File Number S360-32
Form C24-5042-1

DOS
TOS

IBM

Systems Reference Library

IBM System/360 Disk and Tape Operating Systems Utility Macros Specifications

This reference publication describes Multiprogramming Support (MPS) Utility Macro-Instructions for use with the Disk and Tape Operating Systems. The reader should be familiar with the following publications:

1. IBM System/360 Disk Operating System, System Control and System Service Programs, Form C24-5036.
2. IBM System/360 Tape Operating System, System Control and System Service Programs, Form C24-5034.
3. IBM System/360 Disk Operating System, Supervisor and Input/Output Macros, Form C24-5037.
4. IBM System/360 Tape Operating System, Supervisor and Input/Output Macros, Form C24-5035.
5. IBM System/360 Disk and Tape Operating Systems Assembler Specifications, Form C24-3414.
6. For titles and abstracts of other associated publications, see the IBM System/360 Bibliography, Form A22-6822.

PREFACE

This reference publication describes the Multiprogramming Support (MPS) utility macro-instructions available with the Disk and Tape Operating Systems.

It is intended as a guide for the programmer using the utility macro-instructions to create simple file-to-file routines for operation in a multiprogramming environment under the Disk or Tape Control Programs, as described in the System Control and System Service publications listed on the cover page of this publication.

This publication deals with utility macro-instructions, source language program illustrations for generating a utility program via these macro-instructions, and the process for the assembly and initiation of generated utility programs.

Second Edition, October 1966

This edition, Form C24-5042-1, is a major revision of, and obsoletes, Form C24-5042-0. Changes are indicated by a vertical line to the left of the affected text and to the left of affected parts of figures. A dot (●) next to a figure title or page number indicates that the entire figure or page should be reviewed.

Pages that have been affected are: 1, 2, 5, 7, 9, 11-26, 28-37, 39, 41 and 42.

CONTENTS

PURPOSE OF THE MPS UTILITY MACRO-INSTRUCTIONS

The utility macro-instructions provide the user of the System/360 Disk or Tape Operating System with a direct and easy means of generating efficient file-to-file utility programs tailored to the user's specific needs. The generated utility programs are designed to operate as foreground programs within the multiprogramming environment of the Disk or Tape Operating System, but can also operate as background programs.

Multiprogramming Support (MPS) has been provided in the Disk and Tape Operating Systems for the purpose of increasing total system throughput. In a single-program environment, storage available to the system may not be efficiently utilized, I/O channel utilization may not approach capacity, and many I/O devices attached to the system may be idle. Moreover, in a single-program environment, the program under execution frequently cannot utilize the CPU until some event has been satisfied. Most typically, this event is the completion of an I/O request.

The utility programs created by means of the MPS utility macro-instructions perform file-to-file operations concurrently with the execution of other programs. A typical use of a utility program generated by means of the MPS utility macro-instructions would be tape-to-printer operation executed concurrently with normal batched job processing.

Utility programs generated via the utility macro-instructions offer several distinct advantages over other utility programs.
These are:

- ability to add optional user code
- self-relocating capability of generated utility programs
- simple parameter selection

SCOPE OF THE MPS UTILITY MACRO-INSTRUCTIONS

Utility programs can be generated via utility macro-instructions to support any combination of the following media:

$$\left\{\begin{matrix}\text{card}\\\text{tape}\\\text{disk}\end{matrix}\right\}\quad\text{to}\quad\left\{\begin{matrix}\text{card}\\\text{tape}\\\text{disk}\\\text{printer}\end{matrix}\right\}$$

In addition, utility macro-instructions provide the ability to write and/or read using the IBM 1052 Printer-Keyboard.

The customer is given the choice of generating one or more utility programs for each type of file-to-file operation; the number of programs being a function of his requirements.

The generated utility programs do not require control information from the operator or job input stream, unless it is required by optional user-supplied routines.

MACHINE REQUIREMENTS

The minimum machine configuration for the generation of MPS utility programs is the minimum system configuration required to assemble a program in DOS/360 or TOS/360.

FOR SYSTEM RESIDENCE

- IBM 2311 Disk Storage Drive (DOS/360), or
- IBM 2401, 2402, 2403, 2404, and 2415 Magnetic Tape Drives (TOS/360). A 7-track tape drive, with the Data Conversion Feature, may be used, but 9-track is recommended.

FOR CONTROL-STATEMENT LOADING

- IBM 1052 Printer-Keyboard, or
- IBM 1442 Card Reader, or
- IBM 2501 Card Reader, or
- IBM 2520 Card Reader, or
- IBM 2540 Card Read-Punch, or
- IBM 2311 Disk Storage Drive, or
- IBM 2401, 2402, 2403, 2404 and 2415 Magnetic Tape Drives.

Note that foreground programs can only utilize a printer-keyboard and a card reader for control statement loading.

FOR OPERATOR-COMMUNICATION

- IBM 1052 Printer-Keyboard

FOR FOREGROUND UTILITY PROGRAM OPERATIONS

- IBM System/360 processing unit with a minimum of 32K bytes of main storage.
- Input/Output devices required by the specific program. Supported devices include:

  IBM 1052 Printer-Keyboard

  IBM 1442 Card Reader

  IBM 2501 Card Reader

  IBM 2520 Card Reader

  IBM 2540 Card Reader

  IBM 1442 Card Punch

IBM 2520 Card Punch

IBM 2540 Card Punch

IBM 1403 Printer

IBM 1404 Printer (continuous forms only)

IBM 1443 Printer

IBM 1445 Printer

IBM 2311 Disk Storage Drive

IBM 2401, 2402, 2403, 2404 and 2415 Magnetic Tape Drives. For tape input and/or output, the utility macro-instructions require the optional Supervisor "set mode" facility, with mode indicated on the ASSGN statements, where applicable. (Refer to System Control and System Service and System Generation and Maintenance publications listed on the front cover.)

UTILITY PROGRAM GENERATION

This section describes concepts and procedures relating to the generation of a utility program via the MPS utility macro-instructions. The section is divided into three main parts. The first part deals with general programming information relative to the coding of source language statements. The second part presents detailed information on the macro-instruction prototype statements. For convenient presentation, the macro-instructions are divided into an input group and an output group. The third part presents examples of source-language programs for file-to-file operations, illustrating the use of the MPS utility macro-instructions.

The programmer coding required for the generation of a file-to-file utility program is minimal. For example, a self-relocating card-to-card utility program may be assembled from the following:

NAME	OPERATION	OPERAND
	START	0
	INCARD	
	OUTCARD	
	END	

The assembler interprets the macro-instructions and calls in the appropriate macro definition for each macro from the assembler sub-library of the source statement library. The macro definition serves to expand the single macro-instruction into a series of source statements replacing dummy variables with the parameters (if any) specified in the operand field of the macro-instruction.

The object program generated via assembly may be cataloged in the core image library. It can be executed as either a foreground or background program.

MACRO-INSTRUCTION FORMAT

The format of the macro-instruction prototype dictates the form in which the macros must be written in the problem program. The following general rules apply:

1. The name field of the macro-instruction may contain a symbolic address, or may be left blank.

2. The operation field of the macro-instruction must contain exactly the same mnemonic operation code as the prototype (for example, INCARD).

3. The parameters in the operand field of a macro-instruction must be written in the same format as those in the operand field of the prototype.

The MPS utility macro-instruction parameters may be written in any order. Any parameters that are not required may be omitted. Multiple parameters may either be punched on a single card, or on separate cards with an alphameric punch in column 72. The parameters must always be separated by a comma.

The macro-instruction conventions used to illustrate macro-instructions are as follows:

1. Uppercase letters and punctuation marks represent information that must be coded exactly as shown.

2. Lower case letters represent information that must be supplied by the programmer. The letter b always indicates one blank space, and where a parameter variable is concerned, lower case letters represent constants that must be supplied.

3. Options contained within braces { } represent alternatives; one of which must be chosen.

4. Information contained within brackets [ ] represents an option that can be omitted, (depending upon program requirements).

5. Underlined elements $\begin{bmatrix}\underline{\text{NO}}\\ \text{YES}\end{bmatrix}$ represent an assumed option in the event a parameter is omitted.

6. An ellipsis (a series of three periods) indicates that a variable number of items may be included.

7. When registers are permitted as register arguments, $\begin{bmatrix}\text{name1}\\ \text{(r)}\end{bmatrix}$ the register number must be enclosed within parentheses.

UTILITY PROGRAM ORGANIZATION

Each generated utility program may consist of the following components:

- optional processing sections (user-provided)
- input sections
- output sections

The processing sections consist of optional user routines. Typical functions performed by user routines might include selecting an output routine for each record, editing input data, inserting data into the output stream, or initializing parameters for input and output sections.

