60498300

CONTROL DATA

QUERY UPDATE VERSION 3 REFERENCE MANUAL

CDC® OPERATING SYSTEMS:
NOS 2
NOS/BE 1

REVISION RECORD

Revision	Description
A (11/01/75)	Manual released.
B (03/05/76)	This revision documents feature DM157, a new directive and a new parameter in the QU control statement. Minor technical corrections and recently added diagnostics are included.
C (12/06/76)	This revision documents Version 3.1 of Query Update. Feature CP144 includes the relational data base facility, read-only and alternate key update modes of operation, modifications to the USE, RETURN, DESCRIBE, EXHIBIT, ERASE, PREVIEW, COMPILE, and SORT directives, the VIA directive, the SCAN function, and the CURRENT-ANY register. Numerous technical corrections clarify and update directive descriptions. Released at PSR level 439.
D (03/31/78)	This revision documents Version 3.2 of Query Update, incorporating minor maintenance changes and reorganization of the manual content. The entire manual has been reprinted.
E (07/15/79)	This revision incorporates minor maintenance changes.
F (10/31/80)	This revision documents the Query Update ability to access a data base through CDCS. The new directives INVOKE, STORE, REMOVE, and MODIFY (also used in the IMF data base management system) replace the USE, INSERT, DELETE, and UPDATE directives, respectively. The new directives ACCESS and RECOVERY have been added. Released at PSR level 528.
G (05/14/82)	This revision documents Query Update Version 3.4 at PSR level 564. Version 3.4 can be used under NOS 2. Enhancements were made to the DISPLAY UPON, EXTRACT, and IF directives. A FOR DATABASE option was added to the CREATE, INVOKE, and VERSION directives. Numerous technical changes were made to update some directives. Refer to the Preface for specifics on new features and the operating systems to which they apply. This is a complete reprint.
H (02/20/84)	This revision incorporates minor technical clarifications and documents recently added error diagnostics. Released at PSR level 599.
J (06/06/86)	This revision documents Query Update Version 3.4 at PSR level 647. References to NOS 1 and FORTRAN 4 are removed, and Sort/Merge 4 is changed to Sort/Merge 5. The revision corrects user library support, clarifies use of IFSAME, adds two directives, DIRECTORY and LOOKUP, and includes miscellaneous technical changes.

REVISION LETTERS I, O, Q, AND X ARE NOT USED

Address comments concerning this manual to:

CONTROL DATA CORPORATION
Publications and Graphics Division
P. O. Box 3492
SUNNYVALE, CALIFORNIA 94088-3492

or use Comment Sheet in the back of this manual

LIST OF EFFECTIVE PAGES

New features, as well as changes, deletions, and additions to information in this manual are indicated by bars in the margins or by a dot near the page number if the entire page is affected. A bar by the page number indicates pagination rather than content has changed.

Page	Revision	Page	Revision
Front Cover	-	6-3	H
Inside Front Cover	J	6-4	G
Title Page	-	6-5	H
ii	J	6-6	G
iii/iv	J	A-1	F
v	J	A-2	F
vi	J	A-3	E
vii/viii	J	A-4	E
ix	G	B-1	G
x	G	B-2	F
xi/xii	G	B-3	F
xiii	G	B-4	G
1-1 thru 1-6	G	B-5	G
1-7	H	B-6 thru B-13	F
1-8	H	B-14	G
1-9	G	B-15 thru B-17	F
1-10	J	B-18	G
2-1 thru 2-4	G	B-19	F
3-1 thru 3-5	G	B-20	G
3-6	H	B-21	G
3-7 thru 3-10	G	B-22	F
4-1 thru 4-10	G	B-23	F
4-11	J	B-24 thru B-26	G
4-12	J	B-27	F
4-12.1/4-12.2	J	B-28	G
4-13	H	B-29	G
4-14	G	B-30	J
4-15	J	B-31	F
4-16	J	B-32 thru B-34	G
4-17	G	B-35	J
4-18	G	B-36	J
4-19	J	B-36.1/B-36.2	J
4-20	J	B-37 thru B-39	G
4-21	G	B-40	H
4-22	G	B-40.1/B-40.2	H
4-23	J	B-41 thru B-48	G
4-24	H	B-49	H
4-24.1/4-24.2	H	C-1	J
4-25	G	C-2 thru C-4	H
4-26	G	D-1 thru D-12	J
4-27	H	E-1	J
4-28	G	F-1 thru F-5	G
4-29	J	G-1	E
4-30	J	G-2	E
4-30.1/4-30.2	J	G-3 thru G-5	D
4-31	G	G-6	E
4-32	H	G-7 thru G-9	D
4-33 thru 4-37	G	G-10	E
4-38	H	H-1 thru H-9	G
4-39 thru 4-47	G	I-1	G
4-48	J	I-2	J
4-49	G	I-3	J
4-50	G	J-1	F
5-1 thru 5-3	G	Index-1	G
5-4	D	Index-2	J
5-5 thru 5-7	G	Index-3	J
5-8	D	Index-4 thru -6	H
6-1	J	Comment Sheet/Mailer	J
6-2	F	Back Cover	-

PREFACE

This manual describes the Query Update language Version 3.4, which is designed for data storage and retrieval operations. Query Update Version 3.4 operates under control of the following operating systems:

NOS 2 for the CDC® CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; and 6000 Computer Systems

NOS/BE 1 for the CDC® CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; and 6000 Computer Systems

The Query Update reference manual is designed for both the inexperienced data processing user who prepares simple reports, and the experienced programmer who performs complex file manipulation operations.

Query Update accepts and interprets user-oriented directives. Although Query Update was designed for interactive use, directives can be submitted through either a terminal or a deck of punched cards. Individual terminal operation is beyond the scope of this manual; the user should reference the appropriate manual for terminal operation.

Three data base interfaces exist in which Query Update can be used: CYBER Database Control System (CDCS), CYBER Record Manager (CRM), and Information Management Facility (IMF). This manual presents an overview of these interfaces and documents specific interface information in the following ways:

CRM	Complete documentation; discussions on using Query Update are in terms of using this interface.
CDCS	Syntax and brief explanation of syntax are included.
IMF	Syntax and brief explanation of syntax are included.

Detailed information pertaining to the CDCS and IMF interfaces is contained in the CDCS and IMF application programming reference manuals.

Related material is contained in the listed publications. The publications are listed alphabetically within groupings that indicate relative importance to readers of this manual.

The NOS manual abstracts and the NOS/BE manual abstracts are instant-sized manuals containing brief descriptions of the contents and intended audience of all NOS and NOS product set manuals, and NOS/BE and NOS/BE product set manuals, respectively. The abstracts manuals can be useful in determining which manuals are of greatest interest to a particular reader.

The Software Publications Release History serves as a guide in determining which revision level of software documentation corresponds to the Programming System Report (PSR) level of installed site software.

The following manuals are of primary interest:

Publication	Publication Number	NOS 2	NOS/BE 1
DMS-170 CYBER Database Control System Application Programming Reference Manual	60485300	X	X
Information Management Facility Version 2 Reference Manual	60484600	X	
NOS Version 2 Reference Set Volume 3, System Commands	60459680	X	
NOS/BE Version 1 Reference Manual	60493800		X
Query Update Version 3 Programmer User´s Guide	60499000		X
Query Update Version 3 User´s Guide For Use With: CYBER Record Manager	60387700		X

The following manuals are of secondary interest:

Publication	Publication Number	NOS 2	NOS/BE 1
CYBER Record Manager Advanced Access Methods Version 2 Reference Manual	60499300	X	X
CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual	60495700	X	X
DMS-170 CYBER Database Control System Version 2 Data Administration Reference Manual	60485200	X	X
DMS-170 CYBER Database Control System Version 2 Reference Manual	60481800		
DMS-170 DDL Version 3 Reference Manual Volume 1, Schema Definition for Use With: COBOL FORTRAN Query Update	60481900		
DMS-170 DDL Version 3 Reference Manual Volume 2, Subschema Definition for CYBER Database Control System Use With: COBOL Query Update	60482000		
DMS-170 FORTRAN Data Base Facility Version 1 Reference Manual	60482200		
DMS-170 Query Update/CYBER Record Manager Data Administration Reference Manual	60482100	X	X

Publication	Publication Number	NOS 2	NOS/BE 1
NOS Version 2 Manual Abstracts	60485500	X	
NOS/BE Version 1 Manual Abstracts	84000470		X
Software Publications Release History	60481000	X	X

CDC manuals can be ordered from Control Data Corporation, Literature and Distribution Services, 308 North Dale Street, St. Paul, Minnesota 55103.

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or parameters.

CONTENTS

APPENDIXES

INDEX

FIGURES

TABLES

NOTATIONS

Each directive is described in terms of a reference format. When more than one specific arrangement is permitted, the format is separated into numbered formats.

NOTATION USED IN REFERENCE FORMATS

UPPERCASE — Uppercase words are reserved words and must appear exactly as shown. Reserved words can be used in directives only as specified in the reference formats.

UNDERLINED UPPERCASE — Underlined uppercase words or parts of words are required when the format in which they appear is used. Any part word including the underlined part up to the full word is a valid abbreviation.

lowercase — Lowercase words are generic terms that represent the words or symbols supplied by the user. When generic terms are repeated in a format, a number is appended to the term for identification.

[ ] — Brackets enclose optional portions of a reference format. All of the format within the brackets can be omitted or included at the user's option. If items are stacked vertically within brackets, only one of the stacked items can be used.

{ } — Braces enclose one item or several vertically stacked items in a reference format. One of the enclosed items must be used. When one item is enclosed in braces and followed by ellipses, the item must be used once and can be repeated at the user's option.

‖ ‖ — Vertical bars enclose two or more vertically stacked items in a reference format when at least one of the enclosed items must be used. Each of the vertically stacked items can be used once.

... — Ellipses immediately follow a pair of brackets or braces to indicate that the enclosed material can be repeated at the user's option. Ellipses also follow dataname, expression, or file name to indicate that the user-supplied element can be repeated.

[‖ ‖] — Vertical bars within brackets enclose two or more vertically stacked items when each of the stacked items can be used once, or omitted. Any items can be written in any order.

Punctuation symbols shown within the formats are required unless enclosed in brackets and specifically noted as optional. In general, commas and semicolons are optional. One or more spaces separate the elements in a directive.

NOTATION USED IN EXAMPLES

↑ — An up arrow indicates the position of an assumed decimal point in an item.

Δ — A delta indicates a space (blank).

$\overset{+}{n}$ — A plus or minus sign above a numeric character indicates an operational sign is stored in combination with the numeric character.

$\boxed{A}\boxed{B}\boxed{C}\boxed{D}$ — Character positions in storage are shown by boxes.

SYSTEM DESCRIPTION 1

Query Update is a nonprocedural interpretive system that enables individuals with varying levels of technical knowledge to perform data storage and retrieval operations. Query Update can be used to produce special-purpose reports either interactively or through the batch-oriented REPORT utility. Query Update can add new records to files, remove records from existing files, extract data from records within files, or change individual fields within records. Query Update can access data either from a collection of files known as a data base or from non-data-base files.

Query Update can be used to access data base files through the following interfaces:

CYBER Record Manager (CRM)

CYBER Database Control System (CDCS)

Information Management Facility (IMF)

Although IMF terminology refers to permanent files containing physical information as an information base, the term data base replaces the term information base in this manual. Also, the term subschema replaces the term external schema in this manual.

CDC offers guidelines for the use of the software described in this manual. These guidelines appear in appendix J. Before using the software described in this manual, the reader is strongly urged to review the content of this appendix. The guidelines recommend use of this software in a manner that reduces the effort required to migrate software application programs to future hardware or software systems.

QUERY UPDATE ORGANIZATION

Query Update is made up of a series of elements that include the following:

Reserved words

Recognized symbols

Punctuation

User-supplied elements such as names, literals, functions, expressions, conditions, and picture specifications

Query Update parallels the COBOL language convention of grouping these elements into sentence instructions called directives. Directives begin with a unique keyword that identifies the operation to be performed. DISPLAY, STORE, REMOVE, and MODIFY are examples of keywords that begin directives. Directives can be submitted to

Compare data content

Perform arithmetic evaluations

Remove, insert, or otherwise modify data

Design and generate reports

The Query Update directives are listed in table 1-1. This table indicates the directives that can be used with each data management interface and the function of the directive. Detailed descriptions of all the directives and their complete formats are given in section 4 and summarized in appendix D.

TABLE 1-1. QUERY UPDATE DIRECTIVES

Directive	CRM	CDCS	IMF	Description
				Data Base Identification
CREATE	X	X	X	Initiates access to an area for initial insertion of data. When used through IMF, terminates processing through the IMF interface and establishes either the CRM or CDCS interface.
INVOKE	X	X	X	Initiates data base access depending on the specified subschema directory or external schema.
USE[†]	X	X	X	Initiates data base access depending on the specified subschema directory. When used through IMF, terminates the IMF interface and establishes either the CRM or CDCS interface.

[†] This directive is available, but its use is not recommended. See appendix H.

TABLE 1-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	CRM	CDCS	IMF	Description
Data Definition				
DEFINE	X	X	X	Establishes temporary data names and storage requirements.
DESCRIBE	X	X	X	Establishes a directory to the contents of a source data file that is not associated with a data base.
SEPARATOR	X	X	X	Defines a character to be used for delimiting nonnumeric literals.
SPECIFY	X	X	X	Establishes a name for convenient reference to a condition.
UNIVERSAL	X	X	X	Establishes a character that marks a character position to be ignored during comparison testing.
Display Output				
DISPLAY	X	X	X	Displays information from a data base, temporary storage, or designated file and can create a directory of the information displayed.
EXTRACT	X	X	X	Creates a subset of information from a data base, temporary storage, or designated file and can create a directory to the subset.
FOLLOW			X	Specifies an access path and the cosets for accessing records.
IF	X	X	X	Presents a test condition to determine to what extent subsequent directives are to be executed.
Report Output				
ALTER	X	X	X	Identifies retained report directives in the catalog for subsequent modification.
BREAK	X	X	X	Indicates situation that causes interruption of the body of the report to insert footings and headings; interruption can occur when data name content changes or stated conditions are met.
COMPILE	X	X	X	Stores report specifications in encoded form on a table file.
DATE	X	X	X	Specifies use and positioning of system-supplied data information.
DETAIL	X	X	X	Determines report line content and positioning of source data fields, literals, and computed values.
ERASE	X	X	X	Removes one or more report specifications from the current catalog. Removes DEFINE items, SPECIFY items, or DESCRIBE lists.
EVALUATE	X	X	X	Selects the working storage data names for which values are to be calculated when a particular report production step occurs.
FOOTING	X	X	X	Provides content and determines line and column positioning for informative footings.
FORMAT	X	X	X	Initiates grouping and retention of directives in the catalog under a report name for reference by other directives.
HEADING	X	X	X	Provides content and determines line and column positioning for informative headings.
MOVE	X	X	X	Places values in temporary data items.
PAGE-NUMBER	X	X	X	Specifies use and positioning of system-supplied page number.

