File 1401/1460-33
Form C24-3053-3

IBM Systems Reference Library

Merge 6 Specifications and Operating Procedures IBM 1401 and 1460

Program Number 1401-SM-063

This publication, C24-3053-3, describes the Merge 6 program, its capabilities, phases, and merge object program. The control and parameter cards by which a user can tailor the program to his specific needs are also explained.

This publication is intended for use with Form C24-1489, *Input /Output Control System (on Disk) Specifications for IBM 1401/1460.*

The user should also become familiar with the following SRL bulletins:

- *Autocoder (on Disk) Language Specifications for IBM 1401, 1440, and 1460,* Form C24-3258.
- *Autocoder (on Disk) Program Specifications and Operating Procedures for IBM 1401, 1440, and 1460,* Form C24-3259.

Other related publications and their abstracts are listed in the IBM 1401/1460 Bibliography SRL, Form A24-1495.

This edition, Form C24-3053-3, is a major revision of an earlier edition, and incorporates information from Form C24-3213-0 and Technical Newsletter N24-0217-0. This new edition obsoletes Forms C24-3053-2, C24-3213-0, and N24-0217-0.

Copies of this and other IBM publications can be obtained through IBM Branch Offices. A form is included at the back of this manual for readers' comments. If this form has been removed, address comments to: IBM Corporation, Product Publications, Dept. 245, Rochester, Minn. 55901.

Contents

Merge 6 Specifications 5

Introduction 5
- Additional Programs Required 6
- Machine Requirements 6

Merge 6 Generation 7
- Merge Prephase 7
- Object Program 7

Merge 6 Operating Procedures 14

Introduction 14
- Card Decks 14
- Update Job 14
- Library Jobs 14

Generation of Assignment and Merge Phases 15

Parameter Deck 15

Prephase Diagnostics 16

Merge Object Program Specifications 18

Introduction 18

Machine Requirements 18

Record Configurations 18

Labeling Procedures 19
- Disk Label Processing 19
- Disk Processing Limits 20
- Disk Label Track 20
- Tape Label Processing 20
- RDLIN Cards 23

Object Program Features 25
- Padding (Blocked Tape Output Only) 25
- Cylinder Overflow 25
- Control Field Data 25
- Deletions by Control Data 25
- Selections or Deletions by Class 25
- Card Additions 26
- Sequence Check and Hash Total 26
- Object Program Control Cards 27

Merge Object Program Operating Procedures 29

Object Deck Preparation 29
- Disk RDLIN Cards 29
- Tape RDLIN Cards 29
- DATE Card 29

Running the Object Program 29
- Merge 6 Assignment Phase Messages 30
- Merge Phase Halts 30
- Merge Phase Messages 30

Appendix I 36

Timing Charts 36

Appendix II 38

Core Storage Requirements Charts 38

Introduction

Merge 6 for the IBM 1401/1460 Data Processing System allows the user to combine previously sorted files into one continuous file. The collating sequence (either ascending or descending) used to sort the original files must also be used in the merge operation.

Merge 6 is a generative program designed for incorporation into the 1401, 1440, 1460 Autocoder library. By selecting the parameter cards that define his particular merging application, the user can produce the desired object program with Autocoder and the macro generator. The parameter cards determine the content of the generated object program.

When generated, the object program is a general program. It can be tailored to specific needs, however, by inserting control cards. Input (except card additions) and output to the object program are controlled by IBM 1401/1460 Input/Output Control System (IOCS) on Disk. Hence, the user should become familiar with both the IBM 1401, 1440, 1460 Autocoder (on Disk), and the 1401/1460 Disk IOCS.

Merge 6 has two major phases, prephase and generation phase. Merge 6 object program consists of two minor phases, assignment phase and the merging phase. Figure 1 is a schematic diagram of the Merge 6 program.

Before the prephase, the user describes in parameter cards the operations he wants the object program to perform. During the prephase, these parameter cards are read and analyzed, and a number of diagnostic functions is performed to make certain the necessary parameter cards have been included in their correct form. The prephase translates the information from the parameter cards into information that can be used by the macro generator. The macro generator uses this information to set the permanent switches that select the Merge 6 library routines used to form the object program. Thus, a program containing the operations specified by the parameter cards is made ready during the generation phase.

During the generation phase, Merge 6 uses the 1401, 1440, 1460 Autocoder (on Disk) macro generator to select and translate the specified program characteristics into an object program in machine language.

The object program produced by Merge 6 performs the following functions:

1. Merges blocked or unblocked fixed-length records in move mode.
2. Merges either numeric or alphameric records.
3. Merges according to control information contained in up to ten fields of each record.
4. Utilizes IOCS described in the 1401, 1460 disk IOCS bulletin with accompanying error procedures.
5. Provides the ability to perform deletions by class or by control-data field information (deletions by class and by control-data field information cannot be generated simultaneously), and additions from cards if specified by a selected set of parameter cards.
6. Provides the ability to merge by selection.
7. Allows program modification.
8. Allows input files (on disk pack, magnetic tape or in cards) to be in either ascending or descending order.

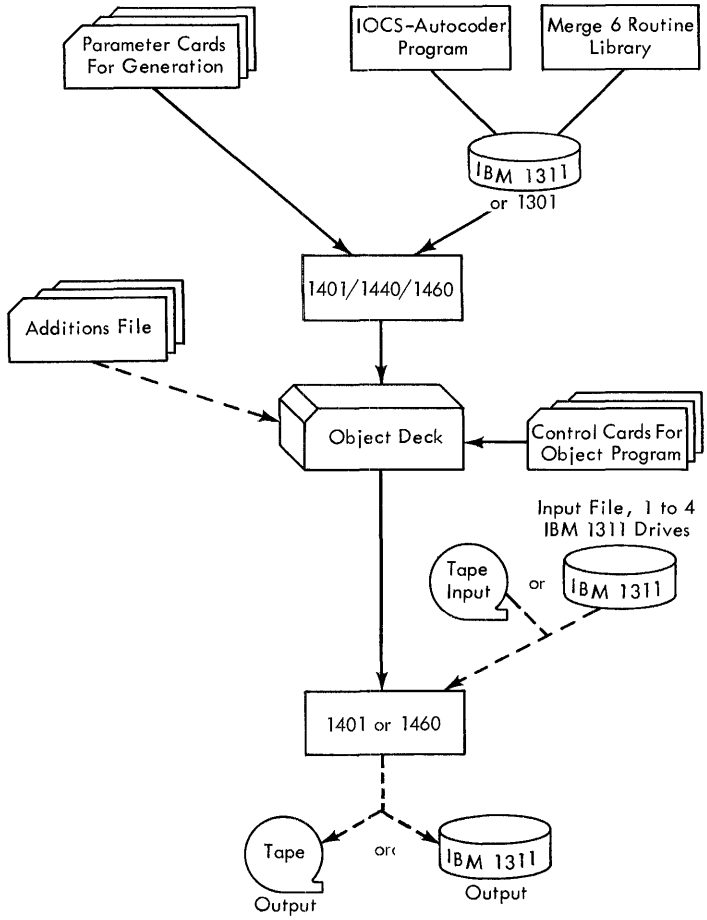


Figure 1. Merge 6 General Operation

Columns	Indicates	Punch (Meaning)	Assumptions if the Columns Are Left Blank
16-18	Mnemonic operation code	CTL	
21	Object machine size	1 (4K) 2 (8K) 3 (12K) 4 (16K)	4K
22	Modify address feature available	1 (Yes)	No, if the object machine has 4K; Yes, if the object machine has 8, 12, or 16K.
23	This column must be left blank		
24	Multiply-Divide feature available	1 (yes)	No
25	Object machine	0 (1401) 6 (1460)	Processor machine
26	Punch device	P (1402)	P if the object machine is a 1401 or 1460
27	Read device	P (1402)	P if the object machine is a 1401 or 1460
28	Print device	P (1403)	P if the object machine is a 1401 or 1460
29	Disk device	1 (1311)	1311
30	Source Statement Diagnostics	1 (yes) N (no)	Yes
31	Label Table or Cross Reference Listing	L (label table) N (neither)	Cross reference listing
32-36	Read-in area	00001	00001
37-41	Loader location	Assignment Phase - 00334 Merge Phase - 00087	

Figure 2. CTL Card Format

9. Allows output files (on disk pack or magnetic tape) to be in either ascending or descending order.
10. Provides full labeling capabilities as in 1401/1460 disk IOCS.
11. Performs a sequence-check of the output records if specified by the user.
12. Counts the records merged during out-put.

Additional Programs Required

In addition to the Merge 6 program, the following programs are required for generation of a merge object program:

- 1401, 1440, 1460 Autocoder (on Disk), Program Number 1401-AU-008
- 1401, 1460 Input/Output Control System (on Disk), Version 2, Program Number 1401-IO-068

Machine Requirements

Any of the following system configurations can be used for generating the object program:

1. IBM 1401 Data Processing System with:
   - A minimum of 4000 positions of core storage
   - High-Low-Equal Compare feature
   - IBM 1402 Card Read-Punch
   - IBM 1403 Printer, Model 2 or 3
   - IBM 1311 Disk Storage Drive
2. IBM 1440 Data Processing System with:
   - A minimum of 4000 positions of core storage
   - IBM 1442 Card Reader

- IBM 1443 Printer
- IBM 1301 Disk Storage or IBM 1311 Disk Storage Drive

3. IBM 1460 Data Processing System with:
   - A minimum of 8000 positions of core storage
   - IBM 1402 Card Read-Punch
   - IBM 1403 Printer, Model 2 or 3
   - IBM 1301 Disk Storage or IBM 1311 Disk Storage Drive

Merge 6 Generation

A Merge 6 generation is controlled by the 1401, 1440, 1460 Autocoder system. Only one phase of the merge object program can be generated per job. The input for the generation is a deck of Autocoder control cards, parameter cards for the phase to be generated, and any user routines that are to be incorporated in the phase. All user routines must be written in the Autocoder language.

The merge prephase, which is part of the Autocoder system, checks the parameter cards and translates the information into a format recognized by the Autocoder macro generator. The macro generator selects the appropriate library routines for the phase being generated and modifies the routines according to parameters supplied by the user. Although the Merge 6 library routines are selected by the macro generator, they cannot be used as subroutines in other Autocoder programs because the merge prephase sets permanent switches that are used by the merge library routines.

When the phase is being generated, user routines, if included in the job deck, are assembled. Thus, the result of each generation job is an object program that contains one merge phase and any user routines that are to be executed during that phase of the merge job.

If the general characteristics of the users merge application(s) change, only those phases affected by the change need to be regenerated.

Merge Prephase

The merge prephase must be incorporated in the preprocessor portion of the 1401-1440-1460 Autocoder system. The prephase is transferred from cards to the disk unit (SYSTEM file) by the Autocoder update job. Because a fixed area on the SYSTEM file is reserved for the merge prephase, the update job need be performed only once.

All the control and data cards required for the update job are included in the Merge 6 program deck. The first card in the prephase section of the deck is an AUTOCODER RUN; the last card is an END card.

Object Program

Only the assignment library routine and the single merge library routine MERGE (63M41) are needed to generate the assignment-phase object deck. No IOCS library routines are used.

Three IOCS library routines are required for generation of the merge phase object deck: DTFFI, DTFTP, and IOCS. The following merge library routines are also required: MERGA, MERGB, MERGC, MERGD, MERGE, MERGF, and MERGG.

The system pack can accommodate all the IOCS library routines, although only the three mentioned need be incorporated for a merge phase generation.

Job Deck

A job deck is required for each phase generation. The deck must contain Autocoder control cards and parameter cards, and can contain user routines that are written in the Autocoder language.

Autocoder Control Cards

Autocoder cards required for the generation of each phase are RUN, CTL, MERGE, and END. A JOB card may be inserted ahead of the CTL card.

RUN *Card* is punched as follows:

Columns	*Contents*
6-14	AUTOCODER
16-18	RUN
21-24	THRU
26-31	OUTPUT

NOTE: AUTOCODER RUN THRU OUTPUT is the conventional assembly that yields an object program deck in the condensed-loader format.

JOB *Card* is punched as follows:

Columns	*Contents*
16-18	JOB
21-72	*Any identification*
76-80	*Any identification*

CTL *Card* is shown in Figure 2.

MERGE *Card* is punched as follows:

Columns	*Contents*
16-20	MERGE

This card causes the merge prephase to be read into core storage from the SYSTEM file.

END *Card* is punched as follows:

Columns	*Contents*
16-18	END

Parameter Cards for Merge 6

Parameter cards containing specifications punched by the user are required by Merge 6 to produce the object program. Figure 3 gives a summary of the parameter cards that can be used. If a particular card is indicated as being mandatory for a particular phase, the card must be used.

NUMBER	LABEL	OPER-ATION	OPERANDS	MANDATORY FOR ASSIGNMENT PHASE	MANDATORY FOR MERGE PHASE	REMARKS
1		MERGE		Yes	Yes	This is the first card read, and indicates to the Autocoder that the cards following are to generate the object program.
2	MERGEORDER		X	No	Yes	The user must specify the order of merge that will be generated. X denotes the order of merge (1, 2, 3, or 4).
3	PHASE		X	Yes	Yes	The user must specify which phase is to be generated (A for assignment phase and M for merging phase).
4	NOCTLFLDS		XX	No	Yes	The user must specify the maximum number of control fields to be compared. XX denotes the number of control fields (01-10).
5	SYSTEMSPEC		DIRECT, CONPRINT, CYLOFLOW	No	No	If the user has the direct-seek feature, he should specify DIRECT. If he has a console printer he can specify CONPRINT. If cylinder overflow occurs on input or output, he must specify CYLOFLOW.
6	EXITS		EXIT1, EXIT2, EXIT3, EXIT4, EXIT5, EXIT6, EXIT7, EXIT8	No	No	The user can specify the exits to be used during the execution of the object program. EXITS 1, 2, 3, or 4 can be used after a record block is read into core storage from the corresponding input file (i.e., EXIT 1 can be used after a record block is read from input 1.). EXIT 5 can be used in the merging phase before the deletions routine occurs. EXIT 6 can be used in the merging phase before a block of records is written in disk storage. EXIT 7 can be used to call in the next program after messages have been printed at the end-of-job. EXIT 8 can be used at the beginning of the merging phase. Note: IF EXIT 5 is specified the SELECTDLET parameter card must be specified.
7	SELECTDLET		SELDELCLASS, CONTROLDATA	No	No	At object time, if the user desires selections or deletions by class, he can specify SELDELCLASS. At object time, if he desires deletions by control information, he can specify CONTROLDATA. If CONTROLDATA is specified as an operand, the user must not include the ADDITIONS card.
8	CHECKS		SEQHASH	No	No	The user can specify that routines to compute hash totals and to sequence-check the output file are included in the object program.
9	USERAREA		YES	No	No	The user can specify that a user written routine is to be assembled during the generation of the object program.
10	DSKLABLIN		ALL	No	No	The user can specify that he desires the object program to contain routines for checking input disk labels.
11	DSKLABLOUT		ALL	No	No	The user can specify that he desires the object program to contain routines for checking output disk labels.
12	TAPLABLIN		STANDARD, NONSTANDARD, TM, ALL, A, B, or C	No	No	The user must specify the kind of labels, whether standard or nonstandard, that he has on his input files and whether a tape mark follows each header label. If the input files contain standard labels, the user must further specify that they are type A (120-character label), or type B (80-character label), or type C (84-character label).