The only restriction imposed by the utility macro-instructions on a user routine is that it normally must save and restore registers 0, 1, 13, 14, and 15 used for communication between the input and output sections. Register 13 is reserved for future expansion of the utility macro-instructions.

For foreground operations, optional user processing routines incorporated into the utility programs must comply with the conditions set forth for foreground programs, and must interface with the system as defined in DOS/360 and TOS/360 Control Programs reference publications cited on the cover page of this document.

The input sections of each utility program will normally consist of the coding generated by one or more INCARD, INTAPE, INDISK, or INLOG macro-instructions.

Each input macro-instruction generates all required instructions for initialization, data input, and buffer area definitions. The initialization which primarily consists of the coding required for self-relocation, is within the input areas, except for INLOG. INLOG accesses a user-provided area for input data, and does not generate a buffer area.

The output sections will normally consist of coding generated by one or more OUTCARD, OUTAPE, OUTDISK, OUTPRT, or OUTLOG macro-instructions.

Each output macro-instruction generates an independent output routine including instructions for initialization, data output, and buffer area definitions. The initialization, which primarily consists of the coding required for self-relocation, is within the output areas, except for OUTLOG. OUTLOG accesses a user-provided area, and does not generate a buffer area.

INTERFACE BETWEEN PROGRAM SECTIONS

A generated utility program will normally be entered at the initialization routine of the input section, unless the input section is preceded by an optional user routine. Following each read operation, control will pass to the output initialization routine or to an optional user routine. The address of the input routine reentry point is stored in register 14. The output routine normally will do a write operation, and then return to an input or user routine via register 14. Provision is made for the user to specify multiple outputs by specifying the RETURN=NO parameter in the output macro-instruction.

Figure 1 shows sample techniques for passing control between the generated utility program routines.

INTERFACE		PROCEDURE
User Routine 1	To Input Routine	via next sequential instruction
Input Routine	To User Routine 2	via next sequential instruction
	To Output Routine	via next sequential instruction or through User Routine 2
User Routine 2	To Output Routine	via next sequential instruction
	To Input Routine	via address in register 14
Output Routine	To User Routine 3	via next sequential instruction with RETURN=NO specified
	To Input Routine	via address in register 14
User Routine 3	To User Routine 2	via branch
	To Input Routine	via address in register 14
Note: All user routines are optional and user-supplied.		

●Figure 1. Techniques for Passing Control Between Utility Program Routines

REGISTER USAGE CONVENTIONS

General registers 0, 1, 14, and 15 are used by the MPS utility macro-instructions for both information transmission and control. Registers 2 through 12 are restored by each utility macro-generated routine. Register usage may be summarized as follows:

REGISTER	INPUT ROUTINE	OUTPUT ROUTINE
0	Set to record address	Not modified
1	Set to length of the record	Not modified
2-12	Not modified	Not modified
13	Reserved	Reserved
14	Set to return address	Not modified
15	Used as base register	Used as base register

Control of the output routines is via zero and/or nonzero values in registers 0 and 1 as follows:

Reg. 0	Reg. 1	Output Routine Response	Comments
≠0	≠0	Write output record.	Normal exit from input routine.
0	0	CLOSE tape or disk file and end job <u>or</u> End job (printer or card output).	Normal end of data signal from input routine.
≠0	0	Do not write record.	Requires a user routine to generate this condition.
0	≠0	Same as for Reg 0=0 and Reg 1=0 except no end of job.	Requires a user routine to generate this cond. tion.

Normal processing is accomplished by not modifying the contents of registers 0 and 1. End of input is indicated by zeros in registers 0 and 1. To cancel the output of a record, the user sets register 1 to zero, or he returns directly to the input routine (via BR 14). If the output routine finds that only register 1 contains zero, it immediately returns to the input routine. If the user wishes to signal end of file without signaling end of job, he sets register 0 to zero and register 1 to some non-zero value. This will cause the output routine to empty its buffer areas, CLOSE its output file, and return.

SELF-RELOCATION

The utility programs generated by the MPS utility macro-instructions are self-relocating. If the optional user routines incorporated into the program are not self-relocating, then the resulting utility program is not self-relocating and must be linkage-edited for subsequent execution at a fixed location of main storage. A self-relocating program is one which can be executed at any location in main storage. A self-relocating program normally is given a linkage-edit address of zero. For example:

```
   SOURCE STATEMENTS
          REPRO
          PHASE EXAMPLE, +0            +0 ORIGIN IMPLIES SELF-RELOCATION
          PRINT NOGEN
PROGRAM   START 0
          BALR  15,0
          USING *,15
*    ROUTINE TO RELOCATE ADDRESS CONSTANTS
          LA    1,PRINTCCW             RELOCATE CCW ADDRESS
          ST    1,PRINTCCB+8            IN CCB FOR PRINTER
          LA    1,TAPECCW              RELOCATE CCW ADDRESS
          ST    1,TAPECCB+8             IN CCB FOR INPUT TAPE
          LA    1,EOFTAPE             *RELOCATE*****
          ST    1,AEOFTAPE            * PROGRAM    *
          LA    1,CHA12               *  ADDRESS   *
          ST    1,ACHA12              ****CONSTANTS*
          IC    2,PRINTCCW             SAVE PRINT CCW OP CODE
          LA    1,OUTAREA              RELOCATE OUTPUT AREA ADDRESS
          ST    1,PRINTCCW              IN PRINTER CCW
          STC   2,PRINTCCW             RESTORE PRINT CCW OP CODE
          LA    1,INAREA               RELOCATE INPUT AREA ADDRESS
          ST    1,TAPECCW               IN TAPE CCW
          MVI   TAPECCW,2              SET TAPE CCW OP CODE TO READ
*    MAIN ROUTINE...READ TAPE AND PRINT RECORDS
READTAPE  LA    1,TAPECCB              GET CCB ADDRESS
```

```
          EXCP   (1)                       READ ONE RECORD FROM TAPE
          WAIT   (1)                       WAIT FOR COMPL. OF I/O
          L      10,AEOFTAPE               GET ADDRESS OF TAPE EOF ROUTINE
          BAL    14,CHECK                  GO TO UNIT EXCEPTION SUBROUTINE
          MVC    OUTAREA(10),INAREA        EDIT RECORD
          MVC    OUTAREA+15(70),INAREA+10  IN
          MVC    OUTAREA+90(20),INAREA+80  OUTPUT AREA
          LA     1,PRINTCCB                GET CCB ADDRESS
          EXCP   (1)                       PRINT EDITED RECORD
          WAIT   (1)                       WAIT FOR COMPL. OF I/O
          L      10,ACHA12                 GET ADDRESS OF CHAN 12 ROUTINE
          BAL    14,CHECK                  GO TO UNIT EXCEPTION SUBROUTINE
          B      READTAPE
CHECK     TM     4(1),1                    CHECK FOR UNIT EXEC. IN CCB
          BCR    1,10                       YES-GO TO PROPER ROUTINE
          BR     14                         NO-RETURN TO MAINLINE
CHA12     MVI    PRINTCCW,X'8B'            SET SK TO CHAN 1 OP CODE
          EXCP   (1)                       SK TO CHAN 1 IMMEDIATELY
          WAIT   (1)                       WAIT FOR COMPL. OF I/O
          MVI    PRINTCCW,9                SET PRINTER OP CODE TO WRITE
          BR     14                        RETURN TO MAINLINE
EOFTAPE   EOJ                              END OF JOB
          CNOP   0,4                       ALIGN CCB'S TO FULL WORD
PRINTCCB  CCB    SYS004,PRINTCCW,X'0400'
TAPECCB   CCB    SYS001,TAPECCW
PRINTCCW  CCW    9,OUTAREA,X'20',110
TAPECCW   CCW    2,INAREA,X'20',100
AEOFTAPE  DC     A(EOFTAPE)
ACHA12    DC     A(CHA12)
OUTAREA   DC     CL110' '
INAREA    DC     CL100' '
          END    PROGRAM
```

The input/output routines also adjust references to optional user-supplied locations, the optional "name" operands of macro parameters (such as STCTL, RECSIZ, LBL), etc.

Except for the macro-instructions INLOG and OUTLOG, the code required for self-relocation is placed in an I/O buffer and subsequently overwritten. All relocation takes place prior to the start of I/O operations.

INTERFACE WITH THE DATA MANAGEMENT FACILITIES

The standard error-recovery facilities of the DOS/360 or TOS/360 systems are used for card readers, 1442 card punch, and printers. For the 2520 and 2540 card punches, the utility macro-instructions provide error recovery. For tape and disk I/O, the system attempts error recovery and signals an error to the utility routine only if the error is not corrected.

