three. One must ensure that these constants properly describe the installation's machine configuration before assembling the Phase 1 or Phase 2 source decks or by altering the assembled relocatable decks with Replace cards at object time or by punching information in the END card of the Phase 2 self-loading deck. Note that a card's position in the source deck should not be altered during the modification process.

USING THE TWO-PHASE DUMP PROGRAM

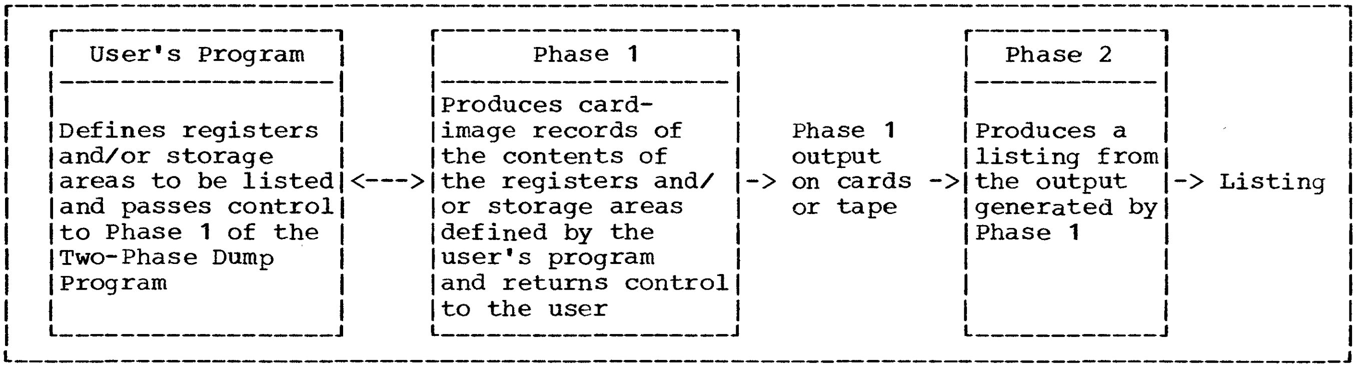
Each phase of the Two-Phase Dump Program has its own set of operating procedures. These procedures are described in the following text.

Phase 1

Phase 1 is used in essentially the same way as the Single-Phase Dump Program (see the topic "Using the Single-Phase Dump Program"). The two differ only with respect to their output. Phase 1 produces card or tape output for subsequent use by Phase 2. The Single-Phase Dump Program produces listings. Note that if tape is used for output, this tape is only rewound at end-of-reel by Phase 1. This enables the user to place the dump output of more than one program on a single reel for later processing by Phase 2.

Phase 2

Phase 2 produces listings of the contents of registers and/or storage from the output generated by Phase 1.



User's Program

Defines registers and/or storage areas to be listed and passes control to Phase 1 of the Two-Phase Dump Program

<--->

Phase 1

Produces card-image records of the contents of the registers and/or storage areas defined by the user's program and returns control to the user

-> Phase 1 output on cards or tape ->

Phase 2

Produces a listing from the output generated by Phase 1

-> Listing

Figure 9. The Two-Phase Dump Program

Phase 2, as supplied by IBM, is a self-loading version. To use the self-loading deck, the following must be supplied:

1. The type of output device and its address.

2. The type of input device and its address.

3. The address of the IBM 1052 Printer-Keyboard (if one is available for operator messages).

The user supplies this information by taking out the END card from the self-loading deck of Phase 2 of the two-phase dump and punching this card as follows:

Cols.	Description
17-20	The address of the output device -- a printer or an IBM 1052 Printer-Keyboard.
21	0 if a printer is to be used. 1 if an IBM 1052 Printer-Keyboard is to be used.
22-25	Address of the input device to be used.
26	0 if the input device is a tape unit. 1 if the input device is a card reader.
27-30	The address of the IBM 1052 Printer-Keyboard if one is available. FFFF if no IBM 1052 Printer-Keyboard is available.

I/O error messages are only displayed on the console during error waits when the self-loading deck supplied by IBM is used.

