File Number S360 Mod 195 S370-30
Order Number GC21-5097-0

OS

IBM Systems Reference Library

OS and OS/VS Programming Support for the IBM 3505 Card Reader and IBM 3525 Card Punch

This publication is intended for application programmers who are familiar with the BSAM and QSAM access methods for OS or OS/VS systems. It describes the macro instructions and services which are provided to support the 3505 Card Reader and the 3525 Card Punch.

Before using this publication you should be familiar with the information contained in:

IBM 3504 Card Reader/IBM 3505 Card Reader and IBM 3525 Card Punch Subsystem, GA21-9124, and the data management services and data management macro publications for your system.

PREFACE

This publication contains descriptions of the OS and OS/VS support for the 3505 Card Reader and 3525 Card Punch and the optional features for both devices.

It is divided into three parts:

3505 Card Reader — This section describes the data sets and access methods used with the 3505 and provides information about the data management macro instructions pertinent to the device.

3525 Card Punch — This section describes the data sets and access methods used with the 3525 and provides information about the data management macro instructions pertinent to the device. Special emphasis is placed on the usage of associated data sets for the various combinations of read, punch, and print operations.

Appendixes — This section includes information about the diagnostics available for the 3505/3525. It also contains a sample program demonstrating the use of the 3525 for a read, punch, and print job with program controlled line positioning.

Prerequisite Publications

OS or OS/VS Data Management Macro Instructions (GC26-3794 or GC26-3793)

OS or OS/VS Data Management Services Guide (GC26-3746 or GC26-3783)

IBM 3504 Card Reader/IBM 3505 Card Reader and IBM 3525 Card Punch Subsystem (GA21-9124)

First Edition (July 1973)

This publication corresponds to OS Release 21, VS1 Release 2, and VS2 Release 1 and to all subsequent releases. It contains the special programming considerations for the 3505 Card Reader and the 3525 Card Punch that were previously described in the Appendix of the *OS Data Management Services Guide,* GC26-3746-1 and in the Appendix of the *OS/VS Data Management Services Guide,* GC26-3783-1.

Information in this publication is subject to change from time to time. Any such change will be reported in subsequent revisions or technical newsletters. Before using this publication in connection with the operation of IBM systems, consult the latest IBM System/360 and 370 SRL Newsletter, GN20-0360, for the editions that are applicable and correct.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM Branch Office serving your locality.

Forms are provided at the back of this publication for reader's comments. If they have been removed, comments may be addressed to International Business Machines Corporation, Department D58, Building 706-2, Box 390, Poughkeepsie, New York 12602.

CONTENTS

The IBM 3505 is a medium-speed serial 80-column card reader. It comes in two models, B1 and B2. The maximum throughput on the model B1 is 800 cards per minute; on the model B2 it is 1200 cards per minute. Both models are channel attached to System/370 and System/360 Model 195.

The 3505 provides faster read capabilities and more innovative functions than either the IBM 2501 Card Reader or the read portion of the IBM 2540 Card Read Punch. Standard features include:

- An 80-column card image buffer which allows both a reread function and its associated channel error recovery.
- One logical stacker that consists of two physical stacker mechanisms called stacker 1 left and stacker 1 right.
- Card reading in either EBCDIC or column binary mode.
- Read Column Eliminate (RCE), which suppresses the reading of data from specified card columns.

Program support provided for existing readers applies to the standard 3505 with no modifications. Buffering is the same, including the buffering for normal reading and stacker selection. This includes compatible 2540 channel programs. The 3505 is supported as a system input device.

3505 OPTIONAL FEATURES

Optional features on the 3505 Card Reader include:

- A selective stacker (stacker 2). This includes a third stacker and a stacker wait station.
- Optical Mark Read (OMR), which provides the ability to read up to 40 columns per card of pencilled marks (#2 lead or softer), and machine-printed, non-reflective ink marks from cards.

The 3505 is not supported as a system input device when using OMR or RCE.

3505 DATA SETS

The 3505 has the same BSAM and QSAM support as other IBM readers. The read data set must be designated as input according to the methods specified in OS or OS/VS Data Management Macro Instructions (GC26-3794 or GC26-3793). Buffering for normal reading and stacker selection remains unmodified. Data management also supports the OMR and RCE features. (See *3505 Optical Mark Read* and *Read Column Eliminate*.)

BSAM/QSAM programs currently used with 2540, 2501, 2520, or 1442 may be executed without change to the logic or JCL except where a device has been explicitly specified (such as UNIT=2540 in the DD statement). However, if the DD statement is modified to indicate 3505, the current user program will execute identically, assuming the 3505 has identical features.

If column binary mode is used, it must be specified in either a DCB statement at assembly time or in the DCB subparameter of a DD statement at execution time. Validity checking is suspended in column binary mode because all characters are considered valid. If data mode is not specified, EBCDIC is assumed. Once data mode is established, it cannot be changed during execution of the program, except at the EXCP level.

Read Column Eliminate must also be specified in either a DCB statement at assembly time or in the DCB subparameter of a DD statement at execution time. (See *Read Column Eliminate*.) The RCE format does not become effective until the data set is opened and remains in effect until the data set is closed.

The MODE operand of the DCB statement is used to indicate that RCE or OMR is to be used by the program for the 3505.

The user is responsible for assigning, via a DD statement, a 3505 which has the feature required for the job. The Scheduler will not allocate a 3505 by feature. However, the Scheduler will allocate a 3505 as a reader without concern for any optional features.

3505 ACCESS METHODS

The operating system, via the assembler, uses a set of macro instructions to initiate the data management access methods used by the 3505. These instructions initiate:

- the Basic Sequential Access Method (BSAM)
- the Queued Sequential Access Method (QSAM)

A detailed description of the macros that are used for these access methods can be found in OS or OS/VS Data Management Macro Instructions (GC26-3794 or GC26-3793).

Both access methods and the macro instructions are used with the two major configurations of the IBM System/360 or 370 operating system:

- Multiprogramming with a Fixed number of Tasks (MFT), or VS1.
- Multiprogramming with a Variable number of Tasks (MVT), or VS2.

The data control block (DCB) macro instruction defines the data set for BSAM and QSAM.

MACRO INSTRUCTIONS (3505)

The macro instructions used with BSAM on the 3505 for input operations are READ, CHECK, CNTRL, OPEN, and CLOSE. (See the OS or OS/VS Data Management Macro Instructions, GC26-3794 or GC26-3793 for OPEN and CLOSE.)

The macro instructions which are used with QSAM for 3505 input operations are GET, CNTRL, OPEN, and CLOSE. The CNTRL, OPEN, and CLOSE macro instructions are the same for QSAM and BSAM (except a test for completion of input operations is not required for the CNTRL macro instruction for QSAM before it is issued).

READ

The READ macro instruction initiates an input operation. After the instruction is issued, control is returned to the problem program so that it may perform operations which do not involve the input buffer specified in the READ instruction. A data event control block (DECB) is constructed as part of the READ macro expansion. A DECB:

1. passes parameters to the controlling program; and

2. aids in controlling the READ operation.

The READ macro instruction for BSAM is written as follows:

[symbol]	READ	DECB name, type, DCB address, area address, {length / 'S'}

- DECB name – the name assigned to the DECB created during the macro expansion.

- type – the type of READ operation; SF (normal retrieve) must be coded for 3505 programs.

- DCB address – the address of the DCB for the data set which is read.

- area address – the address of the area into which the record is placed.

- length – the number of data bytes which are read (maximum 32,760); if 'S' is coded, the number of bytes is taken from the DCB. (The length operand is ignored if the records are either in F or V format.)

CHECK

The CHECK macro instruction checks for a completed input operation. If the operation is complete, the instruction checks for error indications. If the operation is incomplete, the task waits for completion. The CHECK macro instruction is written as follows:

[symbol]	CHECK	DECB address

- DECB address – the address of the data event control block, which is either created by the associated READ macro instruction or used by the associated input operation.

CNTRL (FOR STACKER SELECTION)

The CNTRL macro instruction is used to control stacker selection for both QSAM and BSAM. Under BSAM, all input operations must be tested for completion before the CNTRL macro instruction is issued. If stacker selection is required:

1. the CNTRL macro instruction must be issued whenever it is necessary to read a new card.

2. the BUFNO field of the DCB must be coded as one.

The CNTRL macro instruction is written as follows:

[symbol]	CNTRL	DCB address,SS,{1 / 2}

- DCB address — the address of the DCB for the card reader.

- SS — the stacker select option (one or two).

Note: See the OS or OS/VS Data Management Macro Instructions, GC26-3794 or GC26-3793 for a more detailed description of the CNTRL macro instruction.

GET

The GET macro instruction causes the control program to retrieve the next record. The instruction can be issued in either the locate, move, or substitute mode. The GET macro instruction is written as follows:

[symbol]	GET	DCB address [,area address]

- DCB address — the address of the DCB for the data set which is being retrieved.

- area address — there are three modes of retrieval; these may not be intermixed within a specified DCB.

1. In locate mode, the area address is omitted. The address which is returned in register 1 references the buffer which contains the record.

2. In move mode, the area address specifies the address of the area in the problem program into which the system will move the record.

3. In substitute mode, the area address specifies the address of an area in the problem program that will be exchanged for the buffer containing the record. The system returns the address of the buffer containing the record in register 1.

3505 READ WITH OPTIONAL STACKER

Stacker selection is specified in the MACRF or STACK operand of the DCB macro instruction, or in the STACK operand of the DCB subparameter in the DD statement. If stacker selection via the CNTRL macro instruction is specified by the MACRF operand, either stacker 1 or 2 must be subsequently specified in the CNTRL instruction. If no stacker selection is specified, the cards are stacked in logical stacker 1.

Stacker 1 is the default stacker if the 3505 has only one logical stacker and a command is issued indicating stacker 2 or 3. Stacker 2 is the default stacker if a command is issued indicating stacker 3 and the 3505 has the optional stacker. Stacker selection default is not diagnosed.

If stacker control is required, BUFNO=1 must be specified. This is the only way the user can be sure a particular card is going to the desired stacker.

3505 OPTICAL MARK READ

OMR provides the facility to read up to 40 columns of marked data with the OMR data physically located on the card in alternating columns. If OMR is specified, a format descriptor card must be provided as the first card of the data deck. The format descriptor card specifies the columns from which optical marks are to be read. Abnormal termination results if a format descriptor card is not supplied. If checkpoint/restart is used, the format of the OMR data set must be re-established when the job is restarted.

Format Descriptor Card: The word FORMAT must be coded starting in column 2 of the first card of the data deck (column 1 must be blank), followed by a blank and the parameters that specify the columns to be read in OMR mode. For example, if columns 3, 5, 7, 9, 70, 72, 74, 76, 78, and 80 are to be read in OMR mode, the format descriptor card would be coded as follows:

```
 FORMAT (3,9),(70,80)
```

Continuation cards can be coded if:

1. a continuation character (non-blank) is entered in column 72 of the card; and

2. the continued field begins in column 16 of the next line.

A continued field is coded in one of two ways:

1. the operand field is coded through column 71 with no blanks, and is then continued in column 16 of the next card, or

2. the operand field is truncated by a comma and then continued in column 16 of the next card.

OMR Data Records: The following rules apply to coding an OMR record.

- Mark columns within a field must be separated by at least one blank (not a punch or mark).

- Mark and punch fields must be separated by at least one blank (not a punch or mark).

- Mark fields in odd columns and mark fields in even columns must be separated by at least two blanks (not punches or marks).

- Mark or punch fields may begin in any column, so long as the coding conforms to the first three rules.

Although OMR data is physically located on the card in alternating columns, the data is compressed in the channel. The blank following an optical mark is not transferred to the input buffer by the channel.

The format data is stored in the system and may contain either a digit (0 through 9) or a letter (A through Z) for each card column that is to contain OMR data. All other format bytes are blanks (X'40'). Figure 3505-1, Format of OMR Data, illustrates how the data appears in the card, the system, and in the channel and input buffer.

Card Column	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Card Data	H_1	H_2	Ⓑ	M_4	Ⓑ	M_6	Ⓑ	Ⓑ	M_9	Ⓑ	M_{11}	Ⓑ	H_{13}	H_{14}	H_{15}
Format Data	b	b	b	F_4	–	F_6	–	b	F_9	–	F_{11}	–	b	b	b
		switch from hole to mark					switch from even to odd marks					switch from mark to hole			
Channel Data	H_1	H_2	b	M_4	M_6	b	M_9	M_{11}	H_{13}	H_{14}	H_{15}				

Ⓑ must not have hole or mark data.
b hexadecimal 40
– may be either character or blank
H_x hole data in Column X
M_x mark data in Column X
F_x format data for Column X

Figure 3505-1. Format of OMR Data

When a marginal mark, weak mark, or poor erasure is detected, that column's data is replaced with X'3F' in EBCDIC mode or with X'3F3F' in column binary mode. X'3F' is also placed in column 80 for EBCDIC and column 160 for column binary. The user is responsible for checking for OMR reading errors.