The standard OPEN and CLOSE transient routines provided in TOS/360 and DOS/360 are used by the generated utility routines for tape and disk files. (Refer to the Supervisor and Input/Output Macros publications.) These functions include standard label processing, and for files bearing standard labels, alternate tape units, multiple extent sequential disk files, and multi-volume tape files. Tape files should be properly positioned prior to OPEN and are not repositioned following CLOSE. Tape files not bearing standard labels may include multiple-volume files and multiple-file volumes; in those cases the operator must control processing of additional volumes and/or files.

The utility programs handle all files as if they were sequentially organized. Tape and disk utility macro-instructions process fixed- and undefined-length records, blocked or unblocked. The MPS utility macros generate instructions for buffer rotation and for optional blocking or deblocking. Also, an option is provided in the INTAPE macro-instruction for either using checkpoint information as data or for bypassing checkpoint records.

INPUT MACRO-INSTRUCTIONS

Except for INDISK, the macro-instructions described as follows apply equally to the DOS/360 Control Program and the TOS/360 Control Program.

The following apply to all the input macro-instructions, except INLOG:

1. The values contained in the "namex" parameters may be set by an optional user initialization routine prior to processing each file via the macro input routine.

2. SYSnnn is a programmer logical unit, i.e., nnn is three digits.

INCARD (CARD INPUT MACRO-INSTRUCTION)

The INCARD macro-instruction causes records to be read from the designated card reader. A 160 byte buffer is generated for two input records. The user's card input is assumed to be in BCD or EBCDIC format. End-of-card input is signaled by an EOF condition on the card reader.

NAME	OPERATION	OPERAND
[name]	INCARD	$\left[\text{UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$

$\left[\text{UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$

This parameter specifies the symbolic name of the input logical unit.

INTAPE (TAPE INPUT MACRO INSTRUCTION)

The INTAPE macro-instruction generates initialization, a file OPEN, a tape read routine, an end-of-file test, and a file CLOSE.

If a tape not bearing standard labels is indicated but the tape contains label(s), the labels are considered to be data. This permits the user to check a nonstandard label via an optional routine.

Upon encountering each tape mark on a volume not bearing standard labels, the operator is given the option to initiate processing of additional files or to signal end-of-input.

NAME	OPERATION	OPERAND
[name]	INTAPE	BUFSIZ=n $\left[\text{,RECSIZ=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$ $\left[\text{,UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$ $\left[\text{,FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}\right]$ $\left[\text{,LBL=name2}\right]$ $\left[\text{,ERROR=}\begin{Bmatrix}\text{SKIP}\\ \underline{\text{IGNORE}}\\ \text{name3}\end{Bmatrix}\right]$ $\left[\text{,CHKPT=}\begin{Bmatrix}\text{NO}\\ \text{name4}\end{Bmatrix}\right]$

BUFSIZ=n

This required parameter specifies the maximum size of an input block and is used to allocate buffer space for two blocks of that length. Shorter records can be read, in which case, actual record length is determined by the input routine. Longer records will be truncated. The first time record truncation occurs, the operator is given the option of canceling the job. The minimum buffer size is 20 bytes.

$\left[,\text{RECSIZ=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$

This parameter specifies the size of logical records within blocks. If omitted, records are assumed to be unblocked.

If RECSIZ=m, then m specifies the logical record size and the value is assembled into the program. A value of zero for m indicates unblocked records, and is equivalent to omitting the RECSIZ operand.

If RECSIZ=name1, the content of the fullword symbolic location "name1" is interrogated at execution time when the input file is OPENed. The value found will be used in subsequent deblocking, unless the value is zero. If the value is zero, records are treated as unblocked.

If records are blocked, any incomplete logical record at the end of a block will be transmitted without modification. The first time this condition occurs, the operator is given the option of canceling the job. If the operator chooses to continue, an optional user routine may modify or delete the record.

$\left[,\text{UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$

This parameter specifies the symbolic name of the input logical unit.

$\left[,\text{FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}\right]$

This parameter specifies that the file bears standard labels. If omitted the file is treated as unlabeled. A register, (r), may contain the address of the eight character filename. (Refer to the // VOL statement described in either one of the <u>System Control and System Service</u> publications listed on the front cover.)

[,LBL=name2]

This parameter specifies what label processing is to be performed. If omitted, no user label processing is attempted. If LBL=name2 is specified, the FILE parameter must be provided and the full-word area designated "name2" will be interpreted to determine user label processing procedures. If the area contains a zero value, the file must bear standard labels and any optional user-header record(s) will be skipped in positioning the tape past the next tape mark. If the area contains a negative value, the file is treated as unlabeled. A positive, nonzero value is assumed to be the address of a user routine to process user-header or trailer records. This address will be relocated, if necessary by the INTAPE macro. The user routine releases control via the LBRET macro instruction. (Refer to the <u>Supervisor and Input/Output Macros</u> publications.)

$\left[,\text{ERROR=}\begin{Bmatrix}\text{SKIP}\\ \underline{\text{IGNORE}}\\ \text{name3}\end{Bmatrix}\right]$

The ERROR parameter specifies error-handling procedures after the supervisor has detected an error. SKIP causes the input routine to bypass any unreadable record. IGNORE causes the error condition to be ignored and the record to be processed. If ERROR=name3 is specified, then "name3" must be the symbolic location of a user-provided routine to which control is passed when an unreadable record is encountered. When this occurs registers 0 and 1 will contain the block address and length, respectively. If the user returns (via BR 14) with register 1=0, the record is bypassed, but if register 1 $\neq$ 0, the record is processed as though no error occurred.

If records are blocked, a user-provided routine (name3) receives control only once for each unreadable block.

$\left[,\text{CHKPT=}\begin{Bmatrix}\text{NO}\\ \text{name4}\end{Bmatrix}\right]$

This parameter specifies the handling of checkpoint records. If omitted, checkpoint records will be bypassed. If a tape does not contain checkpoint records, CHKPT=NO should be indicated. If the full-word area, symbolically designated "name4" is specified, it will be interrogated at execution time, and checkpoint records will be treated like data only if the value is nonzero; otherwise checkpoint records will be bypassed. For further information on checkpoint, refer to the Supervisor and Input/Output Macros publications.

INDISK (DISK INPUT MACRO INSTRUCTION)

The INDISK macro instruction generates all required initialization, a file OPEN, a disk read routine, and an end-of-file test, and a file CLOSE. The input unit is determined by the required label information. Any key fields on disk input are ignored.

NAME	OPERATION	OPERAND
[name]	INDISK	BUFSIZ=n $\left[,\text{RECSIZ=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$ $,\text{FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}$ $\left[,\text{LBL=name2}\right]$ $\left[,\text{ERROR=}\begin{Bmatrix}\text{SKIP}\\ \underline{\text{IGNORE}}\\ \text{name3}\end{Bmatrix}\right]$

BUFSIZ=n

This required parameter specifies the maximum size of an input block and will be used to allocate buffer space for two blocks of that length. Shorter records can be read, since actual record length is determined by the input routine.

Longer records will be truncated. The first time record truncation occurs, the operator has the option of canceling the job. Regardless of the physical length of records, each record will be read from each track within the extents of the file.

$$\left[\ ,\mathrm{RECSIZ}=\begin{Bmatrix}\mathrm{m}\\ \mathrm{name1}\end{Bmatrix}\right]$$

This parameter determines the size of logical records within blocks. If omitted, records are assumed to be unblocked.

If RECSIZ=m then m specifies the logical record size whose value is assembled into the program. A value of zero for m indicates unblocked records, and is equivalent to omitting the RECSIZ operand.

If RECSIZ=name1, the content of a full-word, symbolic location, "name1", is interrogated at execution time, when the input file is OPENed. The value found will be used in subsequent deblocking, unless the value is zero. If the value is zero, records are treated as unblocked.

If records are blocked any incomplete logical record at the end of a block will be transmitted without modification. The first time this condition occurs, the operator has the option of canceling the job. If the operator chooses to continue, an optional user routine may modify or delete the record.

$$,\mathrm{FILE}=\begin{Bmatrix}\mathrm{filename}\\ \mathrm{(r)}\end{Bmatrix}$$

This required parameter identifies label information. A register (r) may contain the address of the eight character filename. Refer to the // VOL statement in either System Control and System Service publications listed on the front cover.

[,LBL=name2]

This parameter specifies user label processing procedures. The disk file must bear a standard label. If omitted, any additional user labels are treated as data. If LBL=name2 is specified, the full-word area symbolically designated "name2", will be interpreted prior to OPEN, to determine label processing procedure. If the area contains zero or is negative, files must bear standard labels and additional user labels will be treated as data, just as though the parameter were omitted. Otherwise, the area must contain the address of a user routine to process additional user labels. This address will be relocated, if necessary by the INDISK macro. The user routine releases control via the LBRET macro instruction. (Refer to the Supervisor and Input/Output Macros publications.)

$$\left[\ ,\mathrm{ERROR}=\begin{Bmatrix}\mathrm{SKIP}\\ \underline{\mathrm{IGNORE}}\\ \mathrm{name3}\end{Bmatrix}\right]$$

This parameter specifies error-handling procedures after the supervisor has detected an error. SKIP causes the input routine to bypass any unreadable record. IGNORE causes the error condition to be ignored and the record to be processed. If ERROR=name3 is specified, then "name3" must be the symbolic location of a user-provided routine to which control is passed when an unreadable record is encountered. When this occurs, registers 0 and 1 contain the block address and length respectively. If the user returns with register 1 = 0 (via BR 14), the record is bypassed, but if register 1 $\neq$ 0, the error condition is ignored.