To use Phase 2 of the Two-Phase Dump Program in its self-loading version, the following steps must be performed:

1. Run cards out of the card reader.

2. Place the properly initialized self-loading deck of Phase 2 in the card-read hopper.

3. Place the Phase 1 output on the appropriate unit. This unit address was defined to Phase 2 in the END card. If tape is used, the tape will be rewound at the beginning of execution. If card reader was used for Phase 1 output, this output should be followed by at least one blank card to ensure that the last listing will print.

4. Set the load unit switches on the system control panel to address the card reader and press the load key.

A user with a machine larger than 8K can make more efficient use of Phase 2 of the two-phase dump by altering the source program for residence in higher storage and increasing the buffer size. The assembled deck can then be loaded by either the absolute or relocating loader. In order to use Phase 2 in this form (an assembled relocatable version), the following are required:

1. A properly prepared and assembled Phase 2 program.

2. The output generated by Phase 1. The output can be on cards or tape.

3. A self-loading loader on cards.

To execute Phase 2 in its assembled relocatable version, perform the following steps:

1. Run cards out of the card reader.

2. Prepare the self-loading Absolute or Relocating Loader to read from the device containing Phase 2 of the two-phase dump program. The method of initialization is described in the section "The Absolute and Relocating Loaders."

3. Place the self-loading loader in the card read hopper.

4. Place Phase 2 of the two-phase dump program on the appropriate device.

5. Place the Phase 1 output on the appropriate unit. This unit address was defined to Phase 2 during the Phase 2 initialization process (see the topic "Initializing the Two-Phase Dump Program"). If tape, the tape will be rewound at the beginning of Phase 2. If card, Phase 1 output should be followed by at least one blank card to ensure that the last listing will print.

6. Set the load-unit switches on the system control panel to address the card reader, and press the load key.

The self-loading dump program is essentially the single-phase dump program in self-loading form. It produces listings of all of storage (excluding the area in lower storage occupied by the dump program -- approximately 3,000 bytes) and the contents of all general registers. However, the loading process destroys the original contents of 11 general registers but leaves registers 0, 7, 9, 10, and 15 intact. The listings are produced in full-word hexadecimal format.

Since the program is self-loading, it can be loaded and executed at any time. Its primary purpose is to provide a means of obtaining a dump when a program comes to an abnormal end.

USING THE SELF-LOADING DUMP PROGRAM

In order to use the self-loading dump program, perform the following steps:

1. Run cards out of the card reader.
2. Place the properly initialized self-loading dump program deck in the card read hopper. (Initialization of the self-loading dump program is described below.)
3. Initialize the load-unit switches on the system control panel to address the card reader and press the load key.

INITIALIZING THE SELF-LOADING DUMP PROGRAM

The self-loading dump program is supplied by IBM in self-loading form on punched cards (that is, an object deck in self-loading form). Before the program can be used, it must be initialized to operate on the installation's machine. This initialization consists of modifying the END card (that is, the last card) in the IBM-supplied deck.

The END card initialization is described below:

Cols.	Description
17-20	0 followed by the three-digit hexadecimal address of the unit used to produce the listing.
21	0 if the listings are to be produced on a printer. 1 if the listings are to be produced on a 1052 Printer-Keyboard.
22-26	The storage size of the installation's machine in hexadecimal code. Valid sizes are: Decimal Value — Hexadecimal Equivalent 08192 — 02000 16384 — 04000 32768 — 08000 65536 — 10000
27-30	0 followed by the three-digit hexadecimal address of the machine's printer-keyboard (used for the end-of-job message). FFFF if a printer-keyboard is not available.

Note: I/O Support Package error indications will not be typed but will appear in the PSW. (See Appendix A.)

THE ABSOLUTE AND RELOCATING LOADERS

Two load programs are available from IBM: the Absolute Loader and the Relocating Loader. Both are designed to load assembled programs (from cards or tape) into storage for execution. They are available in two versions: one is assembled to occupy lower storage and the other to occupy higher storage on an 8K configuration. Certain installations may want loaders that reside elsewhere in storage and/or disable the printing of I/O error messages. For these reasons, both loaders are available from IBM in symbolic form on punched cards. See the description of the Loader Generator program for information on generating self-loading loaders. (Use of the Relocating Loader is recommended for users with greater than 8K main storage.)