Figure 3. Merge 6 Parameter Cards (Part 1 of 2)

NUMBER	LABEL	OPER-ATION	OPERANDS	MANDATORY FOR ASSIGNMENT PHASE	MANDATORY FOR MERGE PHASE	REMARKS
13	TAPLABLOUT		STANDARD, NONSTANDARD, A, B, or C	No	No	The user can specify the kind of labels that are to be generated and written on each output tape if the tape has expired. If the output tapes already have standard header labels, normal IOCS type checking will be performed on each label.
14	INPUTMEDIA		DISK, TAPE	No	Yes	The user must specify the kind of input medium.
15	OUTPTMEDIA		DISK, TAPE	No	Yes	The user must specify the kind of output medium.
16	TAPE		UNLOADIN, UNLOADOUT, DUMP, SCAN	No	No	The user can specify whether he desires any of the following tape rewind options: 1. rewind and unload each input tape when an end-of-file condition is recognized; 2. rewind and unload each output tape when either an end-of-reel indicator is sensed or the last record of the file is written. The user must specify the procedure to be followed for unreadable records by including either the operand DUMP or the operand SCAN. If DUMP is specified, unreadable records are written on tape. If SCAN is specified, the user can attempt to correct the unreadable block.
17	ADDITIONS		CARD	No	No	The user can specify that he desires additions from cards during the execution of the object program.
18		END		Yes	Yes	This card is the Autocoder end card and signals that the last parameter card has been read, or that the last card of the user routine at assembly time has been read.

Figure 3. Merge 6 Parameter Cards (Part 2 of 2)

All parameter cards used to generate Merge 6 object programs must be punched in a specific format. There is no specific order in which the parameter cards must appear in the deck, but the parameter deck must be preceded by a card with MERGE punched into columns 16 through 20 (Figure 4).

The parameter cards must be punched as follows:

Columns	*Contents*
1- 5	These columns should be left blank.
6-15	These columns must contain the parameter card label. The label must begin in column 6 and may extend through column 15. If the label does not contain 10 characters, the unused columns must be left blank.
16-20	These columns must be left blank.
21-80	The allowable operands in each card must begin in column 21. A comma must separate each entry in the operand field of cards with multiple operands, but the last operand on each card must not be followed by a comma. Blank columns must not occur within operands or between commas, but blanks must follow the last operand punched into each card.

If the user supplements the Merge 6 program with his own subroutine and the subroutine is assembled along with the object program, a LTORG* card (LTORG* punched into columns 16 through 21) must be inserted directly behind the last parameter card (see Figure 4). The LTORG* card causes the Autocoder processor to incorporate the user's subroutine into the object program. The user's symbolic subroutine cards follow the LTORG* card, and an END card is placed behind the subroutine. The END card indicates that the last parameter card has been read. If no user subroutines are to be incorporated into the object program, the END card follows directly behind the last parameter card.

MERGEORDER *Card* indicates the order of merge (i.e., the number of files to be used as input to the object program) to be generated. Column 21 must contain a 1 for a 1-way merge, a 2 for a 2-way merge, a 3 for a 3-way merge, or a 4 for a 4-way merge. The order of merge must not exceed 4. Card additions are not considered when specifying the order of merge.

PHASE *Card* indicates the phase of the object program to be generated. Only one phase can be generated

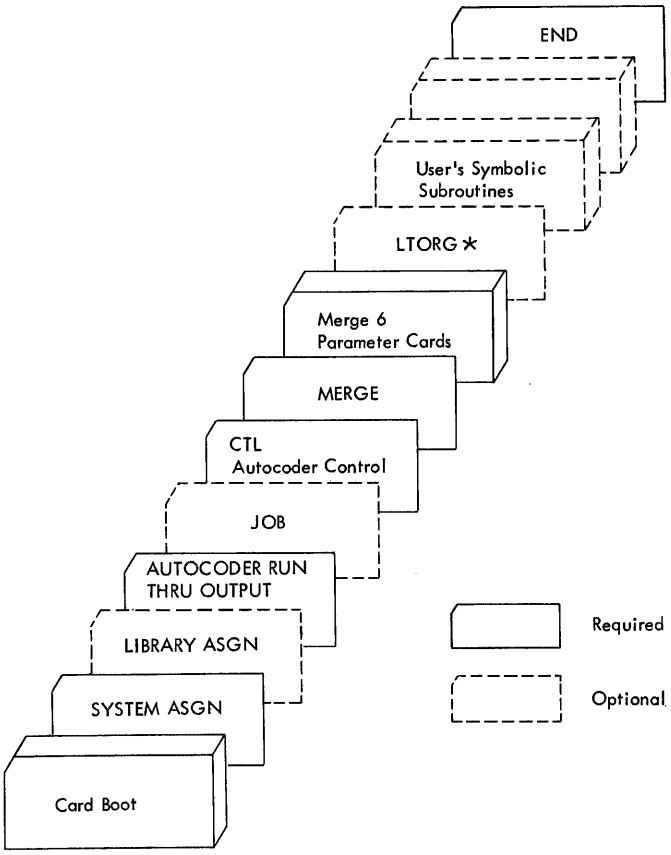


Figure 4. Source Deck for Generating Object Decks

at a time. Therefore, an A punched into column 21 indicates generation of the assignment phase; an M punched into column 21 indicates generation of the merge phase.

NOCTLFLDS *Card* indicates the number of control-data fields to be compared within each record. If the records contain only one control-data field or if the records are to be compared on only one field, punch 01 into columns 21 and 22. If the number of control-data fields to be compared is greater than one, punch the number into columns 21 and 22. A maximum of ten control-data fields can be compared at object time. Any value greater than 01 will cause a multiple control-data field compare routine to be generated.

SYSTEMSPEC *Card* can cause object time messages to be printed by the console printer (if the printer is attached to the system used to execute the object program) by punching CONPRINT in the operand field.

If the object-time 1311 has the direct-seek feature, the user can punch DIRECT in the operand field. If disk cylinder overflow can occur during execution of the object program, the user must punch CYLOFLOW in the operand field. See *Cylinder Overflow* in this publication.

EXITS *Card* provides the user with a simple means of incorporating as many as eight user exits into the object program. By punching EXIT*x* (*x* is the exit number) into the operand field of this card, the user can generate entries into his own routines. For example, a particular merge operation may require the use of two user subroutines. The operand entry would be: EXIT1, EXIT2 for this parameter card, and the Merge 6 program would generate entry into these two routines. If any exits (EXIT1-EXIT8) are specified, but associated routines are not assembled at generation time, the unused labels will be undefined. This card has no bearing on the generation of IOCS exits for processing nonstandard labels. Additional information about the use of these exits is given later in this section.

USERAREA *Card* indicates that a user routine is to be assembled with and incorporated into the specified object program. All user routines will be assembled and incorporated into the merge phase, and access to them can be gained only through the established exits (see EXITS parameter card). If any routines are to be assembled, punch YES in the operand field of this card. This card has no bearing on the generation of routines for processing nonstandard labels. For additional information, refer to the section on *User Programming* in this publication.

NOTE: If a parameter card is detected by the prephase, the user routines will not be assembled.

SELECTDLET *Card* has two allowable operands, but only one can be specified for any generation run. If a specific class of records, including card additions, is to be selected or deleted from an input file, punch SELDELCLASS. If records, excluding card additions, are to be deleted from input files on the basis of control data in cards, punch CONTROLDATA.

CHECKS *Card* has only one allowable operand. If hash totals are to be accumulated from records merged into the output file, and/or a sequence-check routine is to be performed during the object run, punch SEQHASH in the operand field of this card. By including this card with the proper operand into the prephase, the hash-total and sequence-check routines will be generated.

DSKLABLIN *Card* must be included with ALL punched into the operand field if the header labels of input files contained in disk storage are to be checked. The inclusion of this card will cause the prephase

to generate routines capable of checking labels in full or in part (file-identification field only). The generated label-checking routine will check fields 1 through 7 of the standard disk header label to make certain the correct file is about to be processed.

DSKLABLOUT *Card,* if included, with ALL punched into the operand field, causes a routine for checking header labels on disk output files to be generated. This label routine will check, at object time, all the standard disk header labels on the label track against information in a RDLIN card to be sure the correct pack is about to be written on, and to be sure no other files are within the limits specified in the RDLIN card. If the new file area is available, a header label will be created from the information in a RDLIN card and a date card for the output file or file section.

TAPLABLIN *Card* indicates the type of label on all tape input files, if labels are present, and determines whether routines to check these labels will be generated. The following is a list of allowable operands:

1. STANDARD. Used when tape input files will contain standard labels.
2. NONSTANDARD. Used when tape input files will contain nonstandard labels. IOCS exits, enabling the user to link his label checking routines to the program, will be generated.
3. TM. Used when a tape mark follows the input header labels. The operand is not required for Type A standard labels.
4. A, B, or C. Used when STANDARD is specified. One of these operands must be included to indicate the type standard label on the input files: A = 120-character standard label, B = 80-character standard label, C = 84-character standard label.
5. ALL. Used when standard labels are to be checked in full or in part (file identification field only).

NOTE: Whenever standard tape labels are specified, all input files *must* contain labels at object time.

TAPLABLOUT *Card* indicates the type of label to be created for tape output files. The following is a list of allowable operands:

1. STANDARD. Used when standard labels are to be created from information punched into a tape RDLIN card for output files.
2. NONSTANDARD. Used when nonstandard labels are to be written on the output tape(s) by a user routine. IOCS generated exits will enable the user to link his label routines to the object program.
3. A, B, or C. Used when the STANDARD operand is specified. One of these must be punched to indicate the type of standard label to be written on the output tape(s). A tape mark is written after a Type A label only.

NOTE: When standard labels are specified, Merge 6 will generate a routine to check the retention period of the header label on the output tape(s) at object time.

INPUTMEDIA *Card* specifies the input medium (magnetic tape or disk) to be used at object time. Only one medium may be specified by the user per generation, because the program is not designed to process mixed input or output media. Punch DISK into the operand field of this card if the input medium is to be disk. Punch TAPE into the operand field if magnetic tape is to be used as the input medium. Although card additions can be processed by the object program, they are not considered an input medium and must not be indicated in this card.

OUTPTMEDIA *Card* indicates the output medium at object time. The two allowable operands are DISK and TAPE; however, only one may be specified per generation run. Punch DISK in the operand field if the output medium is to be disk; punch TAPE if the output medium is to be magnetic tape.

TAPE *Card* indicates the desired tape rewind options, and the procedure to be followed if unreadable or wrong-length tape blocks are encountered during execution of the merge phase. The following is a list of allowable operands:

1. UNLOADIN. Use this operand if input tapes are to be rewound and unloaded when an end-of-file condition occurs.
2. UNLOADOUT. Use this operand if the output tape(s) is to be rewound and unloaded after the tape(s) is closed.
3. DUMP. Use this operand if unreadable input tape blocks and wrong-length input tape blocks are to be written on tape unit 6.
4. SCAN. Use this operand only if unreadable input tape records are to be corrected through use of the scan option. (See Systems Reference Library publication *Input/Output Control System (on Disk) for IBM 1401/1460: Specifications,* Form C24-1489, for the procedure to follow.)

NOTE: If neither DUMP nor SCAN is punched when tape input is specified, the SCAN option is assumed.

ADDITIONS *Card* with CARD punched into the operand field, causes Merge 6 to generate a routine for accepting input from cards. The additions routine, which will process cards in the 70-character or 80-character format, will be incorporated into the object program at assembly time.

Programming Exits

A maximum of eight exits can be incorporated into the object program at various points to facilitate the inclusion of user-written routines. All user routines, except those associated with nonstandard tape labels, must be placed in upper core storage. The amount of core storage available for EXITS 1 through 8 depends upon the following:

1. Size of the object machine.
2. Size of the object program.
3. Amount of core storage required for input/output areas, including the storage area set aside for processing card additions.

The user can calculate the amount of core storage required for any object program by using the statistics cited in *Appendix II.* The amount of required storage can then be deducted from the core capacity of the object machine to determine the low limit (lowest core position) of the storage area available for the user's routines. The high limit of the available area is the last core position in the object machine.

As an example, assume that the object program will be executed on an 8k system, the user's object program requires 2000 core-storage positions, and the input/output areas (including the area for card additions) utilize another 4000 core-storage positions. The program, therefore, requires 6000 storage positions. The remaining core-storage positions, 6001 to 7999, are available for the necessary exits. The user may place his routines anywhere within the limits of this 2000-position area.

NOTE: Only those exits specified in the EXITS parameter card at generation time will be available in the object program.

Exit1 through Exit4 are available immediately after a block of input records has been read into core storage and the associated error checks have been performed. The availability of these exits depends upon the order of merge specified at object time. For example, if the user has generated a 2-way merge, exits 1 and 2 will be available at object time; if he has generated a 3-way merge, exits 1, 2, and 3 will be available at object time.

If the user wishes to perform any kind of manipulation on the input records after they have been read into core storage, he need not know the exact storage locations of the records. The input records are labeled to facilitate user programming. The address of the core locations preceding the high-order characters (first character in the record block) of the record blocks read from input files 1, 2, 3, and 4 will be contained in the 3-character DCW fields labeled A1, A2, A3, and A4, respectively. The actual address of the first character in the respective input blocks is, therefore, the contents of A1 increased by 1, the contents of A2 increased by 1, the contents of A3 increased by 1, and the contents of A4 increased by 1. If during the execution of a user routine the user wishes, for example, to modify the fifth character in a block of records read into storage from input file 3, he simply addresses his modification to the contents of A3 increased by 5.

The core-storage location of the last character in the last record block read into storage from input files 1, 2, 3, and 4 will be contained in the 3-character DCW field labeled A1END, A2END, A3END, and A4END, respectively.

The first instruction of a user routine entered through one of these four exits must be labeled *EXITx* (*x* is the exit number). The return instruction of each routine must be a branch to the common re-entry point labeled ENTRY1.

Exit5 is available before a deletions-by-class routine or a deletions-by-control-data routine occurs and before the record being processed is merged into the output block. Any manipulation of the records that might be deleted can be made at this time.

For example a user might wish to write a subroutine that will test for a particular condition in the records; and if the condition is present, the record might be tagged with a character that will cause it to be deleted. It is not necessary for the user to know the exact location of the record being processed. He has access to the record through use of its label, MOVE + 10. This field, labeled MOVE + 10, contains the core-storage address of the low-order character (the last character) of the record.

When a deletions by control data routine has been generated, CARD is the label associated with the high-order position of the deletions-by-control-data card presently in core.

The first instruction of the user routine associated with exit 5 must be labeled EXIT5; the return instruction must be a branch to the re-entry point labeled ENTRY5.

Exit6 is available immediately before a record block is to be written out on magnetic tape or disk storage.

Output records can be modified with a routine made available through this exit. The exact core-storage location of the record block can be determined by the user. The 3-character DCW field labeled EOFOUT contains the core-storage address of the core location preceding the high-order position (first character) of the record block. The exact address of the low-order position (last character) of the output block will be the contents of the 3-character DCW field labeled OUTEND.

If the user chooses to use disk storage as his output medium, the program will automatically write a 1EORb trailer label in the last record block to be written in a file section, and a 1EOFb trailer label as the last record to be written in a file area.

A 1EORb label is always the last record in the last record block in an output file section. The presence of a label in a multisection file can be tested by the user, but he should compare each character in the label individually, because each character can contain a word-mark. The 3-character DCW field labeled NXTOUT will contain the storage address of the low-order character (the blank) in the trailer label.

A 1EOFb trailer label always follows immediately behind the last logical record.