If records are blocked prior to processing of the block, a user-provided routine (name3) receives control only once for each unreadable block.

INLOG (PRINTER-KEYBOARD INPUT MACRO INSTRUCTION)

The INLOG macro-instruction provides the facility to read information from the IBM 1052 Printer-Keyboard assigned to SYSLOG.

NAME	OPERATION	OPERAND
[name]	INLOG	$\text{BUFFER=}\left\{\begin{matrix}\text{name1}\\(r_b)\end{matrix}\right\}\left[\text{,COUNT=}\left\{\begin{matrix}n\\(r_c)\end{matrix}\right\}\right]$

$\text{BUFFER=}\left\{\begin{matrix}\text{name1}\\(r_b)\end{matrix}\right\}$

This required parameter designates the location of the user's buffer area (name1) or a register (r_b) containing the buffer address.

$\left[\text{,COUNT=}\left\{\begin{matrix}n\\(r_c)\end{matrix}\right\}\right]$

This parameter specifies the number of characters, or the register (r_c) containing the number of characters, to be read. If BUFFER=name1, the COUNT parameter may be omitted, in which case the number of characters read will be equal to the buffer length. The number of characters may not exceed 256. If the number of typed characters exceeds the indicated count, the message is truncated. Register 1 contains the number of characters read, not including end of block Ⓑ .

Note: No check will be made to ensure that SYSLOG is assigned to a readable 1052.

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
INCARD	$\left[\text{UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$	If other than SYS001 is wanted.	nnn = the programmer logical unit containing input. A 160 byte buffer is generated.
INTAPE	BUFSIZ=n	For each file.	n=maximum size of input block. A buffer of 2n bytes is generated.
	$\left[\text{,RECSIZ=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$	If logical record size within blocks is to be specified.	m=logical record size name1=symbolic location containing logical record size. If omitted or value=0, records are treated as unblocked.
	$\left[\text{,UNIT=}\begin{Bmatrix}\underline{\text{SYS001}}\\ \text{SYSnnn}\end{Bmatrix}\right]$	If other than SYS001 is wanted.	nnn = the programmer logical unit containing input.
	$\left[\text{,FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}\right]$	For each file bearing standard labels.	Identifies the filename for label processing. r=register pointing to the 8 character filename.
	[,LBL=name2]	If special label processing is desired.	name2=full-word area to be interpreted. If value=0, standard labels are assumed and user labels are bypassed. If value is positive, value= address of user HDR record processing routine. If value=negative, or the parameter is omitted no label processing is attempted.
	$\left[\text{,ERROR=}\begin{Bmatrix}\text{SKIP}\\ \underline{\text{IGNORE}}\\ \text{name3}\end{Bmatrix}\right]$	If read errors are not to be entirely ignored.	SKIP=bypass any unreadable block. IGNORE= ignore error condition. name3=symbolic address of a user's routine for unreadable block processing.
	$\left[\text{,CHKPT=}\begin{Bmatrix}\text{name4}\\ \text{NO}\end{Bmatrix}\right]$	If checkpoint records are to be treated as data. •	name4=full-word area to be interpreted. If value=non-zero, checkpoint will be treated as data. If value=zero or the parameter is

Figure 2. Summary of Input Macro Instructions (Page 1 of 2)

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
			omitted, checkpoint records are bypassed. If CHKPT=NO, check-point records are treated as data.
INDISK	BUFSIZ=n	For each file.	n=maximum size of input block. A buffer of 2n bytes is generated.
	$\left[,\text{RECSIZ=}\begin{Bmatrix}\text{m}\\\text{name1}\end{Bmatrix}\right]$	If logical record size within blocks is to be specified.	m=logical record size name1=symbolic address containing logical record size. If omitted or value=0, records are treated as unblocked.
	$,\text{FILE=}\begin{Bmatrix}\text{filename}\\\text{(r)}\end{Bmatrix}$	For each file.	Identifies the filename for label processing. r=register pointing to an 8 character filename.
	[,LBL=name2]	If additional user label processing is desired.	name2=full-word area (designated name2) to be interpreted. If value=0, additional user labels are treated like data. Otherwise, value is address of user's HDR label processing routine.
	$\left[,\text{ERROR=}\begin{Bmatrix}\text{SKIP}\\\underline{\text{IGNORE}}\\\text{name3}\end{Bmatrix}\right]$	If read errors are not to be entirely ignored.	SKIP=bypass any unreadable block. IGNORE= ignore error condition. name3=symbolic address of a user's routine for unreadable block processing.
INLOG	$\text{BUFFER=}\begin{Bmatrix}\text{name1}\\(r_b)\end{Bmatrix}$	For each input via the 1052.	name1=symbolic location of the user's buffer area. r_b=register containing the buffer address.
	$\left[,\text{COUNT=}\begin{Bmatrix}\text{n}\\(r_c)\end{Bmatrix}\right]$	If BUFFER=(r_b).	n=number of characters to be read. r_c=register containing the number of characters to be read.

Figure 2. Summary of Input Macro-Instructions (Page 2 of 2)

Except for OUTDISK, the macro-instructions described as follows apply equally to the DOS/360 Control Program and the TOS/360 Control Program.

The following apply to all output macros except OUTLOG:

1. The values contained in the "namex" parameter operands may be set by an optional user initialization routine prior to processing each file via the output macro instruction.

2. SYSnnn is a programmer logical unit, i.e., nnn is three digits.

OUTCARD (CARD OUTPUT MACRO-INSTRUCTION)

The OUTCARD macro-instruction causes records to be punched on cards.

NAME	OPERATION	OPERAND
[name]	OUTCARD	$\text{UNIT=}\left\{\begin{matrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{matrix}\right\} \left[\text{,STCTL=}\left\{\begin{matrix}\underline{\text{NO}}\\ \text{YES}\\ \text{name1}\end{matrix}\right\}\right] \left[\text{,DEVICE=}\left\{\begin{matrix}\underline{\text{1442}}\\ \text{2520}\\ \text{2540}\end{matrix}\right\}\right]$ $\left[\text{,RETURN=}\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}\right]$

$$\left[\text{UNIT=}\left\{\begin{matrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{matrix}\right\}\right]$$

This parameter specifies the symbolic name of the output unit.

$$\left[\text{,STCTL=}\left\{\begin{matrix}\underline{\text{NO}}\\ \text{YES}\\ \text{name1}\end{matrix}\right\}\right]$$

This parameter controls stacker selection. If STCTL=NO, unconditional selection of pocket 2 is implied, and input records will be punched with the first character in column 1.

If STCTL=YES, stacker selection is controlled by the first character of each input record (V or W, for pockets 1 and 2 respectively), and the input records will be punched with the second character in column 1 of the output records.

If STCTL=name1, then "name1" specifies the symbolic location of a full-word area which is interpreted at execution time to determine whether stacker selection is to be provided. If the word contains zero, selection of pocket 2 is unconditional, as though STCTL=NO had been specified. If the word contains a nonzero value, stacker selection becomes effective, as though STCTL=YES had been specified. The value is checked prior to punching each card.

$\left[\text{,DEVICE=}\begin{Bmatrix}\underline{1442}\\ 2520\\ 2540\end{Bmatrix}\right]$

This parameter controls error-recovery procedures. The parameter must be specified if the card punch is a 2520 or 2540, so that the appropriate error-recovery code will be provided in the output routine. The 1442 error recovery routines are supplied by the DOS/360 or TOS/360 control program.

Notes:

1. A buffer will be generated for 2 output records (160 bytes) for 1442 and 2520, or for 3 output records (240 bytes) for 2540.

2. When stacker selection is in effect, any leading character other than V (pocket 1) or W (pocket 2) will result in selection of pocket 2, and the leading character will be truncated. The first time this occurs, the operator has the option of canceling the job. Stacker-select characters V and W are not punched.

3. If a record that is to be punched exceeds 80 characters, not including stacker select control (first character), the record will be truncated. The first time this condition occurs, the operator has the option of canceling the job.

$\left[\text{,RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\right]$

This parameter governs the return to the input section or optional user section. If RETURN=YES, a return to the input or user routine is provided via register 14. RETURN=NO signifies that the user's own coding or additional output utility macros will follow.