PREPARING THE LOADERS FOR USE

The Absolute and Relocating Loaders are available from IBM in self-loading form on punched cards. Before either program can be used, it may require modification for operation on the installation's machine. This modification consists of altering the program's END card.

The END card is the last card in the deck. It must include the following:

Cols.	Description
17-20	Blank if the program to be loaded is on the same device as the loader. 0 followed by the three-digit hexadecimal address of the unit from which the program is to be loaded if it is on a different unit from the loader.
21-24	0 followed by the three-digit hexadecimal address of the installation's printer or printer-keyboard (used to produce operator messages). Blank if neither device is available.

LOADER OPTIONS AND MODIFICATIONS

The loader source programs available from IBM are designed for residence in lower storage, beginning at location 128. The programs can be broken into the following general groups:

Introduction
I/O Routines
Loader Routines
Initial Entry Routine (IER)

They are organized as such so that the user can overlay the Initial Entry Routine with the beginning or end of his program, if he wishes. After loading, he can overlay the loader routines during execution and still use the loader's I/O routines if he is exercising that option.

If the loaders are to be modified for residence in higher storage, it is recommended that the groups be reorganized to make optimum use of available storage, as described below.

To modify the Absolute Loader for residence in upper storage, the following alterations to the source deck are necessary:

1. Remove the constant IOTA EQU * from the end of the deck. Insert the constant IOTA EQU *-160 in the beginning of the deck, in place of the comment card *IOTA EQU *-160.

2. Move the Initial Entry Routine from the end of the deck to the position specified by the comments following the new constant IOTA.

3. Move the Loader Routines (Hex-Bin Conversion Routine through the end of Constants Area) to precede the I/O Routines. The constants THE END and OMEGA should now precede the END card.

4. Alter the START card to the desired starting address of the new loader.

5. Assemble the modified deck and generate a self-loading deck using the LDRGEN program.

To modify the Relocating Loader for residence in upper storage, the following alterations to the source deck are necessary:

1. Remove the constant IOTA EQU * from the end of the deck. Insert the constant IOTA EQU *-160 in the beginning of the deck, in place of the comment card *IOTA EQU *-160.

2. Move the Initial Entry Routine from the end of the deck to the position specified by the comments following the new constant IOTA.

3. Move the Loader Routine (Hex-Bin Conversion Routine through end of Constants Area) to precede the I/O Routines. The constant OMEGA should still precede the END card.

4. Change the existing constants TOP, BELOW, and CTRSET to the following:

```
TOP         EQU       MON
BELOW       DC        A(LOAD2)
CTRSET      DC        XL4'80'
```

5. Alter the START card to the desired starting address of the new loader.

6. Assemble the modified deck and generate a self-loading deck using the LDRGEN program.

LOADING CAPACITY

The Relocating Loader available from IBM is set for a maximum storage size of 8K. To modify the Relocating Loader designed for residence in lower storage for a larger storage size than 8K, it is necessary to alter the constant TOP; this constant may be altered as described in the listing (the description of this alteration occurs just before the actual constant), or it may be altered to 131071 for 128K storage. The source deck should then be assembled and a new loader generated using the LDRGEN program.

USING THE LOADERS

To load an assembled program into storage for execution, the following two items are required:

1. An Absolute or Relocating Loader in self-loading form on punched cards.

2. The assembled program to be loaded. The program may exist on punched cards or magnetic tape.

To run a job, perform the following steps:

1. Run cards out of the card reader.

2. Place the Absolute or Relocating Loader in the reader hopper. The loader must be initialized as described under "Preparing the Loaders for Use."

3. Place the assembled program on the input unit from which it is to be loaded.

4. Set the load-unit switches on the system control panel to address the card reader, and press the load key.

LOADER GENERATOR PROGRAM

The self-loading Absolute and Relocating Loaders available from IBM reside in lower storage during execution (higher storage in an 8K configuration). They are not in a form suitable for relocation. Since installations may want self-loading loaders that reside elsewhere in storage, IBM supplies a means to create them. This involves the use of the IBM-supplied Loader Generator (LDRGEN) program.