TABLE 1-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	CRM	CDCS	IMF	Description
PAGE-SIZE	X	X	X	Specifies maximum number of vertical lines, horizontal columns, horizontal or vertical sectional page divisions, and multiple copy images.
PREFACE	X	X	X	Causes lines of text or another report to precede the first page of the report generated according to the format specifications.
PREPARE	X	X	X	Initiates execution of report directives.
PREVIEW	X	X	X	Causes sample execution of report directives.
RECAP	X	X	X	States content and positioning of recapitulative information generated at the end of each report page.
SELECT	X	X	X	Indicates alternative DETAIL specifications to be selected when stated conditions are met.
SUMMARY	X	X	X	Causes lines of text or another report to follow the last page of the report generated according to the format specifications.
TABS	X	X	X	Relates tabular references to horizontal column numbers.
TIME	X	X	X	Specifies use and positioning of system-supplied time information.
TITLE	X	X	X	States content and positioning of title to start each page.
				Modification Operations
MODIFY	X	X	X	Modifies data item values of existing records in a data base.
UPDATE†	X	X	X	Identifies and obtains data items to be modified when used through CRM or CDCS; can be used only with temporary items through the IMF interface.
REMOVE	X	X		Removes specific records from a data base.
DELETE†	X	X		Removes specific records from a data base.
STORE	X	X	X	Places a record in a data base.
INSERT†	X	X		Creates a record entry and places it in an area in the data base.
				Catalog Operations
DUPLICATE	X	X	X	Copies recorded sessions or report specifications from one catalog to another.
ERASE	X	X	X	Removes a temporary data name, a recorded directive, a report specification, or a directory for a non-data-base file.
EXHIBIT	X	X	X	Lists information recorded in the current catalog.
FORMAT	X	X	X	Causes the subsequent report specification directives to be retained in the current catalog under a specified report name.
PERFORM	X	X	X	Retrieves and executes transmissions recorded in the current catalog.
PREPARE	X	X	X	Initiates production of a report according to the specifications associated with report name; report name must exist in the current catalog.

†This directive is available, but its use is not recommended. See appendix H.

TABLE 1-1. QUERY UPDATE DIRECTIVES (Contd)

Directive	CRM	CDCS	IMF	Description
RECORDING	X	X	X	Initiates the recording of subsequent transmissions in the current catalog.
VERSION	X	X	X	Attaches a permanent file as the current catalog, or reverts to the default catalog.
				Miscellaneous Operations
ACCESS		X		Specifies the key that allows file access when an access control key is required.
DIAGNOSTIC	X	X	X	Specifies whether or not consecutive duplicate diagnostic messages are to be displayed.
END	X	X	X	Terminates Query Update operations and retures control to the operating system.
EVALUATE	X	X	X	Performs arithmetic operations to compute data name content or a cumulative function result.
EXECUTE	X	X	X	Causes execution of a procedure that is external to Query Update.
EXHIBIT	X	X	X	Lists temporary data names, information about active relations, attributes for a data name, active areas, and limits or special values that Query Update uses in performing operations.
HELP	X	X	X	Presents descriptions of directives or explanations of diagnostic messages.
MOVE	X	X	X	Places values in temporary data names.
NOTE	X	X	X	Allows user comments to be included in transmissions.
OS	X	X	X	Allows the user, in interactive mode, to enter an operating system control statement during a Query Update session.
RECOVERY		X		Establishes a recovery point on the log file.
RETURN	X	X	X	Releases a file, relation, or subschema that is no longer needed by Query Update.
REWIND	X	X	X	Logically positions a nondata base file at the beginning of information.
SORT	X	X	X	Specifies and initiates the resequencing of a source data file.
STOP	X	X	X	Terminates Query Update operations and returns control to the operating system. NOTE: Cannot be used for normal termination when using the NOS interactive facility; END must be used instead.
VERIFY	X	X	X	Specifies data names for terminal display for use with a VETO option or directive.
VETO	X	X	X	Causes a terminal display for data subject to modification or removal.
VIA	X	X		Establishes the relation to be used when an ambiguity exists on a query.

† This directive is available, but its use is not recommended. See appendix H.

INPUT/OUTPUT METHODS

Directives can be submitted to Query Update either in interactive mode through a terminal or in batch mode through an input deck of punched cards.

In either interactive or batch mode, directives can be collected on a catalog file. The catalog file can be submitted to Query Update either during the present Query Update session, or in a future Query Update session.

In both interactive and batch modes, the results of directive operations can be either output to the user terminal, or stored on a file that can subsequently be printed. Results can be printed as an unformatted list or as a fully formatted report.

Query Update input/ouput methods are shown in figure 1-1.

INTERACTIVE MODE

Interactive communication with Query Update consists of the following actions:

Connecting the terminal to the computer.

Requesting Query Update.

Identifying the data base, information base, independent file, or report to be manipulated.

Submitting Query Update directives and reviewing any responses.

Terminating Query Update when tasks are completed.

The appropriate terminal manual should be consulted for detailed operating instructions. The terminal manual should also be consulted to determine the appropriate graphic character representation for the terminal type. All characters indicated in this manual are CDC graphic characters as shown in appendix A. Individual terminals might use other graphics.

Interactive Transmission of Directives

An interactive transmission can consist of a complete directive, or a series of multiple directives. A transmission is executed by Query Update as a unit. A transmission is submitted in a line (or lines) of input that is terminated when the transmit key is struck; for example, the SEND and RETURN keys are transmit keys. A physical transmission can contain a maximum of 150 characters, while a logical transmission can contain a maximum of 1030 characters unless the transmission length parameter (TL) of the Query Update control statement specifies otherwise.

If a transmission cannot be completed on one input line, a continuation character can be entered as the last character of the line. The continuation characters recognized by Query Update are the plus sign (+) and the equal sign (=).

When the last element entered on a line is a complete word (reserved word, name, or literal), a space must be entered before the continuation character, or a space must be entered as the first character in the next line. When a space is not entered on either line, Query Update assumes the word is continued on the next line.

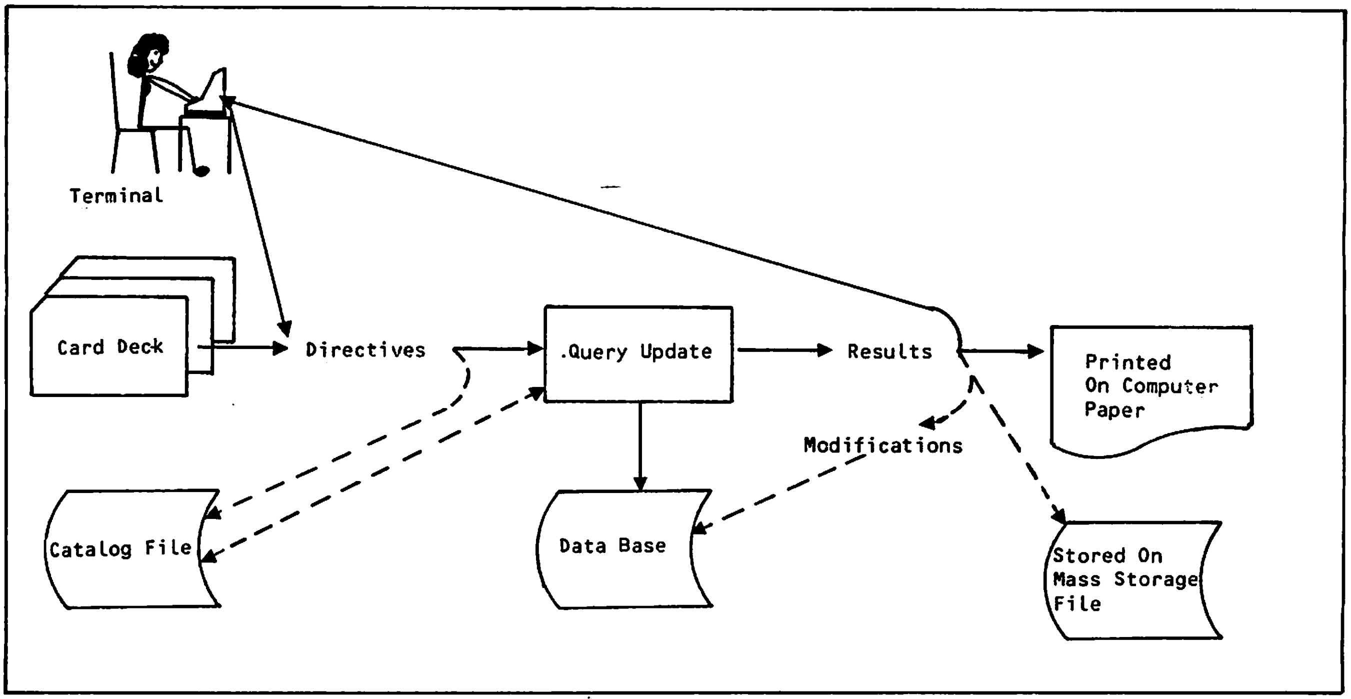


Figure 1-1. Query Update Input/Output Methods

Query Update Interactive Response

Query Update indicates that it is ready to receive a transmission by displaying two hyphens (--). On NOS an additional symbol, the question mark (?), is displayed by the operating system to indicate a request for terminal input.

When Query Update has received a transmission, it either executes or catalogs the transmission. During execution Query Update might send a response to the user. The response can consist of either an error message or a request for information. Information is requested as follows:

> Two greater than signs (>>) indicate that Query Update is requesting data.
>
> Three greater than signs (>>>) indicate that Query Update is requesting permanent file parameters.

BATCH MODE

Batch operations allow the user to submit transmissions to the computer on punched cards. With the exception of the VETO, OS, and VERIFY directives, which are ignored in batch mode, batch input produces the same results as terminal input.

A batch transmission can contain a complete directive, or a series of multiple directives. A transmission is executed by Query Update as a unit. A transmission is submitted on one or more punched cards; only the first 72 columns of each card are read by Query Update. A transmission can contain a maximum of 1030 characters unless the transmission length parameter (TL) of the Query Update control statement specifies otherwise.

If a transmission cannot be completed on one card, it can be continued by punching a continuation character as the last nonblank character of the card or by punching a continuation character in column 1 of the next card. The continuation characters recognized by Query Update are the plus sign (+) and the equal sign (=). When the continuation character is entered as the last nonblank character on a card, it must be preceded by a space unless the directive element punched (reserved word, name, or literal) is being continued on the next card. If the continuation character is entered as the first character on a card, Query Update accepts all 72 columns of the preceding card; therefore, blank columns are included in the transmission as spaces. If a space is not included on either card, Query Update assumes that the word is continued on the next card.

CATALOG OPERATIONS

Query Update can record directives for subsequent execution. These directives are stored on a mass storage file called a catalog. When a Query Update session is initiated, a default catalog is available to record transmissions or report specification directives. The default catalog is not created until recording or report formatting is requested. The catalog can be made permanent at the end of a session. The catalog file can be designated either as the default file (ZZZZZQ2) or as a specific permanent file. In subsequent Query Update sessions, the permanent catalog file is attached through the VERSION directive.

To initialize the default catalog, the user specifies one of the following:

> No VERSION directive, and then RECORDING or FORMAT followed by the directives to be written to the catalog file (assuming no VERSION directive had been specified previously).
>
> VERSION IS DEFAULT, and then RECORDING or FORMAT followed by the directives to be written to the catalog file.

Information can be copied from the current catalog to the default catalog by using the DUPLICATE directive. The default catalog is not returned by the VERSION directive; therefore, it is always available for duplication.

The recording of directives in a catalog can be initiated or terminated by the RECORDING directive. Recorded directives can then be executed using the PERFORM directive.

Information recorded in a catalog can be displayed by using the EXHIBIT directive. Information can be eliminated by using the ERASE directive.

The catalog file is used whenever a Query Update user records a session, performs a session, specifies report directives with ALTER or FORMAT, or prepares a report. A Query Update catalog file can be used through either CRM, CDCS, or IMF. There are two methods of access: CRM catalog mode and CDCS catalog mode.

CRM CATALOG MODE

CRM catalog mode is always available to provide directives when Query Update is used through the CRM, CDCS, or IMF interface.

CRM catalog mode is also available when Query Update is used to access non-data-base files.

In CRM catalog mode, the catalog is independent of the subschema. Therefore, the VERSION directive, which specifies the catalog, is independent of the INVOKE or CREATE directive, which specifies the subschema.

CDCS CATALOG MODE

CDCS catalog mode is available when Query Update accesses data base files through the CDCS interface and when CDCS accesses non-data-base files. The catalog file must be described by the data administrator within the subschema.

QUERY UPDATE PROCESSING

Query Update, whether operating in interactive mode or batch mode, receives directives from the user, performs data manipulation and data base maintenance operations, and produces output. Query Update can perform operations on nondata base files or it can perform operations on data base files through an interface to CRM, CDCS, or IMF.

QUERY UPDATE NON-DATA-BASE FILE ACCESS

Query Update can be used to access files that are not part of a data base. If a non-data-base file is to be accessed, it must be described by a directory. A directory to the contents of a non-data-base sequential file can be established by the DESCRIBE directive.

QUERY UPDATE DATA BASE FILE ACCESS

Query Update operations can access information stored in a data base through interfaces to CRM, CDCS, and IMF. To interface with any of these data management facilities, Query Update must be provided with a directory that describes the data to be manipulated. The appropriate directory for each of these interfaces is as follows:

CRM The directory is a subschema (referred to in this manual as a Query Update/CRM subschema) that contains data descriptions not based on a schema definition.

CDCS The directory is a subschema (referred to in this manual as a Query Update/CDCS subschema) that contains data descriptions based on a schema definition.

IMF The directory is an external schema (either COBOL or FORTRAN) that has been initialized as symbolic.

Both the Query Update/CRM subschema and the Query Update/CDCS subschema are created by a data administrator using Data Description Language (DDL). The IMF symbolic external schema is created by an application administrator using special features of the IMF compiler. Refer to the appropriate reference manual for more information about subschema or external schema definition.

Concurrent Data Base File Access

If concurrent file access is allowed, several Query Update applications can access a data base file at the same time. Data base files can be accessed concurrently through the data base management interfaces as follows:

CRM Query Update attaches files in read mode (allowing concurrent file access) for directives that only read the file. Query Update attaches files in write mode (not allowing concurrent file access) for directives that modify the file. Since each transmission is executed as a unit, Query Update attaches and returns each file before preceeding to the next transmission; therefore, the access allowed for a file can change from transmission to transmission depending on the directive specified. (Refer to the discussion of the AF parameter of the Query Update control statement in section 6 for more information.)

CDCS CDCS provides concurrent file access. The data administrator determines concurrent file access for a particular file. (Refer to the CDCS 2 Application Programming reference manual for more information.)

IMF The application administrator determines in the external schema whether concurrent file access is allowed. (Refer to the IMF Application Programming reference manual for more information.)

Query Update/CRM Interface

When a Query Update application operates through CRM, all input/output operations are handled directly by the CRM Advanced Access Methods. CRM performs all logging and data base procedures as specified in the Query Update/CRM subschema. When this interface is used, the features of CDCS cannot be used. An overview of data base access with CRM is shown in figure 1-2.

To access a data base through CRM, Query Update must be provided with a Query Update/CRM subschema. The subschema describes data base organization and defines data names. The subschema does not contain the actual data; it merely describes the data.

The data administrator names all the elements in the subschema through DDL and decides which file and relation descriptions are to be included in the subschema. Data descriptions in the subschema are organized in structures called areas. When two areas have common data items, they can be joined in a logical structure called a relation. (Refer to section 2 for more information about relations.) Within an area, data items are grouped into a record. A record consists of group data names and elementary items that are named and described in terms of size, type (alphabetic, numeric, alphanumeric), and usage (display, computational, and so forth).

One elementary data name in each record description is specified by the data administrator as the record key. The record key is used to locate a specific record, thus eliminating the need for Query Update to examine all records. Alternate keys can also be specified for a record. When alternate keys are specified, the record key is the primary key; 255 alternate keys can be specified for the record. If alternate keys are defined, an index file is maintained; this file provides lists of primary keys or records that contain specific alternate key values. All alternate key index information for an area is contained in one index file.

The character keys of CRM files used through Query Update are considered to be collated. If an uncollated key is desired, see the data administrator to specify the area collating sequence. No attempt should be made to use KT=U on the FILE control statement; incorrect results will occur.