OUTAPE (TAPE OUTPUT MACRO-INSTRUCTION)

The OUTAPE macro instruction generates initialization, a file OPEN, a tape write routine, an end-of-file test, and a file CLOSE.

If tape not bearing standard labels is indicated, no check is made for labels, and any label would be over-written by the first output record. At end of job the program writes two tape marks. The tape will then be backspaced so that it is positioned immediately after the first tape mark.

Upon encountering end-of-reel on each tape not bearing standard labels, a tapemark is written, the tape is rewound and unloaded, and the operator is instructed to mount an additional volume to be used for further output.

NAME	OPERATION	OPERAND
[name]	OUTAPE	BUFSIZ=n $\left[\text{,BLK=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$ $\left[\text{,UNIT=}\begin{Bmatrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{Bmatrix}\right]$ $\left[\text{,FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}\right]$ [,LBL=name2] $\left[\text{,RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\right]$

BUFSIZ=n

This required parameter specifies the maximum size of an output block and is used to allocate buffer space for two blocks of that length. The value of n must not be less than 20. This is the shortest record that is written. If unblocked records that are to be written are longer than the buffer space provided, they will be truncated. The first time this occurs, the operator is given the option of canceling the job.

$\left[\text{,BLK=}\begin{Bmatrix}\text{m}\\ \text{name1}\end{Bmatrix}\right]$

This parameter determines the number of logical records per output block. If the parameter is omitted, the output records are unblocked.

If BLK=m, then m specifies the blocking factor which is assembled into the program. A value of zero or one for m indicates unblocked records, and is equivalent to omitting the BLK operand.

If BLK=name1, the content of the full-word symbolic location, "name1", is interrogated at execution time when the file is OPENed. The value found is used as the blocking factor, unless the value is zero or one. If it is zero or one, output is unblocked.

Regardless of the blocking factor specified, the accumulated records are written immediately, as a block, if the next record threatens to exceed allocated buffer space. The first time this condition is encountered, the operator has the option of canceling the job.

The final output block may be shorter than previous blocks due to termination of input.

$\left[\text{,UNIT=}\begin{Bmatrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{Bmatrix}\right]$

This parameter specifies the symbolic output unit.

$\left[\text{,FILE=}\begin{Bmatrix}\text{filename}\\ \text{(r)}\end{Bmatrix}\right]$

This parameter specifies that the file bears standard labels. If omitted the file is treated as unlabeled. A register (r) may contain the address of the eight character filename. Refer to the // VOL statement contained in System Control and System Service publications listed on the front cover.

[,LBL=name2]

This parameter specifies what label processing is to be performed. If omitted, no user label processing is attempted. If LBL=name2 is specified, the FILE parameter must be provided and the fullword area designated "name2" will be interpreted to determine user label processing procedures. If the area contains a zero value, no user label processing is attempted and the file must bear an IBM Standard Label. If the area contains a negative value, the file is treated as unlabeled. A positive, nonzero value is assumed to be the address of a user routine to create user-header records or user-trailer labels (EOF and EOV). This address is relocated if necessary by the OUTAPE macro. The user routine releases control via the LBRET macro-instruction. (Refer to the Supervisor and Input/Output Macros publications.)

$\left[,\mathrm{RETURN}=\left\{\begin{matrix}\underline{\mathrm{YES}}\\ \mathrm{NO}\end{matrix}\right\}\right]$

This parameter governs the return to the input routine or to the optional user routine. If RETURN=YES a return to the input or user routine is provided via register 14. RETURN=NO signifies that the user's own coding or additional output utility macro-instructions will follow.

OUTDISK (DISK OUTPUT MACRO-INSTRUCTION)

The OUTDISK macro-instruction generates initialization, a file OPEN, a disk write routine, an end-of-file test, and a file CLOSE. The input unit is determined by the required label information. All records written on disk will have a key length of zero.

NAME	OPERATION	OPERAND
[name]	OUTDISK	$\mathrm{BUFSIZ}=n\ \left[,\mathrm{BLK}=\left\{\begin{matrix}m\\ \mathrm{name1}\end{matrix}\right\}\right]$ $,\mathrm{FILE}=\left\{\begin{matrix}\mathrm{filename}\\ (r)\end{matrix}\right\}\ \left[,\mathrm{LBL}=\mathrm{name2}\right]\ \left[,\mathrm{ERROR}=\left\{\begin{matrix}\underline{\mathrm{IGNORE}}\\ \mathrm{name3}\end{matrix}\right\}\right]$ $\left[,\mathrm{FORMAT}=\left\{\begin{matrix}\underline{\mathrm{FULL}}\\ n\\ \mathrm{name4}\end{matrix}\right\}\right]\ \left[,\mathrm{RETURN}=\left\{\begin{matrix}\underline{\mathrm{YES}}\\ \mathrm{NO}\end{matrix}\right\}\right]$

BUFSIZ=n

This required parameter specifies the maximum size of an output block and allocates buffer space for two blocks of that length. Shorter records can be generated. The value of n must not exceed 3625.

If unblocked records to be written are longer than the buffer space provided, they are truncated. The first time this occurs, the operator has the option of canceling the job.

$\left[,\mathrm{BLK}=\left\{\begin{matrix}m\\ \mathrm{name1}\end{matrix}\right\}\right]$

This parameter determines the number of logical records per output block. If omitted, output records are unblocked.

If BLK=m, then m specifies the blocking factor to be assembled into the program. A value of zero for m indicates unblocked records, and is equivalent to omitting the BLK operand.

If BLK=name1, the content of the full-word symbolic location, "name1" is interrogated at execution time when the file is OPENed. The value found will be used as the blocking factor, unless the value is zero or one. If zero or one, output is unblocked.

Regardless of the blocking factor specified, an accumulation of records are written immediately, as a block, if the next record threatens to exceed allocated buffer space. Unblocked records that are too large for the buffers are truncated. The first time each condition is encountered, the operator has the option of canceling the job. The final output block may be shorter than previous blocks due to termination of input.

$,\text{FILE=}\left\{\begin{matrix}\text{filename}\\ \text{(r)}\end{matrix}\right\}$

This required parameter identifies label information and is required for files bearing standard labels. A register (r) may contain the address of the eight character filename. Refer to the // VOL statement contained in either System Control and System Service publications listed on the front cover.

[,LBL=name2]

This parameter specifies user label-handling procedures. The disk files must bear standard labels with or without additional user labels. If the LBL parameter is omitted, no provision is made for user labels. If LBL=name2 is specified, the full-word area symbolically designated "name2" will be interpreted prior to OPEN, to determine label processing procedure. If the area contains zero, no provision is made for user labels. A nonzero value must be the address of a user routine to create user labels. This address will be relocated, if necessary, by the OUTDISK macro. The user routine releases control via the LBRET macro instruction. (Refer to Supervisor and Input/Output.)

Note: All records written on disk will have a key length of zero.

$\left[,\text{ERROR=}\left\{\begin{matrix}\underline{\text{IGNORE}}\\ \text{name3}\end{matrix}\right\}\right]$

This parameter specifies error-handling procedures after the supervisor has detected an error. If IGNORE is specified, no attempt is made to verify disk output. If ERROR=name3 is specified, then "name3" must be the symbolic location of a user-provided routine to which control is passed when each error occurs. The user may return to the output routine via BR 14. Regardless of user action, the output error will be ignored.

$\left[,\text{FORMAT=}\left\{\begin{matrix}\underline{\text{FULL}}\\ \text{n}\\ \text{name4}\end{matrix}\right\}\right]$

This parameter specifies the number of records to be written on a track. FULL indicates that as many records as possible are to be written (within the 3625 byte capacity) on each track. If FORMAT=n is specified, the n is the fixed number of records to be written per track. If n is so large as to cause a track overflow condition, each track will contain as many records as possible, but the number will not exceed n.

If FORMAT=name4, the content of the full-word symbolic location "name4", is interrogated when the file is OPENed to determine the number of records to be written per track. If the value of n is greater than zero it is assumed to indicate the number of records per track. If the value is less than (or equal to) zero, it is equivalent to specifying FULL or omitting the parameter.

$\left[,\text{RETURN=}\left\{\begin{array}{l}\underline{\text{YES}}\\ \text{NO}\end{array}\right\}\right]$

This parameter governs the return to the input routine or user routine. If RETURN=YES, a return to the input or user routine is provided via register 14. RETURN=NO signifies that user coding or additional output utility macro-instructions follow.

OUTPRT (PRINTER OUTPUT MACRO-INSTRUCTION)

The OUTPRT macro-instruction causes records of the specified size to be output on the line printer.