IBM provides both the Absolute and Relocating Loaders in symbolic form on punched cards. To create a self-loading loader, one must assemble the associated symbolic deck. The assembled loader is then loaded into storage with the LDRGEN program.

The LDRGEN program is designed to regenerate assembled loaders into a form suitable for direct loading into storage -- that is, to make them self-loading. Figure 10 shows the sequence of operations required to produce a self-loading loader.

PREPARING THE LDRGEN DECK FOR ASSEMBLY

The LDRGEN program as supplied by IBM is in symbolic form on punched cards. Before the LDRGEN source deck can be assembled for use, the address of the card reader-punch upon which the self-loading loaders are to be written must be defined. This is accomplished by inserting an Equate card in the LDRGEN source deck just before the END card. The format of the Equate card is:

Name	Operation	Operand
OUTPUT	EQU	A decimal or hexadecimal self-defining value equivalent to the address of the output unit.

Once the Equate card has been inserted in the deck, the LDRGEN program can be assembled.

RUNNING A JOB

In order to produce a self-loading loader, both the assembled loader (to be regenerated in self-loading form) and the assembled LDRGEN program must be loaded into storage. Since neither is self-loading, a separate load program must be used. Neither of these programs can overlay the self-loading loader used to load them. Two such programs are available in self-loading form: the Absolute Loader and the Relocating Loader. The use of each is described in the following text.

Using the Absolute Loader

Since the Absolute Loader loads programs into the storage locations assigned to them by the assembler, care must be taken to ensure that none of the programs to be loaded overlays another.

To use the Absolute Loader, one must have:

1. The Absolute Loader in self-loading form.

2. An assembled LDRGEN program.

3. The assembled loader to be regenerated in self-loading form. Note that the storage locations at which the loader was assembled are the ones assigned to the self-loading loader produced by the LDRGEN program.

4. Several Replace cards. These cards replace data in the LDRGEN program.

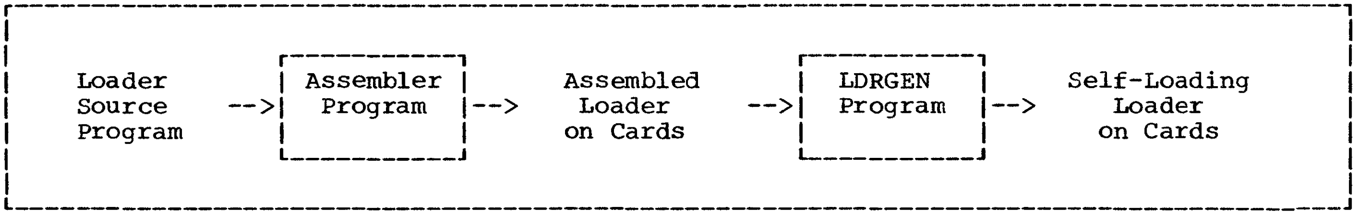


Figure 10. The LDRGEN Program

They define addresses in the assembled loader and, if applicable, specify the number of duplicate self-loading loader decks to be produced by the LDRGEN program. The addresses in the assembled loader that they specify and the places in the LDRGEN program at which these addresses are inserted are shown in the following lists.

Address of Symbol in Assembled Loader	Inserted at Symbolic Location in LDRGEN Program
ALPHA	ALPHAA
ALPHA	ALPHAB
BETA	BETAA
IOTA	IOTAA
OMEGA	OMEGAA

If duplicate self-loading decks are desired, a Replace card is used to insert the number of duplicates desired in a half-word area in the LDRGEN program named CON.

To run a job, the self-loading Absolute Loader is placed in the card read hopper. The assembled loader is placed behind it. The Replace cards are inserted in the assembled LDRGEN deck immediately after the Text cards, and the entire deck is placed behind the assembled loader in the card reader hopper. The card reader-punch upon which the self-loading loader is to be written is prepared for use. Then the load-unit switches on the system control panel are set to address the card reader, and the load key is pressed.