The Query Update user can initially access a data base through CRM by specifying either the INVOKE or CREATE directive with the name of the subschema to

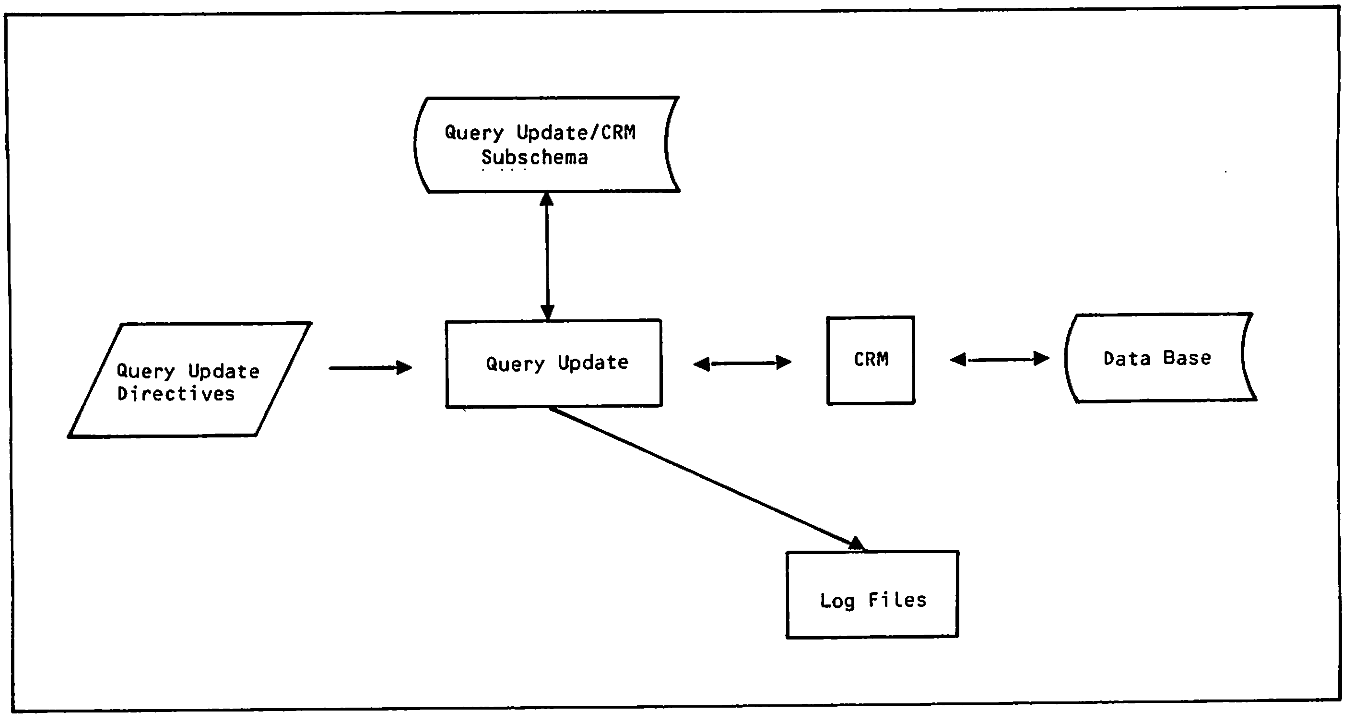


Figure 1-2. Data Base Access With CRM

be used and the permanent file information for the file that contains the subschema directory. The user can access elements within the area or areas joined in a relation by specifying names that have been defined in the subschema. Names and data descriptions can be obtained from a listing of the Query Update/CRM subschema. Alternatively, the EXHIBIT directive can be used to provide this information.

Data can be inserted in the data base after a Query Update/CRM subschema has been established. The data is placed in the data base according to descriptions in the subschema directory.

Records are selected for modification, removal, or display. The criteria for selecting records is specified by the user. The range of records to be processed is identified by the fields specified as selection criteria.

A Query Update/CRM subschema can be used as the directory to access CDCS-controlled data base files if the subschema describes the data exactly as it is described in the CDCS schema.

Query Update/CDCS Interface

When a Query Update application operates through the CDCS interface, CDCS controls all data base access. CDCS uses CRM for input/output operations. Through this interface, the concurrency, privacy checking, logging, and recovery features of CDCS are used. An overview of data base access with CDCS is shown in figure 1-3.

To access a data base through CDCS, Query Update must be provided with a Query Update/CDCS subschema that describes the portion of the data base and relations available to the application. The subschema does not contain the actual data; it merely describes the data and provides for locating and accessing it.

The data administrator names the subschema and decides which file and relation descriptions are to be included in it. Data is organized in a structure called an area in the schema and a realm in the subschema. (In this manual, however, the term area is used for both area and realm.) A set of areas can be accessed as a data base version.

When two areas have common data items, they can be joined in a logical structure called a relation. (Refer to section 2 for a description of relations.) Within an area, data items are grouped into a record. A record consists of group data names and elementary items that are named and described in terms of size, type (alphabetic, numeric, alphanumeric), and usage (display, computational, and so forth). Names and data descriptions are obtained from a listing of the Query Update/CDCS subschema. Alternatively, the EXHIBIT directive can be used to provide this information.

The Query Update user initially accesses the data base through CDCS by specifying either the INVOKE or CREATE directive and including the name of a Query Update/CDCS subschema and the permanent file information required to attach the file containing the subschema directory.

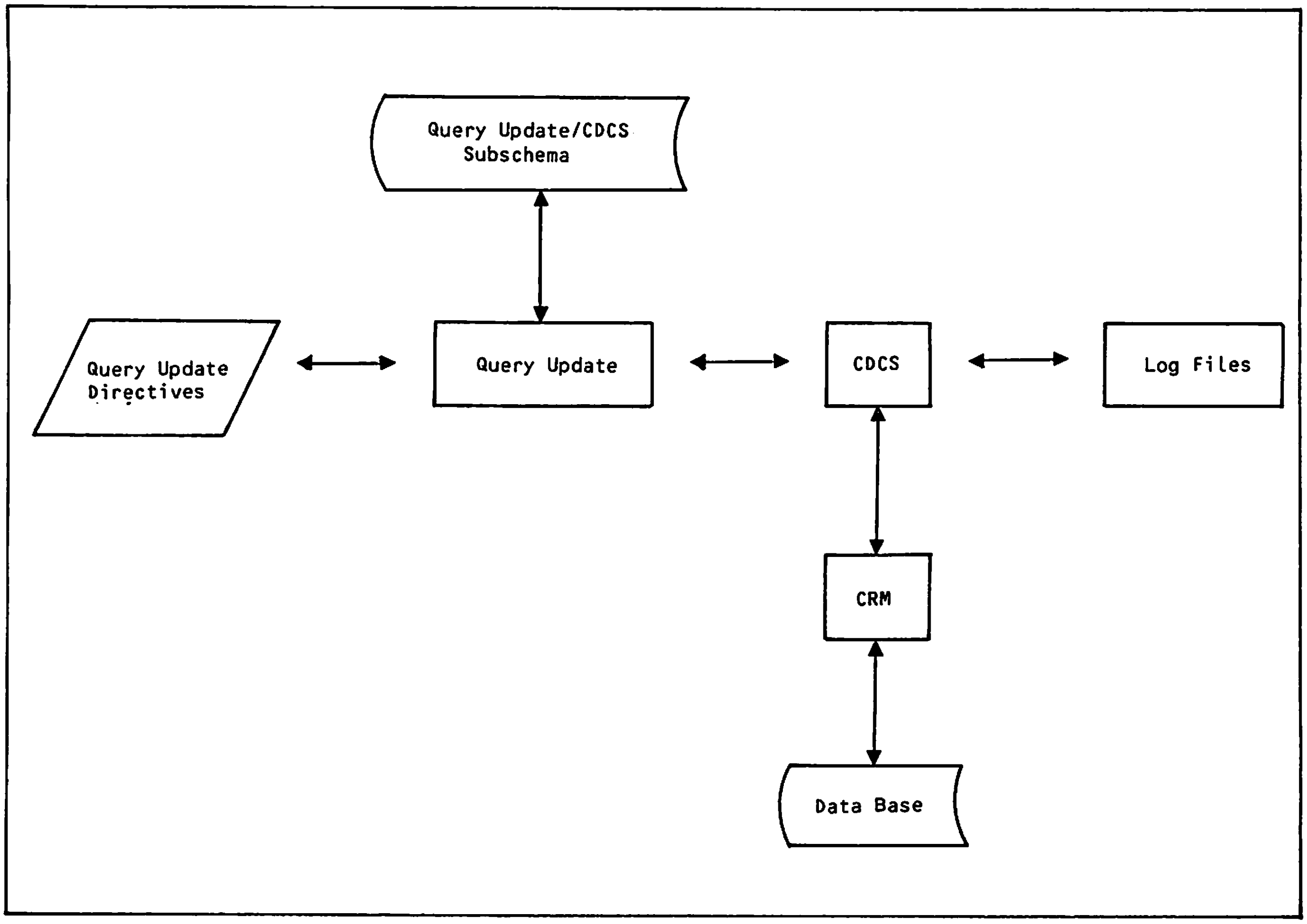


Figure 1-3. Data Base Access With CDCS

If a data base is accessed and manipulated from a previously prepared Query Update session, the catalog file containing the session can be accessed through CDCS if the catalog file is an area included in the subschema.

Query Update/IMF Interface

When a Query Update application operates through the IMF interface, data base access is determined by the application administrator. All data base processing is handled by IMF. An overview of data base access with IMF is shown in figure 1-4.

To access a data base through IMF, Query Update must be provided with a symbolic external schema that describes records and data items. The external schema does not contain the actual data; it merely describes the data and provides for locating and accessing it.

The application administrator defines the external schema and names the record types and data items that are included in it. This individual also defines any cosets (records related by equal values for one or more data items) and access paths (the means of locating a particular record). The application administrator also specifies the permissible operations for each record. Information about data names, access path names, and coset names can be obtained from a listing of the external schema. Alternatively, the EXHIBIT directive can be used to provide this information.

The Query Update user can initially access an information base in IMF mode by specifying the INVOKE directive and including the name of the external schema. The INVOKE directive must also identify the conceptual schema associated with the external schema and must specify the permanent file information required by IMF.

Refer to the IMF Application Programming reference manual for more information about the Query Update interface with IMF.

EXTERNAL FILE ORGANIZATIONS

All data base files accessed through an interface to the data management facilities (CRM, CDCS, IMF) can be read without a FILE control statement.

All Query Update output files and the output files generated by the REPORT utility (intended for print disposition) have the CRM file characteristics of block type C and record type Z. Query Update input files containing directives have the same CRM file characteristics. (Refer to the CRM Basic Access Methods reference manual for a complete description of the meaning of record and block types.)

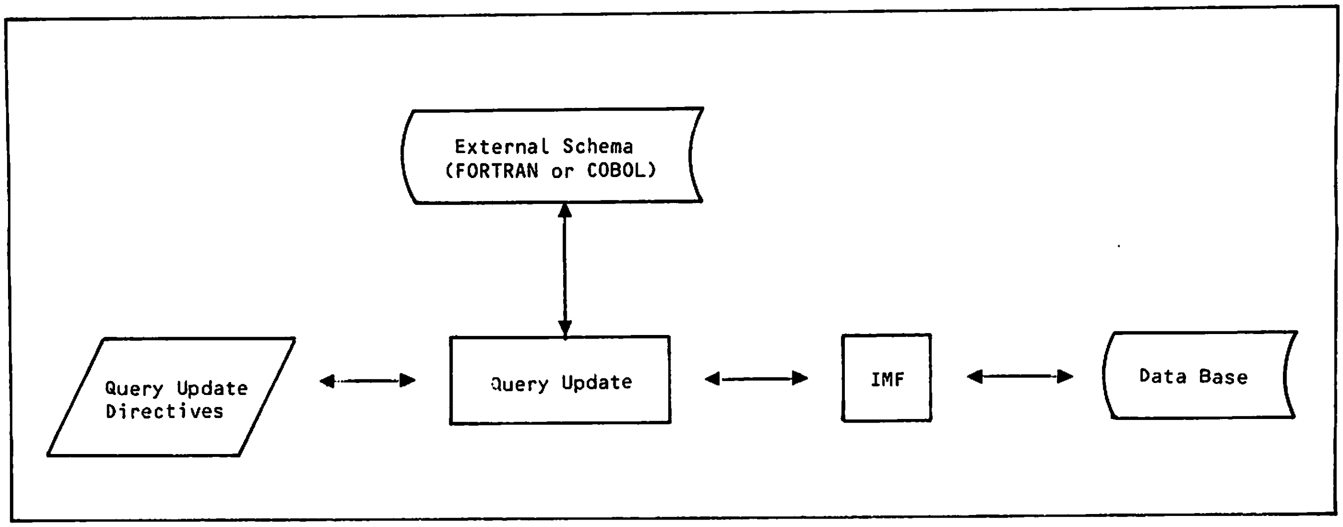


Figure 1-4. Data Base Access With IMF

All other files read or written by Query Update are assumed to have the CRM file characteristics of block type C and record type F by default. (This default can be changed by a FILE control statement; refer to the CRM Basic Access Methods reference manual for information about the FILE control statement.) When writing a file with these file characteristics, Query Update determines the record length unless the record length has been previously specified by a FILE control statement. When reading a file not written in the present session, Query Update requires that either the DESCRIBE directive or a FILE control statement be present to provide information about the file. Query Update retains the file information for the duration of the Query Update session or until either the file is returned or the description is erased.

Query Update communicates all the available file information to the REPORT utility via the COMPILE directive; therefore, a FILE control statement is necessary to describe to the REPORT utility all files that were created after the COMPILE directive was used. The REPORT utility uses by default a CRM block type C and record type Z for the files supplying text in the PREFACE/SUMMARY TEXT IS FROM file-name directives, as well as for the variable file that supplies the names and values of temporary data items that are to be preset before the report is generated.

FILE POSITIONING

The following rules govern the positioning of all files not in a data base, unless otherwise noted for a particular directive:

A file to be read by Query Update (DISPLAY FROM, PREPARE...FROM, MODIFY USING...FROM, and so forth) is rewound before reading, and not rewound after reading. The file cannot be modified before reading. The file is read until end-of-information is encountered.

A file to be written by Query Update (DISPLAY UPON, COMPILE...UPON, and so forth) is not rewound either before or after the write operation.

A file to be read for input data values by Query Update (REMOVE...FROM or STORE...FROM) must be written in display code. The DISPLAY directive can be used for this purpose because all data is converted to display code. The EXTRACT UPON directive is used only if all items in the file are nonnumeric (EXTRACT does not convert items to display code.)

QUERY UPDATE OPERATIONS 2

Query Update can perform the following operations:

Identify temporary data items

Specify condition names

Redefine special characters

Retrieve data

Produce reports

Modify a data base

Retrieve data from a relation

These operations are discussed generally in the following subsections. Specific keywords from the directives are incorporated into the discussion, if appropriate, to show how the user would perform such operations.

TEMPORARY ITEM IDENTIFICATION

Query Update can process directives that operate on temporary items. Unlike permanent items that are created by the data administrator and reside in the data base, temporary items are created by the user and reside in Query Update working storage. Temporary items are defined by the user through the DEFINE directive. This directive establishes data names for temporary items and specifies initial values or methods for evaluating data name content.

CONDITION REFERENCES

A condition can be named for convenient reference through the SPECIFY directive. Subsequent use of the condition name in a Query Update directive causes the condition associated with the name to be tested. Naming conditions can be effective when a long conditional expression is frequently needed.

SPECIAL CHARACTERS

Query Update recognizes special characters other than the continuation character described in section 1. A special character acts as either a delimiter or a universal character.