Before the 1EOFb trailer label can be written, the contents of the 1-character DCW fields labeled FILCNT (containing a count of the input files that have been depleted) and OM (containing a number that corresponds to the order of merge) must be equal. The exact address of the low-order character (the blank) in the label will appear in the field labeled EOFOUT.

The first instruction of the user routine associated with exit 6 must be labeled EXIT6; the return instruction must be a branch to the re-entry point labeled ENTRY6.

Exit7 is available after end-of-job messages have been printed out at the conclusion of the merge phase. It provides the user with easy access to another program. The first instruction of the additional program must be labeled EXIT7. There is no provision for re-entry into the Merge 6 program.

Exit8 is available immediately before the first record block is read into core storage. Exit 8, therefore, can be used only once during each object run. The user might use this exit to initialize other exits.

The user must label the first instruction of his routine EXIT8, and the return instruction must be a branch to ENTRY8.

User Routines

Several conditions must be met if the user desires to include his routines in the merge phase. The user routines must be punched into cards in the Autocoder symbolic language format, (see *Autocoder [on Disk] Language Specifications for IBM 1401, 1440, and 1460,* Form C24-3258) and the cards must be correctly located in the parameter card deck during generation of the merge phase. An Autocoder LTORG (punched in columns 16 through 20) with an asterisk in the operand field (column 21), must follow the last parameter card.

The user's decks and any additional LTORG* cards that might be required will be inserted between the first LTORG* card and the END card in the following order:

1. Routine for Nonstandard Labels
2. LTORG* card
3. Routines using EXIT1 through EXIT6 and EXIT8
4. LTORG* card
5. Routine using EXIT7.

Execute cards, LTORG* cards, and END cards must not appear within the user routines, and any group-marks with word-marks required by the user should be cleared before returning to the object program.

If any user routine requires use of index registers, the print area, or the punch area, the user must make provisions for storing the information already in these areas when his routine is activated. He must also provide a means of restoring this information in the affected areas before re-entering the object program. If the contents of these areas are saved by writing them on tape, any word-separator characters in the area must be individually restored.

Merge 6 Operating Procedures

Introduction

Card Decks

Three card decks for updating the Autocoder system pack are sent to the user of the Merge 6 Program. The three decks are identified by the labels punched into columns 76 through 80.

Deck	*Label*
1. Merge 6 Prephase	
First Overlay	MERGE
Second Overlay	SHORT
2. Merge 6 Assignment Phase	
ASSMT Routine	63A01
3. Merge 6 Merge Phase	
MERGA Routine	63M01
MERGB Routine	63M11
MERGC Routine	63M21
MERGD Routine	63M31
MERGE Routine	63M41
MERGF Routine	63M51
MERGG Routine	63M61

The cards in the prephase deck are numbered consecutively in columns 72 through 75. Each routine of the other decks is numbered sequentially in columns 1 through 5.

Update Job

An update job transfers the merge prephase to its fixed location on the System file. In addition to the control and data cards supplied in the program deck, the card boot, SYSTEM ASGN, and HALT cards are required for the job (Figure 5).

Figure 6 shows the arrangement of cards for an update job. An AUTOCODER RUN card is the first card in the sort-prephase section of the program deck; an END card is the last.

Library Jobs

The Merge 6 routines and 1401, 1460 disk IOCS library routines for each generation job must be in the same Autocoder library. The library routines for the phase(s) to be generated also must be in the library.

To transfer the appropriate routines to an Autocoder library a library change run must be performed. The model statements that make up a library routine are stored in the Autocoder library as follows: The model statement is compressed by eliminating

Formats

Label Field (Columns 6-15)	Operation Field (Columns 16-20)	Operand Field (Columns 21-72)	Remarks
SYSTEM	ASGN	1301 UNIT 0 1311 UNIT n	n represents unit number and can be 0, 1, 2, 3, 4. The SYSTEM ASGN card immediately follows the card boot. The card boot, supplied with the Autocoder system deck, is required for all system operations.
LIBRARY	ASGN	1301 UNIT n, START nnnnnn, END nnnnnn 1311 UNIT n, START nnnnnn, END nnnnnn	n represents the number of the unit and can be 0, 1, 2, 3, or 4. nnnnnn is a disk address; it must be a multiple of 20.
AUTOCODER	RUN		
INITIALIZE	OPTN		
LIBRARY	OPTN		
	END		
	HALT		This must be the last card in a stack of jobs.

Note: Leave blanks in the operand field as indicated above.

Figure 5. Control Card Formats

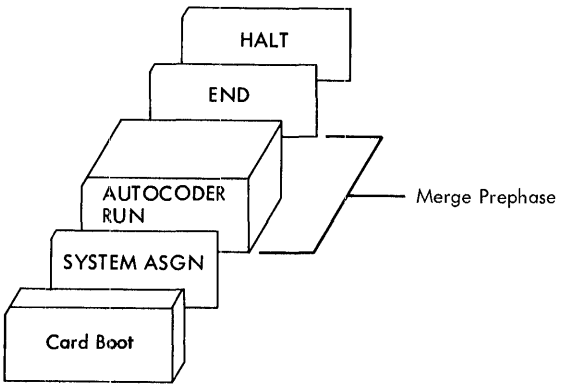


Figure 6. Update Job

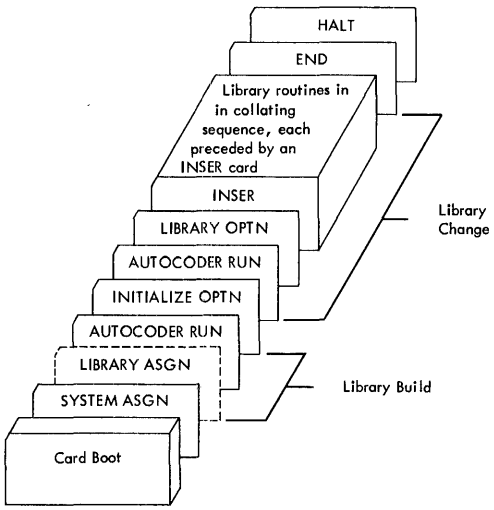


Figure 7. Preparing a New Library

high-order blanks. The statements are stored as variable-length records in 2-sector blocks. The library table requires 12 sectors.

If disk storage is available, the most efficient method of preparing the library is to define a new library and insert the required IOCS and merge routines.

If disk storage is at a minimum, delete routines from an existing library and then insert the required IOCS and merge routines. If the required IOCS routines are currently in the library, they need not be deleted and re-inserted.

New Library

The required IOCS and all merge routines will fit on a single system pack. In addition to the CARD BOOT, SYSTEM ASGN, and HALT cards required for every stack of Autocoder jobs, the following control cards are required to build the new library and to insert the routines:

Library Build:	AUTOCODER RUN
	INITIALIZE OPTN
Library Change:	AUTOCODER RUN
	LIBRARY OPTN
	INSER (an INSER card precedes each library routine in the program deck)
	END

The formats of the cards used to build and change a library are shown in Figure 5. The arrangement of the input cards is shown in Figure 7.

Modifying an Existing Library

Deletions and additions to a library can be made either concurrently or separately, but because routines must be deleted and added in collating sequence, the capacity of the library is less apt to be exceeded when all deletions are made first, then the additions. The size and content of the library and the number of routines to be inserted dictates the method. The arrangement in a library is shown in Figure 8.

Generation Assignment and Merge Phases

Either the assignment phase object deck or the merge phase object deck can be generated first. The following can be used as a guide for generating a phase:

1. Perform a 1401-1440-1460 Autocoder library run to incorporate the desired IOCS routines and the library routines of the phase to be generated. A LIBRARY ASGN card must precede the library routines if the library has been relocated and an END card must follow the last library card.
2. Insert forms into the printer and, if necessary, install an appropriately punched carriage tape.
3. Place blank cards in the punch hopper of the IBM 1402 Card Read-Punch and turn the punch on.
4. Place on the assigned drives the disk pack(s) containing the 1401-1440-1460 Autocoder processor with the routines required for the phase to be generated on the assigned drive(s).
5. Turn sense switch A on.
6. Turn the disk write switch on.
7. Turn the ILO check stop switch off.
8. Place the parameter deck in the read hopper of the IBM 1402 Card Read-Punch.
9. Follow the normal procedure for loading a card deck.

Parameter Deck

Generation of either the assignment phase or the merge phase of the object program, requires that the following cards be read in the IBM 1402 Card Read-Punch in this sequence:

1. IBM 1401-1440-1460 Autocoder card boot routine (loader-cards).
2. SYSTEM ASGN card.

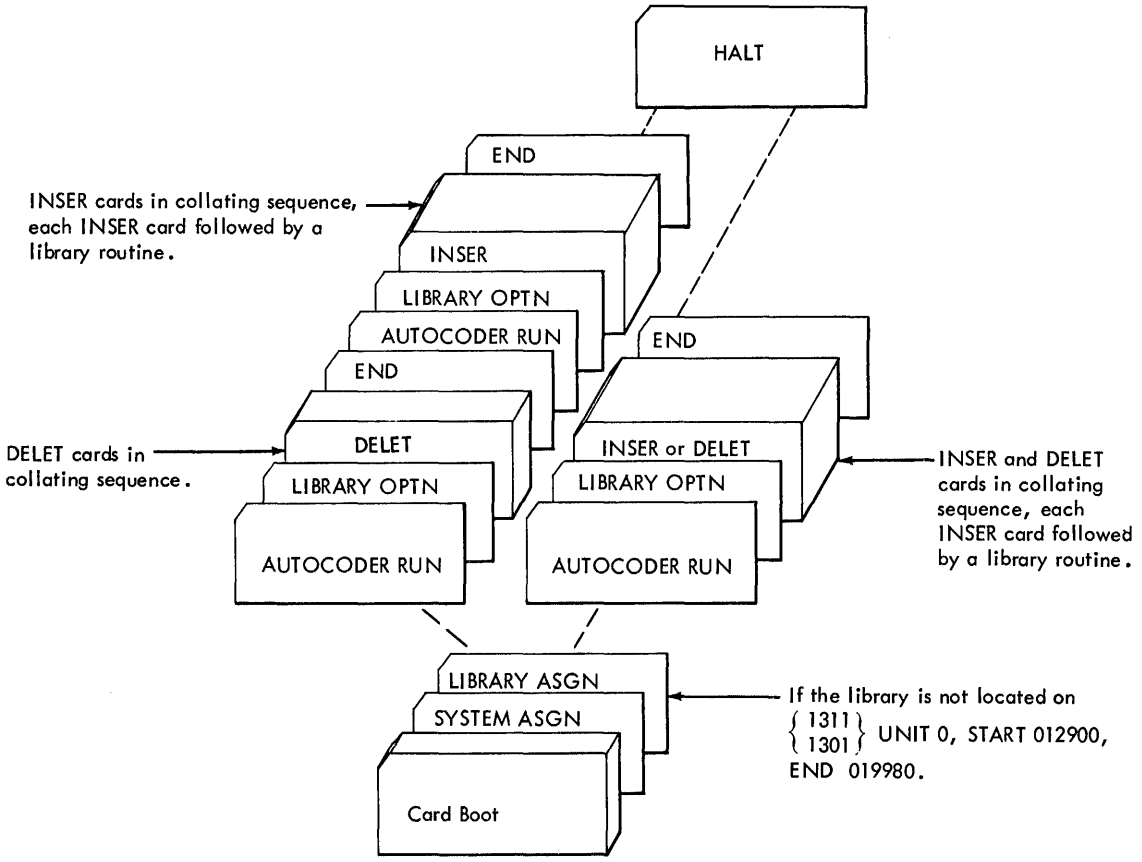


Figure 8. Modifying an Existing Library.

3. LIBRARY ASGN card (if the library has been relocated).
4. AUTOCODER RUN THRU OUTPUT card.
5. Autocoder CTL card.
6. MERGE parameter card.
7. Remaining parameter cards. (Only the PHASE card is required for the assignment phase, unless the console printer is to be used.)
8. User-written routine preceded by a LTORG* card to be assembled during the merge phase.
9. Autocoder END card.
10. Autocoder HALT card.

Specific information can be printed at the top of each page in the program listing by inserting an Autocoder JOB card before the Autocoder CTL card. The information punched into the JOB card will be listed as page headings, and the program identification, columns 76-80, will be punched into each card in the condensed object deck.

Although advanced programming features are not incorporated into the object program, the modify-address instruction is used for object machines with more than 4000 positions of core storage.

Prephase Diagnostics

The Merge 6 prephase has no associated standard diagnostic message routine; however, it is capable of incorporating diagnostics, in the form of comments, into the Autocoder listing that results from generation of the merge-phase object deck. The following procedure can be used to detect errors:

1. The contents of all parameter cards used as input to the prephase should be printed exactly as they are punched in the cards.
2. Certain diagnostics can be attached to the operand field, columns 21 through 72, of the printed parameter field. The card will be printed verbatim, and the attached comment will specify any errors with the following comments:
   a. ERROR-INCORRECT LABEL

      This comment will appear in the operand field of the printed parameter cards that have an unidentified or illegal label punched into columns 6 through 15.
   b. ERROR-REDUNDANT LABEL

      If the user inadvertently duplicates a param-

eter card label, this comment will appear in the operand field of the printout of the second card in which the label appears.

c. ERROR IN OPERAND FIELD
This comment occurs in the parameter card printout if the card contains an error in the operand field. The specific error is not indicated.

3. The following diagnostics occur as separate comments, most of which refer to specific errors associated with Merge 6 parameter cards:
   a. ERROR-STANDARD TYPE LABEL A*B*C WAS OMITTED
   This comment refers to either the TAPLABLIN or TAPLABLOUT parameter card associated with tape input or output. If STANDARD is specified as an operand, the user must specify either A, B, or C.
   b. THIS PARAMETER CARD WAS OMITTED-MANDATORY ENTRY
   This comment appears if a required parameter card is omitted. The label of the missing card is printed.
   c. SELECTDLET THIS PARAMETER CARD WAS OMITTED—EXIT5 REQUIREMENT
   This comment indicates that the user has specified Exit 5 in the EXITS card, but has failed to include the SELECTDLET card, or has failed to specify one of the deletion options in the operand of the SELECTDLET card.
   d. LTORG* ERROR—THIS CARD MUST PRECEDE A USER ROUTINE
   This comment occurs when the user has specified assembly of a user routine but has neglected to insert a LTORG* card immediately behind his last parameter card.
   e. USERAREA ERROR—THIS CARD IS INVALID, NO EXIT SPECIFIED
   This comment indicates that the user has specified assembly of a user routine but has not specified one of the exits, EXITS 1 through 8.
   f. *STANDARD* OPERAND HAS BEEN ASSUMED IN THIS CARD
   This comment appears if the user has specified A, B, or C as an operand in the TAPLABLIN and/or TAPLABLOUT parameter card(s), but has omitted STANDARD as an additional operand. The prephase will correct this condition.
   g. TAPLABLIN ERROR—THIS CARD IS INVALID FOR DISK INPUT
   h. TAPLABLOUT ERROR—THIS CARD IS INVALID FOR DISK OUTPUT
   i. ERROR—ADDITIONS PLUS DELETIONS BY CONTROLDATA IS INVALID
   This comment occurs if the user includes an ADDITIONS parameter card in a deck that contains a SELECTDLET parameter card with CONTROLDATA specified as the operand. Merge 6 cannot generate additions and deletions by control data simultaneously.
   j. TAPE ERROR—INVALID ENTRY/S IN THIS CARD
   This comment occurs if the user specifies DISK in either the INPUTMEDIA or OUTPTMEDIA card and has included an inappropriate operand in the TAPE card. DUMP, for example, would be an inappropriate operand for the TAPE card if the user has specified disk input.
   k. TAPE ERROR**DISK I/O**THIS CARD IS INVALID
   This comment indicates that disk input and output has been specified and a TAPE parameter card has also been included.
   l. UNREADABLE RECORD PROCEDURE NOT SPECIFIED —SCAN OPTION ASSUMED
   This comment occurs if the user specifies tape input but fails to specify his unreadable record procedure (either SCAN or DUMP) in the TAPE parameter card.
   m. DSKLABLIN ERROR—THIS CARD IS INVALID FOR TAPE INPUT
   n. DSKLABLOUT ERROR—THIS CARD IS INVALID FOR TAPE OUTPUT
   o. SYSTEMSPEC ERROR—INVALID ENTRY/S IN THIS CARD
   This comment occurs if the user has specified DIRECT and/or CYLOFLOW in the SYSTEMSPEC card after he has specified tape input, tape output, and no label checking. Under these conditions, the only valid operand is CONPRINT.