Using the Relocating Loader

Since the Relocating Loader loads programs into storage at the locations specified by Set Location Counter (SLC) cards, care must be taken when specifying these cards so as to ensure that the programs to be loaded do not overlay one another. SLC cards are described in the publication IBM System/360 Basic Programming Support Basic Utilities, Form C28-6505.

To use the Relocating Loader, one must have:

1. The Relocating Loader in self-loading form.

2. An assembled LDRGEN program.

3. The assembled loader to be regenerated in self-loading form. Note that the storage locations into which this program is loaded by the Relocating Loader are the ones assigned to the self-loading loader produced by the LDRGEN program.

4. A single Replace card, if duplicate self-loading decks are to be produced by the LDRGEN program. The Replace card inserts the number of duplicates in a half-word area in the LDRGEN program called CON. Note that the Replace cards that define addresses when the Absolute Loader is used are not required in this case. The Relocating Loader performs this function automatically.

To run a job, the self-loading Relocating Loader is placed in the card read hopper. The assembled loader is placed behind it. If applicable, the Replace card (for duplicate decks) is inserted in the assembled LDRGEN deck immediately after the Text cards, and the entire deck is placed behind the assembled loader in the card read hopper. A Load Terminate card, with LDRGEN in columns 17-22, is then placed in the card read hopper. The card reader-punch upon which LDRGEN will write the self-loading program(s) is prepared for use. The load-unit switches on the system control panel are set to address the card reader and the load key is pressed.

INPUT/OUTPUT SUPPORT PACKAGE

IBM supplies a group of routines designed to provide the programmer with the coding required to use input/output devices. This group of routines is called the Input/Output Support Package.

The routines are available in symbolic and assembled form on punched cards. The use of the IBM-supplied decks is exclusively the task of the programmer and, therefore, will not be described in this publication. For detailed information on the Input/Output Support Package, refer to the publication IBM System/360 Basic Programming Support Basic Utilities, Form C28-6505.

APPENDIX A: PROGRAM WAITS AND OPERATOR MESSAGES

A program wait occurs whenever the Basic Assembler or Basic Utility programs find it necessary to communicate with the operator. A program wait is indicated by the wait light on the system control panel.

When a program wait occurs, the three low-order bytes of the current PSW contain a three-character code, each character consisting of eight bits. This code identifies the reason for the program wait. This code can be displayed on the system control panel through use of the storage-select switch and the address switches. The storage-select switch is set to display the current PSW. The address switches are set to display the three low-order bytes in the PSW. Smaller System/360 models may display only the last byte or the last two bytes.

The first character of the code identifies the program being executed when the program wait occurred. The characters and the programs with which they are associated are shown in the following:

Character	Program
A	Assembler (both phases)
1	Assembler (Phase 1 only)
2	Assembler (Phase 2 only)
D	Dump Programs
G	Loader Generator
I	I/O Support Package
L	Loaders

The third character of the code can be one of the following:

A Operator action is necessary. No decision on the part of the operator is required.

D Operator action is necessary. The operator must, however, make a decision on the course of action to be taken.

S A program wait has occurred because of a machine error. The job cannot continue. SEREP interface has been set up, and the SEREP program should be loaded and executed. Save the SEREP[1] printout for Field Engineering analysis. If attention is required, Field Engineering should be notified. Once SEREP has completed its processing, the operator must re-initialize the system to rerun the error-interrupted job or to proceed with the next job.

W A program wait has occurred because of a program check. The job cannot continue.

I Operator information only

If the installation has provisions to print operator messages, the three-character code is printed on the output device. In some cases, the code is followed by a descriptive message. In others, it is followed by a string of hexadecimal characters which define the conditions that exist as the result of an erroneous I/O operation.

The following is a list (in alphabetical order) of all possible message codes and their hexadecimal equivalents. It is provided to enable easy translation of the display on the system control panel into the proper message code.