If OMR is used, it must be specified in either a DCB statement at assembly time (MODE=O), or the DCB subparameter of a DD statement at execution time

(MODE=$\left\{\begin{matrix}E\\C\end{matrix}\right\}$ O). If the MODE operand for a 3505 is specified in the DCB subparameter of a DD statement, either C or E (for column binary or EBCDIC), must be specified if O is specified.

The OMR format becomes effective only after the data set is opened. The OPEN macro instruction must be followed by a READ and CHECK, or a GET macro instruction. A CNTRL macro instruction must also be issued if stacker selection is required. If stacker selection is used, BUFNO=1 must be specified. The format remains in effect until the data set is closed. When the CLOSE macro instruction is issued or when the device runs out of cards at end of file (EOF), the device is automatically reset to read all 80 columns in normal mode.

Stacking more than one job utilizing card columns 1 or 2 for OMR is not recommended. The normal file delimiter (/*), which occupies these columns, cannot be recognized if OMR uses these columns.

The following example illustrates how OMR can be specified via the DD statement:

```
READD    DCB    DDNAME=OMR,DSORG=PS,MACRF=GMC, . . . .
                                              | | CNTRL
                                              | Move Mode
                                              GET

//OMR     DD     UNIT=3505,DCB=(MODE=EO,BUFNO=1, . . . . . .
```

where MODE=EO specifies EBCDIC and OMR mode, and BUFNO=1 is specified because the CNTRL macro is being used.

Note: Chain channel scheduling and the user totaling facility is not supported when using OMR.

READ COLUMN ELIMINATE (3505/3525)

If RCE is specified, a format descriptor card must be provided as the first card of the data deck. The format descriptor card specifies the columns which are to be eliminated. Abnormal termination results if a format descriptor card is not supplied. If checkpoint/restart is used, the format of the RCE data set must be reestablished when the job is restarted.

Format Descriptor Card: The word FORMAT must be coded starting in column 2 of the first card of the data deck (column 1 must be blank), followed by a blank and the parameters that specify the columns to be eliminated in RCE mode.

For example, if columns 20 through 30 and 52 through 76 are not to be read, the format descriptor card would be coded as follows:

```
 FORMAT (20,30),(52,76)
```

Continuation cards can be coded if:

1. a continuation character (non-blank) is entered in column 72 of the card; and
2. the continued field begins in column 16 of the next card.

A continued field is coded in one of two ways:

1. the operand field is coded through column 71 with no blanks, and is then continued in column 16 of the next card, or
2. the operand field is truncated by a comma and then continued in column 16 of the next card.

If RCE is used, it must be specified in either a DCB statement at assembly time (MODE=R), or in the DCB subparameter of a DD statement at execution time

(MODE= $\{^{C}_{E}\}$R). If the MODE operand for a 3505 or 3525 is specified in the DCB subparameter of a DD statement, either C or E (for column binary or EBCDIC), must be specified if R is specified.

The RCE format becomes effective only after the data set is opened. The OPEN macro instruction must be followed by a READ and CHECK, or a GET macro instruction. A CNTRL macro instruction must also be issued if stacker selection is required. If stacker selection is used, BUFNO=1 must be specified. The format remains in effect until the data set is closed. When the CLOSE macro instruction is issued, or when the device runs out of cards at end of file (EOF), the device is automatically reset to read all 80 columns in normal mode.

Stacking more than one job utilizing card columns 1 or 2 for RCE is not recommended. The normal file delimiter (/*), which occupies these columns, cannot be recognized if RCE uses these columns.

The following example illustrates how RCE can be specified via the DCB statement:

```
READD     DCB    DDNAME=RCE,DSORG=PS,MACRF=GL,MODE=R, . . . . .
                                              | |
                                              | Locate Mode
                                             GET
//RCE     DD     UNIT=3525, . . . . . .
```

where MODE=R specifies RCE mode (EBCDIC is the default for data mode).

Note: Chain channel scheduling and the user totaling facility is not supported when using RCE.

The IBM 3525 is an 80-column card punch. The 3525 models and their respective performance capabilities are:

- 3525-P1 Card Punch — 100 Cards per Minute Maximum
- 3525-P2 Card Punch — 200 Cards Per Minute Maximum
- 3525-P3 Card Punch — 300 Cards Per Minute Maximum

All of these models can be attached to System/360 Model 195 and System/370.

The 3525 must be used in conjunction with the 3505. Its attachment to the using system is through a channel attachment to a control unit in either model of the 3505.

The 3525 has one hopper from which it passes its data cards. There are two program-controlled selectable stackers, stacker 1 (on the right) and stacker 2 (on the left). A card traversing the feed path will pass from the hopper to a dummy station (card read station if the option is installed), to the card punch station, then to another dummy station (card print station if the option is installed), and into one of the two program selectable stackers. If a card is found to be mispunched (during a punch only job), It is passed to a third stacker and two cards are punched automatically. One of the two cards is fed to the third stacker with the error card and the second card replaces the one in error in the normal stacker. The repunching of cards is a hardware function.

3525 OPTIONAL FEATURES

All models of the 3525 can include the following optional features:

Card Read

This feature provides the ability to read punched-hole data, under program control, from a card during a single pass of the card through the machine. The data can be EBCDIC or column binary. Read Column Eliminate is also included with this feature (see *Read Column Eliminate*).

Two-Line Print

This feature provides the ability to print two lines of data, under program control, on a card during a single pass of the card through the machine. The data printed on the card can be the same as or different than the data punched into the card. Print line one is located above punch row 12 of the card and the second print line (actually printed on print line three) is located between punch rows 12 and 11. Each line of print is 64 characters in length. A 63-character graphic set, plus a blank, is provided.

Multiline Print

This feature provides the ability to print up to 25 lines of data, under program control, on a card during a single pass of the card through the machine. This feature has the same character set and print span as the Two-Line print feature.

3525 DATA SETS

The punch and read data sets must be designated as output and input data sets, according to the methods specified in OS or OS/VS Data Management Macro Instructions (GC26-3794 or GC26-3793). The print and interpret punch functions must be designated by the FUNC operand in either a DCB statement or the DCB subparameter of a DD statement. (Also see the *3525 Associated Data Sets* section of this book.)

BSAM/QSAM programs currently used with a 2540, 2501, 1442, or 2520 may be executed without change to the logic or JCL, except where a device has been explicitly specified on the DD statement (such as UNIT=2540). If the DD statement is modified to indicate 3525, the current program will execute identically, assuming the 3525 has identical features.

Punch

The 3525 has the same BSAM and QSAM support as other IBM punches. Data management also supports the previously discussed optional features. The 3525 is supported as a SYSOUT device for punch only data sets.

Read

The 3525 has standard IBM reader support for BSAM and QSAM. Data management also supports the RCE feature. If RCE is used, it must be specified in either a DCB statement at assembly time or in the DCB subparameter of a DD statement at execution time. A first card descriptor statement must be used to specify which columns are to be eliminated. For a detailed discussion of RCE mode, see *Read Column Eliminate*. The 3525 is supported as a SYSIN device for read only data sets if RCE is not specified.

Print

The 3525 has BSAM and QSAM support for printing. Macro support includes CNTRL and PRTOV with each card regarded as a print page.

Interpret Punch

The interpret punch data set is supported by both BSAM and QSAM. The support, via a single data set, includes both punching and printing of graphically printable punched characters on print lines one and three of the card. Line one includes the first 64 characters and line three includes the last 16 characters (right justified). Extraneous characters are printed for nongraphic eight-bit codes.

Special Considerations

If column binary mode is used, it must be specified in either a DCB statement at assembly time or in the DCB subparameter of a DD statement at execution time. If data mode is not specified, EBCDIC is assumed. Once the data mode is established, it cannot be changed during execution of the program, except at the EXCP level.

Operands which are new or have been modified for use with the 3525 include:

MODE – The MODE operand of the DCB statement is also used to indicate whether RCE mode is to be used.

FUNC – The FUNC operand of the DCB statement defines the type of data set to be opened. It must be used with print only and associated data sets, and can be used with read only and punch only data sets. See *3525 Associated Data Sets* for more information.

FCB – The FCB operand of a DD statement is used to indicate that Data Protection Image (DPI) is associated with a particular data set. See *Associated Data Sets* for more information.

3525 ACCESS METHODS

The operating system, via the sasembler, uses a set of macro instructions to initiate the data management access methods of the 3525. These instructions initiate:

- the Basic Sequential Access Method (BSAM)
- the Queued Sequential Access Method (QSAM)

A detailed description of the macros that are used for these access methods can be found in OS or OS/VS Data Management Macro Instructions (GC26-3794 or GC26-3793).

Both access methods and the macro instructions are used with the two major configurations of the System/360 Model 195 and System/370 operating systems:

- Multiprogramming with a Fixed number of Tasks (MFT), or VS1
- Multiprogramming with a Variable number of Tasks (MVT), or VS2

The data control block (DCB) macro instruction defines the data set for BSAM and QSAM.

MACRO INSTRUCTIONS (3525)

The macro instructions used with BSAM for the 3525 are:

Input – – – – – – – – – – – – READ, CHECK, and CNTRL
Output – – – – – – – – – – – – WRITE, CHECK, CNTRL, and PRTOV

The macro instructions for READ, CHECK, and CNTRL (for stacker selection), are discussed under *Macro Instructions* in the 3505 section of this book.

The macro instructions used with QSAM for the 3525 are:

Input – – – – – – – – – – – – GET and CNTRL
Output – – – – – – – – – – – – PUT, CNTRL, and PRTOV

The GET and CNTRL (for stacker selection) macro instructions are discussed under *Macro Instructions* in the 3505 section of this book.
See OS or OS/VS Data Management Macro Instructions, GC26-3794 or GC26-3793, for OPEN and CLOSE.

WRITE

The WRITE macro instruction initiates an output operation. After the instruction is issued, control is returned to the problem program so that it may perform operations which do not involve the buffer specified in the particular instruction. A data event control block (DECB) is constructed as part of the write macro expansion. A DECB:

1. passes parameters to the controlling program;
2. aids in the controlling of the write operation; and
3. receives indication of the success or failure of the write operation.

Note: The operating system returns control to the program before the write operation is complete, so that output operations can be overlapped with CPU processing.

The WRITE macro instruction for BSAM is written as follows:

[symbol]	WRITE	DECB name,SF,DCBaddress,area address,{length / 'S'}

- DECB name – the name which is assigned to the DECB.
- SF – is codes as shown.
- DCB address – the address of the DCB for the data set which is created.
- area address – the address of the main storage area which contains the block.
- length – the operand which is used for unspecified-length records; this operand specifies the number of data bytes to be written (maximum 32,760). 'S' is coded to indicate that the length in the block size (BLKSIZE) field of the DCB is to be used.

CNTRL

The CNTRL macro instruction is used to control printing and stacker selection on the 3525 under both BSAM and QSAM (CNTRL for stacker selection is discussed under *Macro Instructions* in the 3505 section of this book). Under BSAM, all output operations must be tested for completion before the CNTRL instruction is issued.

The numbers of the lines on an individual card correspond to the channel numbers as designated in the following figure.

Line Number	Channel Number
1	1
2	
3	2
4	
5	3
6	
7	4
8	
9	5
10	
11	6
12	
13	7
14	
15	8
16	
17	9 (overflow)
18	
19	10
20	
21	11
22	
23	12 (overflow)
24	
25	

The channels all correspond to odd numbered lines; an even numbered line can be designated by skipping to the channel which immediately precedes it and then spacing.

The CNTRL macro instruction is written as follows:

[symbol]	CNTRL	DCB address, {SP, {1/2/3} / SK, {1-12}}

- DCB address — the address of the DCB for the output device
- SP — space (one, two, or three) lines on a card.
- SK — skip to a channel (1 through 12) on a card.

Note: A line can also be designated for printing using standard ASCII control characters or machine code control characters. These methods of control are described in *3525 Print Only*, Printing Control.

PRTOV

The PRTOV macro instruction tests for overflow channels nine and twleve. An overflow condition on either of these channels causes:

1. a transfer of control to the overflow processing routine (if the address of this routine is specified in the overflow exit address entry); or
2. a skip to channel one to begin printing on the next card for print only.

The PRTOV macro instruction is written as follows:

[symbol]	PRTOV	DCB address, {9 / 12} [, overflow exit address]

- DCB address — the address of the print DCB.
- {9 / 12} — tests for either channel nine or twelve.
- overflow exit address — the address of the user-supplied routine which gains control when an overflow condition is detected on the specified channel. If this address is omitted, the 3525 feeds a new card (by performing a skip to channel 1), before executing the next WRITE or PUT instruction. (This condition would abnormally end with associated data sets.) When the overflow routine is given control, the contents of:

- registers 0 and 1 are destroyed.
- registers 2 through 13 remain the same.
- register 14 is the return address.
- register 15 is the overflow exit routine address.