The conditions that cause these diagnostics, with the exception of the items f and l, will prevent generation of the object program and user routines.

Merge Object Program Specifications

Introduction

The two phases of the Merge 6 object program, assignment and merge, each require a separate generation.

Before object time, the user inserts into the assignment phase object deck the control cards needed to make the program specific. The control cards are read and analyzed during the assignment phase. They are displayed on the printer, and their contents are examined for invalid or contradictory data. The object program is examined for the presence of all options specified in the control cards. Constants to be used by the merging phase are computed and stored. Should any errors be detected during execution of the assignment phase, they will be listed and a programmed halt will occur at the end of the phase. The end of the assignment phase is indicated by a printed message. If no errors are detected, execution of the merging phase begins.

The merging phase, using data and constants supplied by the assignment phase, merges presorted input files into a composite output file. The output file is in the same collating sequence (ascending or descending) as the input files. Completion of the merging phase is indicated by printed messages and a programmed halt.

Machine Requirements

Either of the following system configurations can be used for executing the object program:

1. IBM 1401 Data Processing System with:
   - A minimum of 4000 positions of core storage.
   - High-Low-Equal Compare feature.
   - IBM 1402 Card Read-Punch.
   - IBM 1311 Disk Storage Drive and/or IBM 7330 or IBM 729 Magnetic Tape Unit(s).
   - IBM 1403 Printer, Model 2, or IBM 1404 Printer, Model 2, or IBM 1407 Console Inquiry Station, or IBM 1447 Console Printer, Model 3.
2. IBM 1460 Data Processing System with:
   - A minimum of 8000 positions of core storage.
   - IBM 1402 Card Read-Punch.
   - IBM 1311 Disk Storage Drive and/or IBM 7330 or IBM 729 Magnetic Tape Unit.
   - IBM 1403 Printer, Model 2, or IBM 1404 Printer, Model 2, or IBM 1447 Console Printer, Models 1, 2, or 4.

If the user desires tape input or output, he may use a minimum of one and a maximum of six IBM 7330 or IBM 729 Magnetic Tape Units. If he desires disk input or output, he may use a minimum of one and a maximum of five IBM 1311 Disk Storage Drives. The number of disk storage drives and/or magnetic tape units is determined by the requirements of the user's object program.

Record Configurations

The object program accommodates fixed-length records, blocked or unblocked, in the move mode. The input records must have been stored previously in disk storage or on magnetic tape (or in cards, if additions).

A block is a group of characters handled as one unit. Blocking is the combining of two or more data records into a block. Thus, to process three sectors from disk as a unit, the block length is 300 characters. If the data record contains 80 characters, the maximum blocking factor is 3. It is to the user's advantage to organize his file so there is as little unused disk storage as possible.

In most cases, it is a good idea to process as large a block as possible. This reduces the rotational delay time if the external storage media is disk, and the start-stop time in the magnetic tape units if the external storage media is magnetic tape. The blocking factor of the input file is referred to as the input blocking factor (B_i). The blocking factor for the output file is referred to as the output blocking factor (B_o).

If the input records are blocked, the number of records per block must be constant for each job. The maximum input and/or output blocking factor for disk input and output is 80 for 4k, 240 for 8k, 400 for 12k, and 600 for 16k. The maximum input and/or output blocking factor for magnetic tape input and output is 40 for 4k, 120 for 8k, 200 for 12k, and 300 for 16k.

The maximum input and output block sizes depend on the number of positions of core storage set aside for the internal operations after the program is initialized. The object program does not make adjustments for variances in input and output block length, but takes it for granted that this information (supplied by the user) is correct.

The amount of core storage required for running the program may be calculated with the following formulas and with the information cited in *Appendix*

II at the end of this publication:

1. Magnetic tape input and magnetic tape output
   $PS + MB_iL + B_oL + 2(M + 1)$
2. Magnetic tape input and magnetic disk output
   $PS + MB_iL + 100B_oS + 2(M + 1) + 10$
3. Magnetic disk input and magnetic tape output
   $PS + 100MB_iS + 10M + B_oL + 2(M + 1)$
4. Magnetic disk input and magnetic disk output
   $PS + 100MB_iS + 100B_oS + 12(M + 1)$

PS = program size
M = order of merge
B_i = input blocking factor
B_o = output blocking factor
L = record length
B_iS = input sector blocking factor
B_oS = output sector blocking factor

NOTE: If the record-mark option is chosen, the output record length is increased by one character. If card additions is specified, the total core storage requirement will be increased by the input record length.

The minimum record length for the object program is 5 characters for disk storage and 13 characters (unblocked only) or 10 characters (blocked) for magnetic tape. The maximum record length for a system with 4000 positions of core storage is 400 characters. The maximum record lengths for systems with 8000, 12000, or 16000 positions of core storage are 1200, 2000 and 3000 characters, respectively.

The maximum blocking factor is equal to the maximum block length divided by the record length.

The input and output block lengths have the same limitations as the record lengths.

Labeling Procedures

The Merge 6 object program allows full labeling capabilities as provided in the 1401, 1460 disk IOCS program regardless of core-storage capacity. The object program through IOCS provides the capability of processing either labeled or unlabeled files. For both disk and tape input/output, labeled files are defined by RDLIN cards (see RDLIN Cards).

Disk Label Processing

Header Labels

Partial or Indent Check: Partial checking of 1311 disk header labels applies to input files* only. All files designated to have only the file-identification field of the header label checked must be limited to a single section per disk pack, because the object program has no way to check correctly any header label of a multisection file contained on a single disk pack.

Full or Complete Check: All input files designated to be checked in full should be limited to a single file section per disk pack. The object program will check fields 2 through 7 of the header label against the label information punched into the user's RDLIN card to be sure the correct file section is about to be processed.

All output files designated to have their labels checked completely should also be limited to a single file section per pack. A header label, as specified by the user in a RDLIN card and a date card (covering fields 2 through 7 and 19 through 20 of the standard 1311 disk header label), will be written in the first available sector on the label track of each on-line pack pertaining to the given file or file section.

Trailer Labels

End-of-Records (1EORb) Trailer Labels: For input files, an end-of-records trailer label is mandatory to indicate the end of each section of a multisection file. There should be only one file section contained on a disk pack. This 1EORb trailer label must be located in the first five positions of a logical record. However, the record containing the trailer label may be any record within the last block of a file section.

This requirement of an end-of-records trailer label also applies to multipack on-line processing of a given file or file area. Whenever a file area extends from one pack to another (with a contiguous addressing scheme) as defined by the area limits punched in a RDLIN card, a 1EORb trailer label must be located within the last record block that can be contained on a given disk pack. If the trailer label is not located in this last block, it will be construed as the end of a file section, and a RDLIN card will be required to define a new file section.

For output files, an end-of-records trailer label will be written when the user's upper limit address (as punched in a RDLIN card) is reached, or when the last available block that can be written on a pack is reached, if the file area extends over more than one pack (with contiguous addressing). The 1EORb trailer label, in any case, will be written in the first five positions of the last record within that last block.

*The following terms are defined as they apply to the Merge 6 program:

FILE. Any group of records (serially in ascending or descending sequence) contained in an area or areas of disk storage or in cards or in tape.

FILE SECTION. Any part of the file contained in disk storage and terminated by a 1EORb trailer label.

AREA. This is referred to as file area and is that portion of a file defined by a single set of limits. The entire area must be on-line and must have contiguous addresses throughout. It can extend from one pack to another.

End-of-File (1EOFb) Trailer Label: For input files, the object program interprets an end-of-file trailer label as indicating the end of a logical file. If, however, the user's upper limit is encountered prior to a 1EOFb or 1EORb label, the program will interpret this as the end of a logical input file. All 1EOFb trailer labels must appear in the first five positions of the last logical record of the file.

For output files, the program will write an end-of-file trailer label immediately following the last logical record that was merged into the output file. This 1EOFb label will be written in the first five positions of the record. The rest of the record and the block will be filled with the contents of the previous block.

Disk Processing Limits

Processing limits, defined by the user in columns 55-66 of a RDLIN card, are the limits within which a logical file or logical file area is contained. These limits are interpreted as follows:

1. Lower Limit (columns 55 through 60). The lower limit specified by the user must be the starting sector address that contains, in the case of input files, or is to contain, in the case of output files, the first record or block of records of a logical file or a logical file area. It is the sector address at which the object program is to begin processing a logical file or a logical file area.

2. Upper Limit (columns 61 through 66). The upper limit has a different meaning for input files than it has for output files. For input files, the upper limit means the end of a logical file, the same meaning that is applied to a 1EOFb trailer label. The user's upper limit indicates to the program the last record or block of records of a logical file. The only case in which reaching the upper limit does not signify an end-of-file condition is when a 1EORb trailer label appears within the block defined by the upper limit. Detection of the EOR condition overrides detection of the upper limit.

   The upper limit of an output file indicates the end of a logical file section or logical file area. It is the same as the end-of-records condition, as it applies to output files. When the program reaches the upper limit, it writes a 1EORb trailer label in the block of records defined by the upper limit. In all cases, with input or output files, the upper limit is interpreted as the first sector of the last record or block of records of a logical file or logical file area.

   The maximum file area available on each pack at object time for a disk input or output file is determined by the characteristics of the program generated. The following table indicates the available areas:

Generated	*Available Area*
1. Disk input and/or disk output with no disk label checking.	0X0000-0X9999*
2. Disk input and disk output with any disk label checking.**	0X0000-0X9919*
3. Disk input or disk output with disk label checking and tape input or tape output.	0X0000-0X9979*

*The upper limit specified in the RDLIN card must always be less than this figure by at least the sector blocking factor minus one ($B_iS - 1$ for input and $B_oS - 1$ for output). The value of X in each of the listed figures will be determined by the addressing range of the disk pack involved. If, for example, the addressing range is 000000-019999, and if the first set of conditions in 1 is generated, the upper limit specified in the RDLIN card will be no greater than $019999 - (B_oS - 1)$ for output.

**The three tracks immediately preceding the label track of each pack must be reserved for storing the label routines.

Disk Label Track

If the user has generated disk label checking, the object program will assume that each disk pack used in running the program contains a label track. Therefore, if the object program has disk label checking incorporated into it, all disk packs used for input and/or output must contain a label track, even if the user decides not to check disk labels at object time. If disk label checking was not generated, the disk packs used as input and/or output must not contain a label track.

Tape Label Processing

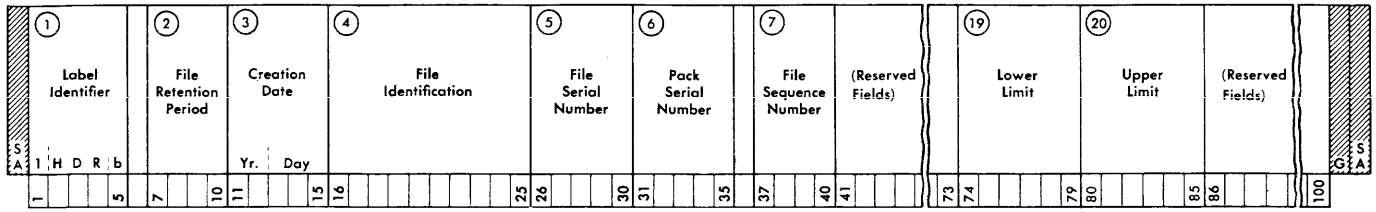
Standard Labels

Although Merge 6 permits the use of standard 1401, 1460 disk IOCS labels, Type A, B, or C, on input/output files, all labeled input files must have both header and trailer labels in the same format (See Fig. 9).

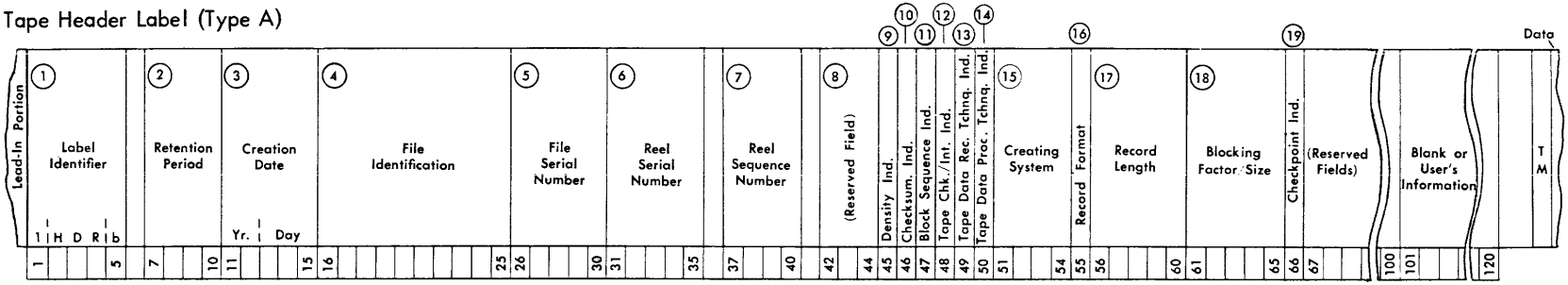
Input Files: If label checking has been generated, the user has the option at object time of specifying full checking, partial checking, or no checking at all. The type label specified determines which fields of the input header are to be checked when the user requests *full* checking. The first seven fields of Type A labels are checked; fields 9 through 19 are not checked. All fields of Types B and C labels are checked. If partial checking is specified, only the file-identification field of all three types is checked. Both full and partial checking require the use of RDLIN cards.