NAME	OPERATION	OPERAND
[name]	OUTPRT	$[\text{BUFSIZ=n}]\left[,\text{UNIT=}\left\{\begin{array}{l}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{array}\right\}\right]\left[,\text{RETURN=}\left\{\begin{array}{l}\underline{\text{YES}}\\ \text{NO}\end{array}\right\}\right]\left[,\text{FORMS=}\left\{\begin{array}{l}\text{A}\\ \text{B}\\ \text{C}\\ \text{D}\end{array}\right\}\right]$

[BUFSIZ=n]

This parameter specifies the maximum size of an output line, including any forms-control character. If this parameter is omitted, the value of n is set to 144. Shorter records can be printed.

If the records that are to be printed are longer than the indicated maximum size, the record is truncated. The first time this condition occurs, the operator has the option of canceling this job.

A buffer will be generated for two output lines (2n bytes).

$\left[,\text{UNIT=}\left\{\begin{array}{l}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{array}\right\}\right]$

This parameter specifies the symbolic name of the output unit.

$\left[,\text{RETURN=}\left\{\begin{array}{l}\underline{\text{YES}}\\ \text{NO}\end{array}\right\}\right]$

This parameter governs the return to the input routine or user routine. If RETURN=YES, a return to the input or user routine is provided via register 14. RETURN=NO signifies that user coding or additional output utility macro-instructions follow.

$\left[,\text{FORMS=}\left\{\begin{array}{l}\text{A}\\ \text{B}\\ \text{C}\\ \text{D}\end{array}\right\}\right]$

This parameter pertains to first-character forms control. If this parameter is omitted, all output lines are single spaced. If no forms control is specified, the first character of each record is printed as data. The significance of each of the forms options follows.

Type A

Indication of Type A allows the user to use the character that is the command-code portion of the System/360 Channel Command Word used in printing a line or spacing the forms. If the character read is not one of the following characters, the line will be printed with single spacing after printing and no error indication will be given.

8-Bit Code	Punch Combination	Function
00000001	12,9,1	Write (no automatic space)
00001001	12,9,8,1	Write and space 1 line after printing
00010001	11,9,1	Write and space 2 lines after printing
00011001	11,9,8,1	Write and space 3 lines after printing
10001001	12,0,9	Write and skip to channel 1 after printing
10010001	12,11,1	Write and skip to channel 2 after printing
10011001	12,11,9	Write and skip to channel 3 after printing
10100001	11,0,1	Write and skip to channel 4 after printing
10101001	11,0,9	Write and skip to channel 5 after printing
10110001	12,11,0,1	Write and skip to channel 6 after printing
10111001	12,11,0,9	Write and skip to channel 7 after printing
11000001	12,1	Write and skip to channel 8 after printing
11001001	12,9	Write and skip to channel 9 after printing
11010001	11,1	Write and skip to channel 10 after printing
11011001	11,9	Write and skip to channel 11 after printing
11100001	11,0,9,1	Write and skip to channel 12 after printing
00001011	12,9,8,3	Space 1 line immediately
00010011	11,9,3	Space 2 lines immediately
00011011	11,9,8,3	Space 3 lines immediately
10001011	12,0,8,3	Skip to channel 1 immediately
10010011	12,11,3	Skip to channel 2 immediately
10011011	12,11,8,3	Skip to channel 3 immediately
10100011	11,0,3	Skip to channel 4 immediately
10101011	11,0,8,3	Skip to channel 5 immediately
10110011	12,11,0,3	Skip to channel 6 immediately
10111011	12,11,0,8,3	Skip to channel 7 immediately
11000011	12,3	Skip to channel 8 immediately
11001011	12,0,9,8,3	Skip to channel 9 immediately
11010011	11,3	Skip to channel 10 immediately
11011011	12,11,9,8,3	Skip to channel 11 immediately
11100011	0,3	Skip to channel 12 immediately
00000011	12,9,3	No op

Type B

Type B allows the user to use the d-modifier character of the IBM 1401 carriage-control instruction used in printing a line or spacing forms with a 1401 system. If the character read is not one of the valid characters, the line will be printed with single spacing after printing and no error indication will be given. The codes are as follows:

d	immediate skip to	d	Skip after print to
1	channel 1	A	channel 1
2	channel 2	B	channel 2
3	channel 3	C	channel 3
4	channel 4	D	channel 4
5	channel 5	E	channel 5
6	channel 6	F	channel 6
7	channel 7	G	channel 7
8	channel 8	H	channel 8
9	channel 9	I	channel 9
0	channel 10	?	channel 10 (EBCDIC or BCDIC)
#	channel 11	.	channel 11
@	channel 12	⌑	channel 12 (EBCDIC or BCDIC)

d	immediate space	d	after print-space
J	1 space	/	1 space
K	2 spaces	S	2 spaces
L	3 spaces	T	3 spaces

Type C

Type C allows the use of the following codes as first-character, forms-control characters. If the character read is not one of the valid characters, the line will be printed with single spacing after printing.

Code	Space or Skip Action
plus (EBCDIC or BCDIC)	Suppress space and print
blank	Print and single space
zero	Double space, print, and space
-	Triple space, print, and space
1-9 or J-R	Immediate skip to channel 1-9 (that is, 1 or J=skip to channel 1; 2 or K=skip to channel 2; etc), print, and then space.

Type D

Type D allows the use of the ASA FORTRAN first-character, forms-control set. If the character read is not one of the valid characters, the line will be printed with single spacing before printing.

This character control set is used with printed output of TOS/360 or DOS/360 components.

Code	Space or Skip Action
blank	Space one line before printing
0	Space two lines before printing
-	Space three lines before printing
+ (EBCDIC or BCDIC)	Suppress space before printing
1	Skip to Channel 1 before printing
2	Skip to Channel 2 before printing
3	Skip to Channel 3 before printing
4	Skip to Channel 4 before printing
5	Skip to Channel 5 before printing
6	Skip to Channel 6 before printing
7	Skip to Channel 7 before printing
8	Skip to Channel 8 before printing
9	Skip to Channel 9 before printing
A	Skip to Channel 10 before printing
B	Skip to Channel 11 before printing
C	Skip to Channel 12 before printing

OUTLOG (PRINTER-KEYBOARD OUTPUT MACRO INSTRUCTION)

The OUTLOG macro instruction provides the facility to write information onto the IBM 1052 Printer-Keyboard assigned to SYSLOG.

NAME	OPERATION	OPERAND
[name]	OUTLOG	BUFFER= $\begin{Bmatrix}\text{name1}\\(r_b)\end{Bmatrix}$ $\left[,\text{COUNT=}\begin{Bmatrix}n\\(r_c)\end{Bmatrix}\right]$ $\left[,\text{RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\\text{NO}\end{Bmatrix}\right]$

BUFFER= $\begin{Bmatrix}\text{name1}\\(r_b)\end{Bmatrix}$

This required parameter specifies the location of the user's buffer area, name1, or a register, (r_b), containing the buffer address.

$\left[,\text{COUNT=}\begin{Bmatrix}n\\(r_c)\end{Bmatrix}\right]$

This parameter specifies the number of characters, or a register, (r_c), containing the number of characters to be written. If BUFFER=name1, the COUNT parameter may be omitted, in which case the number of characters read will be equal to the buffer length. The number of characters may not exceed 256.

No check will be made to ensure that SYSLOG is assigned to a 1052 or any other printer.

$\left[,\text{RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\\text{NO}\end{Bmatrix}\right]$

This parameter governs the return to the input section or optional user section. If RETURN=YES, a return to the input or user routine is provided via register 14. RETURN=NO signifies that the user's own coding or additional output utility macro-instructions follow.

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
OUTCARD	$\left[\text{UNIT=}\left\{\begin{matrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{matrix}\right\}\right]$	If other than SYS002 is wanted.	nnn=programmer logical unit for output. A buffer of 160 bytes is generated for the 1442 and 2520 (240 bytes for the 2540).
	$\left[\text{STCTL=}\left\{\begin{matrix}\underline{\text{NO}}\\ \text{YES}\\ \text{name}\end{matrix}\right\}\right]$	If stacker selection is desired.	Omission, or NO=pocket 2; YES=stacker selection is controlled by first character of each record; name1=symbolic location of area to be interpreted: 0=pocket 2; nonzero=YES.
	$\left[\text{,DEVICE=}\left\{\begin{matrix}\underline{\text{1442}}\\ \text{2520}\\ \text{2540}\end{matrix}\right\}\right]$	If other than a 1442 card punch is used.	Specifies error-recovery procedures.
	$\left[\text{,RETURN=}\left\{\begin{matrix}\underline{\text{YES}}\\ \text{NO}\end{matrix}\right\}\right]$	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTAPE	BUFSIZ=n	For each file.	n=maximum size of output block. A buffer of 2n bytes is generated.
	$\left[\text{,BLK=}\left\{\begin{matrix}\text{m}\\ \text{name1}\end{matrix}\right\}\right]$	If record blocking is desired.	m=blocking factor; value of 0 indicates unblocked records. name1=symbolic location to be interrogated. A value of 0 indicates no blocking is performed. Otherwise value=blocking factor.
	$\left[\text{,UNIT=}\left\{\begin{matrix}\underline{\text{SYS002}}\\ \text{SYSnnn}\end{matrix}\right\}\right]$	If other than SYS002 is wanted.	nnn=programmer logical unit for output.
	$\left[\text{,FILE=}\left\{\begin{matrix}\text{filename}\\ \text{(r)}\end{matrix}\right\}\right]$	For each file bearing standard labels.	Identifies label information. r=register pointing to an 8 character filename.
	[,LBL=name2]	If special label processing is desired.	name2=fullword area symbolically designated name2 to be interpreted. If value=0, standard labels are assumed and user labels are bypassed. If value is negative, no label processing is attempted; if value is positive, value=address of user HDR processing routine.