PUT

The PUT macro instruction causes the control program to write a record in a sequential data set.

The controlling program uses the length which is specified in the record length (LRECL) field of the DCB as the length of the record which is to be written.

The PUT macro instruction is written as follows:

[symbol]	PUT	DCB address [,area address]

- DCB address — The address of the DCB for the data set.
- area address — There are three modes of retrieval; these may not be intermixed within a specified DCB.
  - In locate mode the area address is omitted. When a PUT macro instruction is used in locate mode, the address of the buffer for the first record or segment is obtained by issuing a PUT macro instruction. QSAM returns the address of the next available buffer in register 1, but the record is not written until the next PUT macro instruction is issued. For this reason a dummy PUT is required before processing starts.

- In move mode, the area address refers to the main storage area, which contains the record which is to be written.

- In substitute mode, the area address specifies the address of a main storage area in the problem program that contains the next record to be written. The area is exchanged for an empty buffer. The address of the empty buffer is returned in register 1.

3525 INTERPRET PUNCH

If the Interpret Punch function is designated, via the FUNC operand in a DCB or the DCB subparameter in a DD statement, an existing output data set will be interpreted as well as punched. Specifying the Interpret Punch function in the DD statement as shown below, allows the function to be done without reassembling the program.

```
//PUNCH     DD     UNIT=3525,DCB=(FUNC=I), . . . . .
```

One macro instruction is all that is needed to both punch and interpret the punches on a card.

The Punch Output Writer takes advantage of the interpret punch function. That is, the writer has the ability to interpret an output file with no changes to the routine. The DCB can be modified via the console as follows:

```
using unit address   — S WTR,013,DCB=(FUNC=I) or,
using device ID      — S WTR,3525,DCB=(FUNC=I)
```

It can also be specified via a DD statement as follows:

```
//OUTPUT     DD     SYSOUT=(B,INTRPRT)
```

Where INTRPRT is a user written routine that is cataloged on the system. FUNC=I must be specified in the DCB in the cataloged writer routine. The writer routine can be keyed via the console as follows:

```
S  WTR.INTRPRT, 3525
```

Note: The output record must be 80 bytes, or 81 bytes if first character control is being used.

3525 PRINT ONLY

Two print options are available under 3525 support: (If printing using associated data sets, see *3525 Associated Data Sets.)*

Two-Line Print Option

The two-line print option is specified in the FUNC operand in the DCB statement or the DCB subparameter of the DD statement. The operand must be coded FUNC=WT for print only as shown in the following example:

```
//PRINT  DD  UNIT=3525,DCB=(FUNC=WT,LRECL=64,BUFNO=10)
```

The T in FUNC=WT specifies that the information will be printed on lines one and three regardless of whether the hardware supports the two-line or the multiline print feature.

Multiline Print Option

The multiline print option is specified in the FUNC operand in the DCB statement or the DCB subparameter of the DD statement. The operand is coded FUNC=W. (Omitting the T for the two-line-print option automatically selects the multiline print option.)

Printing Control

Line positioning and card feeding can be controlled by using the CNTRL macro instruction, ASCII control characters, machine code control characters, or automatic line positioning. The numbers of the lines on an individual card correspond to the channel numbers as designated in the following figure.

Line Number	Channel Number
1	1
2	
3	2
4	
5	3
6	
7	4
8	
9	5
10	
11	6
12	
13	7
14	
15	8
16	
17	9 (overflow)
18	
19	10
20	
21	11
22	
23	12 (overflow)
24	
25	

The channels all correspond to odd numbered lines; an even numbered line can be designated by skipping to the channel which immediately precedes it and then spacing.

CNTRL

The CNTRL macro instruction is described under *Macro Instructions* in the 3525 section of this book.

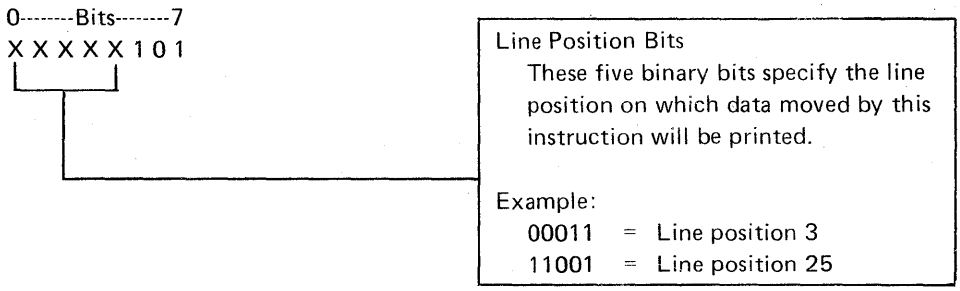
ACSII Control Characters

A line can be designated for printing by specifying standard ASCII character control in the RECFM operand of the DCB. The CNTRL macro instruction cannot be issued if first character control is specified in the DCB. If ASCII control characters are used and an attempt is made to print beyond line 3 (two-line print option), or beyond line 25 (multiline print option), a new card is fed and the information is printed on the first line of that card.

When using ASCII control characters, spacing occurs before the line is printed. For line 1 positioning, the user must specify a blank as the ASCII control character. This is necessary because hardware automatically positioned the card at line 1 when it was fed. The blank ASCII control character allows data management to increment the internal line counter it uses, to monitor line position, from 0 to 1, without changing card position for line 1 only.

Machine Code Control Characters

A line can also be designated for printing through the use of machine operation codes. The use of such codes must be designated in the RECFM operand of the DCB. The CNTRL macro instruction cannot be issued if machine codes are designated. The print operation command codes are as follows:



If machine code control characters are used, these conditions cause the print line command to be rejected:

- A line position other than 1 or 3 is specified for a 3525 equipped with a two-line card print feature.
- A line position greater than 25 has been specified.
- The 3525 is not equipped with either card print feature.

Note: See *3504 Card Reader/IBM 3505 Card Reader and IBM 3525 Card Punch Subsystem,* GA21-9124, for the machine code control characters for punching and printing.

Automatic Line Positioning

If printing control is not specified by the CNTRL macro instruction or by using control characters. print lines are positioned automatically. If consecutive WRITEs or PUTs are issued, output is automatically single-spaced. If T is specified in the FUNC operand (for the two-line print option), consecutive WRITEs or PUTs print on lines 1 and 3.

Under automatic line positioning for a print only job, card feeding varies according to the print option being used. If the two-line feature is in effect, a WRITE or PUT following a WRITE or PUT on line 3 causes a card feed, and the information for the last WRITE or PUT is printed on line 1 of the following card. If the multiline feature is being used, a WRITE or PUT following a WRITE or PUT on line 25 causes a card feed, and the information for the last WRITE or PUT is printed on line 1 of the following card.

Note: Card feeding must be controlled by the read or punch data set when associated data sets are used. This automatically positions the card at line 1 (see *Associated Data Sets*).

Program-Controlled Line Positioning Summary

The program may control printing through either the CNTRL macro instruction, ASCII control characters, or machine code control characters.

If printing is program-controlled, the program is totally responsible for all line positioning (skipping and spacing). Under these methods of control, the two-line and multiline print supports are identical. However, printing is not allowed on any lines other than 1 or 3 with the two-line print option.

Skipping to a channel of a number equal to or less than the present channel position results in line positioning at that channel location on the following card. Skipping to a channel of a number greater than the present channel position results in line positioning at that channel location on the same card. Attempts to overprint by using space suppress (printing twice on the same line,) will result in a space and print.

No user facility is provided to vary stacker selection for a print only operation. However, the user can specify stacker selection via the STACK=operand in the DCB. If STACK=is not specified, the default is stacker 1.

The following options are not supported for 3525 print:

1. Chain channel scheduling.

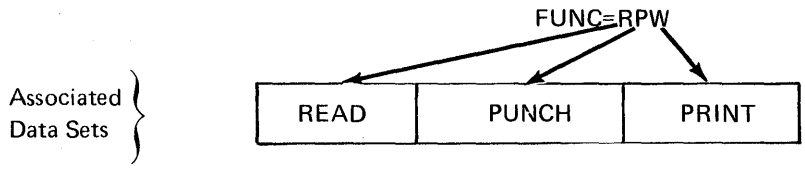
2. Fixed blocked, variable or undefined records.

3. The user totaling facility.

Records used for print only must be fixed (RECFM=F).

3525 ASSOCIATED DATA SETS

More than one operation can be performed on each card during the execution of a program. For example, you may read a card, punch additional data into the card, and print data on the same card. The data used for each operation must be in an individual data set. Each data set must have a separate DCB. In this case three DCBs must be specified. These three data sets are called *associated data sets.* That is, three separate data sets, but all for the same device. Associated data sets are specified by coding the proper character(s) for FUNC in each DCB. This example would be coded as follows:



See OS or OS/VS Data Management Macro Instruction (GC26-3794 or GC26-3793), and the following pages for other FUNC characters to be coded for other associated data sets.

The unit affinity operand of the DD statement must also be used to indicate that the data sets are associated with the same device. On the first DD statement for an associated data set the coding is UNIT=3525, or UNIT=unit-address; on the DD statements for the other associated data sets the coding is UNIT=AFF=READ, where READ is the name of the first DD statement. Records used with associated data sets must be fixed (RECFM=F).

A particular sequence of BSAM and QSAM macro instructions must be followed when associated data sets are being used. Only four combinations of 3525 functions are allowed for associated data sets and must be performed in the sequence shown in the following for each:

- READ and PUNCH
- READ, PUNCH and PRINT
- READ and PRINT
- PUNCH and PRINT

A print operation can be omitted or repeated, but the first line on a card cannot be printed until the card has been punched or, if the card is not to be punched, until the card has been read.

Note: When using locate mode for the punch or printer, the initial PUT (to retrieve the buffer address), is not considered an I/O sequencing factor. Therefore, the user must issue an extra PUT to print or PUNCH the last record of the data set.

FUNC Operand

The FUNC operand defines the type of data set which is to be opened. The valid characters are:

P	PUNCH
R	READ
W	PRINT
I	INTERPRET PUNCH

The following characters are used to further define the data set:

X	PRINTER
D	DATA PROTECTION
T	TWO-LINE PRINT

An X is used to distinguish the 3525 printer output data set from the 3525 punch output data set, when either a read, punch, and print or a punch and print associated data set is opened. (See examples on the following pages.)

A D indicates the use of the data protection feature. Data protection is used to protect data which might otherwise be obliterated by punching in columns which already contain data.

A T indicates the use of the two-line print option. If it is not specified, the multiline print option is used automatically.

Valid combinations of characters which indicate the use of the 3525 features include:

P	PUNCH ONLY
R	READ ONLY
W[T]	PRINT ONLY
I	INTERPRET PUNCH
RP	READ and PUNCH
RW [T]	READ and PRINT
PW[XT]	PUNCH and PRINT
RPW [XT]	READ, PUNCH, and PRINT
RP [D] RPW [D]	DATA PROTECTION

READ and PUNCH

READ and PUNCH associated data sets enable the 3525 to punch additional data into cards after they have been read. The data which is to be read is specified as an input data set; the data which is to be punched is specified as an output data set. The data control block for each associated data set is then opened.

The FUNC operand must be coded in either a DCB or the DCB subparameter of a DD statement. The unit must also be specified in the DD statement. In the following example, READ is the DDNAME of the input data set; PUNCH is the DDNAME of the output data set.

```
//READ    DD    UNIT=3525,DCB=(FUNC=RP)
//PUNCH   DD    UNIT=AFF=READ,DCB=(FUNC=RP)
```

Read and punch associated data sets are supported by normal control characters for dynamic stacker selection for a punch output data set. If static stacker selection is used, the STACK= $\{^{1}_{2}\}$ operand of the DCB should be specified. The data protection option may also be used with the PUNCH data set.

READ, PUNCH, and PRINT

Read, punch, and print associated data sets enable the 3525 to read data cards, punch additional data into them, and print on them. The data to be read must be specified as an input data set; the data to be punched, and that to be printed, must be specified as individual output data sets. The data control block for each associated data set must be opened. If dynamic stacker selection is used, standard ASCII control characters should be specified for the punch output data set. If static stacker selection is used, the STACK= $\{^{1}_{2}\}$ operand of the DCB should be specified.

Line positioning for the cards being printed can be specified by either first character control in the print record, or through the CNTRL macro instruction. Automatic line positioning may also be used. The card is then regarded as a print page with the channel assignments shown in the *3525 Print Only* section of this book. A maximum of 25 lines can be printed on a single card.

The FUNC operand must be coded in either a DCB or the DCB subparameter of a DD statement. The unit must also be specified in the DD statement. In the following example, READ is the DDNAME of the input data set; PUNCH and PRINT are the DDNAMES of the two output data sets.

```
//READ    DD    UNIT=3525,DCB=(FUNC=RPW)
//PUNCH   DD    UNIT=AFF=READ,DCB=(FUNC=RPW)
//PRINT   DD    UNIT=AFF=READ,DCB=(FUNC=RPWX)
```

READ and PRINT

Read and print associated data sets enable the 3525 to read cards and then print on them. The cards to be read are specified as an input data set; the cards to be printed are specified as an output data set. The data control block for each associated data set must be opened.