There is no difference between full and partial checking of input trailer labels. Input trailer labels of all three types are checked only in the label-identifier and block-count fields. However, the number of reels of input per file, specified in columns

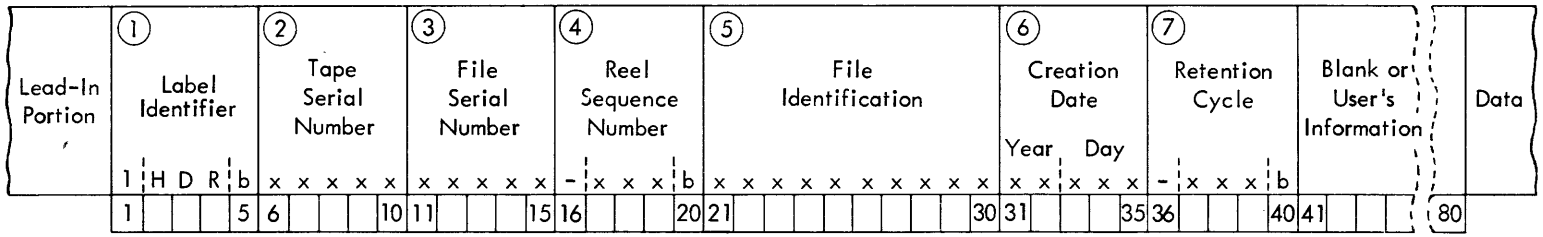
Disk Header Label



Tape Header Label (Type A)



Tape Header Label (Type B)



Tape Header Label (Type C)

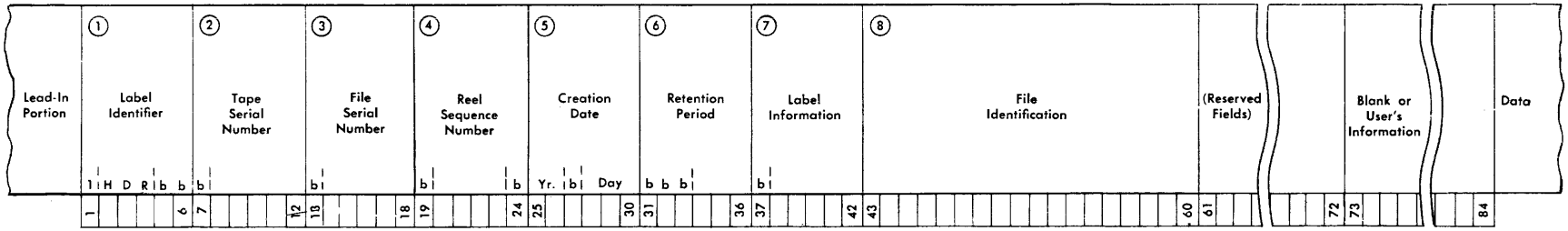


Figure 9. Schematics of Standard Disk and Tape Header Labels (Part 1 of 2)

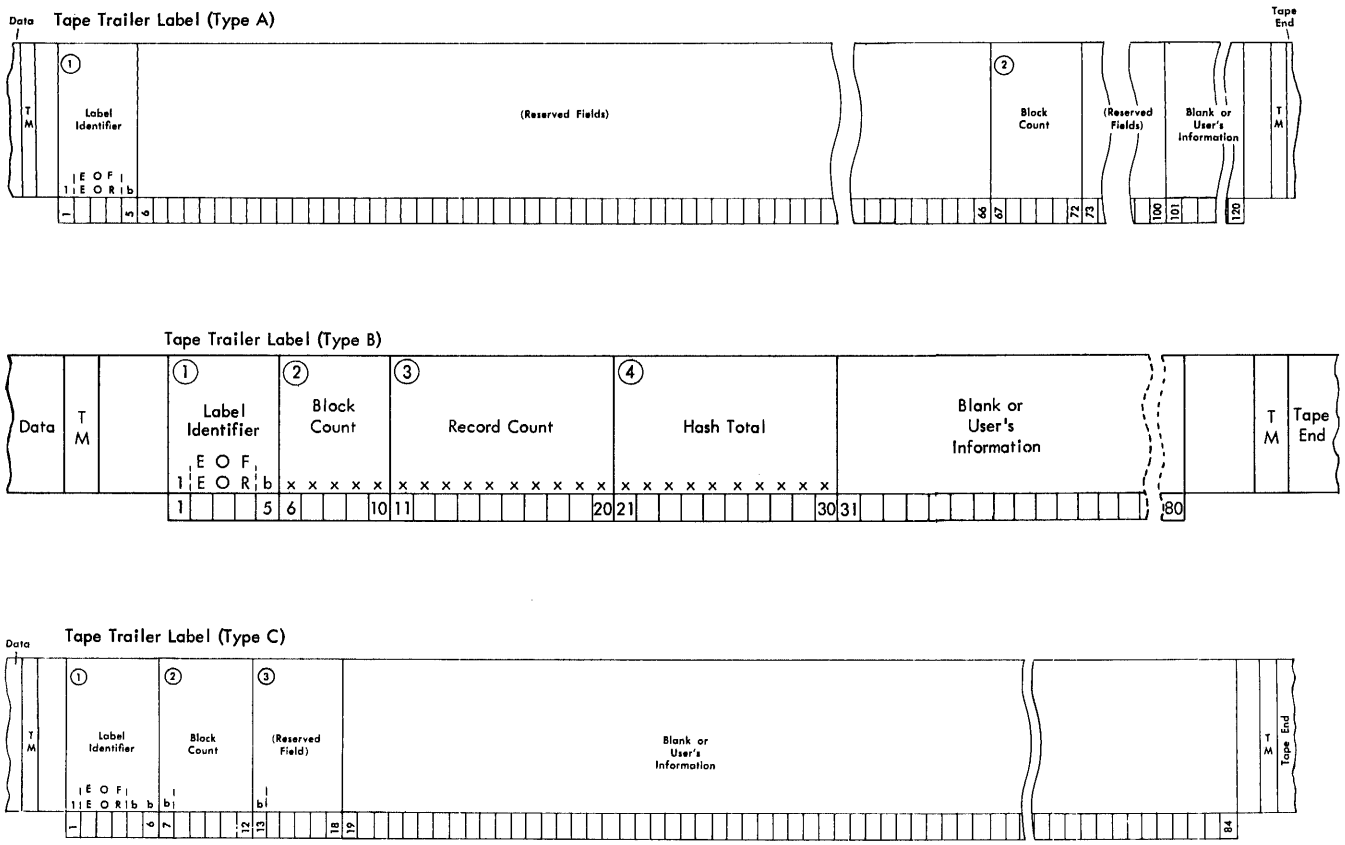


Figure 9. Schematics of Standard Tape Trailer Labels (Part 2 of 2)

37 through 40 of control card 1 (see *Control Cards*), takes precedence over detection of an end-of-file identifier.

If label-checking routines have been generated, a blank in column 35 of control card 1 indicates no label checking, and the object program simply bypasses the existing labels.

Output Files: Some special consideration must be given to the output file. If standard output labels are to be written, three requirements must be met:

1. Column 36 of control card 1 must contain an F.
2. Standard output labels must have been specified at generation time.
3. A RDLIN card and a date card must be inserted into the object deck.

The existence of a standard header on the output file is assumed, and its retention cycle is automatically checked.

The fields written in the output labels are the fields that are checked in the input labels when full checking is specified.

Nonstandard Labels

For nonstandard labels, the Merge 6 object program only provides exits; the user must provide the routines to open and close his files. The exits available, two per file, are the same as those provided by IOCS for nonstandard label processing: Exits 7 and 6 for input files; Exits 5 and 2 for the output file.

IOCS Exit 7 is used to read and check nonstandard input header labels. The first instruction of the user supplied routine must be labeled IHDRn (n denotes the input tape unit number 1, 2, 3, or 4). The return instruction must be a branch to COCSRE. The user must provide his own area for entering the header information and must program to check or bypass the input label before returning to the Merge 6 object program.

IOCS Exit 6 is used to read and check nonstandard input trailer labels. The first instruction of the user supplied routine must be labeled ITLRn. The user must provide his own area for entering the trailer information, and must program a test for an end-of-reel or end-of-file condition. On an end-of-reel condition, the user returns to the Merge 6 object program by branch-

ing to COCSRE; on an end-of-file condition he must return by branching to IEOFN.

The four-position DCW labeled REELS contains information pertaining to the number of input reels (see *Control Card 1: Columns 37-40*). REELS minus 3 pertains to the first input file.

IOCS Exit 5 is used to build and write nonstandard output header labels. The first instruction of the user supplied routine must be labeled OHDR; the return instruction must be a branch to COCSRE. The user must provide his own area for building the header label, and he must provide the programming to write it on tape.

IOCS Exit 2 is used to build and write nonstandard output trailer labels. The user must provide his own area for building the trailer label, and he must provide the programming to write it on tape. The first instruction of his routine must be labeled OTLR. The return instruction must be a branch to COCSRE.

When processing nonstandard output labels, the user can determine when the entire output file has been processed. The core-storage location referenced by the label coc5cz will contain a word-mark when the last output trailer label is ready to be processed. Should the user wish to modify this last label, he may do it at this time.

The four IOCS exits just discussed are distinct from user Exits 1 through 8 discussed elsewhere. Their use is never optional; their availability is governed solely by the specification of nonstandard labels at generation time.

The user's nonstandard label routines are originated by Merge 6 during generation and must precede any additional user routines. The user must clear any group-mark with a word-mark in his routine after processing a nonstandard label.

The area available for the nonstandard label routines depends upon the type object program generated. (Refer to *Appendix II* in this publication for a table of core-storage available for nonstandard labels.)

NOTES:
1. Columns 35 and 36 of control card 1 (see *Control Cards*) are never applicable for nonstandard labels. The user assumes full responsibility for their handling.
2. Columns 41-47 of control card 1 must always be specified for nonstandard labels because the user supplied label routines are stored on disk.

RDLIN Cards

Tape

Tape RDLIN cards are required whenever standard input or output header labels are to be checked. A tape RDLIN card must contain information pertaining to the contents of a standard header label for a given file. The RDLIN card can contain information for checking a standard header label on an input file, or for creating a standard header label on an output file. The data punched in the card must correspond to the fields of the particular standard label to be checked or created. The format of the RDLIN card associated with each of the three types of standard header labels is given in the following lists.

RDLIN card for Type-A standard header label, when fields 9-19 are not processed:

Columns	*Header Information*	*Header-Label Field Number*
16-20	RDLIN	
21-24	Retention Period	2
25-29	Creation Date	3
30-39	File Identification	4
40-44	File Serial Number	5
45-48	Reel Sequence Number	7
80	"A", if DIOCS LABELDEF specified label size "D"	

RDLIN card for Type-B standard header label:

Columns	*Header Information*	*Header-Label Field Number*
16-20	RDLIN	
21-25	File Serial Number	3
26	−(minus sign)	4
27-29	Reel Sequence Number	4
30	blank	4
31-40	File Identification	5
41-45	Creation Date	6
46	−(minus sign)	7
47-49	Retention Period	7
50	blank	7
80	"B", if DIOCS LABELDEF specifies label size "D"	

RDLIN card for Type-C standard header label:

Columns	*Header Information*	*Header-Label Field Number*
16-20	RDLIN	
21	blank	3
22-26	File Serial Number	3
27	blank	4
28-31	Reel Sequence Number	4
32	blank	4
33-34	Creation Date: Year	5
35	blank	5
36-38	Creation Date: Day	5
39-41	blank	6
42-44	Retention Period	6
45-50	Label Information: blank Density Character Coding Checksum Block Sequence Checkpoint Record	7
51-68	File Identification	8
80	"C", if DIOCS LABELDEF specifies label size "D"	

Disk

Disk RDLIN cards are required to define the processing limits and drive-pack relationship of each section of a given disk file. Whenever 1311 disk header labels are to be checked or created, the RDLIN card must

also contain information which corresponds to the specific header label of a given file or file section.

Disk RDLIN cards are punched in the following format:

Columns	*Information*
11	Drive on which Pack 0 is mounted
12	Drive on which Pack 2 is mounted
13	Drive on which Pack 4 is mounted
14	Drive on which Pack 6 is mounted
15	Drive on which Pack 8 is mounted
16-20	RDLIN
21-24	File Retention Period
25-29	Creation Date
30-39	File Identification
40-44	File Serial Number
45-49	Pack Serial Number
50	Blank
51-54	File Sequence Number
55-60	Lower Limit
61-66	Upper Limit

Multipack Online Processing applies to 1311 disk files only. Whenever a file area extends over more than one pack and has contiguous addressing (entire file area must be available to the system for processing), only one RDLIN card is necessary to process the defined file area, for both input and output files. The following conditions must be fulfilled if the object program is to process a multipack file area:

1. The processing limits punched in the RDLIN card must extend over more than one disk pack.
2. The addressing scheme between these limits must be contiguous.
3. The drive-pack table (columns 11 through 15 of the RDLIN card) must contain a drive number for every on-line pack. Example: If the processing limits range from 000000-059000, columns 11 through 13 of the RDLIN card must be punched with the respective drive numbers.
4. An end-of-file (1EOFb) trailer label (for input files only) must not appear within the processing limits.
5. For input files, an end-of-records (1EORb) trailer label must appear within the last record block that can be processed on all packs except the last pack containing the file area. For output files, the program will write a 1EORb trailer label in the last record block.

The following is an explanation of the three possible areas within which a 1EORb trailer label must be located in regard to input files. In all three cases assume that the end-of-records label is situated within the last record block on a disk pack and that the following symbols have these meanings:

X = the last sector address on a disk pack
L = the sector address which contains the trailer label record
B_iS = the input sector blocking factor

A. When no label checking has been generated, the trailer label record must appear within the area calculated with the formula:

$X \geq L \geq (X - 2\,[B_iS - 1])$

Example: Let $X = 039999$ and $B_iS = 04$:

Then: $039999 \geq L \geq (039999 - 06)$

$039999 \geq L \geq 039993$

L can be said to range from sector 039993 to sector 039999. Therefore, the user's trailer label record must be located within one of the seven sectors in that range. If a 1EORb trailer label is not encountered within that area, the program will halt and a RDLIN card will be required to define a new file area.

NOTE: In the preceding example, the last block of records cannot start lower than sector 039993, or higher than sector 039996.

B. When disk input, disk output, and label checking are generated, the trailer label record must be located within the range calculated with the formula:

$X - 80 \geq L \geq (X - 80 - 2\,[B_iS - 1])$

Example: Using the same values as in the preceding example:

Then: $039919 \geq L \geq 039919 - 06$

$039919 \geq L \geq 039913$

Therefore L has the range from sector 039913-039919. Once again, the user's trailer label record must be located within this area, or a RDLIN card will be required to define a new file area.

C. The last possible location of the 1EORb record is governed by the generation of any kind of label checking plus tape output. Then the trailer label record must appear within the range calculated with the formula:

$X - 20 \geq L \geq (X - 20 - 2\,[B_iS - 1])$

Date Card

If output label checking will be performed at object time, today's date must be loaded into storage positions 082 through 086. A card with the following contents will load the required information:

Columns	*Punch*
1-3	082
4-5	05
6	Word separator (0-5-8)
7-11	Year/Day (xx/xxx)

This card must be placed in the merge-phase object deck immediately after the 6 card loader when label checking is to be performed.

Object Program Features

Padding (Blocked Tape Output Only)

When the last data record from the input files has been merged into the output file, the object program determines whether the last used output block is full. If the last block is not full, the program generates and moves in a sufficient number of records to fill it. This process is called padding. These records will consist of all nines if the merged records are in ascending sequence, or all blanks if the merged records are in descending sequence. They will always contain a terminal record mark, and will always be equal in length to the output record length.

Cylinder Overflow

Cylinder overflow occurs when a disk block extends from one cylinder to the next. For example, if the block length is three sectors and the first block begins in sector zero of the first track, 66 complete blocks fit into the first cylinder with two sectors left over. If another block is started in these last two sectors, this block overflows into the first sector of the next cylinder. This block cannot be sought, read, or written by the use of single I/O instructions. If cylinder overflow can occur on input or output, the SYSTEMSPEC parameter card with CYLOFLOW as an operand must have been included at generation time.

Control Field Data

The collating sequence used by the Merge 6 program is the standard IBM 1401 collating sequence. (See *System Operation Reference Manual,* Form A24-3067.) The object program merges data records in ascending or descending sequence, whichever is specified in the first control card. As many as ten data-control fields, each assigned a relative value in the order of priority, may be specified. The sum of the control-data field characters may not exceed 999 in any single record, and the control-data fields must not be overlapped.