A delimiter is used to enclose nonnumeric literals. The delimiter allows Query Update to distinguish nonnumeric literals from data names. The default delimiter is the dollar sign ($); the user can designate a different delimiter, if desired, with the SEPARATOR directive.

A universal character is used to designate character positions that are to be ignored during comparison testing. The character is specified in the UNIVERSAL directive; a suggested character is the equivalence sign (≡). The only restriction is that it cannot be the same character as the delimiter character. When Query Update is initiated, no universal character is in effect.

SELECTION CRITERIA

Directives can be selected for execution through the IF directive. When this directive is specified, permanent or temporary items can be examined for a specified condition. If each item satisfies the specified condition, the associated directive is executed. The IF directive is not used during report production; its function can be accomplished through use of the SELECT directive.

RETRIEVAL OPERATIONS

Query Update can obtain data through retrieval operations. These operations consists of obtaining records and either placing the data items in a local file or displaying them at the terminal.

RETRIEVAL SPECIFICATION

Data is retrieved by either the DISPLAY or EXTRACT directives. The major difference between the two directives is that DISPLAY converts data to a format that can be displayed on a terminal while EXTRACT does not.

DISPLAY or EXTRACT is used to retrieve data from a data base, a Query Update temporary storage area, or a non-data-base file (described by a DESCRIBE, DISPLAY UPON, or EXTRACT directive). The DISPLAY UPON and EXTRACT directives also establish a directory of the data base.

OUTPUT SPECIFICATION

Query Update can produce output that is either formatted or unformatted. Both types of output can be displayed at a terminal. Formatted output is used to produce output in the form of a report.

Unformatted Output

Unformatted output is obtained from a data base or temporary storage and, in combination with literal values, can be output as line images by Query Update. Unformatted output is produced by either the EXTRACT or the DISPLAY directive in conjunction with operating system utilities. Use of the DISPLAY directive in producing unformatted output is discussed in the following paragraphs.

The DISPLAY directive retrieves specified records from working storage, a non-data-base file with a directory, or a data base. The records retrieved can be selected by an IF directive; these records are then subject to the action of the DISPLAY directive.

The EVALUATE and DISPLAY directives can be used together to evaluate expressions and display results, much like a desk calculator.

The DISPLAY directive organizes unformatted output into lines, which in turn are grouped into convenient pages for display on a terminal. The next page can be requested by entering Y or N in response to the Query Update indication that more pages are available. The page length and the width of displayed information can be changed by specifying the PL and PW parameters of the Query Update control statement described in section 6. The paging facility can be disabled by specifying the value zero for the PL parameter. When the paging facility is disabled, all lines of output are transmitted for display.

Formatted Output

Query Update directives can be used to generate a report. Report-oriented directives provide the information necessary to generate the report. Layout directives determine the appearance of report pages with respect to page size, content and positioning of headings and footings, and detail lines. Other directives generate a sample report or the actual report. A COMPILE directive can be used to store report specifications on a table; these specifications can subsequently be executed by the batch-oriented REPORT utility.

Query Update prepares a full page image of the report in memory before issuing any output. A report can contain multiple line formats; these formats are selectable on a variety of conditions. A report can have page-oriented titles and recap lines (refer to the description of the RECAP directive in section 4) in addition to control break headings and footings. A report can have prefaces and summaries that are in themselves full reports; this permits the inclusion of distribution lists and bibliographies. The layout of a report can be specified, verified, and saved for subsequent use.

To generate a report, Query Update has two requirements: a valid data file and a report specification. Report generation involves four phases: extraction, specification, verification, and preparation.

The extraction phase involves providing Query Update with a valid data file. The data file must be a nondata base sequential file that contains the content of the report. The data file must be described during this phase. If the information for the report does not exist within a data base, the required file description can be provided by the DESCRIBE directive. If the information for the report exists within a data base, both the required data file and the corresponding file description can be obtained by using the EXTRACT directive. If the order of information within the data file is not appropriate for the desired report, records within the file can be reordered by the SORT directive.

In the specification phase, the user prepares a layout for the report, or a report specificaton. Once the specification has been created, it is saved by Query Update for subsequent use. Report specification begins with a FORMAT directive that names the report. Subsequent report specification directives are retained under that report name.

In the verification phase, the EXHIBIT directive can be used to examine the retained report specification directives. The PREVIEW directive can be used to output a small portion of the report for visual verification of layout. This same output could aid in the alignment of preprinted forms.

In the preparation phase, the source data file and the report to be used are specified by the PREPARE directive. Within the source data file, data is located through a directory that is prepared during the extraction phase by a DESCRIBE, DISPLAY UPON, or EXTRACT directive. The same source data file can be used with different reports, each report referencing only the information it requires. Different source data files can use the same report format, as long as each of the source data files contains all of the referenced data names.

After the report has been produced during the preparation phase, the generated report can be examined by operating system utilities. Report format or content can be changed or altered through ALTER, ERASE, and appropriate report specification directives. Refer to section 5 for a more detailed description of report generation.

DATA BASE ACCESS

Query Update can perform data base operations through the CYBER Record Manager (CRM), the CYBER Database Control System (CDCS), or the Information Management Facility (IMF). When using a data base interface, Query Update is processing in data base access mode, for example, CRM data base access mode. The two following directives have an effect on data base access.

The INVOKE directive determines whether Query Update accesses a data base in CRM data base access mode, CDCS data base access mode, or IMF data base access mode. When the INVOKE directive is used to access a data base through either CRM or CDCS, it makes all areas and relations described in the subschema available to the Query Update user. When used in CRM data base access mode, the INVOKE directive does not cause the actual data base area to be attached; each area is attached when needed to satisfy a directive referring to the area. When used in CDCS data base access mode, the INVOKE directive causes CDCS to attach all data base areas referenced by the subschema. When used in IMF data base access mode, the INVOKE directive makes the records described in the external schema available to Query Update.

The CREATE directive determines whether subsequent directives are performed through CRM data base access mode or CDCS data base access mode. When used in either CRM data base access mode or CDCS data base access mode, the CREATE directive initiates access to an empty area for insertion of data; the area must be described in the subschema. When used in IMF data base access mode, the CREATE directive terminates processing through IMF and initiates processing through CRM or CDCS.

DATA BASE MODIFICATION

The directives used to modify a data base perform the same functions whether they are used through the interface to CRM, CDCS, or IMF. A data base can be modified by the directives MODIFY, STORE, and REMOVE. These directives can be used to add or remove entire records, or to modify data item values within records. When using these directives in interactive mode, the user can request the VETO operation, which requires a user response before a record is added, removed, or modified.

MODIFYING DATA

The MODIFY directive allows modification of existing records in the data base and of temporary data items in Query Update working storage. Records being retrieved for updating can be selected by the USING option or the IF directive.

There are two techniques for supplying the data items to be modified. The first technique involves using the SETTING option to supply the data names; Query Update then either prompts the user for data values or reads the data values from a local file. The second technique involves using the MOVE option to supply both the data names and the data item values.

STORING DATA

Data item values can be added to the data base by creating a record and inserting a new data item value. The STORE directive generates a record and, using either the MOVE or the SETTING option, inserts a new data item value.

There are two techniques for supplying the data items to be stored. The first technique involves using the SETTING option to supply the data name; Query Update then either prompts the user for data values or reads the data values from a local file. The second technique involves using the MOVE option to supply the data names and the data items to be stored.

REMOVING DATA

Entire records can be removed from the data base. The records to be removed can be specified by various options of the REMOVE or IF directives. If no qualification is specified, Query Update does not execute the REMOVE directive, thus preventing the entire data base from being eliminated.

RELATION PROCESSING

When processing in either CRM data base access mode or CDCS data base access mode, the Query Update user can retrieve data distributed across several areas in a directed relationship by using a single query. The areas to be queried simultaneously must be described in a subschema with an appropriate relation description entry. (Refer to the Query Update/CYBER Record Manager Data Administration Reference Manual for a description of the relation entry in the subschema. The Query Update Version 3 Programmer User's Guide gives an example of a relation entry in the subschema.)

In the subschema, two areas are joined in a logical structure called a relation if a data item or key in the record description for each area is specified as a join term. For the areas to be linked in a relation, data items must be identical in size and characteristics.

Once a particular record is retrieved, Query Update uses the value of one join term to search the next area for records that contain the same value. If the next area does not contain any records with that same value or if the conditions of the RESTRICT clause in the subschema have not been met, Query Update returns null values (blanks or zeros) for that field and for any other fields in areas joined to that area in the relation.

The logical relationships among areas can best be explained by an example. Assume that a data base consists of four areas: CONTRACTS, PRODUCTS, SERVICES, and EMPLOYEES. The following two relationships have been established in the subschema describing the data base:

RELATION1	CONTRACTS-PRODUCTS-EMPLOYEES
RELATION2	CONTRACTS-SERVICES-EMPLOYEES

Figure 2-1 shows the records from the four data base areas, the primary keys, the alternate keys, and the join terms. In the figure, connecting lines indicate the direction of the relationships for the example. Solid arrows indicate RELATION1, and broken arrows indicate RELATION2. Even though the example shows two defined relationships, only one relation can be available in a single transmission using Query Update.

The user views the related records as one projected record containing data from each area in a relation. The record selection and query directives apply at the projected record level, and Query Update performs the task of searching relationships. Only when the data must be updated or when human intervention is required to resolve ambiguities between multiple relations must the user be aware of the underlying relationships in the data base. Generally, the Query Update user is not aware that the data being queried is distributed across several areas.

Retrieving information distributed across several files through the IMF interface is discussed in the IMF Application Programming reference manual.

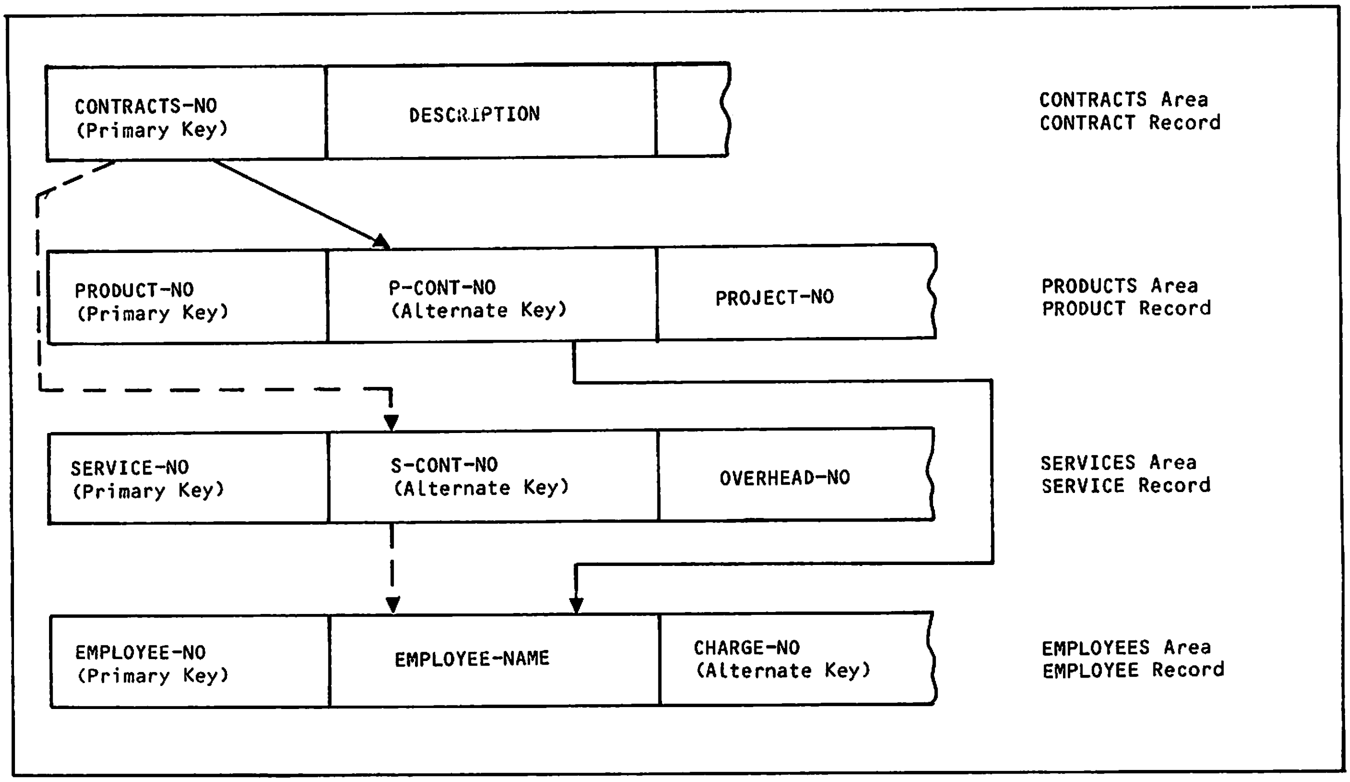


Figure 2-1. Relational Data Base Example

LANGUAGE ELEMENTS 3

Query Update directives are composed of elements such as names, literals, functions, expressions, conditions, and picture specifications. Commas and spaces can be used to separate elements for clarity. Symbols for arithmetic and logical operations must be preceded and followed by at least one space except when the equals sign is within the parentheses of a permanent file parameter list.

DATA TYPES

The data types supported by Query Update are shown in figure 3-1.

CHARACTER	For representing text
LOGICAL	For representing truth values
INTEGER	For representing numbers
FLOATING	
DOUBLE	
COMPLEX	
FIXED	
NUMERIC	

Figure 3-1. Query Update Data Types

The different data types used to represent numbers allow for flexibility in representation and number of significant digits. Data conversion is not allowed between data types used to represent different classes of data. For example, data conversion is allowed between NUMERIC and INTEGER, but not allowed between INTEGER and LOGICAL.

Although the FIXED data type is supported by Query Update, temporary data items cannot be defined with this data type. The FIXED data type is supported for the description of independent files and manipulation of data base items only.

CHARACTER

Data type CHARACTER is used to represent text. It is stored internally as a left-justified display code character with blank fill on the right.

LOGICAL

Data type LOGICAL is used to represent truth values (TRUE, FALSE). It is stored internally as a 60-bit quantity: zero for false or nonzero for true.

INTEGER

Data type INTEGER is used to represent numbers without fractional parts. It is stored internally as a 60-bit binary integer; however, only 48 bits are used. It is capable of retaining 14 significant digits. This storage format is equivalent to COBOL 5 COMP-1.

FLOATING

Data type FLOATING is used to represent numbers with fractional parts. It is stored internally as a normalized 60-bit quantity; 48-bit mantissa and 12-bit exponent. It is capable of retaining 14 significant digits. This storage format is equivalent to REAL in FORTRAN, and COMP-2 in COBOL 5 and DDL.

DOUBLE

Data type DOUBLE is used to represent numbers with fractional parts which require more significant figures than allowed by data type FLOATING. Data type DOUBLE is stored internally as a normalized 120-bit quantity, with a 96-bit mantissa and a 12-bit exponent. It is capable of retaining 29 significant digits.

COMPLEX

Data type COMPLEX is used to represent numbers which have both a real part and an imaginary part. It is stored internally as a pair of normalized 60-bit quantities, each with a 48-bit mantissa and a 12-bit exponent. This data type is capable of retaining 14 significant digits in each of the real and imaginary parts.

FIXED

Data type FIXED is used to represent numbers with fractional parts which have a specified fixed number of positions to the right of the decimal point (for example, representing amounts of money such as $6.98, in which two positions occur to the right of the decimal point, and the need does not exist for representing fractions smaller than .01). Data type FIXED is stored internally as an unnormalized 60-bit quantity, with a 48-bit mantissa and 12-bit exponent. It is capable of retaining 14 significant digits. This storage format is equivalent to COMP-1 in DDL.