If dynamic stacker selection of the cards read is required, it must be specified in the CNTRL macro instruction of the input data set. Line positioning for the cards being printed can be specified by either control characters in the print record, or through the CNTRL macro instruction. Automatic line positioning may also be used.

The FUNC operand must be coded in either a DCB or the DCB subparameter of the DD statement. The unit must also be specified in the DD statement. In the following example, READ is the DDNAME of the input data set; PRINT is the DDNAME of the output data set.

```
READ    DD    UNIT=3525,DCB=(FUNC=RW)
PRINT   DD    UNIT=AFF=READ,DCB=(FUNC=RWX)
```

PUNCH and PRINT

Punch and print associated data sets are supported in the same manner as read, punch, and print associated data sets, except that no input data set is specified. Both the cards which are to be punched and the cards which are to be printed must be specified as individual output data sets. The data control block for each associated data set must be opened.

As in the cases of read, punch, and print; and read and print associated data sets, line positioning for the print data set can be specified by control characters, or through the CNTRL macro instruction. Automatic line positioning may also be used.

The FUNC operand must be coded in either a DCB or the DCB subparameter of a DD statement. The unit must also be specified in the DD statement. In the following example, PUNCH and PRINT are the DDNAMES of the two output data sets.

```
//PUNCH    DD    UNIT=3525,DCB=(FUNC=PW)
//PRINT    DD    UNIT=AFF=PUNCH,DCB=(FUNC=PWX)
```

Data Protection

Data protection is used to protect pre-punched data that might otherwise be obliterated during a punch operation. An 80-byte Data Protection Image (DPI) must be stored in SYS1.IMAGELIB for use with data protection. The DPI must contain blanks in the colums to be protected an alphameric characters in the columns to be punched. The 80-byte format descriptor corresponds to the 80 column card on a column-per-column basis.

The member name for the image in the CSECT and LKED statements cannot exceed eight bytes. The first four characters of this member name must be FORM. The characters that follow FORM identify the FCB image and are referred to as the image identifier. The image identifier must be specified on a DD statement to load the image in the FCB buffer.

Figure 3525-1, Coding for Data Protection, illustrates how to add a data protection image to SYS1.IMAGELIB.

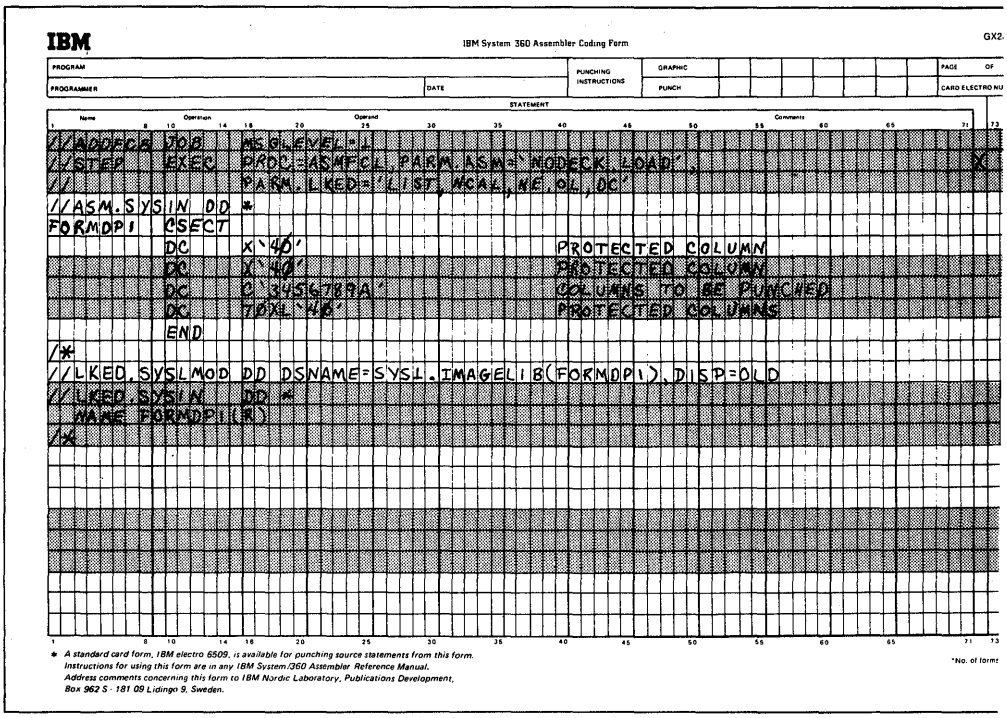


Figure 3525-1. Coding for Data Protection

The data card used with the data protection image shown in Figure 3525-1 would be in the format shown in Figure 3525-2, Data Card Format.

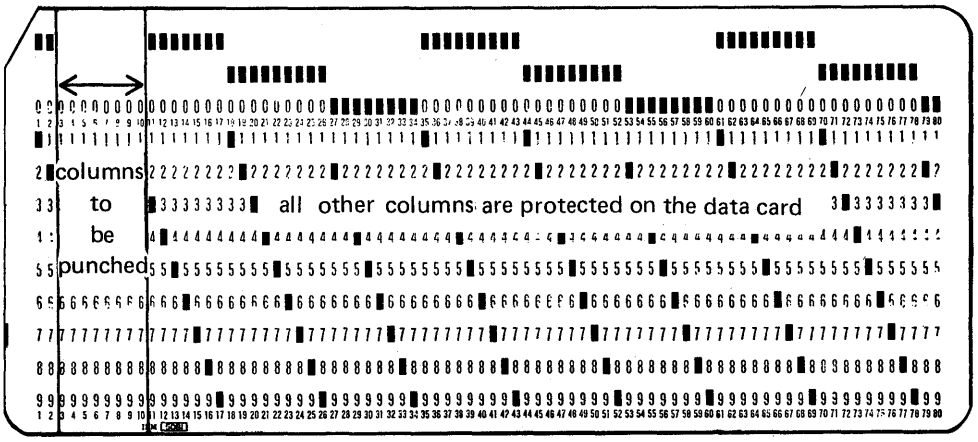


Figure 3525-2. Data Card Format

The DPI is associated with a particular data set through the use of the FCB operand of a DD statement. The FCB operand must be coded FCB=xxxx, where x represents the one to four characters being a unique image identifier. This image identifier may be any combination of alphameric characters.

Data protection can be specified as shown in the following example:

```
//READ      DD      UNIT=3525,DCB=(FUNC=RP,LRECL=80,BUFL=
  80,BUFNO=1)
//PUNCH     DD      UNIT=AFF=READ,DCB=(FUNC=RPD,LRECL=
  80,BUFNO=1),FCB=DPI
```

The D (in the DCB operand FUNC=RPD), which specifies the use of data protection, follows the description of the function which is to be performed; it must be either READ and PUNCH; or READ, PUNCH, and PRINT. If D is specified, data management assumes that the data protection image has been stored in SYS1.IMAGELIB.

The data protection option is not supported for column binary mode.

Opening Associated Data Sets

Associated data sets can be opened in any order, but an associated data set cannot be processed unless all associated data sets are open.

Closing Associated Data Sets

Associated data sets can be closed in any order, but once an associated data set is closed, I/O operations cannot be requested for any of the associated data sets.

CLOSE causes a feed command to be issued when the associated data set that causes feeding is closed. This ensures that the last data card is moved from the card transport to the stacker. If the data set that causes card feeds is not closed before the job terminates, the last data card will remain in the card transport. The operation which causes a card feed (via CLOSE) for 3525 data sets and associated data sets is as follows:

Data Set	Operation Causing A Card Feed
Read	Read
Punch	Punch
Print	Print
Interpret Punch	Punch
Read and Punch	Read
Read, Punch, and Print	Read
Read and Print	Read
Punch and Print	Punch

If a data delimiter card is used for the input data set, the user program must check for it and branch to the EODAA routine. After the program reads a data delimiter card, do not attempt to punch or print on it; this will cause the first card of the following job to be lost. The data delimiter card remains in the card transport until the following job causes a card feed or until nonprocess runout is performed by the operator.

The following restrictions will prevent the loss of the last data card when data sets are closed:

- All associated data sets must be closed before termination of the job step without intervening I/O operations for any of the associated data sets.
- If any data set is reopened, the appropriate associations must be reestablished.
- If the data set was read in read column eliminate (RCE) mode, a card feed will be issued to reset the RCE mode to normal mode when the data set is closed.

Note: In QSAM locate mode, the user must issue a PUT for the last record of an associated data set using the 3525 punch and/or print.

Restrictions for Associated Data Sets

- I/O operations on a single card must be completed in read, punch, print sequence. Two reads in succession or two punches in succession cause abnormal termination if performed on the read and punch data sets or the read, punch, and print data sets. A print operation can be omitted or repeated, but the first line on a card cannot be printed until the card has been punched or, if the card is not to be punched, until the card has been read. (If using locate mode, the initial PUT precludes this restriction.)
- You can use either BSAM or QSAM to process associated data sets, but the same access method must be used on all associated data sets for the same 3525.
- The FUNC operand must be coded in a DCB or the DCB subparameter of a DD statement for each associated data set.
- Associated data sets cannot be allocated to SYSIN or SYSOUT. You must request the specific device type or unit address in the UNIT operand of the DD statement.
- BUFNO=1 must be specified for a read or punch associated data set.
- When one of the associated data sets is to be punched, stacker selection can occur only with the punch data set. You can accomplish this by using the CNTRL macro, control characters, or the STACK operand of the DCB macro instruction. For the read and print associated data sets where no punch data set is used, stacker selection can be specified only with the read data set through the CNTRL macro instruction or STACK operand.
- Data protection applies only to the read, punch, and print; and read and punch associated data sets. It is requested via the FUNC operand of the DCB macro instruction and by specifying the image to be used via the FCB operand.
- All associated data sets must be opened before I/O is performed for any of them.
- A data delimiter card must neither be punched nor printed on.
- An attempt to space suppress and print on the same line (overprint) will result in abnormal termination.
- An attempt to feed a card via the print operation causes abnormal termination. The following commands cause card feeding and should not be used in conjunction with associated data sets:

1. Skip to 1. (If ASCII control characters are being used to control line position, specify a blank (X'40') as the line 1 control character.)

2. Skipping to a channel of a number equal to or less than the present channel position.

3. An attempt to print past line 3 (two-line print option), or line 25 (multiline print option).

4. A space after printing on line 3 (two-line print option), or line 25 (multiline print option).

5. The use of control characters to position the card at line 1 of the next card. Card feeding is controlled by the read or punch data set when associated data sets are used. The card is automatically positioned at print line 1 of the following card. For ASCII control characters, data management monitors line position internally with a counter that is set to zeroes at each card feed for ASCII only. The user of ASCII control characters specifies a blank as the control character to print line 1. This causes the data management internal line counter to be advanced by 1 without changing the hardware card position.