Figure 3. Summary of Output Macro Instructions (Part 1 of 3)

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
OUTAPE (Cont'd)	$\left[,\text{RETURN}=\left\{\begin{matrix}\underline{\text{YES}}\\\text{NO}\end{matrix}\right\}\right]$	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTDISK	BUFSIZ=n	For each file.	n=maximum size of output block. A buffer of 2n bytes is generated.
	$\left[,\text{BLK}=\left\{\begin{matrix}\text{m}\\\text{name1}\end{matrix}\right\}\right]$	If record blocking is desired.	m=blocking factor; value of 0 indicates unblocked records. name1=symbolic location to be interrogated. A value of 0 indicates no blocking is performed. Otherwise value=blocking factor.
	$\text{FILE}=\left\{\begin{matrix}\text{filename}\\\text{(r)}\end{matrix}\right\}$	For each file.	Identifies label information. r=register pointing to an 8 character filename.
	[,LBL=name2]	If user label processing is desired.	name2=fullword area symbolically designated name2 to be interpreted. If value=0, no user HDR label processing is attempted; if value is nonzero, value=address of user HDR record processing routine.
	$\left[,\text{ERROR}=\left\{\begin{matrix}\underline{\text{IGNORE}}\\\text{name3}\end{matrix}\right\}\right]$	If a write check is desired.	IGNORE=ignore error condition. name3=symbolic address of user routine for error-recording.
	$\left[,\text{FORMAT}=\left\{\begin{matrix}\underline{\text{FULL}}\\\text{n}\\\text{name4}\end{matrix}\right\}\right]$	When a specified number of records are to appear on each track.	FULL=fill track as much as possible. n=number of records per track. name4=symbolic address containing number of records per track value. If value is 0, FULL is assumed.

Figure 3. Summary of Output Macro Instructions (Part 2 of 3)

OPERATION	OPERAND	MUST BE INCLUDED	REMARKS
OUTDISK (Cont'd)	$\left[,\text{RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\right]$	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTPRT	[BUFSIZ=n]	If n should be less than 144.	n=maximum size of output block.
	$\left[,\text{UNIT=}\begin{Bmatrix}\text{SYS002}\\ \text{SYSnnn}\end{Bmatrix}\right]$	If other than SYS002 is wanted.	nnn=programmer logical unit for output.
	$\left[,\text{FORMS=}\begin{Bmatrix}\text{A}\\ \text{B}\\ \text{C}\\ \text{D}\end{Bmatrix}\right]$	If other than single-spaced lines are desired.	Refer to the description of carriage control codes provided with macro description.
	$\left[,\text{RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\right]$	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.
OUTLOG	$\text{BUFFER=}\begin{Bmatrix}\text{name1}\\ (r_b)\end{Bmatrix}$	For each output on the 1052.	name1=symbolic location of the user's buffer area. r_b=register containing the buffer address (absolute or symbolic).
	$\left[,\text{COUNT=}\begin{Bmatrix}\text{n}\\ (r_c)\end{Bmatrix}\right]$	If BUFFER=(r_b)	n=number of characters to be written. r_c= register containing the number of characters to be written.
	$\left[,\text{RETURN=}\begin{Bmatrix}\underline{\text{YES}}\\ \text{NO}\end{Bmatrix}\right]$	If user routine or additional output utility macros follow.	NO=continue to next sequential statement. YES=return to input routine or to optional user routine.

Figure 3. Summary of Output Macro Instructions (Part 3 of 3)

CARD-TO-CARD

A program of this type can be used for detection of an arbitrary data delimiter card for card input.

NAME	OPERATION	OPERAND	
	START	0	
	INCARD		
	BALR	9,0	(establish base register)
	USING	*,9	
	LR	2,0	(put address of input area into
*			register 2)
	CLC	0(2,2),A	(compare for data delimiter)
	BNE	B	(continue with output if not
*			delimiter)
	BAL	14,B	(punch delimiter card)
	SR	0,0	(set registers zero and one to
*			zero to signal end of job)
	SR	1,1	
B	OUTCARD		
A	DC	C'**'	(sets up the arbitrary data
*			delimiter)
	END		

Notes: Input unit: SYS001
Output unit: SYS002; no stacker selection.
A user exit provides for cards to be read until a ** is encountered in card columns 1 and 2 or until end-of-file is reached on the reader.

CARD-TO-DISK

A program of this type can be used for building a SYSIN file on disk.

NAME	OPERATION	OPERAND	Col. 72
	START	0	
	INCARD		
	OUTDISK	BUFSIZ=80,FILE=DASDOUPT,ERROR=	x
		IGNORE,FORMAT=25	
	END		

Notes: Input unit: SYS001

Output unit: symbolic unit obtained from XTENT card; no user labels: unblocked output, 80 characters per record; write check disk output; ignore write errors; 25 records per track: filename DASDOUPT

No user routines

CARD-TO-PRINTER AND CARD

A program of this type is used to both print and punch the input records. Note the use of the RETURN operand.

NAME	OPERATION	OPERAND
	START	0
	INCARD	
	OUTPRT	BUFSIZ=80,RETURN=NO
	OUTCARD	UNIT=SYS003,DEVICE=2540
	END	

Notes: Input unit: SYS001

Printer output unit: SYS002; 80 characters per line; single space

Card output unit: SYS003; all output into pocket 2; 2540 card punch

No user routines

CARD-TO-TAPE

A program of this type can be used to build a SYSIN file on tape.

NAME	OPERATION	OPERAND
	START	0
	INCARD	
	OUTAPE	BUFSIZ=80
	END	

Notes: Input unit: SYS001

Output unit: SYS002; no labels; unblocked output;
80 characters per record

No user routines

DISK-TO-CARD

A program of this type can be used to punch a SYSPCH file from disk.

NAME	OPERATION	OPERAND
	START	0
	INDISK	BUFSIZ=81,FILE=DASDINPT
	OUTCARD	STCTL=YES
	END	

Notes: Input unit: Symbolic unit obtained from XTENT card; no user label; ignore read errors; unblocked input records not longer than 81 characters; filename DASDINPT

Output unit: SYS002; stacker selection via ASA control characters

No user routines

DISK-TO-PRINTER

A program of this type can be used to print a SYSLST file from disk.

NAME	OPERATION	OPERAND
	START	0
	INDISK	BUFSIZ=121,FILE=INPTDASD
	OUTPRT	BUFSIZ=121,FORMS=D
	END	

Notes: Input unit: Symbolic unit obtained from XTENT card; no user label; ignore read errors; unblocked input records not longer than 121 bytes; filename INPTDASD

Output unit: SYS002; forms control via ASA control characters

No user routines

TAPE-TO-CARD

A program of this type can be used to punch a SYSPCH file from tape.

NAME	OPERATION	OPERAND
	START	0
	INTAPE	BUFSIZ=81,UNIT=SYS002
	OUTCARD	STCTL=YES,UNIT=SYS003
	END	

Notes: Input unit: SYS002; no labels; ignore read errors; unblocked input records not more than 81 characters in length; bypass checkpoint records

Output unit: SYS003; stacker selection; 1442 card punch

No user routines

TAPE-TO-PRINTER

A program of this type can be used to print a SYSLST file from tape.

NAME	OPERATION	OPERAND
	START	0
	INTAPE	BUFSIZ=121
	OUTPRT	BUFSIZ=121,UNIT=SYS005,FORMS=D
	END	

Notes: Input unit: SYS001; no labels; ignore read errors: unblocked input records not more than 121 bytes long; bypass checkpoint records, if any

Output unit: SYS005; forms control via ASA control characters

No user routines

TAPE-TO-PRINTER AND/OR CARD

A program of this type will be used to print and punch combined SYSLST/ SYSPCH output. A user routine provides for selection of printer or punch. Note that the data is assumed to be on a single tape reel and preceded by a tapemark.