NUMERIC

Data type NUMERIC can be used to represent numbers without fractional parts or, by the use of an appropriate picture specification, it can be used to represent numbers with fractional parts which have a specified fixed number of positions to the right of the decimal point (for example, a picture of 9(5).99 could be used to represent money values such as $6.98). Data type NUMERIC is stored internally, as display code digits, right-justified with leading display code zeros. If the value is negative, the sign is overpunched on the rightmost digit. (Thus, a punched 4 digit with an overpunched sign becomes an 11-4 punch, or the character M which is treated as -4.) Data type NUMERIC is capable of handling 14 significant digits. It is equivalent to COMPUTATIONAL in COBOL 5 and DDL.

NAMES

Names are user-supplied directive elements that identify data base areas, files, report specifications, catalogs, sessions, temporary items, and permanent items. Data base areas, relations, and permanent items are named when the subschema for the data base is created. Other names are assigned by the Query Update user in temporary working storage.

A name can contain letters, digits, and hyphens. The hyphen cannot be the first or last character of a name; consecutive hyphens cannot be used in a name. Spaces and commas are not allowed in a name. A name cannot be the same as a reserved word or an abbreviated part of a reserved word. Appendix E lists all Query Update reserved words.

Seven types of names can be specified: data names, file names, report names, register names, function names, condition names, and session names. Each type of name is used in a specific circumstance. Data names require qualification when common to more than one active area.

Data names refer to the stored contents of temporary or permanent items. A data name can contain up to 30 letters, digits, or hyphens. At least one letter must be used in the data name.

File names refer to non-data-base files on mass storage. A file name can contain one to seven characters, the first of which must be alphabetic.

Report names refer to the report specifications retained in the current catalog under the specified name. The report name is assigned by the user and contains 1 to 7 characters; the first character must be a letter, and the name cannot contain hyphens.

Register names refer to data retained by Query Update in locations known as registers. The content of a register is made available by including the register name in a directive. (Refer to table 3-1.)

TABLE 3-1. QUERY UPDATE REGISTERS

Register Name	Register Content
CURRENT-DATE	System date as defined by the installation.
CURRENT-TIME	System clock time in the form hh.mm.ss with leading and trailing blanks.
CURRENT-REPORT	Name of the current report (1 to 7 characters).
CURRENT-LINE	Current report line number; 3-digit display coded; can be referenced as integer data.
CURRENT-PAGE	Current report page number; 5-digit display coded; can be referenced as data of type NUMERIC.
CURRENT-SESSION	Current 6-character session identifier.
CURRENT-ANY	The 5-digit integer corresponding to the ordinal of a repeating group occurrence that satisfied a conditional test using the figurative subscript ANY; the register is initialized at the beginning of the conditional test and is incremented with each reference so that the ordinal of the repeating group occurrence is available with the satisfaction of the conditional test.
USER-ID	The 10-character name used by CDCS to identify the QU user. Initialized to 'QU△△△△△△△△'.

Function names refer to system-supplied results produced from software routines provided in Query Update. The results are obtained by including the function name in a directive. Thirteen system functions are available. (Refer to the description of functions for the specific function names.)

Condition names refer to user statements of tests to be performed on data by the system. The condition name is assigned by the user and contains 1 to 30 characters (letters, digits, or hyphens). At least one character must be a letter.

Session names identify sequences of directives recorded in a catalog. A session name is assigned by the user and contains 1 to 6 characters; the first character must be a letter, and the name cannot contain hyphens.

DATA NAME SUBSCRIPTS

Data name subscripts are used when a matrix or a repeating group is referenced. A matrix is a series of data items that are referenced by one data name. A repeating group is a collection of data items that is repeated a number of times in each record. In the following discussion, the term matrix is used to explain data name subscripts. A subscript must be an integer literal, a data name containing an integer, or one of the figurative subscripts (ALL, ANY, LAST, or NEXT). The range of values allowed for subscripts is 1 through 4095.

A specific data item in a matrix is denoted by a subscript following the matrix data name. The subscript is specified by enclosing in parentheses the integer number (or the data name of an item containing the integer number) of the desired data item. If the subscript is an item, it cannot be in character or floating-point format; it must be in integer format. For instance, the following example refers to the third item in the matrix NAMES:

```
NAMES(3)
```

The subscript value must be within the range defined by the OCCURS clause recorded in the subschema or the ITEMS clause of the DEFINE directive. The value can be limited further by a DEPENDING ON clause.

When a matrix is within a larger matrix, the data names of both matrices must be specified and must be subscripted. Up to three matrix levels can be specified. The following example refers to the second item of JOB-CLASS, which is the fifth item of DEPT:

```
JOB-CLASS(2) OF DEPT(5)
```

If only one occurrence of JOB-CLASS exists for each DEPT, the reference would be the following:

```
JOB-CLASS OF DEPT(5)
```

Four figurative subscripts are available for use:

ALL References all items in the matrix.

ANY References any matrix item that meets the specified conditional test. Each item in the matrix is tested until the condition is met. (Refer to the CURRENT-ANY register description.)

LAST References the last item in a variable length matrix. This subscript can be used only with occurring items that have been defined with a DEPENDING ON clause.

NEXT References the next item in a variable length matrix only. This can be used in conjunction with the figurative LAST to reference the next available space in the matrix. This subscript can be used only with occurring items that have been defined with a DEPENDING ON clause.

A condition can contain only one figurative subscript. The figurative subscript NEXT can be used only for the destination item in the MOVE and STORE SETTING directives. ANY is restricted to the IF, BREAK, SELECT, SPECIFY, and PERFORM UNTIL directives in which a condition can be specified.

The use of figurative subscripts is illustrated in the following examples:

```
DISPLAY NAMES(LAST)

MOVE $SMITH, J$ TO NAMES(NEXT)
```

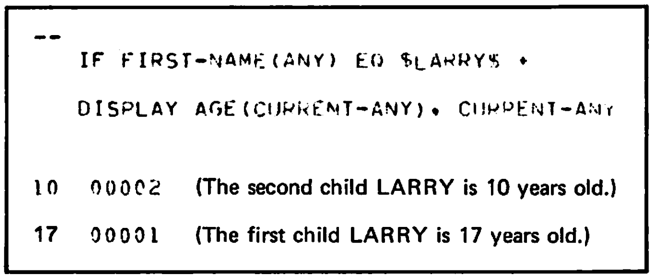
The first directive causes the last name in the matrix to be displayed. The second directive places the name SMITH, J after the last item in the matrix; it then becomes the last item.

The figurative subscript ANY and the CURRENT-ANY register are used to obtain information from a matrix called CHILDREN. Items in the matrix are AGE and FIRST-NAME. In two record occurrences in the file, the actual values for AGE and FIRST-NAME are:

```
Record 1:  12 JEFF     10 LARRY    02 CEDRIC

Record 2:  17 LARRY    15 CHERYL
```

A query using this matrix and the output displayed are shown in figure 3-2.



```
--
   IF FIRST-NAME(ANY) EQ $LARRY$ .

   DISPLAY AGE(CURRENT-ANY). CURRENT-ANY


10   00002   (The second child LARRY is 10 years old.)

17   00001   (The first child LARRY is 17 years old.)
```

Figure 3-2. ANY and CURRENT-ANY Example

DATA NAME QUALIFICATION

Data name qualification is used to make a data name unique when the same name has been used more than once in the data descriptions referenced by Query Update. Data names are used in the subschema to name items described there; they are used to name items established by DEFINE and SPECIFY directives and to reference items created by DESCRIBE, DISPLAY UPON, and EXTRACT directives. Query Update separates these data names into the three categories just described. When a user supplies a data name, Query Update must obtain the attributes of the data item to which the name applies by searching appropriate data item lists created for each category.

Certain qualification is inherent in the type of directive specified by the user. For instance, data names specified in report directives must be in the list of temporary items or in the directory for the file from which the report is prepared. Data names used in updating directives must be in the subschema or in the list of temporary items.

Data names that have been described by a DESCRIBE, DISPLAY UPON, or EXTRACT directive cannot be mixed in a directive with data items in the subschema. For the designated applications, Query Update sequentially searches data item lists for data item attributes as indicated below.

For a data item used within a report directive:

1. The temporary data item list

2. The directory established by the DESCRIBE, DISPLAY UPON, or EXTRACT directive for the specified logical file name

For a data item associated with a cumulative function for a purpose other than evaluation or for one used as a value in a DEFINE directive:

1. The temporary data item list

2. All directories established by DESCRIBE, DISPLAY UPON, or EXTRACT directives (searched in alphabetical order)

3. Subschema directories

For a data item used in all other contexts:

1. The temporary data item list

2. The directory established by the DESCRIBE, DISPLAY UPON, or EXTRACT directive for the specified logical file name

3. Subschema directories

Duplicate data names are not allowed for data names specified in a DEFINE or SPECIFY directive. When a data name is entered in one of these directives, the list of data names is scanned for an identical name; a diagnostic is issued if an identical name is found, and the name is rejected. Defined or described data names are not checked in this manner; therefore, the same data name can exist in more than one of the records in use or a data name defined in a subschema description can be the same as a data name described by the DEFINE or SPECIFY directive. Such can be the case when a subschema is activated by an INVOKE directive after temporary items have been defined, and the subschema contains names that duplicate temporary data names. When duplicate names exist, a reference to the data name alone would obtain the temporary data item. The subschema item name can be referenced if it is qualified.

Qualification of a data name can be made by specifying the word OF with one or more group names and/or either the record name or the file name if it is a described item. Qualification at the record name level is the highest level of qualification possible. Up to five levels of qualification are permitted. If the record name itself duplicates a temporary item name, then any data name that requires qualification with the record name is inaccessible. Only the minimum amount of qualification to make the name unambiguous is required. More qualification than is needed is not treated as an error. An example of qualification for two record descriptions containing identically named items is shown in figure 3-3.

LITERALS

A literal is a string of characters that represents a specific value. Literals are used for comparisons, arithmetic operations, and replacement of items referenced by data name. A literal can be either numeric or nonnumeric.

NONNUMERIC LITERALS

A nonnumeric literal is enclosed by a pair of delimiters. The default delimiter is the dollar sign ($). The SEPARATOR directive can be used to assign a different delimiter. (Refer to section 4.) Delimiters allow the system to distinguish between data names and literals. Some examples of nonnumeric literals are as follows:

$SMITH, J$

ABC

123

```
Record Description 1                  Qualification

RECORD-NAME IS PAYROLL                EMP-NAME OF PAYROLL
     02   EMP-NAME    PIC X(20)
     02   ADDRESS     PIC X(50)

Record Description 2                  Qualification

RECORD-NAME IS EMPLOY                 EMP-NAME OF EMPLOY
     02   ID-NO       PIC X(6)                 or
     02   EMPLOYEE                    EMP-NAME OF EMPLOYEE
          03   EMP-NAME    PIC X(25)           or
          03   GRADE       PIC 99     EMP-NAME OF EMPLOYEE OF EMPLOY
```

Figure 3-3. Data Name Qualification Example

If the character used as the delimiter is to be included in the literal, the character must be specified twice for each occurrence. For example, $A$$B$ yields the literal A$B. A nonnumeric literal must not exceed 255 characters and cannot be used for computation. If the size of a nonnumeric literal exceeds that of an item into which it is being stored, the extra rightmost characters of the literal are truncated.

Octal values and character masks can be specified as nonnumeric literals. These literals are formed as follows:

> An octal value is specified by the letter O immediately preceding the delimiter. For example, O$37746$ represents an octal value. An octal digit (0 through 7) represents 3 bits. Octal values are right-justified and zero filled; the maximum size is 20 octal digits.

> A character mask is specified by the letter M immediately preceding the delimiter. For example, M$YYYNN$ represents a character mask. The mask contains only the characters Y (converted to six 1 bits) and N (converted to six 0 bits). Character masks are left-justified and zero filled; the maximum size is 255 mask characters.

NUMERIC LITERALS

An integer literal is a string of 1 to 14 decimal digits written without a decimal point. It can be positive, negative, or zero. If the integer is positive, the plus sign can be omitted; if it is negative, the minus sign must be present. The format of an integer literal is as follows:

$n_1 n_2 \ . \ . \ . \ n_m$

n is a decimal digit (0 through 9)

$1 \leq m \leq 14$

Examples of integer literals are:

1 0 -1 -456 247342

When floating-point-to-integer conversions are involved, literals are rounded up if the digit following the decimal point is greater than or equal to 5; the literals are truncated when the digit is less than 5.

A single-precision floating-point literal consists of a string of digits with a decimal point and an optional exponent. If the exponent is positive, the plus sign is optional. The range of a single-precision floating-point literal is 10^{-293} to 10^{+322}. A value outside this range is not diagnosed and is treated as a zero. The format of the single-precision floating-point literal is as follows:

$$\pm \begin{Bmatrix} n. \\ n.n \\ .n \end{Bmatrix} [E \pm s]$$

n coefficient ≤ 14 decimal digits

E ± s exponent (base 10, integer)

Examples of the single-precision floating-point literal are as follows:

.007 39.27 -2.54 1.054E3 -.039E-12

A double-precision floating-point literal is written in the same way as a single-precision floating-point literal except that the exponent is specified with a D instead of an E. If the exponent is positive, the plus sign is optional. Double-precision values are represented internally, as two computer words, giving extra accuracy. Double-precision floating-point literals are limited to 29 significant digits. The format of the double-precision floating-point literal is as follows:

$$\pm \begin{Bmatrix} n. \\ n.n \\ .n \end{Bmatrix} D \pm s$$

n coefficient

D ± s exponent (base 10, integer)

Examples of the double-precision floating-point literal are as follows:

.007D0 3937.D-2 1054D+3 -.039D-12

A complex floating-point literal consists of two single-precision floating-point literals separated by the letter I. The first coefficient corresponds to the real part, the second coefficient corresponds to the imaginary part. Both parts must be specified. The format of the complex floating-point literal is as follows:

$$\pm \begin{Bmatrix} n. \\ n.n \\ .n \end{Bmatrix} [E \pm s] \ \mathrm{I} \ \pm \begin{Bmatrix} n. \\ n.n \\ .n \end{Bmatrix} [E \pm s]$$

n coefficient ≤ 14 decimal digits

E ± s exponent (base 10, integer)

Examples of the complex floating-point literal are as follows:

1.I2. 1.E15I4.E10 -1.0I2.E19

Examples of invalid numeric literals are as folows:

1.D5I4.E2	(Double-precision not allowed in complex.)
124E-52	(Decimal point required in coefficient. This is, however, a valid data name.)

If the size of a numeric literal exceeds that of an item into which it is being stored, the most significant digits of each field are truncated to fit.

FUNCTIONS

Functions are software procedures that are supplied by Query Update to produce results. A function can be substituted whenever the directive syntax indicates that an expression can be used. In addition, the SCAN function can be substituted whenever the syntax of a directive indicates that a condition can be used. The general format of a function is as follows; parentheses must be used:

```
                 { literal   }
function-name (  {           } , . . .)
                 { data-name }
```

The number of parameters within the parentheses depends on the function name. Functions are either independent or cumulative.

INDEPENDENT FUNCTIONS

Independent functions produce either a result for each record read during the execution of the directives or a single result for user-defined temporary item data names. Eight independent functions are provided.