Message Code	Hexadecimal Equivalent	Message Code	Hexadecimal Equivalent
AIA	C1C9C1	IOD	C9F0C4
AID	C1C9C4	IOS	C9F0E2
AIS	C1C9E2	I1D	C9F1C4
AMS	C1D4E2	I1S	C9F1E2
APW	C1D7E6	I3S	C9F3E2
DEA	C4C5C1	LAA	D3C1C1
DRA	C4D9C1	LDA	D3C4C1
DTA	C4E3C1	LED	D3C5C4
GCS	C7C3E2	LKA	D3D2C1
GDD	C7C4C4	LOA	D3D6C1
GDS	C7C4E2	LPA	D3D7C1
GEA	C7C5C1	LUA	D3E4C1
GIA	C7C9C1	1EI	F1C5C9
GMS	C7D4E2	2EI	F2C5C9
GNS	C7D5E2	2HA	F2C8C1
IMS	C9D4E2	2SA	F2E2C1

[1] SEREP (System Environment Recording, Editing, and Printing) provides Field Engineering with detailed, accurate information about the system's environment at the time of a machine failure.

The program waits and messages presented in the following paragraphs are grouped according to the programs with which they are associated. Note that the I/O support package is built into each of the utility programs. Therefore, program waits listed under the I/O support package can occur during the execution of any of the utility programs. Where "Not typed" appears in parentheses after the three-character code, the code is displayed in the PSW but not typed on the output device.

TWO-PHASE DUMP PROGRAM

DEA END OF DUMP-PHASE 2

Phase 2 of the two-phase dump program has been completed.

Action: Proceed with the next job.

DRA MT NEXT INPUT REEL

Phase 2 of the two-phase dump program has encountered an end-of-reel condition on its input tape. The reel has been rewound and unloaded.

Action: Mount next reel on the input unit and make the device ready. Then, press the interrupt key on the system control panel to proceed with the job.

DTA MT NEW OUTPUT REEL

Phase 1 of the two-phase dump program has encountered an end-of-reel condition on its output tape. The reel has been rewound and unloaded.

Action: Mount a new work tape on the output unit. Then, press the interrupt key on the system control panel to proceed with the job.

SELF-LOADING DUMP PROGRAM

DEA END OF DUMP

The self-loading dump program has been completed.

Action: Proceed with the next job.

THE BASIC ASSEMBLER

AIA (Not typed)

The assembler has detected an I/O error which can be retried.

Action: Continue the program by depressing the Interrupt key. If after five retries the error still exists, load and execute SEREP.

AID (Not typed)

The assembler is unable to properly perform an I/O operation. The address of the associated I/O unit is contained in the low-order 11 bits of general register 2.

Action: The action taken varies with the type of operation in error.

- Tape Operation - Core location 44 hexadecimal (CSW unit status) should be interrogated.
  1. If the unit exception bit (bit 7) is set, an end-of-file condition on input or an end-of-reel condition on output has occurred. The address of the device causing the unit exception will be located in the lower half of register 2. The operator should change that tape and press the interrupt key to continue the job.
  2. If the unit exception bit is not set, the operator should press the interrupt key to retry the operation. If after five retries the condition still exists, the operator should dump all of storage and discontinue the job.
- Read - If a reader check light is on, the cards in the reader should be run out and reloaded. The operator should then press the interrupt key to retry the operation. If after one retry the condition still exists, the operator should mark the card in error and discontinue the job.
- Punch (1442) - Rerun the job.
- Punch (2540) - Mark bad card and press interrupt key to repeat the punch operation.
- Write Line - Press interrupt key to repeat operation.
- Space or Eject - Press interrupt key to repeat operation.

AIS (Not typed)

The assembler has detected an equipment failure while trying to execute an I/O operation. SEREP interface has been set up.

Action: Load and execute SEREP.

AMS (Not typed)

A machine check has occurred.

Action: Load and execute SEREP.

APW (Not typed)

A program check has occurred. The assembler program has been altered in some way.

Action: Dump all of storage and compare against listing to find the area altered. Correct if possible and rerun the job.

1EI

Phase 1 of the assembler has been completed.

Action: Proceed with Phase 2.

2EI

Phase 2 of the assembler has been completed.