- The following options are not supported for associated data sets:

1. Chain channel scheduling.

2. Fixed blocked, variable or undefined records. Records used with associated data sets must be fixed (RECFM=F).

3. The user totaling facility.

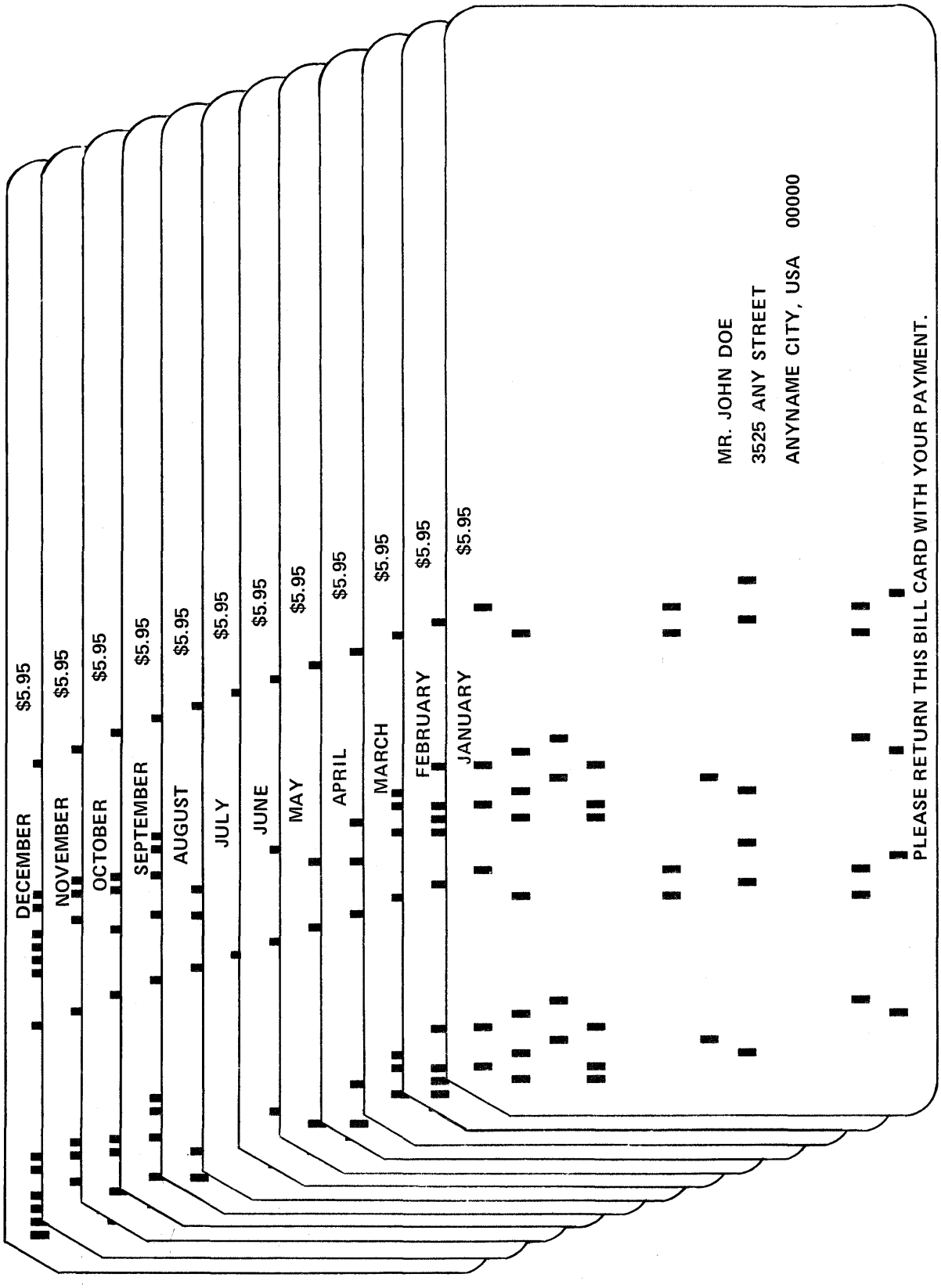
APPENDIX A. MESSAGES AND CODES

COMPLETION CODE	MESSAGE CODE	MESSAGES
003	IEC950I	"003-1 3525 Associated Data Set I/O Sequence Error." This error occurred because of a read I/O sequence error. For read and punch; and read, punch, and print associated data sets, the user must maintain the READ/WRITE (or GET/PUT) sequence for normal operation. Every READ (or GET) must be followed by a WRITE (or PUT) to punch the card. If the user does not want to punch information, he must punch blanks. The initial PUT to punch when using locate mode precludes this restriction.
003	IEC950I	"003-2 3525 Associated Data Set I/O Sequence Error." This error occurred because of a punch I/O sequence error. For read and punch; and read, punch, and print associated data sets, the GET and PUT (or READ and WRITE) for the read and punch portion of the associated data set must be executed in the proper sequence (GET or READ followed by a PUT or WRITE to punch) for every card. The initial PUT to punch when using locate mode precludes this restriction.
003	IEC950I	"003-3 3525 Associated Data Set I/O Sequence Error." This error occurred because of a print I/O sequence error. For a read and print, and punch and print associated data sets, the READ and PUNCH operation for a particular card must be executed before the PRINT operation is executed for that card. However, printing can be performed up to 25 times on that same card. If an attempt is made to skip to another card during the PRINT operation of an associated data set, the program is abnormally ended. The initial PUT to print when using locate mode precludes this restriction.
004	IEC951I	"004 Invalid Format Card or Invalid Device for OMR." Register 15 contains a return code of 5.

COMPLETION CODE	MESSAGE CODE	MESSAGES
004	IEC952I	"004 Conflicting/Invalid DCB FUNC or Related Parameters." The meaning of the return codes in register 15 are as follows: 1 – Indicates invalid DCB FUNC parameters. 2 – Indicates invalid/FUNC/CNTRL combinations. 3 – Indicates conflicting associated data set access methods. 4 – Indicates an invalid DCB for a 3505 or 3525.
004	IEC953I	"004 Data Protection Image Was Not Found." Register 15 contains a return code of 6. Possible reasons for this error are: 1. Image was never loaded in SYS1. IMAGELIB. 2. Volume on which SYS1. IMAGELIB resides was not mounted. 3. SYS1. IMAGELIB data set is not cataloged.