The use of several independent functions is illustrated in figure 3-4. Example 2 shows the use of the DECODE function. When a value cannot be located in the first array, the last item in the second array is used as the result. If DECODE is used to translate a data item in the data base, a data item name would replace $SEPT$ or TEN in the DECODE expression. Another use for the DECODE function is in a detail line specification for a report. Some examples of masking are shown in table 3-2.

```
ABS (data-name)
```

Provides the absolute value of the contents of the data name. The data name can be an item of type NUMERIC, FIXED, INTEGER, or FLOATING. It cannot be data type CHARACTER, LOGICAL, DOUBLE, or COMPLEX.

```
DECODE (data-name-1 ( ALL ),

{ literal     }
{             } ,data-name-3 ( ALL ) )
{ data-name-2 }
```

Translates the literal or data-name-2 according to the data-name-1 matrix to select an item from the data-name-3 matrix. If the specification for data-name-3 is LOGICAL, the DECODE function can be substituted whenever the syntax of a directive indicates that a condition can be used. In a specific parameter list, the parameters cannot be of mixed data type if any one is type CHARACTER or type LOGICAL.

```
      { data-name }
MIN ( {           } , . . . )
      { literal   }
```

Selects the smallest value in the list of parameters; at least two parameters must be specified. In a specific parameter list, the parameters must all have the same data type. Legal data types are CHARACTER, COMPLEX, NUMERIC, DOUBLE, INTEGER, LOGICAL, and FLOATING.

```
      { data-name }
MAX ( {           } , . . . )
      { literal   }
```

Selects the largest value in the list of parameters; at least two parameters must be specified. In a specific parameter list, the parameters must all have the same data type. Legal data types are CHARACTER, COMPLEX, NUMERIC, DOUBLE, INTEGER, LOGICAL, and FLOATING.

```
         { data-name }
JULIAN ( {           } )
         { literal   }
```

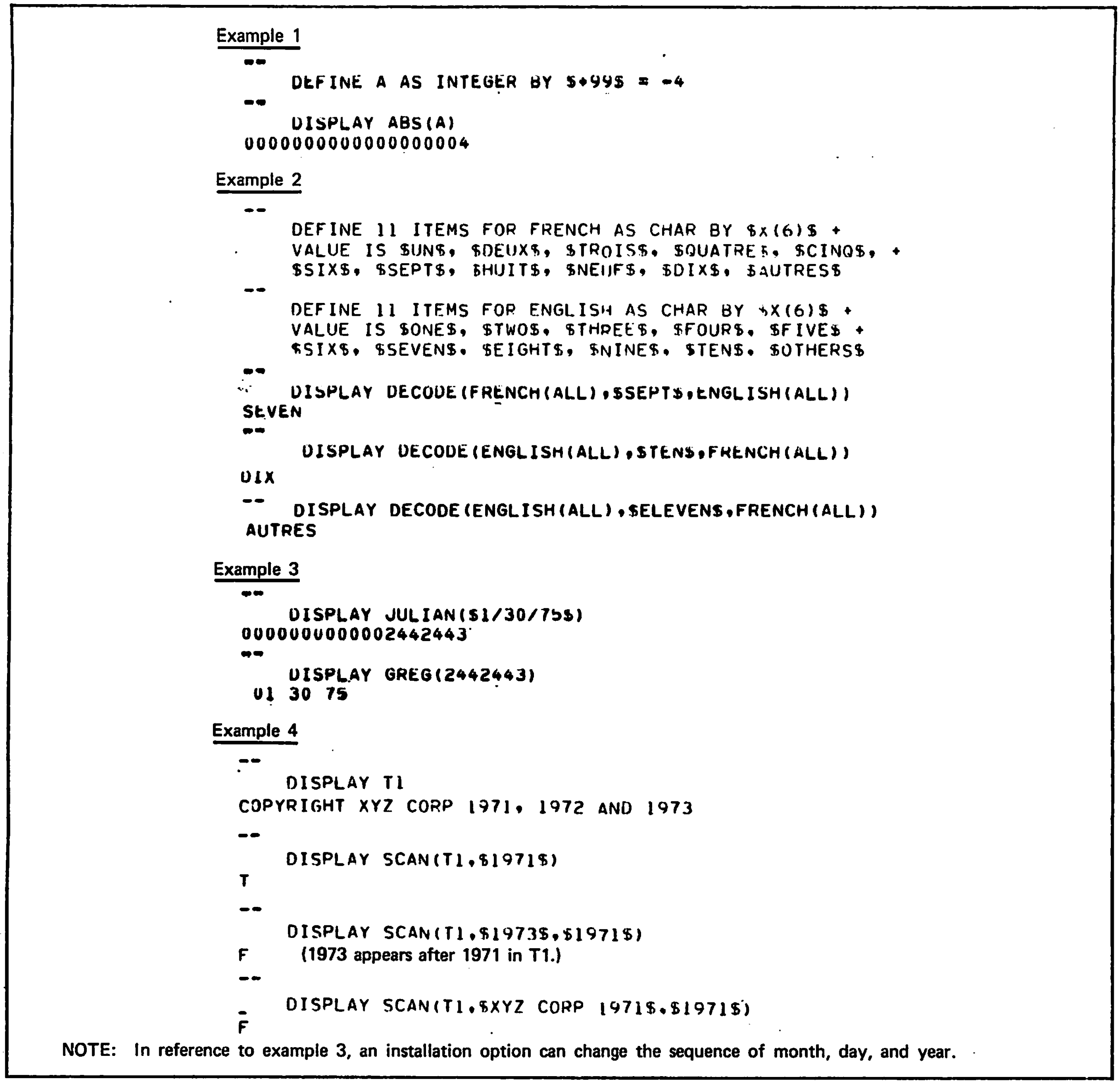
Produces the number of days elapsed from January 1, 4713 B.C. to the specified date. An example of the literal format is MMxDDxYY. Month (MM), day (DD), and year (YY) are two digit numbers whose sequence is determined by the installation. An easy way to find this sequence is to check the date format on the Query Update header. The x's can be any character, or can be omitted.

```
       { data-name }
GREG ( {           } )
       { literal   }
```

Produces a date as MM DD YY when given a data-name or literal of an integer JULIAN date.

```
       { data-name-1 } { data-name-2 }
MASK ( {             },{             } , literal-3 )
       { literal-1   } { literal-2   }
```

Masks the source field (data-name-1 or literal-1) onto the background field (data-name-2 or literal-2) through the mask field (literal-3). All three parameters must be of identical length, with one exception: octal and integer data types can be mixed. The resultant value of the mask function is represented as the same data type as the parameters, with one exception: when the parameters are octal, the resultant is represented as a decimal number. For information on character masks and on the representation of octal values, see the Literals subsection.



Example 1

```
--
     DEFINE A AS INTEGER BY $+99$ = -4
--
     DISPLAY ABS(A)
000000000000000004
```

Example 2

```
--
     DEFINE 11 ITEMS FOR FRENCH AS CHAR BY $x(6)$ +
     VALUE IS $UN$, $DEUX$, $TROIS$, $QUATRE$, $CINQ$, +
     $SIX$, $SEPT$, $HUIT$, $NEUF$, $DIX$, $AUTRES$
--
     DEFINE 11 ITEMS FOR ENGLISH AS CHAR BY $X(6)$ +
     VALUE IS $ONE$, $TWO$, $THREE$, $FOUR$, $FIVE$ +
     $SIX$, $SEVEN$, $EIGHT$, $NINE$, $TEN$, $OTHERS$
--
     DISPLAY DECODE(FRENCH(ALL),$SEPT$,ENGLISH(ALL))
SEVEN
--
      DISPLAY DECODE(ENGLISH(ALL),$TEN$,FRENCH(ALL))
DIX
--   DISPLAY DECODE(ENGLISH(ALL),$ELEVEN$,FRENCH(ALL))
AUTRES
```

Example 3

```
--
     DISPLAY JULIAN($1/30/75$)
000000000002442443
--
     DISPLAY GREG(2442443)
 01 30 75
```

Example 4

```
--
     DISPLAY T1
COPYRIGHT XYZ CORP 1971, 1972 AND 1973

--
     DISPLAY SCAN(T1,$1971$)
T

--
     DISPLAY SCAN(T1,$1973$,$1971$)
F      (1973 appears after 1971 in T1.)
--
     DISPLAY SCAN(T1,$XYZ CORP 1971$,$1971$)
F
```

NOTE: In reference to example 3, an installation option can change the sequence of month, day, and year.

Figure 3-4. Independent Function Examples

TABLE 3-2. MASKING EXAMPLES

MASK Function	Source Field	Background Field	Mask Field	Mask (octal)	Result
MASK($ZYX$,$ABC$,M$YNY$)	ZYX	ABC	M$YNY$	7700770000. . .	ZBX
MASK($123$,$000$,M$NNY$)	123	000	M$NNY$	0000770000. . .	003
MASK(O$334400$,O$707070$,O$443377$)	334400_8	707070_8	O$443377$	443377	100352[†]

[†] This is a decimal number; its octal equivalent is 304000_8.

Masking is used to combine portions of two data fields selectively. The parameter list for the function consists of three fields: source field, background field, and mask field. The presence of a 0 bit in the mask indicates that the corresponding bit in the background field is used in producing the result. The presence of a 1 bit in the mask indicates that the corresponding bit in the source field is used in producing the result. The masking process involves the logical product of the source field and the mask, the logical product of the background field and the complement of the mask, and the logical sum of the two logical products.

$$\underline{\text{SCAN}}\ \underline{(}\ \left\{\begin{matrix}\text{data-name-1}\\ \text{literal-1}\end{matrix}\right\},\left\{\begin{matrix}\text{data-name-2}\\ \text{literal-2}\end{matrix}\right\},\ \left[\begin{matrix}\text{data-name-3}\\ \text{literal-3}\end{matrix}\right]\ \ldots\ \underline{)}$$

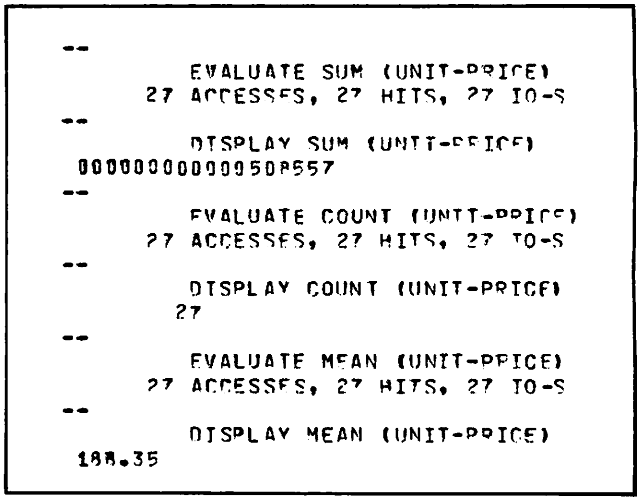
Searches for specified character strings within data items or literals and yields a true or false result (T or F). At least two parameters must be specified. The first parameter is scanned, looking for character strings identical to the character strings specified in the other parameters. Scanning begins at the leftmost character of the first parameter, comparing the first parameter to the second, and sliding to the right one character position at a time to make another comparison if no match is found. If the first parameter ends before a match is found, the result F is returned. If a match is found before the first parameter ends and no more parameters have been specified, the result T is returned. If a match occurs before the end of the first parameter is reached and more parameters have been specified, the scan resumes at the character position following the string that matched, comparing the next parameter to the remainder of the characters in the string, and so on for each subsequent parameter. When more than one parameter is specified in the SCAN function, the scan produces a T result only if all specified character strings are found; the scanning process is terminated when a false condition first results.

The data names can be data items defined in the subschema, in a DEFINE directive, or in an item described in a DESCRIBE, DISPLAY UPON, or EXTRACT directive. Subschema data items and described data items cannot be mixed in a SCAN function. Defined and described items must have the specification CHARACTER. Literals represent character strings and follow the rules applicable to nonnumeric literals. The universal character can be specified within a literal. The first parameter specified cannot have fewer characters than any subsequent parameters; if it does, an automatic result of false is returned. The result of a SCAN function can be moved to a defined data item that has the LOGICAL specification.

CUMULATIVE FUNCTIONS

Cumulative functions produce accumulated results for records read during the execution of a transmission. The function can produce one result after all records have been processed or a separate result for each record that contains a value for a specified data name.

A cumulative function can be specified in the EVALUATE directive with a data name. The data name can be a data item defined in the subschema or in a DESCRIBE, DISPLAY UPON, or EXTRACT directive. The cumulative function is reevaluated each time EVALUATE is executed. It is reevaluated each time EVALUATE is executed. To display the result, the cumulative function and data name must be included in a DISPLAY directive. Values calculated for cumulative functions are stored by Query Update so that they can be displayed upon request during the session. Five cumulative functions are provided. An example of the use of several cumulative functions is shown in figure 3-5.



```
--
          EVALUATE SUM (UNIT-PRICE)
      27 ACCESSES, 27 HITS, 27 IO-S
--
          DISPLAY SUM (UNIT-PRICE)
 000000000000508557
--
          EVALUATE COUNT (UNIT-PRICE)
      27 ACCESSES, 27 HITS, 27 IO-S
--
          DISPLAY COUNT (UNIT-PRICE)
         27
--
          EVALUATE MEAN (UNIT-PRICE)
      27 ACCESSES, 27 HITS, 27 IO-S
--
          DISPLAY MEAN (UNIT-PRICE)
 188.35
```

Figure 3-5. Cumulative Function Example

<u>SUM</u> <u>(</u>data-name<u>)</u>

Accumulates a sum of the value of the data name for the current record and all prior records that have been processed.

<u>COUNT</u> <u>(</u>data-name<u>)</u>

Accumulates a count of all data name occurrences that have been processed.

MEAN (data-name)

Accumulates an arithmetic mean (average) of all the data name occurrences that have been processed. If the functions SUM and COUNT are not specified prior to the specification of MEAN, they are automatically evaluated by Query Update, since the results of these functions are used to calculate the mean.

MINS (data-name)

Retains the smallest value for the data name among the records that have been processed.

MAXS (data-name)

Retains the largest value for the data name among the records that have been processed.

EXPRESSIONS

Expressions are user-supplied elements in various Query Update directives. An expression consists of at least one data name or literal. A complex expression can contain a series of data names and/or literals connected by arithmetic operators. A character literal or data item defined as character usage cannot be included in an arithmetic expression. An arithmetic operator is specified by a word or by a symbol. Allowable arithmetic operators are listed in table 3-3. An arithmetic expression cannot contain a logical variable. Two arithmetic or logical operators cannot be used successively; an operand must intervene. An arithmetic or logical operator cannot be used outside an arithmetic, conditional or logical expression.

TABLE 3-3. ARITHMETIC OPERATORS

Word	Symbol	Operation	Order of Evaluation
PLUS	+	Addition	Third
MINUS	–	Subtraction	Third
TIMES	*	Multiplication	Second
OVER	/	Division	Second
RAISED	**	Exponentiation	First

Expressions are evaluated from left to right according to their order of evaluation. Parentheses can be used to define an explicit order of evaluation. Expressions enclosed in parentheses are evaluated first, beginning with the innermost pair and proceeding to the outermost pair. The limit for nested parentheses is 25. The expressions within parentheses are evaluated according to the order of evaluation.

The format of an expression is:

```
{data-name-1}
{literal-1  }
{function-1 }

[                        {data-name-2}]
[{arithmetic operator}   {literal-2  }] . . .
[                        {function-2 }]
```

When a symbol is used to represent an arithmetic operator, a blank must precede and follow the symbol.

Mixed modes (for example, integers and floating points) should be avoided unless the user is sufficiently knowledgeable to guarantee the evaluation is correct. Arithmetic combination of various numeric types is performed by the system according to the rules of dominance. The dominant type for evaluation is determined as various numeric types are encountered in the left-to-right evaluation of an expression. If parentheses are involved, the innermost parenthetical portion is evaluated first.