Action: Proceed with next job.

2HA

Phase 2 of the assembler requires that blank cards be placed in the 1442 card hopper.

Action: Remove any cards in the 1442 card hopper, insert blank cards, and replace the cards just removed.

2SA

Phase 2 of the assembler requires a blank card at the punch station or blank cards in the 1442 card hopper.

Action: Remove cards from the hopper, run cards out of the 1442, and place them at the bottom of the cards just removed from the hopper. Place blank cards in the hopper and place the cards removed from the 1442 on top of the blanks. Make the unit ready and press the interrupt key on the system control panel to continue.

THE ABSOLUTE AND RELOCATING LOADERS

Several of the program waits associated with the load programs concern load control cards. References to these cards are made in abbreviated form in the descriptions that follow. The abbreviated titles and their equivalent names are given in the following list:

ESD	External Symbol Dictionary card
ICS	Include Segment card
LDT	Load Terminate card
REP	Replace card
RLD	Relocation List Dictionary card
SLC	Set Location Counter card

Note: The preceding cards are described in the publication IBM System/360 Basic Programming Support Basic Utilities, Form C28-6505.

LAA WAIT

The relocating loader has encountered an invalid RLD or ESD card in the program being loaded.

Action: Mark card and discontinue job.

LDA WAIT

The relocating loader has encountered duplicate entry points in the program being loaded.

Action: Discontinue job.

LED WAIT

One of the following situations has occurred:

1. The relocating loader has encountered an end-of-file condition without having read an LDT card.
2. The absolute loader has encountered an end-of-file condition without having read an END card.

Action: Discontinue job if the program is being loaded from tape. If the program is being loaded from cards, make the reader not ready. A card with a 12-2-9 punch in column one and the characters END or LDT (whichever is appropriate) in columns two through four is then placed in the reader hopper. The device is made ready and the interrupt key on the system control panel is pressed.

Note: The programmer should have included the proper LDT or END card in his source program. The operator action described in the preceding does not guarantee proper execution of the user's program.

LKA (Not typed)

The absolute or relocating loader has encountered an invalid SLC, ICS, or REP card in the program being loaded. This message is displayed but not typed for an invalid hexadecimal character.

Action: Mark card and discontinue job.

LOA WAIT

An attempt has been made to load a program into main storage locations reserved for use by the absolute or relocating loader.

Action: Discontinue job.

LPA (Not typed)

A program check has occurred. Note that this wait can occur during the execution of any program loaded into storage by either the Absolute or Relocating Program Loader.

Action: Discontinue job.

LUA WAIT

The relocating loader has encountered an undefined symbol in an SLC, ESD type 2, or LDT card in the program being loaded.

Action: Mark card and discontinue job.

INPUT/OUTPUT SUPPORT PACKAGE

The Input/Output Support Package is used by the IBM-supplied utility programs and by the programmer. In the case of the utility programs, the I/O package is built in prior to their distribution.

When the input/output support routines are unable to properly execute an I/O operation, a program wait occurs to notify the operator of the unusual condition, and SEREP Interface is set up. An operator message accompanies the program wait if the installation has provisions for printing messages.

The Input/Output Support Package has three levels of messages. They are:

1. CCC
2. CCC IOOPSW CSW
3. CCC IOOPSW CSW SBYTES

where:

CCC
is the three-character code which identifies the reason for the message.

IOOPSW
is the contents of the old input/output program status word in hexadecimal notation. The channel and unit number of the I/O device in error is contained in bits 21-31 of this word.

CSW
is the contents of the channel status word associated with the operation in error. It is in hexadecimal notation.

SBYTES
is the contents of the six sense bytes in hexadecimal notation.

All three levels will only appear when the full complement of error message expansions is included. The Basic Utility Programs, other than the Basic I/O Support Package, contain only the first level.

IMS (Not typed)

A machine check has occurred.

Action: Load and execute SEREP.

IOD IOOPSW CSW SBYTES

The input/output support package is unable to properly perform an I/O operation.

Action: The action taken varies with the operation in error:

- Tape - If unit is not ready, make ready and press console Interrupt to retry operation. If retry is unsuccessful, discontinue job.