APPENDIX B. SAMPLE PROGRAM

The following is an example of the data cards used by the sample program.



```
IEF298I  T30102   SYSOUT=B.
//T30102 JOB    (0000,                                                    X
// 9430,366943,15921),'HOLBERT',MSGLEVEL=1
//        EXEC  ASMFCLG,PARM.ASM='DECK,LOAD'
XXASM    EXEC  PGM=IEUASM,PARM='LOAD,NODECK',REGION=78K                  00000010
XXSYSPRINT DD  SYSOUT=A                                                  00000020
XXSYSGO     DD DSNAME=&LOADSET,UNIT=SYSSQ,SPACE=(80,(200,50)),          X00000030
XX             DISP=(MOD,PASS)                                           00000040
XXSYSUT1 DD    DSNAME=&SYSUT1,UNIT=SYSDA,SPACE=(1700,(400,50))           00000050
XXSYSUT2 DD    DSNAME=&SYSUT2,UNIT=SYSSQ,SPACE=(1700,(400,50))           00000060
XXSYSUT3 DD    DSNAME=&SYSUT3,SPACE=(1700,(400,50)),                     X00000070
XX             UNIT=SYSDA                                                00000080
XXSYSLIB DD    DSNAME=SYS1.MACLIB,DISP=SHR                             IEUD 00000090
//ASM.SYSPUNCH DD SYSOUT=B
//ASM.SYSIN    DD *
IEF236I ALLOC. FOR T30102   ASM
IEF237I 133   ALLOCATED TO SYSPRINT
IEF237I 130   ALLOCATED TO SYSGO
IEF237I 131   ALLOCATED TO SYSUT1
IEF237I 132   ALLOCATED TO SYSUT2
IEF237I 133   ALLOCATED TO SYSUT3
IEF237I 130   ALLOCATED TO SYSLIB
IEF237I 250   ALLOCATED TO SYSPUNCH
IEF237I 132   ALLOCATED TO SYSIN
```

```
SYMBOL   TYPE ID  ADDR   LENGTH LD ID

RPWPGM    SD  01 000000 000540
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F01OCT71  6/22/73

000000                             1 RPWPGM   START
                                   2 *
                                   3 **
                                   4 **       THIS PROGRAM CAUSES DATA TO BE PUNCHED FROM THE 2540 PUNCH.
                                   5 **       THEN, THAT SAME DATA IS READ FROM AND PRINTED ON THE BILL
                                   6 **       CARD BY THE 3525 DEVICE.  IT, ALSO, IS PUNCHED IN THE BILL
                                   7 **       CARD AGAIN.  AFTER ALL OF THIS IS DONE, THE NAME AND ADDRESS
                                   8 **       OF THE CUSTOMER IS PRINTED ON THE BILL CARD FOR MAILING
                                   9 **       PURPOSES.
                                  10 **
                                  11 *
                                  12 **
                                  13 ***            EQUATES
                                  14 **
                                  15 *
000001                            16 R1       EQU   1
000002                            17 R2       EQU   2
000003                            18 R3       EQU   3
000004                            19 R4       EQU   4
000005                            20 R5       EQU   5
000006                            21 R6       EQU   6
000007                            22 R7       EQU   7
000008                            23 R8       EQU   8
000009                            24 R9       EQU   9
00000A                            25 R10      EQU   10
00000B                            26 R11      EQU   11
00000C                            27 R12      EQU   12
00000D                            28 R13      EQU   13
00000E                            29 R14      EQU   14
00000F                            30 R15      EQU   15
                                  31 **
                                  32 ***            LINKAGE CONVENTIONS AND ADDRESSABILITY
                                  33 **
                                  34          SAVE  (14,12)
000000                            35+         DS    0H
000000 90EC D00C            0000C 36+         STM   14,12,12(13) SAVE REGISTERS
000004 05C0                       37          BALR  R12,0                    ESTABLISH ADDRESSABILITY OF PGM
000006                            38          USING *,R12                    SET UP BASE REGISTER
000006 50D0 C20E            00214 39          ST    R13,SAVAREA+4
00000A 188D                       40          LR    R8,R13
00000C 41D0 C20A            00210 41          LA    R13,SAVAREA              GET SAVE AREA ADDRESS
000010 50D8 0008            00008 42          ST    R13,8(R8)                STORE SAVE AREA ADDRESS
                                  43 **
                                  44 ***
                                  45 **
                                  46          OPEN  (PCHDCB,(OUTPUT))        OPEN PUNCH DCB TO PUNCH DATA
000014                            47+         CNOP  0,4
000014 4510 C016            0001C 48+         BAL   1,*+8 LOAD REG1 W/LIST ADDR.
000018 8F                         49+         DC    AL1(143) OPTION BYTE
000019 0004E0                     50+         DC    AL3(PCHDCB) DCB ADDRESS
00001C 0A13                       51+         SVC   19 ISSUE OPEN SVC
                                  52          PUT   PCHDCB,DATA1             PUNCH BILL CARD FOR JANUARY
00001E 4110 C4DA            004E0 53+         LA    1,PCHDCB LOAD PARAMETER REG 1
000022 4100 C2F8            002FE 54+         LA    0,DATA1 LOAD PARAMETER REG 0
000026 58F0 1030            00030 55+         L     15,48(0,1) LOAD PUT ROUTINE ADDR.
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F01OCT71  6/22/73

00002A 05EF                       56+        BALR  14,15 LINK TO PUT ROUTINE
                                  57         PUT   PCHDCB,DATA2            PUNCH BILL CARD FOR FEBRUARY
00002C 4110 C4DA          004E0   58+        LA    1,PCHDCB LOAD PARAMETER REG 1
000030 4100 C30C          00312   59+        LA    0,DATA2 LOAD PARAMETER REG 0
000034 58F0 1030          00030   60+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000038 05EF                       61+        BALR  14,15 LINK TO PUT ROUTINE
                                  62         PUT   PCHDCB,DATA3            PUNCH BILL CARD FOR MARCH
00003A 4110 C4DA          004E0   63+        LA    1,PCHDCB LOAD PARAMETER REG 1
00003E 4100 C320          00326   64+        LA    0,DATA3 LOAD PARAMETER REG 0
000042 58F0 1030          00030   65+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000046 05EF                       66+        BALR  14,15 LINK TO PUT ROUTINE
                                  67         PUT   PCHDCB,DATA4            PUNCH BILL CARD FOR APRIL
000048 4110 C4DA          004E0   68+        LA    1,PCHDCB LOAD PARAMETER REG 1
00004C 4100 C334          0033A   69+        LA    0,DATA4 LOAD PARAMETER REG 0
000050 58F0 1030          00030   70+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000054 05EF                       71+        BALR  14,15 LINK TO PUT ROUTINE
                                  72         PUT   PCHDCB,DATA5            PUNCH BILL CARD FOR MAY
000056 4110 C4DA          004E0   73+        LA    1,PCHDCB LOAD PARAMETER REG 1
00005A 4100 C348          0034E   74+        LA    0,DATA5 LOAD PARAMETER REG 0
00005E 58F0 1030          00030   75+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000062 05EF                       76+        BALR  14,15 LINK TO PUT ROUTINE
                                  77         PUT   PCHDCB,DATA6            PUNCH BILL CARD FOR JUNE
000064 4110 C4DA          004E0   78+        LA    1,PCHDCB LOAD PARAMETER REG 1
000068 4100 C35C          00362   79+        LA    0,DATA6 LOAD PARAMETER REG 0
00006C 58F0 1030          00030   80+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000070 05EF                       81+        BALR  14,15 LINK TO PUT ROUTINE
                                  82         PUT   PCHDCB,DATA7            PUNCH BILL CARD FOR JULY
000072 4110 C4DA          004E0   83+        LA    1,PCHDCB LOAD PARAMETER REG 1
000076 4100 C370          00376   84+        LA    0,DATA7 LOAD PARAMETER REG 0
00007A 58F0 1030          00030   85+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
00007E 05EF                       86+        BALR  14,15 LINK TO PUT ROUTINE
                                  87         PUT   PCHDCB,DATA8            PUNCH BILL CARD FOR AUGUST
000080 4110 C4DA          004E0   88+        LA    1,PCHDCB LOAD PARAMETER REG 1
000084 4100 C384          0038A   89+        LA    0,DATA8 LOAD PARAMETER REG 0
000088 58F0 1030          00030   90+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
00008C 05EF                       91+        BALR  14,15 LINK TO PUT ROUTINE
                                  92         PUT   PCHDCB,DATA9            PUNCH BILL CARD FOR SEPTEMBER
00008E 4110 C4DA          004F0   93+        LA    1,PCHDCB LOAD PARAMETER REG 1
000092 4100 C398          0039E   94+        LA    0,DATA9 LOAD PARAMETER REG 0
000096 58F0 1030          00030   95+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
00009A 05EF                       96+        BALR  14,15 LINK TO PUT ROUTINE
                                  97         PUT   PCHDCB,DATA10           PUNCH BILL CARD FOR OCTOBER
00009C 4110 C4DA          004E0   98+        LA    1,PCHDCB LOAD PARAMETER REG 1
0000A0 4100 C3AC          003B2   99+        LA    0,DATA10 LOAD PARAMETER REG 0
0000A4 58F0 1030          00030  100+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
0000A8 05EF                      101+        BALR  14,15 LINK TO PUT ROUTINE
                                 102         PUT   PCHDCB,DATA11           PUNCH BILL CARD FOR NOVEMBER
0000AA 4110 C4DA          004E0  103+        LA    1,PCHDCB LOAD PARAMETER REG 1
0000AE 4100 C3C0          003C6  104+        LA    0,DATA11 LOAD PARAMETER REG 0
0000B2 58F0 1030          00030  105+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
0000B6 05EF                      106+        BALR  14,15 LINK TO PUT ROUTINE
                                 107         PUT   PCHDCB,DATA12           PUNCH BILL CARD FOR DECEMBER
0000B8 4110 C4DA          004E0  108+        LA    1,PCHDCB LOAD PARAMETER REG 1
0000BC 4100 C3D4          003DA  109+        LA    0,DATA12 LOAD PARAMETER REG 0
0000C0 58F0 1030          00030  110+        L     15,48(0,1) LOAD PUT ROUTINE ADDR.
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01OCT71  6/22/73

0000C4 05EF                        111+          BALR  14,15 LINK TO PUT ROUTINE
                                   112 **
                                   113 ***
                                   114 **
                                   115           OPEN  (READIN,(INPUT))        OPEN ASSOCIATED DCBS TO COMPLETE
0000C6 0700                        116+          CNOP  0,4
0000C8 4510 C0CA             000D0 117+          BAL   1,*+8 LOAD REG1 W/LIST ADDR.
0000CC 80                          118+          DC    AL1(128) OPTION BYTE
0000CD 0003E0                      119+          DC    AL3(READIN) DCB ADDRESS
0000D0 0A13                        120+          SVC   19 ISSUE OPEN SVC
                                   121           OPEN  (PRINTOUT,(OUTPUT),PUNCHOUT,(OUTPUT)) INFO ON BILL CARDS
0000D2 0700                        122+          CNOP  0,4
0000D4 4510 C0DA             000E0 123+          BAL   1,*+12 LOAD REG1 W/LIST ADDR.
0000D8 0F                          124+          DC    AL1(15) OPTION BYTE
0000D9 000480                      125+          DC    AL3(PRINTOUT) DCB ADDRESS
0000DC 8F                          126+          DC    AL1(143) OPTION BYTE
0000DD 000430                      127+          DC    AL3(PUNCHOUT) DCB ADDRESS
0000E0 0A13                        128+          SVC   19 ISSUE OPEN SVC
                                   129 **
                                   130 ***       THE FOLLOWING 'PUTS' TO THE PUNCH AND PRINT DATA SETS GET
                                   131 ***       THE ADDRESS OF THE BUFFERS.  THE FIRST 'PUT' IN LOCATE MODE
                                   132 ***       CAUSES NO INPUT/OUTPUT OPERATION.  THEREFORE, NO SEQUENCE
                                   133 ***       CHECKING IS PERFORMED DURING THE FIRST 'PUT' ISSUED FOR THE
                                   134 ***       ASSOCIATED DATA SETS.
                                   135 **
                                   136           PUT   PUNCHOUT                GET ADDR OF BUFFER FOR PUNCH DCB
0000E2 4110 C42A             00430 137+          LA    1,PUNCHOUT LOAD PARAMETER REG 1
0000E6 58F0 1030             00030 138+          L     15,48(0,1) LOAD PUT ROUTINE ADDR.
0000EA 05EF                        139+          BALR  14,15 LINK TO PUT ROUTINE
0000EC 1841                        140           LR    R4,R1                   SAVE BUFFER ADDR IN REGISTER 4
                                   141           PUT   PRINTOUT                GET ADDR OF BUFFER FOR PRINT DCB
0000EE 4110 C47A             00480 142+          LA    1,PRINTOUT LOAD PARAMETER REG 1
0000F2 58F0 1030             00030 143+          L     15,48(0,1) LOAD PUT ROUTINE ADDR.
0000F6 05EF                        144+          BALR  14,15 LINK TO PUT ROUTINE
0000F8 1851                        145           LR    R5,R1                   SAVE BUFFER ADDR IN REGISTER 5
                                   146 **
                                   147 ***
                                   148 **
0000FA                             149 READCARD  EQU   *
                                   150           GET   READIN                  READ THE DATA CARD
0000FA 4110 C3DA             003E0 151+          LA    1,READIN LOAD PARAMETER REG 1
0000FE 58F0 1030             00030 152+          L     15,48(0,1) LOAD GET ROUTINE ADDR.
000102 05EF                        153+          BALR  14,15 LINK TO GET ROUTINE
000104 1831                        154           LR    R3,R1                   SAVE BUFFER ADDR FOR FURTHER USE
000106 D213 4000 C252 00000 00258  155           MVC   0(20,R4),BLANKS         CLEAR FIRST BYTES OF BUFFER
00010C D213 4014 30C0 00014 00000  156           MVC   20(20,R4),0(R3)         MOVE DATA INTO PUNCH BUFFER
                                   157           PUT   PUNCHOUT                PUNCH DATA INTO CARD
000112 4110 C42A             00430 158+          LA    1,PUNCHOUT LOAD PARAMETER REG 1
000116 58F0 1030             00030 159+          L     15,48(0,1) LOAD PUT ROUTINE ADDR.
00011A 05EF                        160+          BALR  14,15 LINK TO PUT ROUTINE
00011C 1841                        161           LR    R4,R1                   SAVE BUFFER ADDRESS
                                   162 **
                                   163 ***
                                   164 **
00011E D213 5000 C252 00000 00258  165           MVC   0(20,R5),BLANKS         CLEAR FIRST BYTES OF BUFFER
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                F01OCT71  6/22/73

000124 D22B 5014 3000 0001A 00000   166          MVC   20(44,R5),0(R3)          MOVE DATA INTO PRINT BUFFER
                                    167          PUT   PRINTOUT                 PRINT FIRST LINE OF DATA ON CARD
00012A 4110 C47A             00480  168+         LA    1,PRINTOUT LOAD PARAMETER REG 1
00012E 58F0 1030             00030  169+         L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000132 05EF                         170+         BALR  14,15 LINK TO PUT ROUTINE
000134 1851                         171          LR    R5,R1                    SAVE BUFFER ADDRESS
                                    172 **
                                    173 ***
                                    174 **
000136 D227 5000 C252 00000 00258   175          MVC   0(40,R5),BLANKS          CLEAR FIRST BYTES OF BUFFER
00013C D217 5028 C27A 00028 00280   176          MVC   40(24,R5),NAME           MOVE PRINT DATA INTO BUFFER
                                    177 **
                                    178 ***      THE 'CNTRL' MACRO IS USED TO REGULATE LINE POSITIONING.
                                    179 ***      SKIPS AND SPACES ARE USED SO THAT PRINTING IS ONLY PERFORMED
                                    180 ***      ON THE DESIRED LINES OF THE BILL CARD.
                                    181 **
                                    182          CNTRL PRINTOUT,SK,8            SKIP TO LINE 15 OF BILL CARD
000142 4110 C47A             00480  183+         LA    1,PRINTOUT LOAD PARAMETER REG 1
000146 4100 0008             00008  184+         LA    0,8(0,0) LOAD PARAMETER REG 0
00014A 1300                         185+         LCR   0,0 INDICATE SK ACTION
00014C 58F1 0054             00054  186+         L     15,84(1,0) LOAD CONTROL ROUT.ADDR
000150 05EF                         187+         BALR  14,15 LINK TO CONTROL ROUT.
                                    188          PUT   PRINTOUT                 PRINT NAME ON CARD
000152 4110 C47A             00480  189+         LA    1,PRINTOUT LOAD PARAMETER REG 1
000156 58F0 1030             00030  190+         L     15,48(0,1) LOAD PUT ROUTINE ADDR.
00015A 05EF                         191+         BALR  14,15 LINK TO PUT ROUTINE
00015C 1851                         192          LR    R5,R1                    SAVE BUFFER ADDRESS
                                    193 **
                                    194 ***
                                    195 **
00015E D227 5000 C252 00000 00258   196          MVC   0(40,R5),BLANKS          CLEAR FIRST 40 BYTES OF BUFFER
000164 D217 5028 C292 00028 00298   197          MVC   40(24,R5),ADDRESS        MOVE CUSTOMER ADDR INTO BUFFER
                                    198          CNTRL PRINTOUT,SK,9            SKIP TO LINE 17 OF BILL CARD
00016A 4110 C47A             00480  199+         LA    1,PRINTOUT LOAD PARAMETER REG 1
00016E 4100 0009             00009  200+         LA    0,9(0,0) LOAD PARAMETER REG 0
000172 1300                         201+         LCR   0,0 INDICATE SK ACTION
000174 58F1 0054             00054  202+         L     15,84(1,0) LOAD CONTROL ROUT.ADDR
000178 05EF                         203+         BALR  14,15 LINK TO CONTROL ROUT.
                                    204          PUT   PRINTOUT                 PRINT ADDRESS ON BILL CARD
00017A 4110 C47A             00480  205+         LA    1,PRINTOUT LOAD PARAMETER REG 1
00017E 58F0 1030             00030  206+         L     15,48(0,1) LOAD PUT ROUTINE ADDR.
000182 05EF                         207+         BALR  14,15 LINK TO PUT ROUTINE
000184 1851                         208          LR    R5,R1                    SAVE BUFFER ADDRESS
                                    209 **
                                    210 ***
                                    211 **
000186 D227 5000 C252 00000 00258   212          MVC   0(40,R5),BLANKS          CLEAR FIRST 40 BYTES OF BUFFER
00018C D217 5028 C2AA 00028 002B0   213          MVC   40(24,R5),CITYST         MOVE CITY/STATE INTO BUFFER
                                    214          CNTRL PRINTOUT,SK,10           SKIP TO LINE 19 OF BILL CARD
000192 4110 C47A             00480  215+         LA    1,PRINTOUT LOAD PARAMETER REG 1
000196 4100 000A             0000A  216+         LA    0,10(0,0) LOAD PARAMETER REG 0
00019A 1300                         217+         LCR   0,0 INDICATE SK ACTION
00019C 58F1 0054             00054  218+         L     15,84(1,0) LOAD CONTROL ROUT.ADDR
0001A0 05EF                         219+         BALR  14,15 LINK TO CONTROL ROUT.
                                    220          PUT   PRINTOUT                 PRINT CITY/STATE ON BILL CARD
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F01OCT71   6/22/73

0001A2 4110 C47A             00480   221+          LA     1,PRINTOUT LOAD PARAMETER REG 1
0001A6 58F0 1030             00030   222+          L      15,48(0,1) LOAD PUT ROUTINE ADDR.
0001AA 05EF                          223+          BALR   14,15 LINK TO PUT ROUTINE
0001AC 1851                          224           LR     R5,R1                    SAVE BUFFER ADDRESS
                                     225 **
                                     226 ***