System selection of numeric type for the result of an expression uses the principle of dominance. Numeric types have a rank number; the largest rank number appearing in the components of the expression is the dominant type used as the format for the result of the expression.

Type	Rank
COMPLEX	4
DOUBLE	3
FLOATING	2
INTEGER	1

The principle of dominance can be stated as follows:

1. If the expression contains any data item with a COMPLEX type, the expression's result is a COMPLEX type.

2. If rule 1 does not apply and the expression contains any data item with a DOUBLE type, the expression's result is a DOUBLE type.

3. If rules 1 and 2 do not apply and the expression contains any data item with a FLOATING type, the expression's result is FLOATING type.

4. If rules 1, 2, and 3 do not apply, the expression's result is an INTEGER type.

The system performs this determination of dominance and type conversion upon operand pairs during expression evaluation. For certain type conversion, rounding is performed (for example, moving 1.5 to an integer item). In these cases, fractional parts less than 0.5 are rounded down, and all others are rounded up. If the receiving item requires scaling, the scaling is performed before rounding takes place (for example, a NUMERIC item with picture 999V99 accepts 6.98, but 6.985 is rounded to 6.99).

CONDITIONS

A condition is used in a directive to test the relationship between pairs of expressions, the result of a SCAN function, or the result of a DECODE function. (See DECODE description.) The directives associated with the conditional directive are executed only when the tested condition is true.

An expression contains one or more literals, data names, or functions. Arithmetic operations can be specified in the expressions. The two expressions are connected by a relational operator. A relational operator can be specified by an abbreviation or by a symbol as indicated in the following list:

EQ	=	Equals
NE	≠	Not equal
GT	>	Greater than
LT	<	Less than
GE	≥	Greater than or equal
LE	≤	Less than or equal

In its simplest form, a condition to be tested is stated as a relational expression. The format for a conditional operation is as follows:

$$\text{expression-1 } \{\text{relational operator}\} \text{ expression-2}$$

The collating sequence used in comparing expressions depends on the items in the expressions. If an expression contains a data item that has been defined within an area, the collating sequence used is the collating sequence for that area. The COBOL collating sequence is the default sequence for an area; however, in the Query Update subschema, another collating sequence can be specified in the SEQUENCE clause. If the expressions do not contain a data item defined within an area, the collating sequence used is the same as display code. If the area items in the expressions have been extracted upon a file, via the DISPLAY UPON or EXTRACT directive, the collating sequence for the area does not apply; the FORTRAN collating sequence is used.

Conditions can be combined or compounded to select data that meets more than one condition. For clarity, parentheses can be used to group relational expressions joined by logical operators. Compound relational expressions are specified in the following format:

$$\text{relational-exp-1}\{\text{logical operator}\}\text{relational-exp 2}$$

Allowable logical operators and symbols are indicated in the following list:

AND	∧	Both conditions must be satisfied.
OR	∨	Either or both conditions must be met.
XOR		One or the other, but not both, conditions must be met.
NOT	¬	The relational expression must not be met.

The logical operators, AND, OR, and XOR, must be preceded and followed by an expression that yields a logical value of true or false.

The logical operator NOT is used to reverse a relational expression. It can be used in conjunction with a relational operator and with another logical operator. The use of the logical operator NOT is shown in the following format:

$$\left[\begin{array}{c}\neg\\ \underline{\text{NOT}}\end{array}\right]\text{relational-exp-1}$$

$$\left[\text{logical operator}\left[\begin{array}{c}\neg\\ \underline{\text{NOT}}\end{array}\right]\text{relational-exp-2}\right]\ldots$$

The sequence in which an expression is evaluated is governed by the following rules, listed in descending precedence:

Query Update cumulative functions and independent functions are expanded.

Expressions delimited by parentheses are evaluated, beginning with the innermost pair and proceeding to the outermost pair.

Expressions defined by arithmetic, relational, and logical operators are evaluated according to the following precedence hierarchy:

**	Exponentiation
/*	Division or multiplication
+-	Addition or subtraction
GT GE LT LE EQ NE	Relational operators
NOT	Logical operator
AND	Logical operator
OR XOR	Logical operators

Expressions containing operators of equal precedence are evaluated from left to right.

Unary addition and subtraction are treated as operations on an implied zero. For example, +2 is treated as 0+2, and -3 is treated as 0-3.

A subscripted item used in an expression requires the evaluation of its subscript.

As an example, the expression A/B-C*D**E is evaluated in the following way:

D is raised to the power of E.

A is divided by B.

C is multiplied by the result of D**E.

The product of C*D**E is subtracted from the quotient of A divided by B.

QUERY UPDATE DIRECTIVES

Over 50 directives are provided for user operation of Query Update. The directives are described in this section and are presented in alphabetic order for convenient reference. It is suggested that the new user initially read the directive descriptions for DEFINE, EXHIBIT, ERASE, DISPLAY, MOVE, IF, SPECIFY, and EVALUATE. The data base oriented directives, including INVOKE, STORE, MODIFY, REMOVE, CREATE, VERIFY, and VETO, can be studied next. See section 1 for a table of all directives.

The directive descriptions assume the use of interactive mode with no parameters specified in the QU control statement. (Refer to section 6.) The applicable term should be substituted for the word terminal under the following conditions:

> Operations are in interactive mode and a different input or output file has been designated in the QU control statement.

> Operations are in batch mode; this applies whether or not a different input or output file has been designated in the QU control statement.

When the RO parameter (denoting read-only mode) is specified in the QU control statement, certain Query Update directives are disabled or modified. Refer to the description of the RO parameter in section 6 for further details.

ACCESS

The ACCESS directive applies only to CDCS data base access mode. The format of the directive is shown in figure 4-1. The literal or data-name must not exceed 30 characters. The ACCESS directive supplies CDCS an access key for areas and catalog files. If access control locks have been defined in the schema for the areas and catalog files, the ACCESS directive must be used to supply the access key.

```
                                      ┌ ║ INPUT  ║ ┐
ACCESS KEY [ IS { literal   } ]  [ ON │ ║ I-O    ║ │ ]
                { data-name }         └ ║ OUTPUT ║ ┘

  [ FOR { AREA area-name . . .        } ] . . . ] . . .
        { CATALOG [catalog-name]      }
        { AREAS                       }
```

Figure 4-1. ACCESS Directive Format

The ACCESS directive must be entered after a subschema has been specified in an INVOKE, CREATE, or VERSION directive and before any area or catalog file with an access control lock has been accessed. If two access keys are required for particular areas or catalog files, both access keys must be specified in access directives before using the areas or catalog files.

Figure 4-2 illustrates examples of the ACCESS directive.

Example 1

```
-- ACCESS KEY IS $READ-PASSWORD$ ON INPUT FOR AREA EMPLOYE
```

Specifies the key READ-PASSWORD to allow reading the EMPLOYE area.

Example 2

```
-- ACCESS KEY
>>>IS $UPDATE-PASSWORD$ ON I-O FOR DEPTMTS
>>>*END
```

Specifies the key UPDATE-PASSWORD to allow updating the DEPTMTS area. Security is improved over example 1 because the key is not printed on output or trace listings.

Figure 4-2. ACCESS Directive Examples

The ON clause indicates the kind of usage the access key is to provide. The following usages can be provided by a single access key:

ON OUTPUT	Indicates update of an area
ON INPUT	Indicates retrieval from an area
ON I-O	Indicates both update and retrieval from an area

The default value is ON I-O.

The FOR clause specifies the area name or the catalog file name whose access is controlled by a particular access key. If AREA is used, the specified access key applies only to the specified area. If AREAS is used, the specified access key applies to all areas in the subschema that can be accessed by the specified access key (but not to the specified catalog file). If CATALOG is used, the specified access key applies only to the catalog file. If the FOR clause is omitted, both the AREAS and CATALOG options are assumed.

The IS clause and subsequent clauses can be omitted. If Query Update encounters a transmission that contains only the word ACCESS or the words ACCESS KEY, terminal input is prompted with the symbols >>>. The user then enters the access key and subsequent optional clauses. Query Update continues to prompt after each transmission until the user enters *END. The prompting feature of this directive ensures the privacy of the access keys because it inhibits the printing of access keys on the output listing or trace file.

ALTER

The ALTER directive specifies the name of the report whose specifications are to be modified. The specified report is established as the current report. The ALTER directive is required only when report directives are added. This directive is not required when subsequent directives remove or replace report specifications; such operations require the ERASE directive, which also establishes the current report. The format of the ALTER directive is shown in figure 4-3.

```
ALTER report-name
```

Figure 4-3. ALTER Directive Format

The report name designated in this directive must have been created by a previous FORMAT directive and must reside on the current catalog. The report specification directives (layout directives) that follow the ALTER directives are added to the existing report format. The ERASE directive must be used to delete a layout directive; specifying the ALTER directive is not necessary. Query Update assumes that the format modification is complete when a directive that is not related to the report format is encountered. After the report specifications have been changed, a PREPARE directive referencing the report name causes the report to be generated in accordance with the altered format.

BREAK

The BREAK directive establishes a situation that causes the body of a report page to be interrupted for footing and heading lines. The situation can be a change of content in a data item or other detail conditions. The format of the BREAK directive is shown in figure 4-4.

```
                                      ( OVERFLOW        )
BREAK level-number [AND EJECT] ON    < ITEM data-name   >
                                      ( condition       )
```

Figure 4-4. BREAK Directive Format

The level number (1 through 63) relates the control break to a numbered heading and/or footing. A hierarchy for the placement of headings and footings is established by the BREAK directive level number; a lower numbered control break condition is examined before a higher numbered control break. The BREAK directives can be specified in any level number sequence; Query Update arranges then in ascending order. Level number 0 cannot be specified in a BREAK directive; level number 0 is associated with the beginning of data condition and the end of data condition. (Refer to the HEADING and FOOTING directives.) A maximum of five level numbers can be specified.

Breaks are tested in ascending order of level numbers. Only those records that satisfy the criteria of a SELECT directive are tested for a break condition. When a break situation occurs, all footings with higher level numbers precede the footing whose level number matches the BREAK directive level number. When a heading also has the same level number, it is placed on the page and all headings with higher level numbers follow it in ascending order before detail lines recur.

If a report specification contains footing and heading level numbers 1, 2, 3, 4, and 5 and the situation of a BREAK 3 directive occurs, the order of the footing and heading lines is as follows:

Detail line (BREAK 3 condition occurs)

Footing 5

Footing 4

Footing 3

Heading 3

Heading 4

Heading 5

Detail line

The EJECT option forces a page eject in conjuction with the break situation. The page eject can take place when an overflow could occur, when the content of a data name changes, or when a specified condition occurs.

The situation that causes a break is determined by the option that follows the word ON.

OVERFLOW

Is detected when the page is filled. The condition is tested upon placement of information into the page and not on reading the source data file; heading, footing, and recap lines as well as detail lines are involved in this determination.

ITEM data-name

Causes a break when the content of data-name changes value.

condition

Causes a break when the specified condition is true.

The testing of a break occurs in a look-ahead process. The situation tested by a BREAK directive is examined while a current detail line is in process even though the next detail line is available for processing. Consequently, a break can occur and choose numbered heading or footing lines while the current detail line is being processed; the heading or footing placement will occur after the current detail line has been processed.

The MOVE and EVALUATE directives can be timed to occur when the break testing is satisfied, that is one detail line after the placement of a heading or footing line. Unless the user has a reason to include information from the first detail line of the next control break group, MOVE and EVALUATE directives should be timed to occur in association with heading or footing placement rather than being timed to occur when the look-ahead break testing occurs. The directives in the first example in figure 4-5 are not recommended unless the next detail line is to be included. Directives in the second example are associated with a footing (to include the last detail line) rather than being timed to occur when the look-ahead break happens.

```
Example 1

    Inclusion of Next Detail

        MOVE BEFORE BREAK  1, A TO B
        EVALUATE AFTER BREAK  1, C

Example 2

    Normal Inclusion of Last Detail

        MOVE BEFORE FOOTING  1, A TO B
        EVALUATE AFTER FOOTING  1, C
```

Figure 4-5. BREAK Directive Examples

COMPILE

The COMPILE directive retrieves selected report specifications from the current catalog and stores then in encoded form on a table file. The COMPILE directive must be the last directive in a transmission. The table file can be subsequently referenced in a REPORT control statement to initiate reports whose specifications do not change between runs. The format of the COMPILE directive is shown in figure 4-6.

```
COMPILE report-name [DESCRIBED IN file-name]

    UPON table-file
```

Figure 4-6. COMPILE Directive Format

The report name specified must be the name of a report whose specifications can be found in the current catalog. Query Update retrieves all report specifications in reportname, builds an encoded table, and writes the table onto the table file. The table file is not rewound or otherwise positioned by Query Update; therefore, several COMPILE directives can be used to accumulate additional report specifications.

A directory described by the DESCRIBE, DISPLAY UPON, or EXTRACT directive must exist for the data file used to produce the report. The DESCRIBED IN option names in directory (named the same as the file it describes) and must be specified if more than one directory has been created during the session. If only one directory exists for the current session, this option can be omitted.

If the specified report name contains preface or summary reports, Query Update displays their names and associated input data files. This serves as a reminder to the user that COMPILE directives must be issued for these reports. Figure 4-7 illustrates the Query Update/user dialogue sequence.

```
--

        COMPILE BUDGET1 UPON TBLFILE

        PREFACES/SUMMARIES

        BUDSUM FROM BUDIN

--

        COMPILE BUDSUM UPON TBLFILE
```

The first COMPILE directive causes the report format named BUDGET1 to be stored on the file TBLFILE. Query Update then informs the user that BUDGET1 specifies a preface or summary named BUDSUM and that the file BUDIN contains input data for the preface or summary.

The second COMPILE directive causes the report specifications for BUDSUM to be stored on TBLFILE.

Figure 4-7. Sample COMPILE Directives

After the table file has been generated by the COMPILE directive, it is available for input to the REPORT utility program. The user provides the utility program with the name of the table file together with the source data to be reported in a REPORT control statement. The REPORT control statement and its required parameters are detailed in section 6.

The user is responsible for the positioning, modification, and disposition of the table file. Utility routines such as ITEMIZE, COPYL, and COPYBR within the NOS/BE and NOS operating systems are available for manipulating the table file. Tables can be added to or deleted from a table file. In addition, a table in an existing table file can be replaced by a table with a matching name in another file. Refer to the appropriate operating system reference manual for instructions regarding these operations.

CREATE

The CREATE directive, shown in figure 4-8, initiates processing in either CRM or CDCS data base access mode. The CREATE directive also identifies a subschema and prepares a data base area for initial insertion of data.

Only one CREATE directive can be in effect at a time. When a CREATE directive is entered, the effect of any prior CREATE or INVOKE directive is terminated. The effect of previously entered access keys is terminated, and subschema files and the tables established to use the subschema are released. In CDCS data base access mode, the CREATE directive can be entered any number of times within a single execution of the QU control statement as long as the same area and version are not duplicated. When used in a transmission with other directives, the CREATE directive must be the last directive in the transmission.

The area-name identifies the area to be created. Area-name must refer to an area included in the subschema referenced by subschema-name.

The subschema-name identifies the subschema being used. Subschema-name must be the name of a Query Update/CDCS or Query Update/CRM subschema. When CDCS catalog mode is active, the subschema must be the same subschema used for the VERSION directive.

The INDEX option is supported in CRM data base access mode only. The INDEX option is required when the area contains primary and alternate keys and the index file was not specified in the subschema. Index-file-name must conform to the operating system convention for naming files.

The FROM LIBRARY option identifies the subschema library that contains the subschema directory being used. If the FROM LIBRARY clause is omitted, the permanent file name of the subschema library is assumed as follows: for NOS, the first seven characters of the subschema name are used, for NOS/BE, the entire subschema name is used.