- Punch Card - If the punch is not ready or out of cards, make it ready and press console Interrupt to retry the punch operation. If retry is unsuccessful, discontinue job.

- Read Card - If the reader is not ready or out of cards, make it ready and press console Interrupt to retry the read operation. If retry is unsuccessful, discontinue job.
- Write a Line - Press interrupt key to repeat the operation. If retry is unsuccessful, discontinue job.
- Skip or Space - Press interrupt key to repeat the operation. If retry is unsuccessful, discontinue job.

IOS IOOPSW CSW SBYTES

The input/output support package is unable to properly execute an operation. The standard retries have been attempted and the error persists.

Action: Load and execute SEREP.

I1D IOOPSW CSW SBYTES

One of the following has occurred:

1. A request to start an I/O operation has been rejected because of a programming error. In this case, the busy bit (bit 35) in the channel status word is off.
2. An overlapped I/O operation has been completed unsuccessfully while an attempt was being made to start a new operation. In this case, the busy bit in the channel status word is on.

Action: If the busy bit in the channel status word is off, press the interrupt key on the system control panel to repeat the request for an I/O operation. If the operation is again rejected, discontinue the job and call the customer engineer. If the busy bit in the channel status word is on, rerun the job.

I1S IOOPSW CSW SBYTES

One of the following has occurred:

1. A request to start an input/output operation has been rejected because of a machine error.
2. An overlapped I/O operation has been completed unsuccessfully while an attempt was being made to start a new operation.

Action: Load and execute SEREP.

I3S IOOPSW CSW SBYTES

The Input/Output Support Package attempted to use an I/O device which was not operational or not available.

Action: Load and execute SEREP.

LOADER GENERATOR PROGRAM

GCS (Not typed)

A channel error has occurred.

Action: Load and execute SEREP.

GDD (Not typed)

The LDRGEN program has attempted to punch a card but the operation resulted in an error.

Action: Mark the erroneously punched card and press the interrupt key to repeat the operation.

GDS (Not typed)

A device failure has occurred.

Action: Load and execute SEREP.

GEA (Not typed)

The LDRGEN program has been executed. This is a normal end-of-job situation.

Action: Proceed with next job.

GIA (Not typed)

The punch unit has run out of blank cards.

Action: Place blank cards in the punch hopper and press interrupt key to continue job.

GMS (Not typed)

Machine check has occurred.

Action: Load and execute SEREP.

GNS (Not typed)

The device specified as the output unit in the LDRGEN program is not available.

Action: Load and execute SEREP.

INDEX

READER'S COMMENTS

Title: IBM System/360 Basic Programming Support
Operating Guide for Basic Assembler and Utilities
360P-UT-017, -018, -019, -020
360P-AS-021

Form: C28-6557-1

Is the material:	Yes	No
Easy to Read?	___	___
Well organized?	___	___
Complete?	___	___
Well illustrated?	___	___
Accurate?	___	___
Suitable for its intended audience?	___	___

How did you use this publication?

___ As an introduction to the subject

___ For additional knowledge

Other ______________________________

fold

Please check the items that describe your position:

___ Customer personnel
___ IBM personnel
___ Manager
___ Systems Analyst
___ Operator
___ Programmer
___ Customer Engineer
___ Instructor
___ Sales Representative
___ Systems Engineer
___ Trainee
Other ______________

Please check specific criticism(s), give page number(s), and explain below:

___ Clarification on page(s)
___ Addition on page(s)
___ Deletion on page(s)
___ Error on page(s)

CUT ALONG LINE

Explanation:

fold

Name ______________________________

Address ______________________________

FOLD ON TWO LINES, STAPLE AND MAIL
No Postage Necessary if Mailed in U.S.A.

staple

old

fol

FIRST CLASS
PERMIT NO. 81

POUGHKEEPSIE, N. Y.

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

POSTAGE WILL BE PAID BY

IBM CORPORATION
P. O. BOX 390
POUGHKEEPSIE, N. Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS
DEPT. D58

old

fold

IBM®

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