                                     227 **
0001AE D209 5000 C252 00000 00258    228           MVC    0(10,R5),BLANKS          CLEAR FIRST 10 BYTES OF BUFFER
0001B4 D235 500A C2C2 0000A 002C8    229           MVC    10(54,R5),NOTE           PUT REPLY NOTE IN PRINT BUFFER
                                     230           CNTRL  PRINTOUT,SK,12           SKIP TO LINE 23 OF BILL CARD
0001BA 4110 C47A             00480   231+          LA     1,PRINTOUT LOAD PARAMETER REG 1
0001BE 4100 000C             0000C   232+          LA     0,12(0,0) LOAD PARAMETER REG 0
0001C2 1300                          233+          LCR    0,0 INDICATE SK ACTION
0001C4 58F1 0054             00054   234+          L      15,84(1,0) LOAD CONTROL ROUT.ADDR
0001C8 05EF                          235+          BALR   14,15 LINK TO CONTROL ROUT.
                                     236           CNTRL  PRINTOUT,SP,2            SPACE DOWN TO LINE 25 OF CARD
0001CA 4110 C47A             00480   237+          LA     1,PRINTOUT LOAD PARAMETER REG 1
0001CE 4100 0002             00002   238+          LA     0,2(0,0) LOAD PARAMETER REG 0
0001D2 58F1 0054             00054   239+          L      15,84(1,0) LOAD CONTROL ROUT.ADDR
0001D6 05EF                          240+          BALR   14,15 LINK TO CONTROL ROUT.
                                     241           PUT    PRINTOUT                 PRINT LAST LINE ON BILL CARD
0001D8 4110 C47A             00480   242+          LA     1,PRINTOUT LOAD PARAMETER REG 1
0001DC 58F0 1030             00030   243+          L      15,48(0,1) LOAD PUT ROUTINE ADDR.
0001E0 05EF                          244+          BALR   14,15 LINK TO PUT ROUTINE
0001E2 1851                          245           LR     R5,R1                    SAVE BUFFER ADDRESS
                                     246 ****************************************
                                     247 *                                      *
0001E4 47F0 C0F4             000FA   248           B      RFADCARD                 BRANCH TO PROCESS ANOTHER CARD
                                     249 *                                      *
                                     250 ****************************************
0001E8                               251 EOJOB     EQU    *
                                     252           CLOSE  (READIN,,PCHDCB)         CLOSE ALL DCB'S
0001E8                               253+          CNOP   0,4
0001E8 4510 C1EE             001F4   254+          BAL    1,*+12 BRANCH AROUND LIST
0001EC 00                            255+          DC     AL1(0) OPTION BYTE
0001ED 0003E0                        256+          DC     AL3(READIN) DCB ADDRESS
0001F0 80                            257+          DC     AL1(128) OPTION BYTE
0001F1 0004E0                        258+          DC     AL3(PCHDCB) DCB ADDRESS
0001F4 0A14                          259+          SVC    20 ISSUE CLOSE SVC
                                     260           CLOSE  (PUNCHOUT,,PRINTOUT)     CLOSE ALL DCB'S
0001F6 0700                          261+          CNOP   0,4
0001F8 4510 C1FE             00204   262+          BAL    1,*+12 BRANCH AROUND LIST
0001FC 00                            263+          DC     AL1(0) OPTION BYTE
0001FD 000430                        264+          DC     AL3(PUNCHOUT) DCB ADDRESS
000200 80                            265+          DC     AL1(128) OPTION BYTE
000201 000480                        266+          DC     AL3(PRINTOUT) DCB ADDRESS
000204 0A14                          267+          SVC    20 ISSUE CLOSE SVC
000206 58D0 C20E             00214   268           L      R13,SAVAREA+4            RESTORE ADDR IN SAVE REGISTER
                                     269           RETURN (14,12)                  RESTORE ALL REGISTERS
00020A 98EC D00C             0000C   270+          LM     14,12,12(13) RESTORE THE REGISTERS
00020E 07FE                          271+          BR     14 RETURN
                                     272 **
                                     273 ***             DC'S AND DCB STATEMENTS
                                     274 **
000210 0000000000000000              275 SAVAREA   DC     18F'0'
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71   6/22/73

000258 4040404040404040              276 BLANKS   DC    CL40'                                        '
000280 D4D94B40D1D6C8D5              277 NAME     DC    CL24'MR. JOHN DOE            '
000298 F3F5F2F540C1D5E8              278 ADDRESS  DC    CL24'3525 ANY STREET         '
0002B0 C1D5E8D5C1D4C540              279 CITYST   DC    CL24'ANYNAME CITY, USA 00000 '
0002C8 D7D3C5C1E2C540D9              280 NOTE     DC    CL54'PLEASE RETURN THIS BILL CARD WITH YOUR PAYMENT.'
0002FE D1C1D5E4C1D9E840              281 DATA1    DC    CL20'JANUARY         $5.95 '
000312 C6C5C2D9E4C1D9E8              282 DATA2    DC    CL20'FEBRUARY        $5.95 '
000326 D4C1D9C3C8404040              283 DATA3    DC    CL20'MARCH           $5.95 '
00033A C1D7D9C9D3404040              284 DATA4    DC    CL20'APRIL           $5.95 '
00034E D4C1E84040404040              285 DATA5    DC    CL20'MAY             $5.95 '
000362 D1E4D5C5404040 40             286 DATA6    DC    CL20'JUNE            $5.95 '
000376 D1E4D3E840404040              287 DATA7    DC    CL20'JULY            $5.95 '
00038A C1E4C7E4E2E34040              288 DATA8    DC    CL20'AUGUST          $5.95 '
00039E E2C5D7E3C5D4C2C5              289 DATA9    DC    CL20'SEPTEMBER       $5.95 '
0003B2 D6C3E3D6C2C5D940              290 DATA10   DC    CL20'OCTOBER         $5.95 '
0003C6 D5D6E5C5D4C2C5D9              291 DATA11   DC    CL20'NOVEMBER        $5.95 '
0003DA C4C5C3C5D4C2C5D9              292 DATA12   DC    CL20'DECEMBER        $5.95 '
                                     293 **
                                     294 ***
                                     295 **
                                     296 READIN   DCB   DDNAME=READD,DSORG=PS,EODAD=EOJOB,BLKSIZE=80,LRECL=80,  C
                                                        BUFNO=1,DEVD=RD,MACRF=GL,RECFM=F,FUNC=RPW


                                     298+**                    DATA CONTROL BLOCK
                                     299+*
0003DE                               300+          ORG   *-16 TO ELIMINATE UNUSED SPACE
0003E0                               301+READIN    DS    0F'0' ORIGIN ON WORD BOUNDARY
0003F0                               302+          ORG   *+16 TO ORIGIN GENERATION

                                     304+**                    READER/PUNCH DEVICE INTERFACE

0003F0 0000                          306+          DC    BL2'0000000000000000' MODE, STACK, DEVT
0003F2 00                            307+          DC    X'00'
0003F3 70                            308+          DC    AL1(112)  FUNC

                                     310+**                    COMMON ACCESS METHOD INTERFACE

0003F4 01                            312+          DC    AL1(1)  BUFNO
0003F5 000001                        313+          DC    AL3(1)  BUFCB
0003F8 0000                          314+          DC    AL2(0)  BUFL
0003FA 4000                          315+          DC    BL2'0100000000000000' DSORG
0003FC 00000001                      316+          DC    A(1)  IOBAD

                                     318+**                    FOUNDATION EXTENSION

000400 00                            320+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
000401 0001E8                        321+          DC    AL3(EOJOB)  EODAD
000404 80                            322+          DC    BL1'10000000' RECFM
000405 000000                        323+          DC    AL3(0)  EXLST

                                     325+**                    FOUNDATION BLOCK

000408 D9C5C1C4C4404040              327+          DC    CL8'READD' DDNAME
000410 02                            328+          DC    BL1'00000010' OFLGS
```

```
  LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71  6/22/73

000411 00                          329+          DC    BL1'00000000' IFLG
000412 4800                        330+          DC    BL2'0100100000000000' MACR

                                   332+*                     BSAM-BPAM-QSAM INTERFACE

000414 00                          334+          DC    BL1'00000000' RER1
000415 000001                      335+          DC    AL3(1) CHECK, GERR, PERR
000418 00000001                    336+          DC    A(1) SYNAD
00041C 0000                        337+          DC    H'0' CIND1, CIND2
00041E 0050                        338+          DC    AL2(80) BLKSIZE
000420 00000000                    339+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
000424 00000001                    340+          DC    A(1) IOBA
000428 00                          341+          DC    AL1(0) NCP
000429 000001                      342+          DC    AL3(1) EOER, EOBAD

                                   344+*                          QSAM INTERFACE

00042C 00000001                    346+          DC    A(1) RECAD
000430 0000                        347+          DC    H'0' QSWS
000432 0050                        348+          DC    AL2(80) LRECL
000434 00                          349+          DC    BL1'00000000' EROPT
000435 000001                      350+          DC    AL3(1) CNTRL
000438 00000000                    351+          DC    F'0' PRECL
00043C 00000001                    352+          DC    A(1) EOB
                                   353 PUNCHOUT  DCB   DDNAME=PCHDD,DSORG=PS,BLKSIZE=40,LRECL=40,BUFNO=1,       X
                                                       DEVD=RD,MACRF=PL,RECFM=F,FUNC=RPW


                                   355+*                     DATA CONTROL BLOCK
                                   356+*
000430                             357+          ORG   *-16 TO ELIMINATE UNUSED SPACE
000430                             358+PUNCHOUT  DS    0F'0' ORIGIN ON WORD BOUNDARY
000440                             359+          ORG   *+16 TO ORIGIN GENERATION

                                   361+*                     READER/PUNCH DEVICE INTERFACE

000440 0000                        363+          DC    BL2'0000000000000000' MODE, STACK, DEVT
000442 00                          364+          DC    X'00'
000443 70                          365+          DC    AL1(112) FUNC

                                   367+*                     COMMON ACCESS METHOD INTERFACE

000444 01                          369+          DC    AL1(1) BUFNO
000445 000001                      370+          DC    AL3(1) BUFCB
000448 0000                        371+          DC    AL2(0) BUFL
00044A 4000                        372+          DC    BL2'0100000000000000' DSORG
00044C 00000001                    373+          DC    A(1) IOBAD

                                   375+*                     FOUNDATION EXTENSION

000450 00                          377+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
000451 000001                      378+          DC    AL3(1) EODAD
000454 80                          379+          DC    BL1'10000000' RECFM
000455 000000                      380+          DC    AL3(0) EXLST
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                    F01OCT71  6/22/73

                                  382+*                      FOUNDATION BLOCK
000458 D7C3C8C4C4404040          384+          DC    CL8'PCHDD' DINAME
000460 02                        385+          DC    BL1'00000010' OFLGS
000461 00                        386+          DC    BL1'00000000' IFLG
000462 0048                      387+          DC    BL2'0000000001001000' MACR

                                  389+*                      BSAM-BPAM-QSAM INTERFACE
000464 00                        391+          DC    BL1'00000000' BERR1
000465 000001                    392+          DC    AL3(1) CHECK, GERR, PERR
000468 00000001                  393+          DC    A(1) SYNAD
00046C 0000                      394+          DC    H'0' CIND1, CIND2
00046E 0028                      395+          DC    AL2(40) BLKSIZE
000470 00000000                  396+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
000474 00000001                  397+          DC    A(1) IOBA
000478 00                        398+          DC    AL1(0) NCP
000479 000001                    399+          DC    AL3(1) EOBR, EOBAD

                                  401+*                      QSAM INTERFACE
00047C 00000001                  403+          DC    A(1) RECAD
000480 0000                      404+          DC    H'0' QSWS
000482 0028                      405+          DC    AL2(40) LRECL
000484 00                        406+          DC    BL1'00000000' EROPT
000485 000001                    407+          DC    AL3(1) CNTRL
000488 00000000                  408+          DC    F'0' PRECL
00048C 00000001                  409+          DC    A(1) EOB
                                 410 PRINTOUT DCB   DDNAME=PRTDD,DSORG=PS,BLKSIZE=64,LRECL=64,DEVD=RD,      X
                                                    MACRF=PLC,RECFM=F,FUNC=RPWX,BUFNO=1


                                  412+*                      DATA CONTROL BLOCK
                                  413+*
000480                           414+          ORG   *-16 TO ELIMINATE UNUSED SPACE
000480                           415+PRINTOUT DS     0F'0' ORIGIN ON WORD BOUNDARY
000490                           416+          ORG   *+16 TO ORIGIN GENERATION

                                  418+*                      READER/PUNCH DEVICE INTERFACE
000490 0000                      420+          DC    BL2'0000000000000000' MODE, STACK, DEVT
000492 00                        421+          DC    X'00'
000493 74                        422+          DC    AL1(116) FUNC

                                  424+*                      COMMON ACCESS METHOD INTERFACE
000494 01                        426+          DC    AL1(1) BUFNO
000495 000001                    427+          DC    AL3(1) BUFCB
000498 0000                      428+          DC    AL2(0) BUFL
00049A 4000                      429+          DC    BL2'0100000000000000' DSORG
00049C 00000001                  430+          DC    A(1) IOBAD

                                  432+*                      FOUNDATION EXTENSION
0004A0 00                        434+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
```

```
 LOC  OBJECT CODE     ADDR1 ADDR2  STMT   SOURCE STATEMENT                                                F01OCT71   6/22/73

0004A1 000001                      435+          DC    AL3(1) EODAD
0004A4 80                          436+          DC    BL1'10000000' RECFM
0004A5 000000                      437+          DC    AL3(0) EXLST

                                   439+*                     FOUNDATION BLOCK

0004A8 D7D9E3C4C4404040            441+          DC    CL8'PRTDD' DDNAME
0004B0 02                          442+          DC    BL1'0000001C' OFLGS
0004B1 00                          443+          DC    BL1'00000000' IFLG
0004B2 004A                        444+          DC    BL2'0000000001001010' MACR

                                   446+*                     BSAM-BPAM-QSAM INTERFACE

0004B4 00                          448+          DC    BL1'00000000' RER1
0004B5 000001                      449+          DC    AL3(1) CHECK, GERR, PERR
0004B8 00000001                    450+          DC    A(1) SYNAD
0004BC 0000                        451+          DC    H'0' CIND1, CIND2
0004BE 0040                        452+          DC    AL2(64) BLKSIZE
0004C0 00000000                    453+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
0004C4 00000001                    454+          DC    A(1) IOBA
0004C8 00                          455+          DC    AL1(0) NCP
0004C9 000001                      456+          DC    AL3(1) EOBR, EOBAD

                                   458+*                         QSAM INTERFACE

0004CC 00000001                    460+          DC    A(1) RECAD
0004D0 0000                        461+          DC    H'0' QSWS
0004D2 0040                        462+          DC    AL2(64) LRECL
0004D4 00                          463+          DC    BL1'00000000' EROPT
0004D5 000001                      464+          DC    AL3(1) CNTRL
0004D8 00000000                    465+          DC    F'0' PRECL
0004DC 00000001                    466+          DC    A(1) EOB
                                   467 PCHDCB    DCB   DDNAME=PCHXX,DSORG=PS,BLKSIZE=20,LRECL=20,BUFNO=1,       X
                                                       MACRF=PM,RECFM=F


                                   469+*                     DATA CONTROL BLOCK
                                   470+*
0004E0                             471+PCHDCB    DC    0F'0' ORIGIN ON WORD BOUNDARY

                                   473+*                     DIRECT ACCESS DEVICE INTERFACE

0004E0 0000000000000000            475+          DC    BL16'0' FDAD,DVTBL
0004F0 00000000                    476+          DC    A(0) KEYLE,DEVT,TRBAL

                                   478+*                     COMMON ACCESS METHOD INTERFACE

0004F4 01                          480+          DC    AL1(1) BUFNO
0004F5 000001                      481+          DC    AL3(1) BUFCB
0004F8 0000                        482+          DC    AL2(0) BUFL
0004FA 4000                        483+          DC    BL2'0100000000000000' DSORG
0004FC 00000001                    484+          DC    A(1) IOBAD

                                   486+*                     FOUNDATION EXTENSION
```

```
 LOC  OBJECT CODE    ADDR1 ADDR2  STMT   SOURCE STATEMENT                                  F010CT71   6/22/73

000500 00                          488+          DC    BL1'00000000' BFTEK,BFLN,HIARCHY
000501 000001                      489+          DC    AL3(1) EODAD
000504 80                          490+          DC    BL1'10000000' RECFM
000505 000000                      491+          DC    AL3(0) EXLST

                                   493+*                     FOUNDATION BLOCK

000508 D7C3C8E7E7404040            495+          DC    CL8'PCHXX' DDNAME
000510 02                          496+          DC    BL1'00000010' OFLGS
000511 00                          497+          DC    BL1'00000000' IFLG
000512 0050                        498+          DC    BL2'0000000001010000' MACR

                                   500+*                     BSAM-BPAM-QSAM INTERFACE

000514 00                          502+          DC    BL1'00000000' RER1
000515 000001                      503+          DC    AL3(1) CHECK, GERR, PERR
000518 00000001                    504+          DC    A(1) SYNAD
00051C 0000                        505+          DC    H'0' CIND1, CIND2
00051E 0014                        506+          DC    AL2(20) BLKSIZE
000520 00000000                    507+          DC    F'0' WCPO, WCPL, OFFSR, OFFSW
000524 00000001                    508+          DC    A(1) IOBA
000528 00                          509+          DC    AL1(0) NCP
000529 000001                      510+          DC    AL3(1) EOBR, EOBAD

                                   512+*                         QSAM INTERFACE

00052C 00000001                    514+          DC    A(1) RECAD
000530 0000                        515+          DC    H'0' QSWS
000532 0014                        516+          DC    AL2(20) LRECL
000534 00                          517+          DC    BL1'00000000' EROPT
000535 000001                      518+          DC    AL3(1) CNTRL
000538 00000000                    519+          DC    F'0' PRECL
00053C 00000001                    520+          DC    A(1) EOB
                                   521           END
```