The user must specify the location of each control field in the control cards. If more than one control field is used, the user must specify which field is to compare first, which second, and so on. Although as many as ten control fields can be used, it is to the user's advantage to limit these fields whenever possible. If fewer fields are to be compared, the processing time can be reduced.

Deletions by Control Data

The user of the Merge 6 program can delete records from his output file by comparing his output records with punched cards containing control data. The control data in the cards is compared with the data in the first control field, containing up to 80 characters, of each logical record about to be merged into the output file. The record will be deleted only when the data in the control field of the record is equal to the data punched in the deletion control card.

If the first control-data field in the record is greater than 80 characters, only the first 80 characters will be compared with the control-data card. If the first control-data field is 80 characters or fewer, it will be compared with an equal number of characters from the control-data card. If the records being merged are in ascending sequence, the first record to compare higher than the deletion control card in storage will cause a new deletion control card to be read into storage by the program.

Conversely, the first low comparison, if the records are in descending sequence, will cause the program to read a new deletion control card into storage. The user must determine the number of control-data cards he needs.

The control-data information is punched starting in column one of the deletion card; it may extend to column 80 only if the length of the first control field is 80 characters or more.

All control-data deletion cards must be inserted into the object deck immediately following the card that is referenced by the label AAAEX3 in the Autocoder assembly listing that is printed out on the printer at the conclusion of the generation process.

Selections or Deletions by Class

If the user specifies the operand SELDELCLASS in the SELECTDLET parameter card at generation time, he has two class-controlled options available to him at object time: selections by class and deletions by class.

Selection of records by class is accomplished by merging into the output file only those records that contain a control character in a specific location. The user specifies both the control character and its location within the record by punching columns 21 through 25 of control card 1.

Deletions by class, the second object-time option available if SELDELCLASS has been generated, is similar to selections by class. A control character is used to determine the records to be deleted. The control character and its location within the record are punched into columns 21 through 25 of control card 1. The records containing the control character will not be merged into the output file.

The user may specify selections by class or deletions by class, but he may not specify both during one object run.

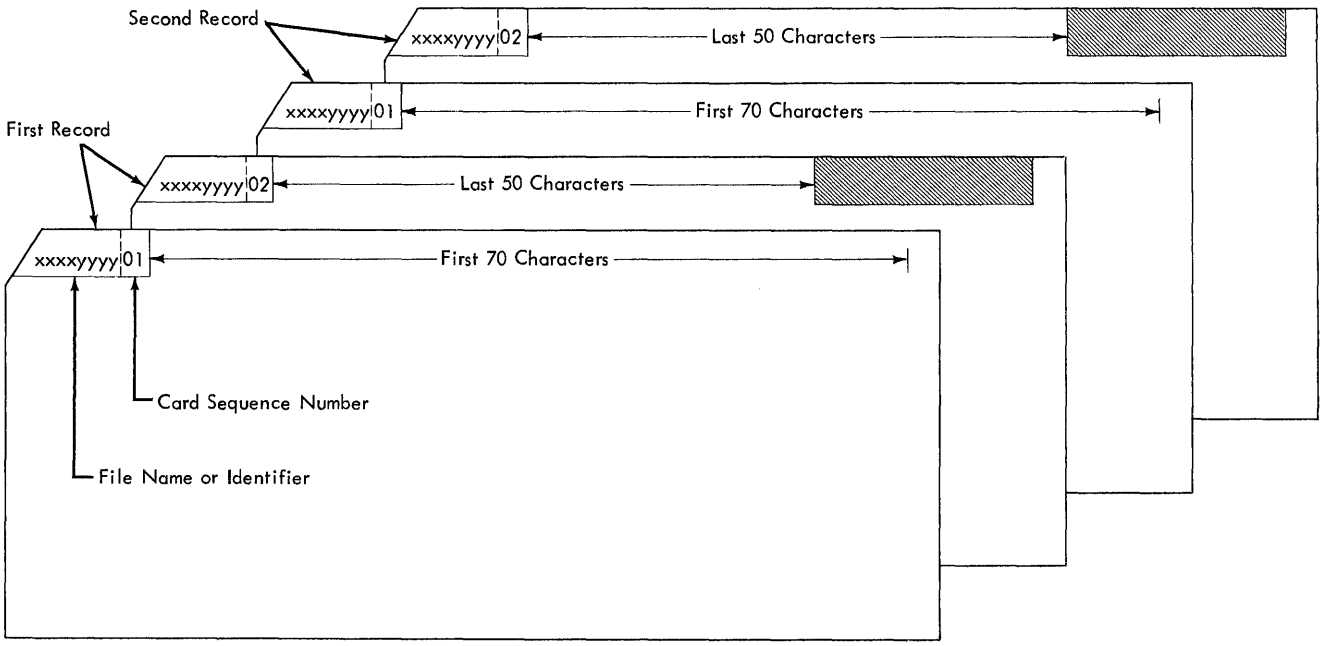


Figure 10. Example 70-Character Format

Card Additions

The Merge 6 program provides the user with the option of using punched cards as an input medium at object time. The card records are merged into the output file on the basis of the control-field information specified in control card 2 (See *Object Program Control Cards).* Each card can contain a maximum of one complete record, or it can be a part of one record; but the complete record must be equal to the input record length specified in columns 13-16 of control card 1.

The entire additions file must be inserted into the object deck immediately behind the card referenced by the label AAAEX3 in the Autocoder assembly listing.

The card additions routine can handle either of the following card record formats:

70-Character Format

By punching a 7 into column 26 of control card 1, the user indicates that each of his additions cards contains an identification field in the first ten columns. See Figure 10. This identification field is divided into two sections:

a. The first section, columns 1 to 8, contains a unique file name which must be identical in each additions card.
b. The second section, columns 9 and 10, contains a sequence number (beginning with 01) that identifies each card of a record. Used to sequence-check the number of cards per record, this section is required only when the specified record length exceeds seventy characters.

80-Character Format

By punching an 8 into column 26 of control card 1, the user indicates that each additions card contains up to eighty characters of data. This card contains no identification field.

Sequence Check and Hash Total

If the user specifies the operand SEQHASH in the CHECKS parameter card at generation time, he has two options available to him at object time: sequence check and hash total.

If the user specifies sequence checking at object time, the control-field data of the logical record about to be merged into the output file will be compared with the corresponding data of the preceding logical record. A programmed halt results if the record about to be merged is out of sequence.

If the user specifies hash total, the object program accumulates a total of up to five characters (the five high-order characters) of the first data-control field of every record merged into the output file. When the merge run is completed, the accumulated total is printed.

These options may be specified individually or simultaneously at object time.

Object Program Control Cards

This section contains a description of the control cards that are used when executing the object program.

Control Card 1

Column	Content	Indicates
1	D or T	*Input Medium.* 1. Punch a D if the input medium is disk storage. 2. Punch a T if the input medium is magnetic tape.
2	D or T	*Output Medium.* 1. Punch a D if the output medium is disk storage. 2. Punch a T if the output medium is magnetic tape. NOTE: a. If the input medium is magnetic tape, the input file(s) 1-4 must be on tape drive(s) 1-4, respectively. b. If the output medium is magnetic tape, the output file must be on tape drive 5. c. If the input/output medium is disk storage, a RDLIN card must be used to supply the file limits for each file section or area.
3-5	001-600	*Input Blocking Factor (B_I).* The number of records in each input block. Unblocked records are indicated by 001.
6-8	001-600	*Output Blocking Factor (B_O).* The number of records in each output block. Unblocked records are indicated by 001.
9-10	01-30	*Number of sectors in each input block from disk storage (B_IS).* This entry must be included when input data is from disk storage.
11-12	01-30	*Number of sectors in each output block to disk storage (B_OS).* This entry must be included when output is to disk storage.
13-16		*Input Record Length (L).*
	0005-3000	1. If input medium is disk storage, the value of L may range from a minimum of 0005 to a maximum of 3000.
	0013 or 0010 -3000	2. If input medium is magnetic tape, the value of L may range from a minimum of 0013 for unblocked records or 0010 for blocked records to a maximum of 3000 for unblocked records.
17	1-4	*Core Storage Capacity of the Object 1401/1460.* 1. Punch a 1 if the 1401 has 4000 positions of core storage. 2. Punch a 2 if the 1401/1460 has 8000 positions of core storage. 3. Punch a 3 if the 1401/1460 has 12000 positions of core storage. 4. Punch a 4 if the 1401/1460 has 16000 positions of core storage.
18	A or D	*Collating Sequence of Data.* 1. Punch an A if the data is in ascending sequence. 2. Punch a D if the data is in descending sequence.
19	1-4	*Order of Merge.* 1. Punch a 1 if the merge is a one-way merge. 2. Punch a 2 if the merge is a two-way merge. 3. Punch a 3 if the merge is a three-way merge. 4. Punch a 4 if the merge is a four-way merge. NOTE: The value punched into this column may be equal to or less than the generated order of merge.
20	b, D, or S	*Delections or Selections by Class.* 1. Leave blank if this option is not desired. 2. Punch a D if the deletions option is desired. 3. Punch an S if the selections option is desired. NOTE: If this column is punched, the next two fields on this card must contain supplementary information.
21	Any valid character	*Controlling Character for Column 20.* Punch any valid character that distinguishes the records to be selected or deleted by the option indicated in column 20.
22-25	xxxx	*Location within Record of Controlling Character for Column 20.* Punch here the position within the data record which will contain the controlling character for column 20.
26	b, 7, or 8	*Additions by Card Indicator.* 1. Leave blank if this option is not desired. 2. Punch a 7 if the card additions contain a ten-character identification field. 3. Punch an 8 if the card additions do not contain an identification field.
27	b or D	*Deletions by Control Data.* 1. Leave blank if this option is not desired. 2. Punch a D if records are to be deleted on the basis of control-data card.
28	b, S, H, or C	*Sequence Check and Hash Total* 1. Leave blank if neither sequence check nor hash total option is desired. 2. Punch an S if sequence checking is desired. 3. Punch an H if hash total checking is desired. 4. Punch a C if both sequence and hash total checking are desired.
29	b or R	*Terminal Record Mark Option.* 1. Leave blank if this option is not desired. 2. Punch an R if each output record is to be increased by the addition of a record mark as the last character.

NOTE: If the input record length, columns 13-16 of this card, contains 3000, this option may not be specified.

30-34	xxxxx	*Storage Address of User's Program.* If the user supplies routines other than those for nonstandard labels, he must indicate the starting address of his program. This applies to exits one to eight.
35	b, I, or F	*Input Label Handling Procedure.* 1. Leave blank if input labels are not checked. 2. Punch an I if the identification field is to be checked. 3. Punch an F if labels are to be checked in full.
36	b or F	*Output Label Handling Procedure.* 1. Leave blank if output labels are not to be checked. 2. Punch an F if labels are to be checked in full.
37-40	1-9	*Number of Tape Reels per Input File.* These four columns, from left to right, indicate the number of tape reels on input files 1 through 4, respectively. Each of the four columns may contain any numeric punch from one to nine. NOTE: This data must be included when column 1 of this card contains a T.
41*	0, 2, 4, 6, or 8	*Disk Drive Used to Store Label Routines.* This column indicates the number of the disk drive that contains a disk pack for storing the label-checking routines during the running of the program.
42-47*	000000-0999xx	*Low Limit Sector Address of Work Area Used to Store Label Routines.* These columns specify the starting sector address at which the program will store label checking routines. From the starting address, a total of 3 tracks (60 sectors) must be reserved by the user for the object program. This entire area must be contained within one cylinder.
48-79	blank	These columns are not used by the Merge 6 object program.
80	1	*Control Card Number.* Punch a 1.

*If the generated program contains any type of label checking and routines to process magnetic tape input and/or output, the user must specify a drive number (column 41) and a sector address at which the program can begin storing label routines. This drive must be available to the system for execution of the object program.

Control Card 2

This card contains the control field information:

Columns	*Content*	*Indicates*
1-2	01-10	The total number of control-data fields in each input record.
3-5	001-999	The sum of characters in all control-data fields.
6-9	xxxx	The location, within each record, of the high-order position of control-data field 1.
10-12	xxx	The number of characters in control-data field 1.
13-16	xxxx	The location, within each record, of the high-order position of control-data field 2.
17-19	xxx	The number of characters in control-data field 2.
20-23	xxxx	The location, within each record, of the high-order position of control-data field 3.
24-26	xxx	The number of characters in control-data field 3.
27-30	xxxx	The location, within each record, of the high-order position of control-data field 4.
31-33	xxx	The number of characters in control-data field 4.
34-37	xxxx	The location, within each record, of the high-order position of control-data field 5.
38-40	xxx	The number of characters in control-data field 5.
41-44	xxxx	The location, within each record, of the high-order position of control-data field 6.
45-47	xxx	The number of characters in control-data field 6.
48-51	xxxx	The location, within each record, of the high-order position of control-data field 7.
52-54	xxx	The number of characters in control-data field 7.
55-58	xxxx	The location, within each record, of the high-order position of control-data field 8.
59-61	xxx	The number of characters in control-data field 8.
62-65	xxxx	The location, within each record, of the high-order position of control-data field 9.
66-68	xxx	The number of characters in control-data field 9.
69-72	xxxx	The location, within each record, of the high-order position of control-data field 10.
73-75	xxx	The number of characters in control-data field 10.
76-79	blank	These columns must be left blank.
80	2	Control Card Number—punch a 2.

The maximum number of control fields allowed for the object program is ten. The total length of the control-data fields (Σ CF) can be no more than 999 characters. The control-data fields must not overlap.

Preparation of the object decks and machine operator procedures for running the merge object program are described in this section.

Object Program Deck Preparation

The cards that comprise the object program deck are punched in the 1401, 1440, 1460 Autocoder condensed card format:

Columns	*Content*
1-3	The three-character machine address of the first core-storage position to be loaded with information from the card.
4-5	The number of characters to be loaded from the card. Word-separator characters are not counted.
6-71	The characters to be loaded. A word separator character (0-5-8 punch) precedes every character that is to have a word mark in core storage.
72-75	The sequence number of the card within the object deck for the particular phase.
76-80	Object-program—deck identification, if an Autocoder JOB card with an identification punched in columns 76 through 80 was included during generation. The object-deck identification is identical to the JOB card identification. These columns are blank if a JOB card was not included.

Before the Merge 6 object program can be run, the assignment-phase object deck and the merge-phase object deck must be combined. First, remove the first card following the 6 card loader from the merge-phase object deck, and insert it immediately behind the assignment-phase card referenced in the Autocoder assembly listing by the label AAAEX0. Insert control cards 1 and 2, in ascending sequence, into the assignment-phase deck immediately behind the card just inserted. Then place the entire merge-phase object deck behind the assignment-phase object deck.

Disk RDLIN Cards

Initial RDLIN cards are inserted into the object deck in the following manner:

1. Locate the label AAAEX1 in the Autocoder assembly listing.
2. Place the disk RDLIN cards (input cards first and the RDLIN card pertaining to the output file last) immediately behind the card referenced by the specified label.

Initial input disk RDLIN cards will be referenced by the program in the order in which they are encountered (i.e., the first one will apply to the user's first file, the second to the user's second file, etc.). This initial placement is significant only when a file is composed of multiple sections.

All subsequent RDLIN cards (if any) will be inserted into the IBM 1402 read hopper upon interruption of the program by a halt (see *Merge Phase Halts:* Numbers 19-23).

NOTE: The user can avoid a nonprocess runout of the object deck by detaching all cards following the card referenced by the label AAAEX3 (as defined by the Autocoder assembly listing). These cards are not required by the object program until after the last disk RDLIN card has been read.

Tape RDLIN Cards

Tape RDLIN cards are inserted into the object deck in the following manner:

1. Locate the label AAAEX2 in the Autocoder assembly listing.
2. Insert the tape RDLIN cards, sequenced (in ascending order) according to the tape drive number on which the files are located, immediately following the card referenced by the AAAEX2 label.

There are no subsequent tape RDLIN cards required by the object program, regardless of the number of reels pertaining to a given file.

Date Card

If output label checking is to be performed, a date card must precede the merge-phase object deck. The date card is placed between the 6 card loader and the first card of the merge-phase object deck.

Running the Object Program

Follow this procedure for running the object program:

1. Prepare the printer by inserting forms on the printing device and installing an appropriately punched carriage tape, if necessary.
2. Prepare the necessary magnetic tape(s) by mounting the correct tape on each drive, by setting the address-select dial on each drive to the correct number, and by setting the density switch on each drive. If tape input has been specified, a drive must be prepared for each input file, and the address-select dials must correspond to the order of merge. A one-way merge with tape input, for example, necessitates setting the address-select dial on the input drive at 1. A

two-way merge with tape input necessitates setting the address-select dials on 1 and 2. The dial settings 1, 2, 3, and 4 are reserved for input files. If tape output is specified, the address-select dial of the tape drive that contains the output file must be set at 5. If DUMP is specified, an additional drive must be prepared, and the dial must be set at 6.

3. Prepare the required disk pack(s) by placing it on the proper drive(s) and by following the standard procedure for making disk units operational.
4. Turn sense switch A on.
5. Turn the I/O check-stop switch off if card-read-error diagnostic halts are desired.
6. Insert the object-program control cards and any additional required cards (RDLIN cards, date card) into their appropriate places in the object program deck.
7. Place the resulting deck in the read hopper of the IBM 1402 Card Read-Punch and follow the normal procedure for loading a card deck.

Merge 6 Assignment Phase Messages

There are a number of messages that might be printed out by the console printer or the 1403/1404 printer (whichever has been specified at generation time) during execution of the assignment phase of the Merge 6 program. Most of these messages signify the presence of an inconsistency between the generated program and the object program specifications, but a number of them simply indicate the presence of options specified by the user. Figure 11 is a complete list of possible messages, accompanied by the reason for the appearance of the message and the procedure to follow in correcting existing errors. The first six messages and the last message in Figure 11 will be accompanied by a programmed halt. The other messages do not cause a halt but must be corrected before the merge phase begins. This prevents repeated stopping and starting during execution of the assignment phase.

Merge Phase Halts

The information given for each halt consists of:

1. The contents of the B-address register, the A-register, or the I-address register when the halt occurs. In determining the cause of a halt, refer first to the B-address register. If the tens and units positions of the B-address register are blank, the A-register (shown in Figure 12 with its BCD equivalent)identifies the halt. Only halt number 35 is indicated by the I-address register. If a tape operation causes a halt, the tape SELECT light that is on identifies the tape unit on which the error that caused the halt occurred.
2. The reason for the halt.
3. The procedure to be followed when the halt occurs.

Merge Phase Messages

The only messages printed as a result of the merge phase will occur after the phase has been completed. Messages 4, 5, and 7 will always be printed. The remaining messages will be printed only if the applicable options were specified at object time. Message 1, for example, will be printed only if the user has specified a hash total.

Following is a complete list of merge-phase messages:

1. XXXXX HASH TOTAL
2. XXX SEQUENCE ERRORS WERE DETECTED
3. XXXXX CARD ADDITION RECORDS PROCESSED
4. XXXXXXX RECORDS PROCESSED
5. XXXXXXX RECORDS MERGED TO OUTPUT FILE
6. LAST OUTPUT BLOCK BEGINS IN SECTOR XXXXXX
7. END OF MERGE PROGRAM

MESSAGE	REASON	PROCEDURE
Error — control cards are out of sequence.	The control cards are either omitted or out of sequence.	Correct and restart.
The card merge phase omitted — insert prior to control card 1 and restart.		
Read error — read control card 1 again. Read error — read control card 2 again.	The specified control card contains an invalid character or was loaded into core storage improperly.	Remove the remaining cards from the hopper. Run the cards out of the reader with the nonprocess runout key. Put the cards in the hopper again, and press START.
Read error — reread generated options card.	The specified card contains an invalid character or was loaded into core storage improperly.	Remove the remaining cards from the hopper. Run the cards out of the reader with the nonprocess runout key. Put the cards in the hopper again, and press START.
Press start to proceed only if object machine has modify address feature. Otherwise, restate object machine size or regenerate object program.	The object program was generated for a system with more than 4K and the modify address feature. If the system used to execute the object program is a 4K system, it must have this special feature.	Follow the procedure indicated in the message.
Control card 1 Column xxx invalid. Control card 2 Column xxx invalid.	The indicated column on the specified card violates the control card format.	Correct and restart.
Specified BI and MS incompatible.	The specified input blocking factor is too large for the specified machine size.	Restate either the input blocking factor or the machine size before restarting.
Error — the following specified option was not generated.	The following option, though specified at object time, was not included during the program generation.	Restate the object time options to conform with generated options or regenerate the program including those options.
** Hash Total **		
** Sequence check and hash total **		
** Sequence check **		
** Deletions by control data **		
** Selections by class **		
** Deletions by class **		
** X way merge **		
** Additions - cards **		
** Multiple control field compare **		
** Disk label checking - input **		
** Disk label checking - output **		
** Tape label checking - input **		
** Tape label checking - output **		
** Tape Input **		
** Tape output **		
** Disk Input **		
** Disk Output **		
Error — record length with record mark exceeds maximum allowed.	Maximum allowable record length has been exceeded by the addition of a record mark on each output record.	Delete the record mark option, and restart.

Figure 11. Assignment Phase Messages (Part 1 of 2)

MESSAGE	REASON	PROCEDURE
Object machine size inadequate for specified conditions.	The specified machine size is smaller than required.	Regenerate with larger machine size and/or fewer options and restart.
Machine size inadequate for specified user area.	The specified user area extends into the core storage required by the program.	Restate the user area and restart.
Specify location of control character for selections or deletions by class.	Columns 22 through 25 of control card must not be blank when column 21 is not blank.	Either supply data in columns 22 through 25, or make column 21 blank.
Location of controlling character exceeds L.	The position of the controlling character stated in control card 1 columns 22 through 25 is incompatible with the record length.	Correct and restart.
User area exceeds machine size.	The value stated in control card 1 columns 30 through 34 is greater than the stated machine size.	Correct and restart.
Control card 1 Columns 1 and 37 — 40 incompatible.	Tape input is specified but the number of input reels is missing or in error.	Correct and restart.
Control card 1 Columns 41 — 47 invalid.	Control card 1 columns 41 through 47 violate conditions described in control card format.	Correct and restart.
Invalid L for Tape input.	The specified record length is invalid for tape input.	Correct and restart.
Input block size incompatible with MS. Output block size incompatible with MS.	The specified block size exceeds the maximum allowed for the stated machine size.	Restate a smaller block size and restart.
Disk BIS omitted - card 1, column 9. Disk BOS omitted - card 1, column 11.	Disk input or output specified but the number of disk sectors per block has been omitted.	State value in control card 1 columns 9 and 10 for input, or columns 11 and 12 for output, and restart.
Error — input block size exceeds capacity of input sector blocking factor. Error — output block size exceeds capacity of output sector blocking factor.	The stated number of disk sectors (disk I/O) is inadequate to contain the stated input and/or output block size.	Restate input and/or output block size to be compatible with input and/or output sector blocking factor and restart.
Specified control field location exceeds record length.		Restate either control field position or record length and restart.
Specified BO and MS incompatible.	The specified output blocking factor is too large for the specified machine size.	Restate either the output blocking factor or the machine size before restarting.
Total length of control field/s in error.	The stated total of control-field characters does not agree with the computed total.	Correct and restart.
Error — control fields xx and xx overlap.	The stated control fields (numbered in the record from left to right) overlap.	Correct and restart.
Total length of control field/s is greater than L.	The specified record length is incapable of containing the specified control fields.	Correct and restart.
End of assignment phase.	Execution of assignment phase completed.	
Sector address in columns 42-47 of control card 1 will cause cylinder overflow.	The routines are to be written on disk in the load mode, and IOCS does not allow cylinder overflow in the load mode.	Correct and restart.
Correct assignment phase errors and restart.	Error (s) detected during execution of assignment phase.	Correct and restart.

Figure 11. Assignment Phase Messages (Part 2 of 2)

NUMBER	A-REG	B-ADD REGISTER	REASON	PROCEDURE
1	1	xxbb	A disk error other than those defined elsewhere in this list has occurred.	Press the start key to retry the disk operation.
2	3 C21	xxbb	The disk light indicates the reason for this halt. Disk light ON: Access inoperable condition on a disk header label read or write operation. Disk light OFF: Disk header label does not contain THDRb in the first five positions.	Press the start key, the start reset key, and the start key again to recheck all header labels on this pack for this logical file.
3	4	xxbb	A parity, wrong-length-record, or unequal-address-compare error was detected while a disk header label was being read or written.	Press the start key to recheck all header labels on this pack for this logical file.
4	5 C41	xxbb	No header label found for the specified disk input file.	Visually check the disk pack and label specifications. Mount a new pack, if required, and press the start key to recheck all header labels. The new disk pack must have the address range already specified by the RDLIN card.
5	6 C42	xxbb	A RDLIN information card for a disk file is missing.	Remove the remaining cards from the read hopper. Run the cards out of the reader with the nonprocess runout key. Place the proper cards, including the correct RDLIN card, in the read hopper. Press the start key, the start reset key, and the start key again to continue processing.
6	? CBA82	xxbb	First halt (parity error) when SCAN has been specified. The error stop switch on the tape-adapter unit must be off. The second halt appears at the symbolic label COCSCN+1 or IOCSCN+1 in the program listing.	Set the tape-select switch to D, and turn off the check stop switch on the auxiliary console. Press the start key to reread the error block for scanning.
7	¤ CBA84	xxbb	A parity error was detected while reading a tape header label.	Press the start key to read the header label again.
8	! B82	xxbb	The expiration date for this tape has not been reached.	Press the start key twice to check the retention period again. Press the start key, the start reset key, and the start key again to use this reel anyway.
9	L B21	xxbb	A parity error was detected while writing a tape header label.	Press the start key to rewrite the header label.
10	$ CB821	xxbb	A tape input header label does not correspond to the user's specifications.	Press the start key twice to recheck the header label. Press the start key, the start reset key, and the start key again to process the reel of data without rechecking the header label.
11	* B84	xxbb	The tape input trailer does not check with the block count accumulated by IOCS.	Press the start key to ignore this situation and continue processing.
12	/ CA1	xxbb	A parity error was detected while reading a tape trailer label.	Press the start key to reread the trailer label.
13	X CA421	xxbb	End-of-reel condition for a tape file.	Mount the next reel for the file, and press the start key to continue processing.
14	, CA821	xxbb	A parity error was detected while writing a tape trailer label.	Press the start key to rewrite the trailer label.
15	% A84	xxbb	Thirty parity-error detections were made during an attempt to write a block of data.	Press the start key to make thirty additional attempts to write the block correctly.

Figure 12. Merge Phase Halts (Part 1 of 3)

NUMBER	A-REG	B-ADD REGISTER	REASON	PROCEDURE
16	# 821	xxbb	Mixed label routines have been generated, but no type A, B, or C label is specified in column 80 of the RDLIN card.	Remove the cards from the read hopper, and run the cards out of the reader with the nonprocess runout key. Place the proper cards, including the corrected RDLIN card, in the hopper. Press the start key, the start reset key, and the start key again to read the RDLIN card.
17	@ C84	xxbb	A parity error was detected during an attempt to write on the dump tape, tape 6.	Press the start key to backspace, erase, and write the record again.
18	. BA821	xxbb	A RDLIN information card for a tape file is missing.	Remove the cards from the read hopper, and run the cards out of the reader with the nonprocess runout key. Place the proper cards, including the correct RDLIN card, in the hopper. Press the start key, the start reset key, and the start key again to read the RDLIN card.
19	A BA1	xxbb	A disk RDLIN card is required to define the next file area for file 1, the first input file.	Remove the cards from the read hopper, and run the cards out of the reader with the nonprocess runout key. Place the correct RDLIN card in the hopper. Press the start key to read the card. The rest of the object deck need not be placed in the read hopper until any additional required disk RDLIN cards have been read.
20	B BA2	xxbb	A disk RDLIN card is required to define the next file area for file 2, the second input file.	Follow the procedure given for halt number 19.
21	C CBA21	xxbb	A disk RDLIN card is required to define the next file area for file 3, the third input file.	Follow the procedure given for halt number 19.
22	D BA4	xxbb	A disk RDLIN card is required to define the next file area for file 4, the fourth input file.	Follow the procedure given for halt number 19.
23	E CBA41	xxbb	A disk RDLIN card is required to define the next file area for the output file.	Follow the procedure given for halt number 19.
24	F CBA42	xxbb	Sequence check has been specified, and the record to be moved to the output file is out of sequence.	Press the start key to accept the record out of sequence and continue processing.
25	G BA421	xxbb	Deletions by control data or additions has been specified, and a card read error has occurred. (The I/O check stop switch is off.)	Remove the cards from the read hopper, and run the cards out of the reader with the nonprocess runout key. Replace the cards in the read hopper in their proper order and press the start key.
26	H BA8	xxbb	ADDITIONS in the 70-character card format has been specified, and either the file name in columns 1 through 8 is not identical to that of the first additions card, or a sequence error has been detected in columns 9 and 10 of the addition card. The latter condition is applicable only when the record length is greater than 70 characters. The sequence number must be in the range of 01 through 43. The card that contains the error is the last card in the stacker when the halt occurs. ADDITIONS in the 80-character card format has been specified, and the total number of cards read was not a multiple of the cards per record.	Remove the cards from the read hopper, and run the cards out of the reader with the nonprocess runout key. Correct the card in error. Replace the cards in the hopper and press the start key. Check the entire additions file for this condition and provide the required number of additional cards. Replace the cards in the hopper, and press the start key if the record in error is the last record. If the error occurs in a record other than the last, the entire merge operation should be rerun.

Figure 12. Merge Phase Halts (Part 2 of 3)

NUMBER	A-REG	B-ADD REGISTER	REASON	PROCEDURE
27		111	A wrong-length record from a tape input file was detected.	Press the start key to delete the record. If the dump option has been specified, the record will be written on tape 6, the dump tape. NOTE: The presence of a group-mark with word-mark in the input area will cause repeated wrong-length record halts.
28		222	Seven unequal-address-compare detections occurred during a disk operation.	Press the start key twice for ten more attempts at processing. NOTE: The A-address register contains the high-order address of the disk-control field.
29		333	Seven parity-error detections occurred during a disk operation.	Press the start key twice for ten more attempts at processing. NOTE: The A-address register contains the high-order address of the disk-control field.
30		444	An access-inoperable error was detected during a disk operation.	Determine the disk drive number by examining the storage position specified by the A-address register. Manually key-in a different number, if required. Press the start key to retry processing with the same or altered drive number.
31		555	Seven wrong-length-record detections occurred during a disk operation.	Press the start key twice for ten more attempts at processing. NOTE: The A-address register contains the high-order address of the disk-control field.
32		666	Disk drive specification missing for a consecutive output file. (This applies to labeled files only.)	Mount a disk pack, if missing. Manually, key-in the disk drive number at the address specified by the A-address register, and press START to recheck all header labels.
33		777	The portion of a pack specified for a disk output file contains active records. This applies to labeled files only, and is detected when the dates in the header label are read.	Mount a different disk pack, and press the start key twice to recheck all header labels. Press the start key, the start reset key, and the start key again to use the original pack without altering it. This automatically deletes the data in the file identification field of the header label.
34		888	The pack serial number in a labeled disk-output file does not check.	Press the start key, the start reset key, and the start key again to write the disk header label with this condition. Press the start key twice to recheck all header labels.

NUMBER	I-ADDRESS	REASON	PROCEDURE
35	Address of symbolic label IOCSCN+1 (or COCSCN+1)	Second halt when SCAN has been specified.	Scan for the character with the parity error and correct it if possible. Reset the tape select switch to N. Turn the check stop switch on, and press the check reset key. Press the start reset key and the start key to process the corrected block, or press the start key to bypass an incorrect block and continue processing.

Figure 12. Merge Phase Halts (Part 3 of 3)

Appendix I

Timing Charts

This section contains five Merge 6 object-program timing charts (Figure 13), for 2-way merges performed on IBM 1401-1311 systems with 4,000, 8,000, 12,000 and 16,000 positions of core storage. Times are shown for object-program runs when the Direct-Seek (DS) feature is used, and for runs when a Normal-Seek (NS) operation is used.

The parameters involved are:

- Input format. Fixed-length records with one control-data field.
- Record size. 20, 40, 80, 100 and 200 characters.
- Approximate number of input records. 10,000, 25,000, 50,000, 100,000, and 200,000. Both input files contain exactly the same number of records.
- The input and output blocking factors equal the maximum blocking factor.

The times are based on the following factors: disk input and output without any additional options, and one disk pack for each input file. All files begin at the lowest possible address. For example, the starting addresses would be 000000 for the input file on drive 0, 020000 for the input file on drive 1, and 040000 for the output file on drive 2.

The processing time is in minutes. The times do not include the time required to load the program. For over-all time, add 1.3 minutes for card-read operations.

For Merge 6 object programs to be run on an IBM 1460, use the values cited in Figure 13 as the maximum times for the same parameters.

2-Way - 20 Character Records

Approximate Number of Input Records	Time in Minutes							
	Machine Size							
	4K		8K		12K		16K	
	DS	NS	DS	NS	DS	NS	DS	NS
10,000	1.9	1.9	1.3	1.3	1.3	1.3	1.2	1.3
25,000	4.7	4.7	3.2	3.2	3.0	3.1	2.9	3.0
50,000	9.2	9.3	6.2	6.4	5.9	6.1	5.7	5.8
100,000	18.2	19.7	12.5	13.9	11.8	12.3	11.2	11.7
200,000	35.1	37.1	24.8	25.9	23.6	24.8	22.4	23.5

2 Way - 100 Character Records

Approximate Number of Input Records	Time in Minutes							
	Machine Size							
	4K		8K		12K		16K	
	DS	NS	DS	NS	DS	NS	DS	NS
10,000	6.1	6.4	3.4	3.6	3.1	3.3	2.7	3.0
25,000	15.0	15.7	8.4	8.9	7.6	8.2	6.8	7.4
50,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
100,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
200,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

2 Way - 40 Character Records

Approximate Number of Input Records	Time in Minutes							
	Machine Size							
	4K		8K		12K		16K	
	DS	NS	DS	NS	DS	NS	DS	NS
10,000	3.0	3.0	1.8	1.9	1.7	1.8	1.6	1.7
25,000	7.2	7.5	4.5	4.6	4.2	4.3	3.9	4.0
50,000	14.5	15.0	8.9	9.3	8.2	8.7	7.6	8.1
100,000	28.6	29.7	17.7	18.8	16.3	17.5	15.2	16.3
200,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

2 Way - 200 Character Records

Approximate Number of Input Records	Time in Minutes							
	Machine Size							
	4K		8K		12K		16K	
	DS	NS	DS	NS	DS	NS	DS	NS
10,000	11.5	12.0	6.0	6.5	5.4	5.9	4.8	5.3
25,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
50,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
100,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
200,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

2-Way - 80 Character Records

Approximate Number of Input Records	Time in Minutes							
	Machine Size							
	4K		8K		12K		16K	
	DS	NS	DS	NS	DS	NS	DS	NS
10,000	5.1	5.2	2.9	3.0	2.7	2.8	2.4	2.5
25,000	12.5	13.0	7.1	7.5	6.5	6.9	5.9	6.4
50,000	25.8	26.3	14.0	15.0	12.9	14.1	11.6	12.7
100,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX
200,000	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX

LEGEND

DS: Direct Seek
NS: Normal Seek
XXX: Requires Multi-Pack Files of Input

NOTES

1. Processing Time is in Minutes. For Over-all Time, Add 1.3 Minutes for Card Read.
2. The Number of Input Records is Approximate.

Figure 13. Timing Charts

Appendix II

Core Storage Requirements Charts

The following information is applicable to the charts (Figures 14, 15, 16 and 17) included in this Appendix:

1. Specification of the following: console printer, user exit 7, user exit 8, and the unload option (tape) requires no additional storage.
2. Presence of the dump option (with tape input only) is assumed in Figures 15 and 16. Use of the scan option requires additional core storage (see Figure 18).
3. Specification of TM (tape mark) for type B or C input tape labels requires 13 additional core-storage positions only if no label checking is generated.
4. Core storage requirements for 1-way merges (reblocking) are not shown. Under all input/output conditions, sufficient storage will be available to accommodate any permitted combination of options.
5. A maximum of 7 core-storage positions must be added to the core requirements cited in Figures 14, 15, and 16, if the object program will be generated for a system with 4,000 positions of core storage.
6. The address constant with the label AAAEX4 will contain the exact program size.

The maximum core storage requirements for a user program can be calculated by using the figures cited in these charts. For example, assume that a user with an 8k system, wishes to generate a 3-way merge with the following:

disk input and disk output with label checking
card additions
cylinder overflow
sequence checking
user exit 1

He should total the following figures:

3-way merge, disk input/output with full label checking	2596	(Figure 14)
card additions	423	(Figure 18)
cylinder overflow	265	(Figure 18)
sequence checking*	158	(Figure 18)
user exit 1	005	(Figure 18)
common entry 1	004	(Figure 18)
	3451	

The total core storage requirement for storing this program is 3451 positions.

*The sum 158 is derived by adding 144 positions for sequence checking during a 3-way merge to the 14 positions required for sequence checking with card additions.

INPUT		INPUT LABELS						OUTPUT		OUTPUT LABELS					CORE-STORAGE REQUIREMENT		
Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	All (T)	Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	2-Way Merge	3-Way Merge	4-Way Merge
X								X							2089	2522	3012
X		X						X		X					2184	2596	3074
X		X						X							2184	2596	3074
X								X		X					2184	2596	3074
X									X						2119	2552	3042
X		X							X						2087	2492	2956
X		X							X		X				2092	2497	2961
X		X							X			X			2092	2497	2961
X		X							X				X		2092	2497	2961
X		X							X					X	2091	2496	2960
X									X		X				2187	2604	2980
X									X			X			2186	2603	2979
X									X				X		2186	2603	2979
X														X	2185	2602	2978

Figure 14. Core Storage Requirements for Object Programs With Disk Input and Disk or Tape Output

INPUT		INPUT LABELS						OUTPUT		OUTPUT LABELS					CORE-STORAGE REQUIREMENT		
Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	All (T)	Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	2-Way Merge	3-Way Merge	4-Way Merge
	X							X							2252	2664	3128
	X							X		X					2157	2489	2875
	X		X					X							2528	3017	3558
	X			X				X							2460	2935	3462
	X				X			X							2464	2939	3466
	X		X				X	X							2231	2558	2939
	X			X			X	X							2229	2555	2935
	X				X		X	X							2229	2555	2935
	X					X		X							2229	2555	2935
	X		X					X		X					2157	2489	2875
	X			X				X		X					2157	2489	2875
	X				X			X		X					2157	2489	2875
	X		X				X	X		X					2157	2489	2875
	X			X			X	X		X					2157	2489	2875
	X				X		X	X		X					2157	2489	2875
	X					X		X		X					2157	2489	2875

Figure 15. Core Storage Requirements for Object Programs With Tape Input and Disk Output

INPUT		INPUT LABELS						OUTPUT		OUTPUT LABELS					CORE-STORAGE REQUIREMENT		
Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	All (T)	Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	2-Way Merge	3-Way Merge	4-Way Merge
	X								X						1537	1941	2397
	X		X						X						1882	2363	2896
	X			X					X						1815	2282	2801
	X				X				X						1823	2290	2809
	X		X				X		X						1997	2326	2709
	X			X			X		X						1995	2323	2705
	X				X		X		X						1995	2323	2705
	X					X			X						1990	2318	2700
	X		X						X		X				1998	2327	2710
	X			X					X			X			1995	2323	2705
	X				X				X				X		1995	2323	2705
	X								X					X	1990	2318	2700
	X		X				X		X		X				1998	2327	2710
	X			X			X		X			X			1995	2323	2705
	X				X		X		X				X		1995	2323	2705
	X					X			X					X	1990	2318	2700
	X								X		X				1996	2324	2706
	X								X			X			1995	2323	2705
	X								X				X		1995	2323	2705
	X		X						X			X			1997	2326	2709
	X		X						X				X		1997	2326	2709
	X		X						X					X	1997	2326	2709
	X		X				X		X			X			1997	2326	2709
	X		X				X		X				X		1997	2326	2709
	X		X				X		X					X	1997	2326	2709
	X			X					X		X				1996	2324	2706
	X			X					X				X		1995	2323	2705
	X			X					X					X	1995	2323	2705
	X			X			X		X		X				1996	2324	2706
	X			X			X		X				X		1995	2323	2705
	X			X			X		X					X	1995	2323	2705
	X				X				X			X			1996	2324	2706
	X				X				X				X		1995	2323	2705
	X				X				X				y		1995	2323	2705
	X				X		X		X		X				1996	2324	2706
	X				X		X		X			X			1995	2323	2705
	X				X		X		X					X	1995	2323	2705
	X					X			X						1996	2324	2706
	X					X			X				X		1995	2323	2705
	X					X			X					X	1995	2323	2705

Figure 16. Core Storage Requirements for Object Programs With Tape Input and Tape Output

INPUT		INPUT LABELS						OUTPUT		OUTPUT LABELS					CORE-STORAGE AVAILABLE			
Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	All (T)	Disk	Tape	Disk	Std. A	Std. B	Std. C	Non-Std.	1-Way Merge	2-Way Merge	3-Way Merge	4-Way Merge
X		X							X					X	1450	1450	1450	1450
X									X					X	1450	1450	1450	1450
	X					X		X							1450	1350	1200	1075
	X					X		X		X					1450	1350	1200	1075
	X					X			X						1875	1775	1625	1500
	X								X					X	1800	1700	1600	1475
	X					X			X					X	1800	1700	1575	1425
	X		X						X					X	1600	1375	1175	0975
	X		X				X		X					X	1300	1050	0700	0400
	X			X					X					X	1600	1450	1250	1050
	X			X			X		X					X	1350	1175	0875	0575
	X				X				X					X	1600	1425	1225	1025
	X				X		X		X					X	1450	1125	0800	0475
	X					X			X		X				1300	1175	1025	0900
	X					X			X			X			1400	1250	1100	0975
	X					X			X				X		1325	1200	1075	0925

Figure 17. Core Storage Available for User Programming of Nonstandard Labels

PARAMETER CARD	OPERAND	APPLIES TO	CORE-STORAGE REQUIREMENT							
			2-Way Merge		3-Way Merge		4-Way Merge		Additions	
			4K	8-16K	4K	8-16K	4K	8-16K	4K	8-16K
ADDITIONS	CARD	Any I/O Media	391	391	423	423	455	455	XXX	XXX
SELECTDLET	CONTROLDATA	Tape I/O	140	140	140	140	140	140	XXX	XXX
		Disk I/O	220	220	220	220	220	220	XXX	XXX
	SELDELCLASS	Tape Input	033	029	040	036	047	043	+007	+007
		Disk Input	040	036	047	043	054	050	+007	+007
SYSTEMSPEC	CYLOFLOW	Disk I/O	269	265	269	265	269	265	XXX	XXX
	DIRECT	Disk I/O	021	021	021	021	021	021	XXX	XXX
CHECKS	SEQHASH	Tape Input	120	116	134	130	148	144	+014	+014
		Disk Input	134	130	148	144	162	158	+014	+014
NOCTLFLDS	02-10	Any I/O Media	142	130	142	130	142	130	XXX	XXX
TAPE	SCAN	Tape Input	014	014	019	019	024	024	XXX	XXX
EXITS *	EXIT1	Tape Input	000	000	000	000	000	000	XXX	XXX
		Disk Input	005	005	005	005	005	005	XXX	XXX
	EXIT2	Tape Input	011	011	011	011	011	011	XXX	XXX
		Disk Input	005	005	005	005	005	005	XXX	XXX
	EXIT3	Tape Input	XXX	XXX	011	011	011	011	XXX	XXX
		Disk Input	XXX	XXX	005	005	005	005	XXX	XXX
	EXIT4	Tape Input	XXX	XXX	XXX	XXX	011	011	XXX	XXX
		Disk Input	XXX	XXX	XXX	XXX	005	005	XXX	XXX
	EXIT5	Any I/O Media	004	004	004	004	004	004	XXX	XXX
	EXIT6	Any I/O Media	004	004	004	004	004	004	XXX	XXX

Note: * Add 4 core-storage positions for the common entry point if any one of the exits 1 through 4 is generated

Figure 18. Core Storage Requirements for Special Parameters Added to Basic Program

READER'S SURVEY FORM

Merge 6 Specifications and Operating Procedures, IBM 1401 and 1460, Form C24-3053-3

- Is the material:

	Yes	*Satisfactory*	*No*
Easy to read?	☐	☐	☐
Well organized?	☐	☐	☐
Fully covered?	☐	☐	☐
Clearly explained?	☐	☐	☐
Well illustrated?	☐	☐	☐

- How did you use this publication?
  - As an introduction to the subject ☐
  - For additional knowledge of the subject ☐

- Which of the following terms best describes your job?

Customer Personnel		*IBM Personnel*	
Manager	☐	Customer Engineer	☐
Systems Analyst	☐	Instructor	☐
Operator	☐	Sales Representative	☐
Programmer	☐	Systems Engineer	☐
Trainee	☐	Trainee	☐
Other _____		Other _____	

- Check specific comment (if any) and explain in the space below:
  (*Give page number*)
  - ☐ Suggested Change (Page) ☐ Suggested Addition (Page)
  - ☐ Error (Page) ☐ Suggested Deletion (Page)

Explanation:

Space is available on the other side of this page for additional comments.
Thank you for your cooperation.

Tear Along Line

C24-3053-3

Staple

Fold

Fold

FIRST CLASS
PERMIT NO. 387
ROCHESTER, MINN.

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

POSTAGE WILL BE PAID BY . . .

IBM Corporation
Systems Development Division
Development Laboratory
Rochester, Minnesota 55901

Attention: Product Publication, Dept. 245

Fold

Fold

Cut Along Line

Merge 6, IBM 1401/1460 Printed in U.S.A. C24-3053-3

IBM®

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

Additional Comments:

Merge 6, IBM 1401/1460 Printed in U.S.A. C24-3053-3

IBM

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