NAME	OPERATION	OPERAND
	START	0
	INTAPE	BUFSIZ=121,CHKPT=NO
	BALR	9,0
	USING	*,9 (establish base register)
	LTR	0,0 (end of input data?)
	BNZ	A (no)
C	NOP	B (skip past first tapemark)
	OI	C+1,X'F0' (change NOP to a branch
*		instruction)
	BR	14 (return to input routine)
B	LA	1,* (set to signal end-of-file)
	BAL	14,PUNCH (clear punch output buffers)
	SR	1,1 (set to signal end-of-job)
	B	PRINT (clear printer buffers and
*		end job)
A	LR	2,0 (record address into register 2)
	CLI	0(2),X'E5' (is control character a V)
	BE	PUNCH (yes--punch card)
	CLI	0(2),X'E6' (is control character a W)
	BE	PUNCH (yes--punch card)
PRINT	OUTPRT	BUFSIZ=121,FORMS=D,UNIT=SYS003
PUNCH	OUTCARD	STCTL=YES
	END	

Notes: Input unit SYS001; no labels; ignore read errors; no checkpoint records; unblocked input records not more than 121 bytes long

Output unit: SYS002 (punch) and SYS003 (print); forms control and stacker select via ASA control characters; 1442 card punch

User exit for selection of output

Before a generated utility program is executed as either a background or foreground program, the utility program must be assembled by the assembler. The assembly can be combined with execution or cataloging, or both, depending on current job requirements and future plans for using the generated program.

The major considerations in determining the job setup are:

1. whether the assembled program is to be cataloged for subsequent use.
2. whether that use is to be a foreground or background program.
3. whether execution is desired immediately after the assembly.

In the examples which follow, a card-to-card program is used to show the deck organization for some of the available options.

The first example illustrates the assembly and cataloging of the program CDTOCD for subsequent execution as a foreground program. The use of zero as a link-edit address is an indication that the program is self-relocating. The program may be executed as a background or foreground job. In the following examples, the job control and linkage editor control statements are explained in the System Control and System Service publications.

Example 1: ASSEMBLE & CATALOG CARD-TO-CARD FOR SUBSEQUENT USE AS A SELF-RELOCATING PROGRAM

```
//  JOB CARDCARD
//  OPTION CATAL
    PHASE CDTOCD,+0
//  EXEC ASSEMBLY
    START 0
    INCARD
    OUTCARD
    END
/*
//  EXEC LNKEDT
/&
```

The second example illustrates the assembly and cataloging of a similar program with user routines, for use as a foreground program.

Example 2: ASSEMBLE & CATALOG CARD-TO-CARD FOR SUBSEQUENT USE AS A FOREGROUND PROGRAM (Not self-relocating, the foreground partition originates at 16384)

```
//  JOB CARDCARD
//  OPTION CATAL
    PHASE CDTOCD,F+16384
//  EXEC ASSEMBLY
    START 0
    .
    .
    .
Initialization Routine (user-provided, not self-relocating)
    .
    .
    .
    INCARD
    OUTCARD
    END
/*
//  EXEC LNKEDT
/&
```

The third example illustrates the assembly and execution of the same utility program as a background job.

Example 3: ASSEMBLE & EXECUTE CARD-TO-CARD AS A BACKGROUND PROGRAM (Not self-relocating, the program is to load at the beginning of the background area)

```
//  JOB CARDCARD
//  OPTION LINK
    PHASE CDTOCD,S
//  EXEC ASSEMBLY
    START 0
    .
    .
    .
Initialization Routine (user-provided, not self-relocating)
    .
    .
    .
    INCARD
    OUTCARD
    END
/*
//  EXEC LNKEDT
//  ASSGN SYS001,X'00C'  (reader)
//  ASSGN SYS002,X'00D'  (punch)
//  EXEC
/&
```

The fourth example combines the cataloging (as in example 2) with immediate execution of the cataloged program.

Example 4: ASSEMBLE, CATALOG, & EXECUTE CARD-TO-CARD AS A BACKGROUND PROGRAM

```
//  JOB CARDCARD
//  OPTION CATAL
    PHASE CDTOCD,+0
//  EXEC ASSEMBLY
    START
    INCARD
    OUTCARD
    END
/*
//  EXEC LNKEDT
//  ASSGN SYS001,X'00C'  (reader)
//  ASSGN SYS002,X'00D'  (punch)
//  EXEC
.
.
.
(card input on reader)
.
.
.
```

The fifth example illustrates the control cards needed to execute a utility which has been cataloged for use as a background program. The cataloging might have been done by either example 1 or 4.

Example 5: EXECUTE PREVIOUSLY CATALOGED CARD-TO-CARD PROGRAM AS A BACKGROUND PROGRAM

```
//  JOB CARDCARD
//  ASSGN SYS001,X'00C' (reader)
//  ASSGN SYS002,X'00D' (punch)
//  EXEC CDTOCD
.
.
.
(card input on reader)
.
.
.
```

The execution of a utility program which has been cataloged as a foreground program is done through SYSLOG and the Foreground Initiator rather than SYSRDR and Job Control.

Example 6: EXECUTION OF A FOREGROUND PROGRAM WHICH HAS BEEN PREVIOUSLY CATALOGED IN THE CORE IMAGE LIBRARY

```
         COL. 16                                   COL. 72
START F1
ASSGN SYS001,X'180'            (tape drive)
ASSGN SYS001,X'181',ALT        (alternate tape drive)
ASSGN SYS002,X'190'            (disk drive)
ASSGN SYS003,X'191'            (second disk drive)
VOL   SYS001,INFILE
TPLAB 'LABEL FIELDS 3-10'                            X
               'LABEL FIELDS 11-13'
VOL   SYS002,OUTFILE
DLAB           'DISKFILE...    1000123',...          X
               0001,66030,66430,'CODEIBM21-3AA'
XTENT 1,0,000017006,000017009,'000123',SYS002
XTENT 1,1,000086000,000089009,'000123',SYS002
XTENT 128,2,000146003,000192007,'000163',SYS003
EXEC  TPTODK                   (previously cataloged tape to disk
                                program)
```

Foreground programs are initiated by the operator through the 1052 assigned to SYSLOG. The operator may initiate a foreground program whenever the specified foreground area does not contain a program.

The operator initiates a foreground program by depressing the 1052 request key.

START $\left\{\begin{matrix}\underline{\text{F1}}\\ \text{F2}\end{matrix}\right\}$ indicates that a foreground program is to be initiated. If the area specified is allocated and does not contain a program, it transfers control to the Foreground Initiator. Otherwise, the operator is notified that he has given an invalid command.

The Foreground Initiator reads subsequent commands required to initiate the program. These commands are used primarily to specify I/O assignments and label information. Such information may be read from the Printer-Keyboard or Card Reader.

When DOS is used, each set of label information is composed into a label information block and written onto system residence for later retrieval and processing by the data management routines. For DOS and TOS, a main storage area for label information may be required, in which case the area is reserved by the Foreground Initiator.

When the EXEC command is encountered, the Foreground Initiator checks to determine if a self-relocating program is to be loaded. (If the load address is zero, the program is self-relocating.) The Foreground Initiator will direct this program to be loaded following the label information area, if any. A non-self-relocating program will be directed to be loaded utilizing the information derived when the program was linkage-edited.

When initial control is given to the user's foreground program, register 2 contains the address of the uppermost byte of storage available to this program. This may be used to calculate the total storage to the program. A foreground program can dynamically determine the storage available to it by storing the contents of this register for later reference. Foreground initiation commands may be referred to in either System Control and System Service publication.

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 COMMENT FORM

IBM System/360
Disk and Tape Operating Systems
Utility Macros Specifications C24-5042-1

- Your comments, accompanied by answers to the following questions, help us produce better publications for your use. If your answer to a question is "No" or requires qualification, please explain in the space provided below. All comments will be handled on a non-confidential basis. Copies of this and other IBM publications can be obtained through IBM Branch Offices.

	Yes	No
• Does this publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

- What is your occupation? ____________________
- How do you use this publication?

As an introduction to the subject?	☐	As an instructor in a class?	☐
For advanced knowledge of the subject?	☐	As a student in a class?	☐
For information about operating procedures?	☐	As a reference manual?	☐

Other ____________________

- Please give specific page and line references with your comments when appropriate.

COMMENTS:

- Thank you for your cooperation. No postage necessary if mailed in the U. S. A.

C24-5042-1

Staple

Fold

Fold

FIRST CLASS
PERMIT NO. 170
ENDICOTT, N. Y.

BUSINESS REPLY MAIL
NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 6
Endicott, N. Y. 13760

Attention: Programming Publications, Dept. 157

Fold

Fold

Cut Along Line

IBM S360 Printed in U. S. A. C24-5042-1

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]

Additional Comments: