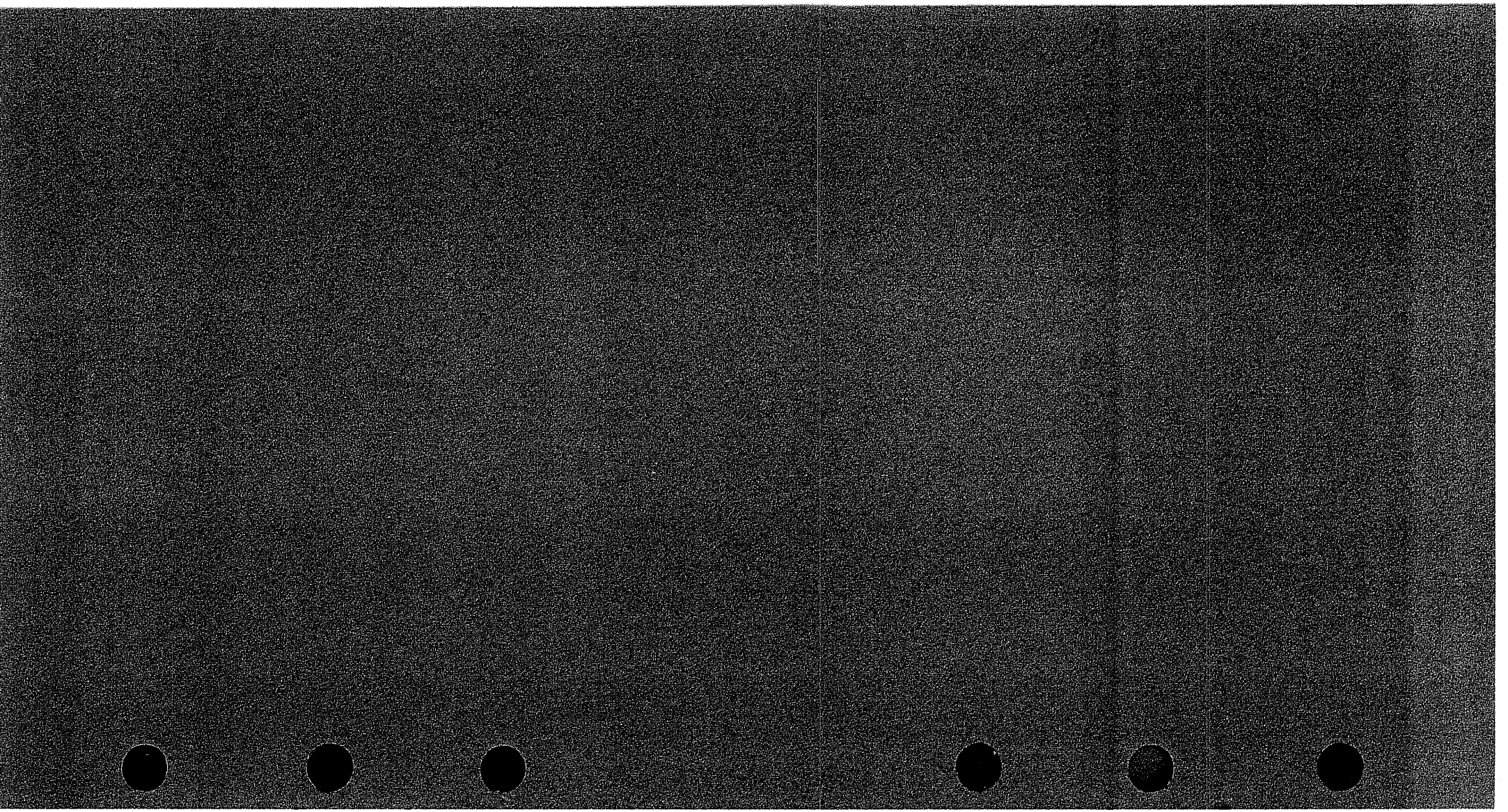
60454000

CONTROL DATA CORPORATION

UPDATE INSTANT

CONTROL DATA®
CYBER 170 SERIES
CYBER 70 SERIES
6000 SERIES
7600
COMPUTER SYSTEMS



CONTROL DATA CORPORATION

UPDATE INSTANT

CONTROL DATA®
CYBER 170 SERIES
CYBER 70 SERIES
6000 SERIES
7600
COMPUTER SYSTEMS

RECORD of REVISIONS

REVISION	NOTES
A	Original printing. This manual is a successor
11-75	publication to publication number 60360200, for users
	of NOS 1.0, NOS/BE 1.0, and SCOPE 2.1 operating
	systems.

AA 3573

Pub. No. 60450000

Printed in the United States of America

Address comments concerning
this manual to:

Control Data Corporation
Publications and Graphics Division
215 Moffett Park Drive
Sunnyvale, California 94806

PREFACE

UPDATE provides a means of maintaining and updating source decks on libraries in compressed symbolic format under the following operating systems and computer systems.

- NOS 1.0 and NOS/BE 1.0 for the CONTROL DATA® CYBER 170 Series, CYBER 70 Series/Models 72, 73, and 74, and 6000 Series Computer Systems
- SCOPE 2.1 for the CONTROL DATA® CYBER 70/Model 76 and 7600 Computer Systems

CONTENTS

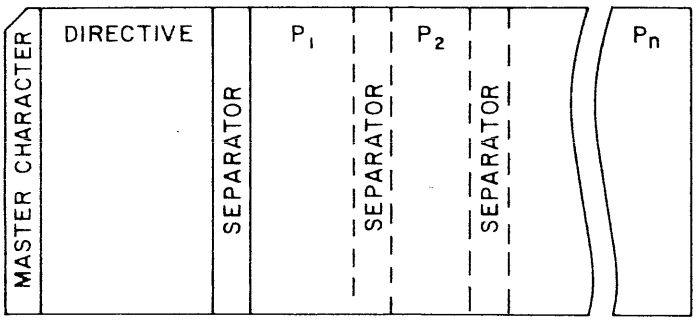
FEATURES

Features of UPDATE include:

- Creation of a program library from source decks.
- Copying of old program libraries from sequential to random format and vice versa.
- Merging of two program library files.
- Updating of source decks by inserting, deleting, and restoring cards according to sequence in the deck or according to correction set.
- Ability to completely and permanently remove correction sets from the program library.
- Generation of a compile file containing corrected output acceptable as input to other processing programs, such as compilers and assemblers. Contents of the compile file and subdivision of the compile file into logical records and files is controlled through UPDATE directives. This file can be formatted as 80- or 90-column card images or can be in compressed form.
- Processing of directives, new text, and new source decks from a file other than the job INPUT file.
- Generation of a new, updated program library.
- Comprehensive list output noting any changes occurring during the run and status of the program library.
- Ability to change the directive card master control character.
- Recognition of abbreviated forms of directives and capability of turning off the search for the abbreviated forms to speed up processing.
- Ability to use full 64-character set, including the colon.
- Checksumming of program library.

DIRECTIVES

DIRECTIVE FORMAT



master control character	In column one. It is defined by UPDATE to be an asterisk but may be changed through use of the * parameter on the UPDATE control statement.
directive name	Starts in column two; terminated by a comma or one or more blanks. Most directives have a full name and an abbreviated name. When the NOABBREV directive is in effect, UPDATE does not recognize the abbreviated forms of the directive names.
separator	Parameters are separated by any character other than A through Z, 0 through 9, or + - * / ( ) $ or =. A separator has a display code value of 55_8 or greater. No embedded blanks are permitted.
p_i	Parameter. Depending on its requirements, a directive may have no parameters or a number of parameters. Numeric parameter fields are decimal.

CARD IDENTIFICATION

The full form of a card identifier is:

ident. seqnum

ident	1-9 character name of a correction set or deck. A period terminates the ident.
seqnum	Decimal ordinal (1 to 262143) of the card within the correction set or deck. Any character other than 0-9 terminates the sequence number.

The shortened forms of card identifiers can be used on BEFORE, INSERT, DELETE, RESTORE, and COPY directives. Shortened forms are expanded as follows:

seqnum	Expands to idname. seqnum where idname is a correction set identifier whether or not it is also a deck name.
. seqnum	Expands to dname. seqnum where dname is a deck name.

To specify an operation which affects all decks in a program library between and including two specified decks, the following format can be used.

*directive $deck_1$. $deck_n$

This type of operation is allowed with PURDECK, SEQUENCE, and COMPILE

DIRECTIVE TYPES

DECK IDENTIFYING DIRECTIVES

DECK	(DK)	Identifies normal deck
COMDECK	(CD)	Identifies common deck

The following directives can be used prior to the first DECK or COMDECK directive in either a creative run or a correction run.

ABBREV	NOABBREV
DECLARE	NOLIST
LIMIT	READ
LIST	REWIND
	SKIP

If any directive other than one or more of the above precedes a DECK or COMDECK directive, the run is considered a correction run.

CORRECTION DIRECTIVES

INSERT	(I)	Insert text after specified card
BEFORE	(B)	Insert text before specified card
DELETE	(D)	Deactivate card and optionally insert text in its place
RESTORE	(R)	Reactivate card and optionally insert text after it
COPY	(C)	Copy and insert text from specified deck
YANK	(Y)	Deactivate correction sets
SELYANK	(SY)	Selectively deactivate correction sets
YANKDECK	(YD)	Deactivate all cards in decks

PURGE	(P)	Permantly remove correction sets
SELPURGE	(SP)	Permanently remove cards belonging to correction sets from specific decks
PURDECK	(PD)	Permanently remove all cards in decks
CHANGE	(CH)	Change correction set name
SEQUENCE	(S)	Resequence decks and purge all inactive cards

FILE MANIPULATION DIRECTIVES

File manipulation directives are illegal when UPDATE is reading from an alternate file. They can be on the primary input file only.

READ	(RD)	Read input stream from specified file
SKIP	(SK)	Skip forward specified number of records on file
REWIND	(RW)	Rewind named file

These operations cannot be performed on the following reserved files or their equivalents:

INPUT	Source of directives and input text
OUTPUT	List output
COMPILE	Compile output
SOURCE	Source output
OLDPL	Old program library
NEWPL	New program library
MERGE	Merge

UPDTSCR UPDTCDK UPDTTPL UPDTEXT UPDTAUD UPDTPMD	UPDATE scratch files

SELECTIVE COMPILE DIRECTIVE

COMPILE	(C)	Identifies decks to be written on compile file

COMPILE FILE DIRECTIVES

WEOR	(W)	Write end-of-record of specified level on compile file
CWEOR	(CW)	Write end-of-record or end-of-file on compile file if the buffer is not empty
CALL	(CA)	Write called common deck onto compile file
IF		Write text onto compile file if condition is defined
ENDIF	(EI)	Optionally used to specify end-of-conditional text
TEXT	(T)	The text following is not to be scanned for directives
ENDTEXT	(ET)	Resume scan for directives
DONT	(DT)	Do not rescind yanks of correction set applying to the following text
DO		Rescind yanks of correction sets applying to the following text

To be recognized while the compile file is being written, these directives must have the same master control character as that defined when the library was created.

SPECIAL DIRECTIVES

/		List Comment
NOABBREV	(NA)	Deselects the abbreviated directives feature
ABBREV		Reselects the abbreviated directives feature
LIMIT	(LT)	Changes maximum size allowed for list output file
NOLIST	(NL)	Inhibits list output
LIST	(L)	Resumes list output
DEFINE	(DF)	Defines names for subsequent use by IF directives
PULLMOD	(PM)	Recreates correction sets
DECLARE	(DC)	Restrict corrections to declared deck

ALPHABETIZED LIST OF DIRECTIVES

In the following, c indicates a card identifier of the form ident.seqnum.

*ABBREV

Check for abbreviated forms of directives.

*ADDFILE file,c
*AF

Read new decks and/or common decks from named file and insert the information after specified card (c). If c is omitted, decks are added at end of library. If file is omitted, the primary input file is used.

*ADDFILE file,dname
*AF

Read new decks and/or common decks from named file and insert the information after specified deck (dname). If dname is omitted, decks are added at end of library. If file is omitted, the primary input file is used.

*BEFORE c
*B

Insert text following this directive into library before card specified by c.

*CALL dname
*CA

Causes text of common deck (dname) to be written on compile file.

*CHANGE idname$_1$, idname$_2$, . . . , idname$_{n-1}$, idname$_n$
*CH

Changes correction set named idname$_i$ to idname$_{i+1}$. idname is 1-9 of the following alphanumeric characters:

A-Z 0-9 + - * / ( ) $ =

*COMDECK dname,NOPROP
*CD

Introduces common deck. dname is 1-9 of the following characters:

A-Z 0-9 + - * / ( ) $ =

Presence of NOPROP specifies that if this deck is modified, decks calling this deck are not to be considered modified.

*COMPILE dname$_1$, . . . , dname$_n$
*C

When F and Q are not selected, identifies decks to be written on compile file, new library, and source file.

*COMPILE dname$_a$.dname$_b$
*C

When F and Q are not selected, identifies range of decks (dname$_a$ through dname$_b$) to be written on new program library, compile file, and source file.

*COPY dname, c
*CY

This form, used during insertion, reads card c from named deck and inserts it.

*COPY dname, c$_a$, c$_b$
*CY

This form, used during insertion, reads cards c$_a$ through c$_b$ and inserts them.

*COPY dname, c_a, c_b, file
*CY

This form of COPY is not restricted to insertion. It is not a correction directive. This directive copies cards c_a through c_b from the named deck (dname) onto the named file.

*CWEOR level
*CW

Conditionally writes end of record (section) or end of file (partition). For NOS 1.0 and NOS/BE 1.0, if level is 0-14, decimal, UPDATE writes an end of record of the specified level. For SCOPE 2, if level is 0-14 UPDATE writes an end of section. If level is 15 or greater UPDATE writes an end of file (end of partition for SCOPE 2).

*DECK dname
*DK

Introduces new source deck of the specified name. Name is 1-9 of the following characters:

A-Z 0-9 + - * / ( ) $ =

*DECLARE dname
*DC

Protects decks other than the named deck from being inadvertently altered. The restriction remains in effect until next DECLARE directive. Omitting dname nullifies a previous DECLARE.

*DEFINE $name_1, \ldots, name_n$
*DF

Defines names for use by IF.

*DELETE c
*D

Deletes specified card and optionally inserts text following.

*DELETE c_a, c_b
*D

Deletes cards c_a through c_b and optionally inserts text following.

*DO idname$_1$,...,idname$_n$

Temporarily rescinds YANK and SELYANK for named correction sets.

*DONT idname$_1$,...,idname$_n$
*DT

Does not rescind YANK and SELYANK for named correction sets.

*END

Provided for compatibility with SCOPE EDITSYM program; ignored by UPDATE

*ENDIF
*EI

Used to indicate end of conditional text when num parameter is omitted from IF directive.

*ENDTEXT
*ET

Indicates end of sequence of text cards that begin with TEXT directive.

*IDENT idname,p_1,p_2,...,p_n
*ID

Identifies new correction set and optionally specifies bias for sequence numbers and/or conditions for including correction set. idname is 1-9 of the following characters:

A-Z 0-9 + - * / ( ) $ =

p_i are any number of the following:

B=num	Bias added to sequence numbers.
K=ident	Specifies ident required before this set can be incorporated.
U=ident	Specifies ident that must not be known if this set is to be incorporated.

*IF type,name,num

*IF -type,name,num

Conditionally writes text. Type indicates type of name. Minus sign preceding type indicates name must not be known. num, if included, specifies number of lines; otherwise, IF is terminated by ENDIF.

Type	Name
DECK	Name is a deck name.
IDENT	Name is a correction set identifier.
DEF	Name is defined through DEFINE directive.

*INSERT c
*I

Inserts cards following this directive after the specified card (c).

*LIMIT n
*LT

Sets new line limit (decimal) for list output.

*LIST
*L

Re-enables list option 4 previously disabled through NOLIST.

*MOVE $dname_1$,$dname_2$
*M

Moves $dname_1$ from its position on the old program library to after $dname_2$ on the new program library.

*NOABBREV
*NA

Turns off the check for abbreviated forms of directives. When NOABBREV is in effect, abbreviated names for directives are recognized as text.

*NOLIST
*NL

Disables list option 4, the list of cards in the input stream. The option is disabled until a LIST directive is encountered.

*PULLMOD idname$_1$,...,idname$_n$
*PM

Recreates named correction sets.

*PURDECK dname$_1$,...,dname$_n$
*PD

Purges all cards in named decks permanently.

*PURDECK dname$_a$.dname$_b$
*PD

Purges all cards in decks dname$_a$ through dname$_b$ permanently.

*PURGE idname$_1$,...,idname$_n$
*P

Purges named correction sets permanently.

*PURGE idname$_a$.idname$_b$
*P

Purges correction sets idname$_a$ through idname$_b$ permanently.

PURGE idname,
*P

Purges correction sets idname and all correction sets introduced after it.

*READ file
*RD

Temporarily stops reading input file and reads insertion text and directives from specified file starting at its current position until an end of record.†

*RESTORE c
*R

Reactivates previously deleted card (c).

*RESTORE c$_a$,c$_b$
*R

Reactivates previously deleted cards in range c$_a$ through c$_b$.

*REWIND file
*RW

Rewinds named file.

*SELPURGE dname$_1$.idname$_1$,...,dname$_n$.idname$_n$
*SP

Causes purging of all cards in a specified deck dname belonging to a specified correction set idname.

† End of section for SCOPE 2.

*SELYANK $dname_1.idname_1,...,dname_n.idname_n$
*SY

Causes yanking of all cards in a specified deck dname belonging to a specified correction set idname.

*SEQUENCE $dname_1,...,dname_n$
*S

Resequences named decks and purges inactive cards from them.

*SEQUENCE $dname_a.dname_b$
*S

Resequences decks $dname_a$ through $dname_b$ and purges inactive cards from them.

*SKIP file
*SK

Skips 1 logical record† forward on file.

*SKIP file,n
*SK

Skips n logical records† forward on file.

*TEXT
*T

The text following is not to be scanned for directives until an ENDTEXT is encountered.

*WEOR level
*W

Writes end of record (section) or end of file (partition). For NOS 1.0 and NOS/BE 1.0, if level is 0-14 decimal, UPDATE writes an end of record of the specified level. For SCOPE 2, if level is 0-14, UPDATE writes end of section. If level is 15 or greater, UPDATE writes an end of file (end of partition for SCOPE 2).

*YANK $idname_1,...,idname_n$
*Y

Temporarily removes effects of named correction sets.

*YANK $idname_a.idname_b$
*Y

Temporarily removes effects of correction sets $idname_a$ through $idname_b$.

*YANKDECK $dname_1,...,dname_n$
*YD

Temporarily removes effects of named decks.

*/ comments

Lists comments.

† Section(s) for SCOPE 2.

MODES

NORMAL SELECTIVE MODE

During a normal UPDATE run (F and Q are not selected on the UPDATE control card), UPDATE writes on the compile file all decks specified on COMPILE directives as well as all decks corrected during the run. COMPILE causes a deck to be written regardless of whether it was corrected or not. In normal mode, decks are written on the compile file in the sequence encountered on the old program library. By selecting the K parameter on the UPDATE control card, the user causes decks to be written in the sequence specified on COMPILE directives followed by any decks for which corrections were made that were not mentioned on COMPILE directives.

FULL UPDATE MODE

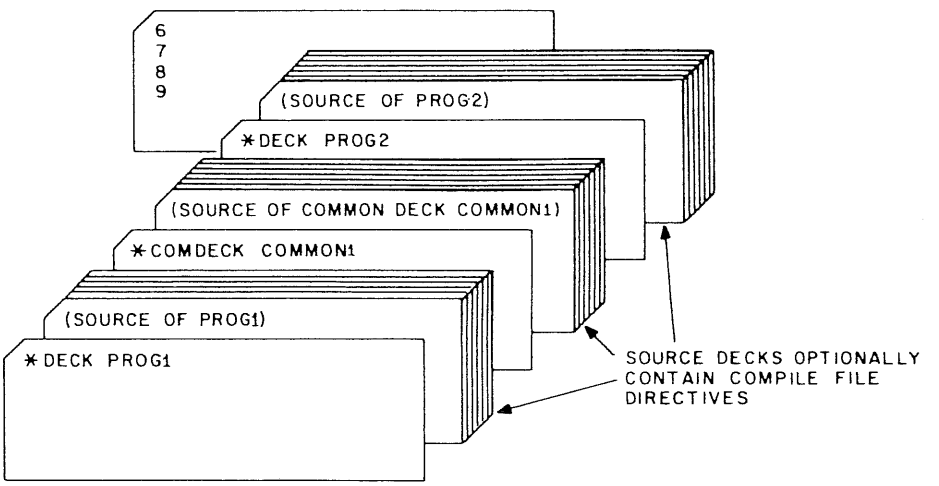
During a full update run (F selected on UPDATE control card), UPDATE ignores COMPILE directives. It updates all decks in the sequence encountered on the library. This sequence cannot be changed through the K option.

QUICK MODE

During a quick UPDATE run (Q selected), only decks specified on COMPILE directives and called common decks are written on the compile file. These decks are written in the sequence encountered on the program library unless the K option has been specified, in which case, they are written in the order specified on the COMPILE directives.

FILES

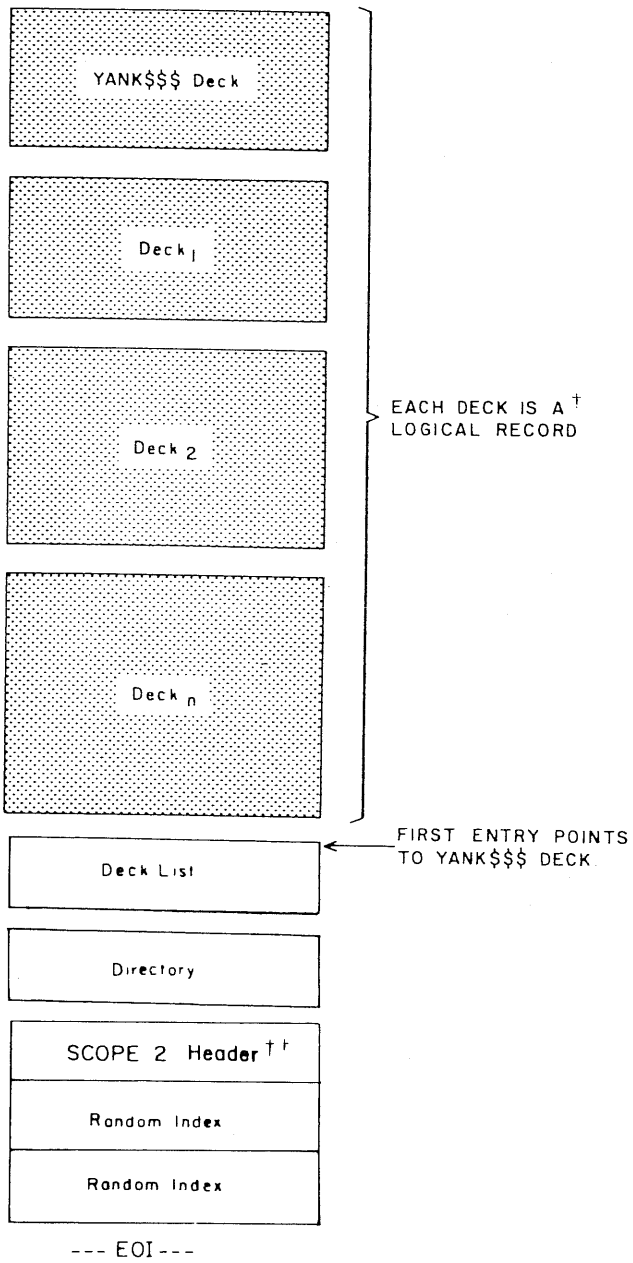
SOURCE DECKS PREPARED BY USER AS INPUT TO UPDATE



SOURCE FILE GENERATED BY UPDATE

The source file contains a copy of all active DECK, COMDECK, TEXT, ENDTEXT, WEOR, CWEOR, CALL, IF, and ENDIF directives and all active cards within each deck. When T is selected on the UPDATE statement, common decks are not written on the source file. The source file is optional output from UPDATE (through use of the S or T options on the UPDATE control card) and, once created, can be used as source input on subsequent UPDATE runs. The source file is a coded file that contains 80-column images.

RANDOM FORMAT

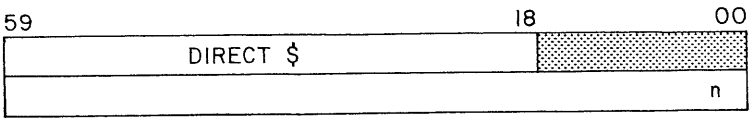


† For SCOPE 2, each deck is a section
†† Header applies to SCOPE 2 only

Under NOS/BE 1.0, the random index record can be obtained by issuing an OPEN function on the file.

Under SCOPE 2, however, to obtain random index, skip to EOI, backspace one record, and read.

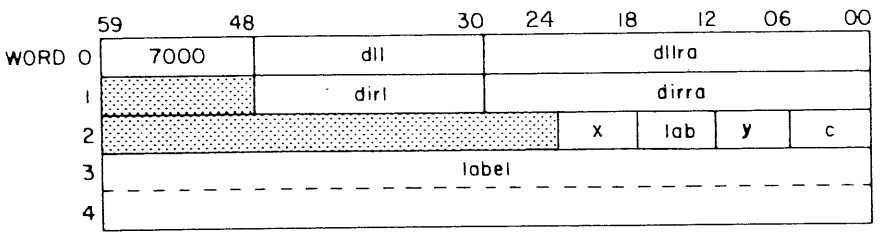
SCOPE 2 Random Index Header



n Number of words in random index; format is shown below

Random Index Format

The system generates two copies of the random index because a CLOSE causes the index to be written again.



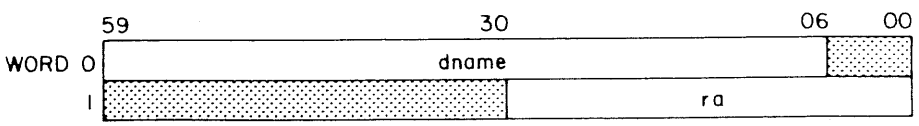
Word	Bits	Field	Description
0	59-48	7000	Identifies random directory record
0	47-30	dll	Length of the deck list in words
0	29-00	dllra	Random address of first word of deck list
1	59-48	none	Unused
1	47-30	dirl	Length of directory in words

Word	Bits	Field	Description
1	29-00	dirra	Random address of first word of directory
2	59-24	none	Unused
2	23-18	x	IP•CSET character set identifier; 3=63 character set, 4=64 character set.
2	17-12	lab	Label flag; if nonzero, words 3 and 4 are present in random directory and contain tape label. SCOPE 2 does not recognize tape labels.
2	11-06	y	Type of program library; must be Y or null to indicate 64-character set.
2	05-00	c	Master control character used when the program library was generated.

Deck List Format

The deck list contains a two-word entry for each deck on the library. The first entry points so the YANK$$$ deck.

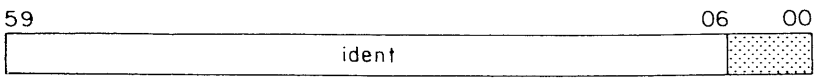
Each entry has the following format:



Word	Bits	Field	Description
0	59-06	dname	1-9 alphanumeric character deck name obtained from DECK or COMDECK directive when deck was placed on library. The first dname is YANK$$$.
0	05-00	none	Unused
1	59-30	none	Unused
1	29-00	ra	Random address of first word of compressed text for the deck.

Directory Format

The directory is a table that contains one entry for each DECK, COMDECK and IDENT that has ever been used for this library. Directory entries each consist of one word containing the 1-9 character identifier in display code, left justified with zero fill.



For a purged ident, bits 59-06 are zeroed and bits 05-00 contain a 20_8.

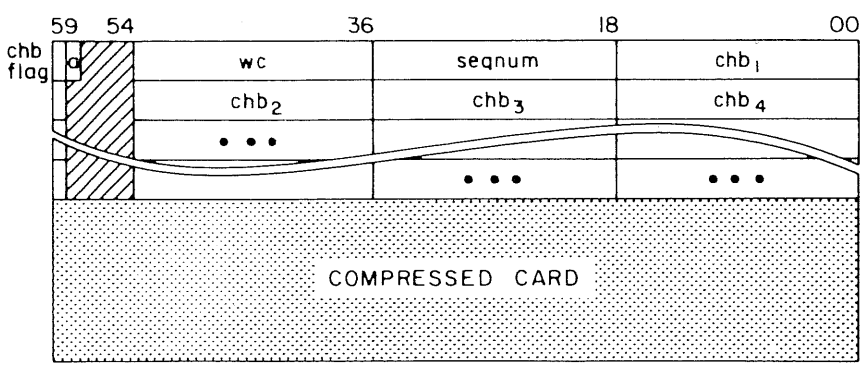
YANK$$$ Deck

The YANK$$$ deck is automatically created on a creation run as the first deck on the program library. On correction runs, UPDATE inserts YANK, SELYANK, YANKDECK, and DEFINE directives into the YANK$$$ deck. These directives acquire identification and sequence information from the correction set from which they originate.

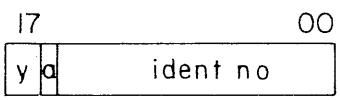
The YANK$$$ deck cannot be yanked or purged. However, cards in the deck can be deleted, yanked or purged.

Compressed Text Format

Text is an indefinite number of words that contain a correction history and the compressed image of each card in the deck. Information for each card is in the following format:



Bits	Field	Description
59	chb flag	Indicates the last word containing correction history bytes. 0 Not last word 1 Last word
58	a	Activity bit for the card. 0 Card is inactive 1 Card is active
57-54	none	Unused
53-36	wc	Number of words of compressed text for this card.
35-18	seqnum	Sequence number of card (octal) according to position in deck or correction set identified by chb_1.
17-00 and subsequent 18-bit bytes	chb_i	Correction history byte. UPDATE creates a byte for each correction set that changes the status of the card. The format of chb_i is:



y	Yank bit 0 Card not yanked 1 Card has been yanked
a	Activity bit 0 Correction set deactivated the card 1 Correction set activated the card
identno	Index to the entry in the directory that contains the name of the correction set or deck that introduced the card or changed the card status.

Compressed card

The compressed image of the card in display code. Single and double spaces are unaltered. Three or more embedded spaces are replaced in the image as follows:

3 spaces replaced by 0002
4 spaces replaced by 0003
5 spaces replaced by 0004

. . .
. . .
. . .

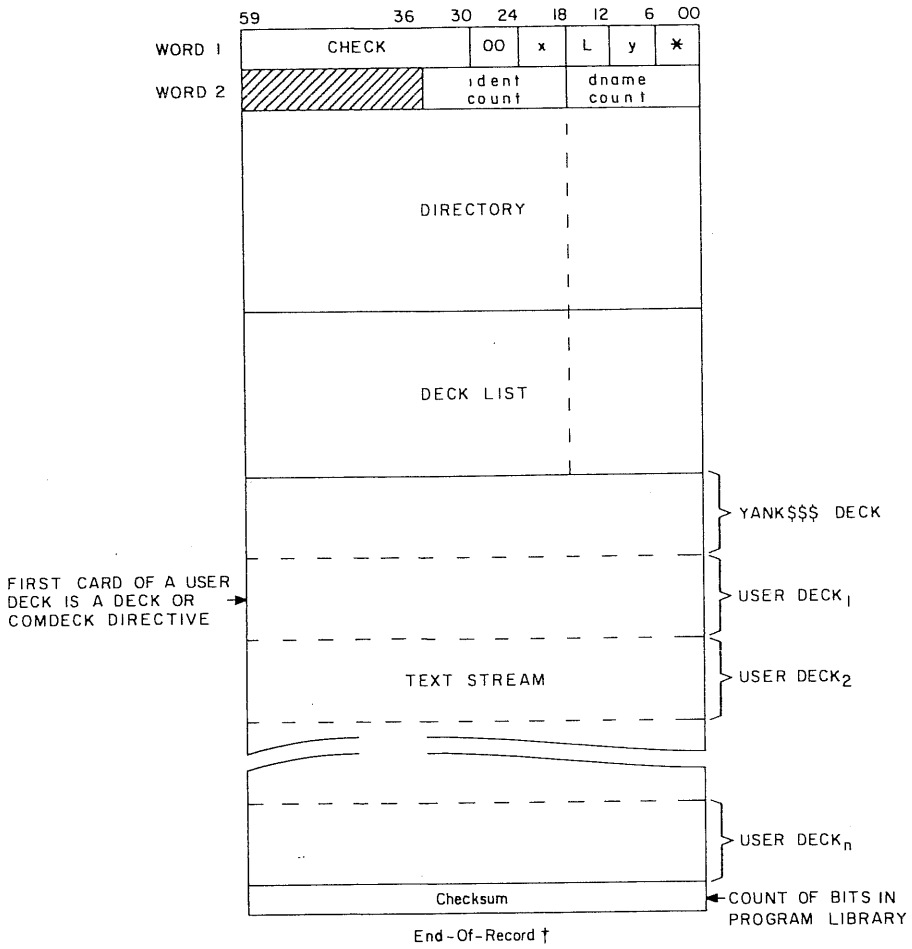
64 spaces replaced by 0077_8
65 spaces replaced by 007755_8
66 spaces replaced by 00775555_8
67 spaces replaced by 00770002_8, etc.

When a space is the first character of a line, it is always represented as 55_8 even when it is part of a string of spaces.

Trailing spaces are not considered as embedded and are not included in the card image. A 4-digit octal code 0000 or word count (wc) reached marks the end of the card. This is conditional on the CHAR64 option.

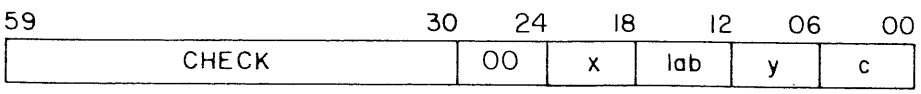
When the full-character set installation option is assembled, a byte of 0001 represents a colon.

NEW SEQUENTIAL FORMAT



† End of section if record type is W. Each W record is 512 words.

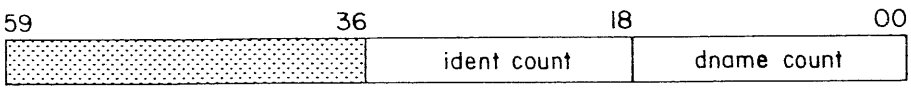
Word One



Bits	Field	Description
59-30	CHECK	Identifies the file as being a new format sequential file. This field contains the word CHECK in display code.
29-24	none	Unused; zero
23-18	x	IP•CSET character set identifier; 3=63 character set, 4=64 character set.
17-12	lab	Presence of L indicates labeled tape. Null indicates unlabeled tape. SCOPE 2 does not recognize tape labels.
11-06	y	Indicates whether or not this library was generated using full 64-character set. Y Yes; colon is supported 00 No; colon is not supported
05-00	c	Indicates master control character in use when this library was created. * First character of directives is asterisk, the conventional master control character. Other First character of directives is character indicated. On a correction run, if the master control character specified on the UPDATE card does not match this character, UPDATE changes the character to c. On a merge run, if the control characters for the two libraries do not match, the run is aborted.

Word Two

The second word of the program library file is composed of two binary counts: the count of the identifiers in the directory, and the count of the deck names in the deck list.



Directory Format

Entries in the directory are in the same format as for a random library.

Deck List Format

The deck list is a table consisting of one entry for each deck on the program library. Each entry consists of one word containing the deck name in display code, left justified, with zero fill. Decks are listed in the order in which they were introduced on the library.

Text Stream

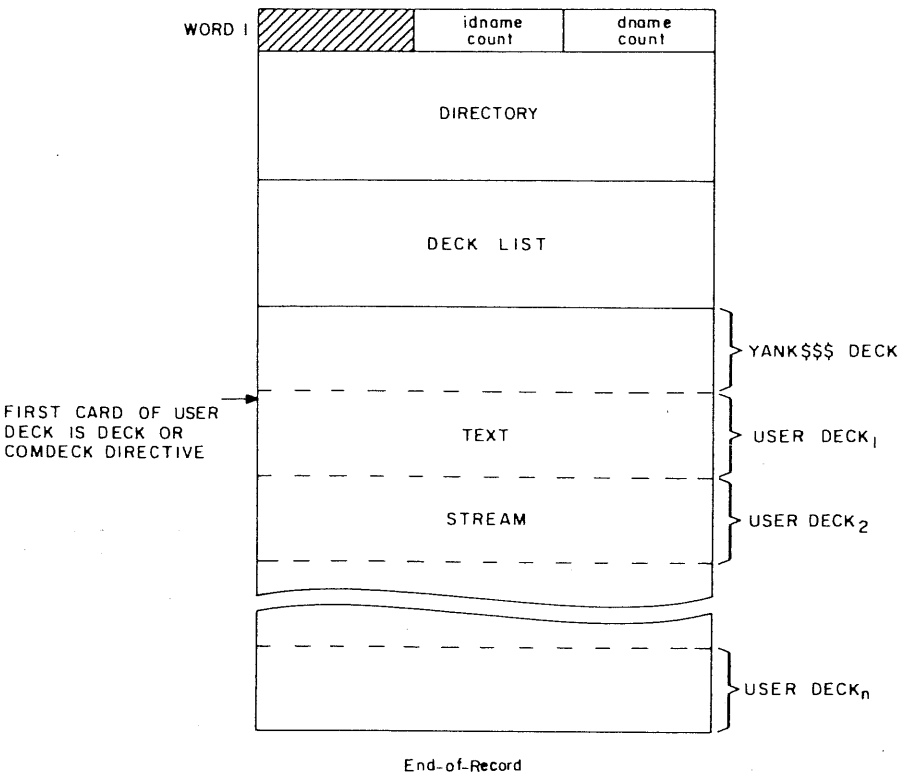
The text stream immediately follows the deck list. Compressed text up to the first DECK or COMDECK directive comprises the YANK$$$ deck. Each subsequent deck begins with a DECK or COMDECK directive and consists of each card up to the next DECK or COMDECK directive or the end-of-record.

Compressed Text Format

Text on the sequential library is compressed in the same manner as on a random library.

OLD SEQUENTIAL FORMAT

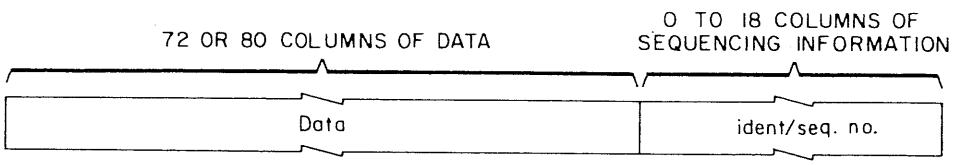
UPDATE optionally accepts old library files in the old (pre-SCOPE 3.3) UPDATE sequential format. These libraries resemble the new sequential format but do not contain the CHECK word or checksum and the text format and chb's are different. Word two on the new format is the same as word one on the old format. UPDATE does not generate this obsolete sequential format.



Expanded Format

The expanded compile file format for each card consists of 72 or 80 columns of data followed by 0 to 18 columns of sequence information. The maximum size of a card image is 90 columns.

Expanded card image:

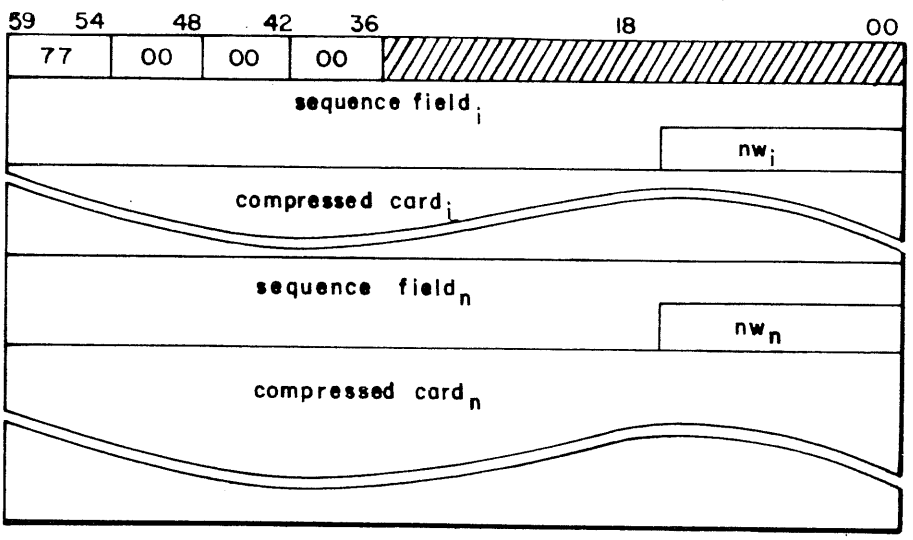


80- or 90- Column Card Image

If the 80 (90) character card image on the compile file has two blanks as the last two characters, these are converted to a 0000 line terminator and the card image is 8 (or 9) words long.

If the last two columns do not contain blanks, a word containing 8 blanks and a zero byte line terminator are added, thus making the card image 9 (or 10) words long. This same procedure is used for creation of the source file.

Compressed Format



sequence field$_1$	17 characters comprising card columns 74-90. Column 73 is always blank.
nw_i	Binary number of words in compressed card$_i$.
compressed card$_i$	Columns 1-72 of a COMPASS source card in compressed form. That is, each 00 character is replaced by the 12-bit value 0001, and three or more consecutive blanks (to a maximum of 64) are replaced by a 12-bit value 0002 through 0077_8. A single blank is represented in display code (55_8); two consecutive blanks are represented by the 12-bit value 5555_8. If the last word is not full, it is padded on the right with binary zeros. Because word count nw_i is present, an extra all-zero word is not required to guarantee 12 zero bits.

UPDATE STATEMENT

UPDATE ($p_1, p_2, \ldots, p_n$)

p_i option
option=filename
option=0 (valid for modes C and L only)

Option	Significance
A - Sequential-to-random copy	
omitted	No special mode
A	When this mode is selected UPDATE copies a sequential old program library to a random new program library. It performs no other UPDATE operations. The only other control card options that can be used, other than those specifying files, are * and /. For SCOPE 2, the new program library cannot be blocked.
B - Random-to-sequential copy	
omitted	No special mode
B	When this mode is selected, UPDATE copies a random old program library to a sequential new program library. It performs no other UPDATE operations. The only other control card options that can be used, other than those specifying files, are * and /.
C - Compile file output	
omitted or C	Compile output decks will be written on file COMPILE; contents are determined by type of UPDATE (F, Q, or normal).
C=filename	UPDATE writes compile output decks on named file; contents are determined by type of UPDATE (F, Q, or normal).

Option		Significance
	C=PUNCH	This is a special form of C=filename. UPDATE writes compile output decks on PUNCH file. Decks are punched according to type of UPDATE (F, Q, or normal). This option also causes the 8 and D modes to be selected.
	C=0	No compile output
D -	Data width	
	omitted	Compile output has 72 columns for data
	D	Compile output has 80 columns for data
E -	Edit; provides a means of cleaning up old program libraries	
	omitted	The old program library is not edited.
	E	UPDATE rearranges the directory to reflect the actual order of decks on the program library. During editing, UPDATE purges any cards other than YANK, SELYANK, and YANKDECK from the YANK$$$ deck.
F -	Full update	
	omitted	F and Q omitted is the normal (selective) UPDATE mode. All regular decks and common decks are processed.
	F	Source and compile files, if specified, contain all active decks in old program library sequence. The contents of the new program library are the same as if F were not specified.

Option	Significance
G - Generate separate PULLMOD output file	
omitted	Output from PULLMOD cards is appended to source file defined by S or T option or to the SOURCE file.
G=filename	Output from PULLMOD cards is written on named file. Any rewind option applying to the source file also applies to this file.
H - Header change	
omitted or H	Character set indicated in old program library header.
H=3	63-character set.
H=4	64-character set.
I - Input	
omitted or I	Input is on job INPUT file.
I=filename	Input comprises next record on named file.
K - COMPILE card sequence (takes precedence over C mode)	
omitted	Output determined by C option.
K	Compile output decks to be written on file COMPILE in COMPILE directive sequence.
K=filename	Compile output decks to be written on named file in COMPILE directive sequence.
L - List options	
omitted	List options A, 1, and 2 are automatically selected on a creation run. Options A, 1, 2, 3, and 4 are automatically selected on a correction run. Options A1 are selected for A or B mode.
L=$c_1c_2\ldots c_n$	Each character in string selects an option.

Option	Significance
L (cont'd)	

Option	Significance
A	Listing of DECK names, IDENT names, COMDECK directives, decks on compile file and known definitions
F	All selections other than 0
0	No listing; zero in a string is equivalent to L=0
1	Cards in error and associated error messages
2	Directives
3	Cards that changed status
4	Input stream
5	All active compile file directives
6	Number of active and inactive cards by deck name and correction set identifier
7	All active cards

Option	Significance
8	All inactive cards
9	Correction history of all cards listed as a result of list options 5, 7, and 8.

M - Merge output; allows two program libraries to be merged and written onto the new program library (see N option).

omitted	No merge file
M	Second old program library on file MERGE
M=filename	Second old program library on named file

N - New program library output

For SCOPE 2, the library is random if the file specified for N is record type W and the UPDATE W option is not specified; the new program library cannot be a blocked file when performing an UPDATE (A). Under NOS 1.0 and NOS/BE 1.0, the new program library is written in random format if the file specified by the N option is on mass storage and the W option is not specified. It is written in sequential format if the file specified by N is not on mass storage (i.e., on magnetic tape or card punch) or if the W option is explicitly specified.

N	New program library to be written on file NEWPL.
N=filename	New program library to be written on named file.
omitted	UPDATE does not generate a new program library.

O - List output file

omitted or O	List output is written on job OUTPUT file. This file is automatically printed.
O=filename	List output is written on named file.

Option	Significance
P - Old program library; ignored on creation run.	
omitted or P	Old program library on file OLDPL.
P=filename	Old program library on named file
Q - Quick update (takes precedence over F)	
omitted	F and Q omitted is the normal (selective) UPDATE mode. See F omitted.
Q	Only decks specified on COMPILE directives and decks added through ADDFILE directives are processed.
R - Rewind files	
omitted	The old and new program libraries, the compile file, and the source file are rewound before and after the UPDATE run.
R	No rewinds are issued for the program libraries, compile file, or source file.
$R=c_1c_2 \cdots c_n$	Each character in string indicates a file to be rewound before and after the UPDATE run.

c	File
C	Compile
N	New program library
P	Old program library and merge library
S	Source and PULLMOD

S - Source output; the contents of the source file are determined by the mode in which UPDATE is operating, the decks named on COMPILE directives, and the type of old program library in use (random or sequential).

1. If Q is not selected (regardless of F), the source file contains all cards required to recreate the library. This recreated library is resequenced.

Option	Significance
2.	If Q is selected and the old program library is sequential, decks written on the source file are those named on COMPILE directives and common decks they call. If Q is selected and the old program library is random, decks written on the source file are those named on COMPILE directives.
omitted	UPDATE does not generate a source output file unless the source output is specified by T.
S	Source output written on file SOURCE
S=filename	Source output written on named file.

T - Source output excluding common decks (takes precedence over S)

omitted	No source output unless source output specified by S.
T	Source output excluding common decks on file SOURCE.
T=filename	Source output excluding common decks on named file.

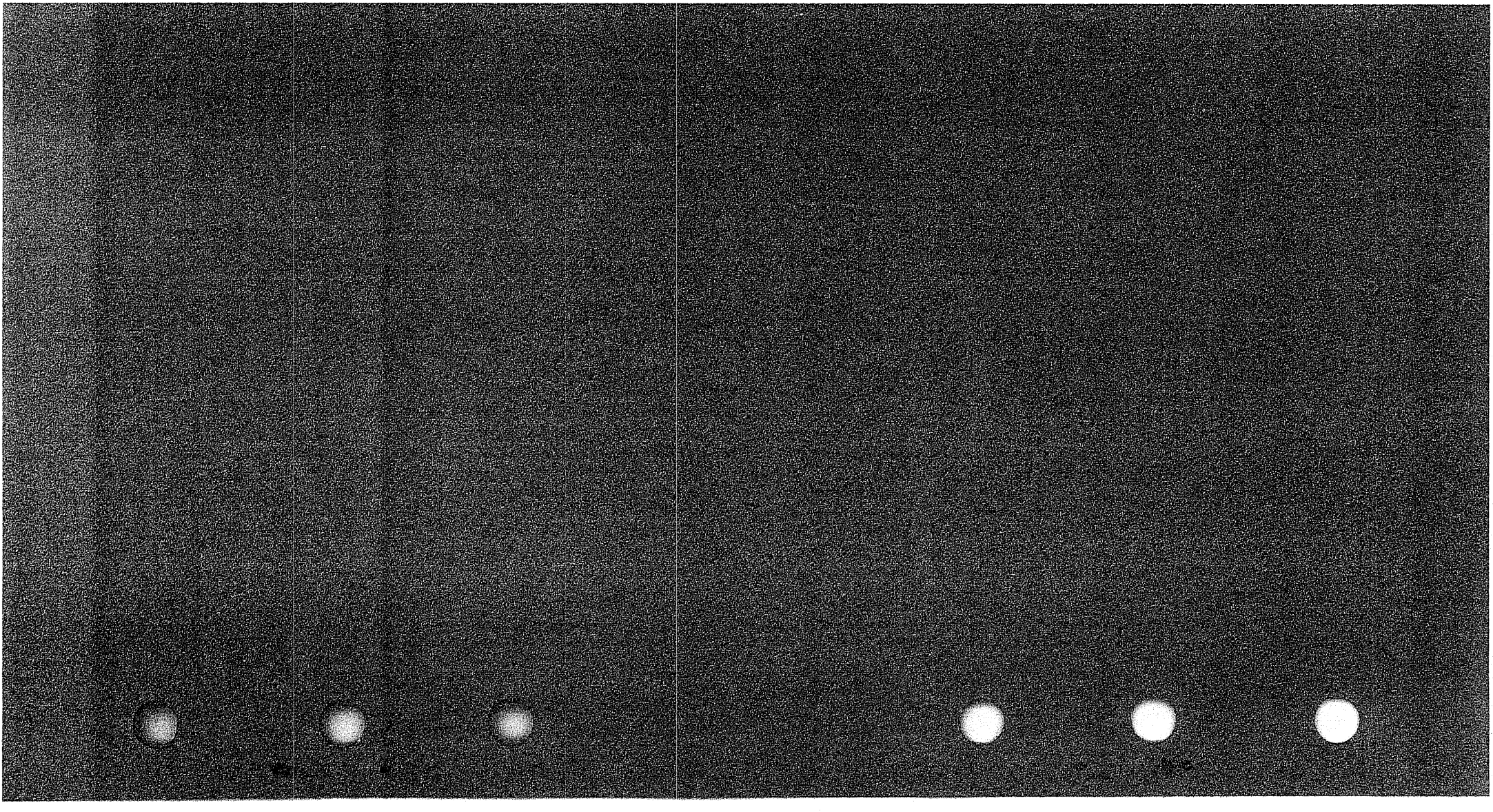
U - Debug mode

omitted	UPDATE execution terminates upon encountering a fatal error.
U	UPDATE execution is not terminated by normally fatal errors.

W - Sequential new program library

omitted	The new program library (see N option) will be determined by characteristics of file specified by N.
W	The new program library (see N option) will be a sequential file.

Option		Significance
X - Compressed compile file		
	omitted	Compile file is not in compressed format.
	X	Compile file is in compressed format.
Z - Compressed input file		
	omitted	The input file (see I option) is a normal, coded file.
	Z	The input file (see I option) is assumed to be in PCS compressed format. This parameter applies to the directives input file only; it does not apply to files specified by READ directives.
8 - 80-column output on compile file		
	omitted	Compile file output is composed of 90-column card images.
	8	Compile file output is composed of 80-column card images.
* - Master control character		
	omitted	The master control character is *.
	*=char	char can be any character having a display code octal value in the range 01-54 except for 51 and 52 (the open and close parentheses).
/ - Comment control character		
	omitted	The comment control character is /.
	/=char	char can be A through Z, 0 through 9 or + - * / $ =



CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440

SALES OFFICES AND SERVICE CENTERS
IN MAJOR CITIES
THROUGHOUT THE WORLD

PRINTED IN U.S.A.

CONTROL DATA CORPORATION