6/22/73

POS.ID	REL.ID	FLAGS	ADDRESS
01	01	08	000019
01	01	08	0000CD
01	01	08	0000D9
01	01	08	0000DD
01	01	08	0001ED
01	01	08	0001F1
01	01	08	0001FD
01	01	08	000201
01	01	08	000401

6/22/73

SYMBOL	LEN	VALUE	DEFN	REFERENCES														
ADDRESS	00024	000298	00278	0197														
BLANKS	00040	000258	00276	0155	0165	0175	0196	0212	0228									
CITYST	00024	0002B0	00279	0213														
DATA1	00020	0002FE	00281	0054														
DATA10	00020	0003B2	00290	0099														
DATA11	00020	0003C6	00291	0104														
DATA12	00020	0003DA	00292	0109														
DATA2	00020	000312	00282	0059														
DATA3	00020	000326	00283	0064														
DATA4	00020	00033A	00284	0069														
DATA5	00020	00034E	00285	0074														
DATA6	00020	000362	00286	0079														
DATA7	00020	000376	00287	0084														
DATA8	00020	00038A	00288	0089														
DATA9	00020	00039E	00289	0094														
EOJOB	00001	0001E8	00251	0321														
NAME	00024	000280	00277	0176														
NOTE	00054	0002C8	00280	0229														
PCHDCB	00004	0004E0	00471	0050	0053	0058	0063	0068	0073	0078	0083	0088	0093	0098	0103	0108	0258	
PRINTOUT	00004	000480	00415	0125	0142	0168	0183	0189	0199	0205	0215	0221	0231	0237	0242	0266		
PUNCHOUT	00004	000430	00358	0127	0137	0158	0264											
READCARD	00001	0000FA	00149	0248														
READIN	00004	0003E0	00301	0119	0151	0256												
RPWPGM	00001	000000	00001															
R1	00001	000001	00016	0140	0145	0154	0161	0171	0192	0208	0224	0245						
R10	00001	00000A	00025															
R11	00001	00000B	00026															
R12	00001	00000C	00027	0037	0038													
R13	00001	00000D	00028	0039	0040	0041	0042	0268										
R14	00001	00000E	00029															
R15	00001	00000F	00030															
R2	00001	000002	00017															
R3	00001	000003	00018	0154	0156	0166												
R4	00001	000004	00019	0140	0155	0156	0161											
R5	00001	000005	00020	0145	0165	0166	0171	0175	0176	0192	0196	0197	0208	0212	0213	0224	0228	0229
				0245														
R6	00001	000006	00021															
R7	00001	000007	00022															
R8	00001	000008	00023	0040	0042													
R9	00001	000009	00024															
SAVAREA	00004	000210	00275	0039	0041	0268												

NO STATEMENTS FLAGGED IN THIS ASSEMBLY
STATISTICS SOURCE RECORDS (SYSIN) = 170 SOURCE RECORDS (SYSLIB) = 3041
OPTIONS IN EFFECT LIST, DECK, LOAD, NORENT, XREF, NOTEST, ALGN, OS, NOTERM, LINECNT = 55
555 PRINTED LINES

```
IEF285I   SYS73173.T150833.SV000.T30102.R0000009       SYSOUT
IEF285I   VOL SER NOS= RPC001.
IEF285I   SYS73173.T150833.RV000.T30102.LOADSET        PASSED
IEF285I   VOL SER NOS= SLIB21.
IEF285I   SYS73173.T150833.RV000.T30102.SYSUT1         DELETED
IEF285I   VOL SER NOS= RPC003.
IEF285I   SYS73173.T150833.RV000.T30102.SYSUT2         DELETED
IEF285I   VOL SER NOS= MACROP.
IEF285I   SYS73173.T150833.RV000.T30102.SYSUT3         DELETED
IEF285I   VOL SER NOS= RPC001.
IEF285I   SYS1.MACLIB                                  KEPT
IEF285I   VOL SER NOS= SLIB21.
IEF285I   SYS73173.T150833.SV000.T30102.R0000010       SYSOUT
IEF285I   VOL SER NOS= PVOL62.
IEF285I   SYS73173.T150833.RV000.T30102.S0000011       SYSIN
IEF285I   VOL SER NOS= MACROP.
IEF285I   SYS73173.T150833.RV000.T30102.S0000011       DELETED
IEF285I   VOL SER NOS= MACROP.
IEF373I STEP /ASM     / START 73173.1511
IEF374I STEP /ASM     / STOP  73173.1513 CPU    0MIN 21.20SEC MAIN  78K LCS   0K
XXLKED   EXEC  PGM=IEWL,PARM=(XREF,LET,LIST,NCAL),REGION=96K,          X00000100
XX             COND=(5,LT,ASM)                                         00000110
XXSYSPRINT DD  SYSOUT=A                                                00000120
XXSYSLMOD DD   DSNAME=&GOSET(GO),UNIT=SYSDA,SPACE=(1024,(50,20,1)),    X00000130
XX             DISP=(MOD,PASS)                                         00000140
XXSYSUT1 DD    DSNAME=&SYSUT1,UNIT=SYSDA,                              X00000150
XX             SPACE=(1024,(50,20))                                    00000160
XXSYSLIB  DD  DSNAME=SYS1.LINKLIB,DISP=SHR                             00000170
XXSYSLIN  DD  DSNAME=*.ASM.SYSGO,DISP=(OLD,DELETE)                     00000180
XX        DD   DDNAME=SYSIN                                            00000190
IEF236I ALLOC. FOR T30102   LKED
IEF237I 132   ALLOCATED TO SYSPRINT
IEF237I 130   ALLOCATED TO SYSLMOD
IEF237I 251   ALLOCATED TO SYSUT1
IEF237I 130   ALLOCATED TO SYSLIB
IEF237I 130   ALLOCATED TO SYSLIN
```

```
F44-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED XREF,LET,LIST,NCAL
        DEFAULT OPTION(S) USED -  SIZE=(100352,18432)

                                        CROSS REFERENCE TABLE

 CONTROL SECTION                        ENTRY
   NAME     ORIGIN  LENGTH               NAME   LOCATION     NAME   LOCATION     NAME   LOCATION     NAME   LOCATION
 RPWFGM       00      540

  LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION            LOCATION  REFERS TO SYMBOL  IN CONTROL SECTION

ENTRY ADDRESS      00
TOTAL LENGTH      540

****GO        DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET
```

```
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS73173.T150833.SV000.T30102.R0000012        SYSOUT
IEF285I   VOL SER NOS= MACROP.
IEF285I   SYS73173.T150833.RV000.T30102.GOSET           PASSED
IEF285I   VOL SER NOS= SLIB21.
IEF285I   SYS73173.T150833.RV000.T30102.SYSUT1          DELETED
IEF285I   VOL SER NOS= PVOL61.
IEF285I   SYS1.LINKLIB                                  KEPT
IEF285I   VOL SER NOS= SLIB21.
IEF285I   SYS73173.T150833.RV000.T30102.LOADSET         DELETED
IEF285I   VOL SER NOS= SLIB21.
IEF373I STEP /LKED    / START 73173.1513
IEF374I STEP /LKED    / STOP  73173.1514 CPU    0MIN 00.57SEC MAIN  96K LCS    0K
IXGO     EXEC  PGM=*.LKED.SYSLMOD,COND=((5,LT,ASM),(4,LT,LKED))          00000200
//GO.SYSUDUMP  DD  SYSOUT=A
//GO.PCHXX  DD  UNIT=00D
//GO.READD  DD  UNIT=3525
//GO.PCHDD  DD  UNIT=AFF=READD
//GO.PRTDD  DD  UNIT=AFF=READD
IEF236I ALLOC. FOR T30102   GO
IEF237I 130   ALLOCATED TO PGM=*.DD
IEF237I 132   ALLOCATED TO SYSUDUMP
IEF237I 00D   ALLOCATED TO PCHXX
IEF237I 013   ALLOCATED TO READD
IEF237I 013   ALLOCATED TO PCHDD
IEF237I 013   ALLOCATED TO PRTDD
```

READER'S COMMENT FORM

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications will delay response, however. For more direct handling of such requests, please contact your IBM representative or the IBM Branch Office serving your locality.

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

What is your occupation? _____

Number of latest Technical Newsletter (if any) concerning this publication: _____

Please indicate in the space below if you wish a reply.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Cut Along Line

Your comments, please . . .

This manual is part of a library that serves as a reference source for system analysts, programmers, and operators of IBM systems. Your comments on the other side of this form will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become property of IBM.

Fold Fold

First Class
Permit 81
Poughkeepsie
New York

Business Reply Mail
No postage stamp necessary if mailed in the U.S.A.

Postage will be paid by:

International Business Machines Corporation
Department D58, Building 706-2
PO Box 390
Poughkeepsie, New York 12602

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

OS and OS/VS Support for the IBM 3505 and 3525 Printed in U.S.A. GC21-5097-0

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)