When permanent file parameters are required to access the subschema, they must be specified in the CREATE directive. The PW option can be specified to protect the security of passwords. When Query Update encounters the characters PW alone, terminal input is prompted with the symbols >>>. The user then enters the appropriate passwords. Query Update continues to prompt after each transmission until the user enters *END. The prompting feature of this option ensures the security of passwords because it inhibits the printing of passwords on the output listing or trace file.

The FOR DATABASE clause is available under CDCS data base access mode only and identifies the data base version being used. If the clause is omitted, the MASTER version is assumed unless a prior VERSION directive that is in effect specified an alternate data base version, then the alternate version is assumed. If a prior VERSION directive is in effect and the FOR DATABASE clause is specified, the same data base must be specified in both the VERSION and CREATE directives. The version name must be a one through seven character name of a data base version included in the master directory as a version of the schema being used; MASTER is the name of the default data base version. The data administrator must provide the user with valid alternate version names when data base versions are being used.

```
     CDCS Format

          CREATE area-name OF subschema-name [FROM LIBRARY permanent-file-name]
                    [( permanent-file-parameters [PW] )] [FOR DATABASE version-name]†


     CRM Format

          CREATE area-name [INDEX IS index-file-name] OF subschema-name
                    [FROM LIBRARY permanent-file-name] [( permanent-file-parameters  [PW] )]


     †Not available on NOS 1
```

Figure 4-8. CREATE Directive Format

In CRM data base access mode, data base files can be made permanent either before the session or after the session is terminated. Figure 4-9 shows a sequence for making data base files permanent before the session on NOS and after the session on NOS/BE.

```
NOS

    DEFINE,permanent-file-name.
    QU.
    CREATE area-name ...


NOS/BE

    QU.
    CREATE,area-name ...
    CATALOG,permanent-file-name.
```

Figure 4-9. Making Files Permanent Under CRM

In CDCS data base access mode, data base files must be defined or cataloged as empty permanent files by the data administrator before beginning the Query Update session that creates the files. If the Query Update user must define or catalog the files, the data administrator must provide permanent file names and other permanent file information to be used to create the files. The permanent file information must be the same as recorded in the master directory for the particular area and data base version. Figure 4-10 shows a sequence for making data base files permanent before the session.

```
NOS

    DEFINE,permanent-file-name.
    RETURN,permanent-file-name.
    QU.
    CREATE,area-name ...


NOS/BE

    REQUEST,permanent-file-name,PF.
    REWIND,permanent-file-name.
    CATALOG,permanent-file-name.
    RETURN,permanent-file-name.
    QU.
    CREATE,area-name ...
```

Figure 4-10. Making Files Permanent Under CDCS

DATE

The DATE directive is a report specification that indicates the vertical and horizontal positioning of the system supplied date. The sequence of digits for month, day, and year (mm/dd/yy) is determined by the installation; a space appears before and after the date. The format of the DATE directive is shown in figure 4-11.

```
        ┌    ┌ TITLE-LINE     ┐   ┌ TAB integer-2    ┐ ┐
DATE    │ AT │ LINE integer-1 │   └ COLUMN integer-3 ┘ │ ...
        └    └ RECAP-LINE     ┘                        ┘
```

Figure 4-11. DATE Directive Format

The DATE directive with no options specified causes the date to appear at line 1 in column 2. Vertical positioning of the date can be specified by one of three options:

TITLE-LINE

Positions the date in the first line of the title. If no title directive was specified, the date is positioned in the first line of each page.

LINE integer-1

Places the date in the line that is integer-1 lines from the top of the page.

RECAP-LINE

Positions the date in the first line of the recap. If no RECAP directive was specified or the BEYOND option was used in the RECAP directive, the date is positioned in the last line of the page.

Horizontal positioning of the date is specified by either the TAB or COLUMN option.

TAB integer-2

Begins the date in the column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

Begins the date in the print position specified by integer-3.

The date can appear more than once on a page through repetition of the AT vertical and horizontal positioning specifications. If either specification is omitted, the date is positioned at the default for that specification:

Vertical default	Line 1
Horizontal default	Column 2

DEFINE

The DEFINE directive establishes a temporary data item in terms of a data name, description, and initial content or method of evaluation. A value for a temporary data name can be stored by including an expression containing only literals in the DEFINE directive or by executing a subsequent MOVE or EVALUATE directive for an expression that does not contain only literals (for example, a register name or function). The format of the DEFINE directive is shown in figure 4-12.

The ITEMS option is used to establish a matrix. A matrix is a series of identically defined data items that are referenced by one data name. The integer specified is the exact number of items for a fixed-length matrix or the maximum number of items for a variable-length matrix; the integer must be within the range of 1 through 4095. The DEPENDING ON option must also be included for a variable-length matrix; data-name-1 must be a previously defined temporary data item that contains an integer.

Data-name-2 must not be a currently defined data name. When the ITEMS option is used, subsequent references to data-name-2 must be subscripted. (Refer to section 3 for a description of subscripts.) Each item in the matrix is described identically. The entire series can be preset by using the VALUE (or =) option and one or more literals or expressions associated with the items in the matrix, beginning with the first element (subscript = 1). If fewer literals or expressions are specified than there are elements in the matrix, the value for the last literal or expression specified is used for all of the remaining elements in the matrix. Constants are the only legal preset values for a matrix (array), being preset by the VALUE IS clause.

The AS option specifies the internal format for the data item in system working storage. Table 4-1 shows the internal representation for each data type. If this option is not included in the DEFINE directive, FLOATING format is assumed by default.

```
DEFINE [integer ITEMS [DEPENDING ON data-name-1] FOR]

               ⎡   ⎧ INTEGER   ⎫ ⎤
               ⎢   ⎪ FLOATING  ⎪ ⎥
               ⎢   ⎪ DOUBLE    ⎪ ⎥
data-name-2    ⎢AS ⎨ COMPLEX   ⎬ ⎥ [BY picture-specification] [{VALUE IS} expression]
               ⎢   ⎪ LOGICAL   ⎪ ⎥                             {   =    }
               ⎢   ⎪ CHARACTER ⎪ ⎥
               ⎣   ⎩ NUMERIC   ⎭ ⎦
```

Figure 4-12. DEFINE Directive Format

TABLE 4-1. DEFINE AND DESCRIBE DIRECTIVE DATA TYPES

Data Type	DISPLAY Option	No DISPLAY Option
CHARACTER	Alphanumeric display code	Alphanumeric display code
NUMERIC	Numeric display code, sign overpunched on rightmost digit	Numeric display code, sign overpunched on rightmost digit
INTEGER[†]	Optional sign preceding string of numeric display code digits	60-bit binary quantity (COBOL 5 COMP-1)
FIXED[††]	Optional sign preceding string of numeric display code digits containing optional decimal point	Unscaled 60-bit binary quantity; 48-bit mantissa, 12-bit exponent (COBOL 4 COMP-1)
FLOATING	Optional sign preceding string of numeric display code digits containing decimal point, followed by E and an optionally signed scale factor	Scaled 60-bit binary quantity; 48-bit mantissa, 12-bit exponent (COBOL 4 and COBOL 5 COMP-2)
DOUBLE	Same as FLOATING but contains a D instead of E	Scaled 120-bit binary quantity; 96-bit mantissa, 12-bit exponent
COMPLEX	Two DISPLAY FLOATING quantities separated by an I	Scaled pair of normalized 60-bit quantities; each with 48-bit mantissa, 12-bit exponent
LOGICAL	One to five characters containing leftmost characters of TRUE or FALSE	Binary 60-bit quantity, nonzero is TRUE

[†] COBOL 5 COMP-1 data items cannot have fractional parts.

[††] FIXED data type does not apply to the DEFINE directive.

The BY option specifies the size and type of the data item in a manner similar to a COBOL picture specification. If the picture specification contains characters other than 9's, it must be enclosed by a pair of separators. Appendix F describes the picture specification. When this option is not included in the DEFINE directive, a default picture specification is used based on the internal format specified in the AS option. Default picture specifications are as follows:

Type	Default Picture Specification
INTEGER	-9(18)
FLOATING	-9(12).9(2)
DOUBLE	-9(12).9(6)
COMPLEX	-9(12).9(2)
LOGICAL	X
CHARACTER	X(10)
NUMERIC	-9(18)

Default pictures containing a negative sign produce a blank for a positive number and a minus for a negative number.

The VALUE option specifies either the actual value for a data name or an expression that provides a method for Query Update to determine the value for a data name. If an actual value is not specified, a value appropriate to the internal format of the data item (either zero or space) is initially assigned. When the expression is a literal, it specifies an actual value. The data name being defined cannot be referenced within the expression. Refer to section 3 for examples of numeric literals. If COMPLEX is specified, two values must be entered; the values must be separated by an I (for example, VALUE IS 3.0I3.1). The first value given is the real part and the second value is the imaginary part of the complex number. Any other form of expression requires evaluation of the expression. Evaluation is performed only when the data name is referenced in a subsequent EVALUATE directive. When an expression contains data names, the removal or redefinition of these data names (via a RETURN, ERASE, DISPLAY UPON, EXTRACT, INVOKE, or DESCRIBE directive) invalidates the expression. A subsequent attempt to evaluate an item that is defined by the use of an invalidated expression would give indeterminate results. The value of an expression can be displayed, but the EVALUATE directive is required to store the value of an expression in a data item.

DELETE

Although the DELETE directive is available to the Query Update user, the REMOVE directive performs the same function, and is preferred because it is more flexible. Refer to appendix H for a description of the DELETE directive.

NOTE

Refer to appendix J for recommendations on the use of this feature.

DESCRIBE

The DESCRIBE directive establishes a directory to the content of a sequential data file that is not within the data base; if a previous directory exists for the file, it is eliminated. This directive allows Query Update to process an existing sequential local file for report production and to display requested data from the file. More than one DESCRIBE directive can be in effect at a time. The format of the DESCRIBE directive is shown in figure 4-13.

The sequence of data names in AND clauses indicates the order of fields in the record. Each AND clause produces an entry in the directory for data-name-2. The AS and BY clauses are subordinate to an AND clause and are required to complete the entry in order to account for all positions in the source file records.

Use of FILLER is recommended when a portion of a record is not to be referenced; FILLER should not be used with the ITEMS option.

The OR clause is used to indicate the beginning of an alternate description (another record type) for the file.

```
DESCRIBE file-name  { {AND} [integer-1 ITEMS [DEPENDING ON data-name-1] FOR]
                    { {OR }

                                     { FIXED     }
                                     { INTEGER   }
                                     { FLOATING  }
        {FILLER     }  AS [DISPLAY]  { DOUBLE    }  BY picture-specification } . . .
        {data-name-2}                { COMPLEX   }
                                     { LOGICAL   }
                                     { CHARACTER }
                                     { NUMERIC   }
```

Figure 4-13. DESCRIBE Directive Format

The ITEMS option is used to indicate that a series of data items (a matrix) is to be described. For a fixed-length matrix, the exact number of items in the matrix is specified by integer-1; integer-1 must be within the range of 1 through 4095. For a variable-length matrix, the maximum number of items is specified by integer-1 and the DEPENDING ON option is included with the ITEMS option. The item referred to by data-name-1 must be an integer and it must appear in the record before the variable-length matrix; data-name-1 designates the exact number of items for the record. A variable-length matrix must be the last item described for a record.

The choice between FILLER or data-name-2 determines whether or not the data can be referenced during a Query Update operation. If a data name is entered, the data can be processed by reference to that data name. Data identified by FILLER cannot be referenced by Query Update unless the data is defined by another record description for the same file; FILLER provides a directory entry for Query Update to locate fields relative to the beginning of the record.

The type description of the data is specified by AS followed by a word designating the data type. This also determines any conversion required by Query Update. The DISPLAY option, which immediately follows AS, informs Query Update that the item being described is in display format. Table 4-1 shows the data representation for each data type with or without the DISPLAY option.

The BY picture specification clause must be included in each entry. The size of the data item is determined by this clause. If the picture specification contains characters other than 9's, it must be enclosed by a pair of separators. Refer to appendix F for a description of the picture specification.

If the DISPLAY option is used, the picture specification clause is used only to determine the size of the data item. The data item is displayed or printed in a report exactly as it appears in the data file. If the data is NUMERIC, INTEGER, FIXED, FLOATING, DOUBLE, or COMPLEX; it must be right-justified within the field. For example, if a DISPLAY DOUBLE field is 12 characters long, it could contain the data:

Δ	Δ	Δ	-	1	4	.	3	2	1	D	1

The picture in this example could be specified as 9(12) or X(12).

A directory can be eliminated by:

Issuing the ERASE directive with the DESCRIBE option.

Issuing the Query Update RETURN directive to release the local file with which the directory is associated.

Using the DISPLAY directive with the UPON option, the SORT directive with the UPON option, or the EXTRACT directive.

When the ERASE directive with the DESCRIBE option is used, the associated file remains available but its contents cannot be referenced by the names that were in the directory that was just eliminated. The DISPLAY directive with the UPON option, the SORT directive with the UPON option, or the EXTRACT directive eliminates the directory and establishes a new directory for the file.

Figure 4-14 illustrates a DESCRIBE directive used to define two different record formats.

```
DESCRIBE MYFILES AND COUNTRY AS CHAR BY $X(30)$ +
                 AND FILLER AS CHAR BY $X(7)$ +
                 AND UN-MEMBER AS DISPLAY LOGICAL BY $X$ +
                 AND CONTINENT AS NUMERIC BY $99$ +
                 AND SIZE AS INTEGER BY $ZZ,ZZZ,ZZ9$ +
                 AND POPULATION AS INTEGER BY $Z,ZZZ,ZZZ,ZZ9$ +
                 AND CITIES AS CHAR BY $X(20)$ +
                 AND DENSITY AS FLOAT BY $ZZ,ZZ9.99$ +
                 OR FIRSTLETTER AS CHAR BY $X$ +
                 AND FILLER AS CHAR BY $X(29)$ +
                 AND EX-FILLER AS CHAR BY $X(7)$
```

The data file MYFILES is described twice. The first description defines eight fields in each record; the FILLER field in the first record description cannot be referenced except if a second record description defines a method. The second, or alternate, description defines three fields in each record; the first character of the record, a FILLER field to position past the remaining 29 characters of COUNTRY, and the EX-FILLER field used to access the field defined as FILLER in the first record description are used in the second record description.

Figure 4-14. Sample DESCRIBE Directive

DETAIL

The DETAIL directive is a report specification that determines the content and positioning of data items, literals, and results of expressions for a detail line on a report page. Specific detail lines can be referenced by SELECT directives. The format of the DETAIL directive is shown in figure 4-15.

The tag number option is included when a SELECT directive chooses a particular detail line specification upon satisfaction of a condition. Inclusion of a tag number (1 through 63) requires an identically numbered SELECT directive. Omission of a tag number prohibits use of the SELECT directive and permits only a single unnumbered DETAIL directive in the format specification.

The AT LINE option specifies the vertical position of the detail line. An absolute line number (AT LINE integer-1) places the detail line on the specified line on the report page. The DETAIL directive cannot specify an absolute line number when the PAGE-SIZE directive for the report has specified vertical sections. A relative line number (AT LINE integer-1 BEYOND) positions the detail line in relation to the line generated immediately preceding execution of the DETAIL directive. Generally, absolute line numbers are not used in detail line specifications because of the overwriting conflict; however, a report involving forms such as checks or other fixed line documents would reference absolute line numbers. A fixed line document can be visualized as a one-page report that repeatedly begins and ends with each detail source data record. For this type of report, the user must specify that a page eject be performed after each record is processed. The BREAK directive with the EJECT and ITEM options specified can be used to control page ejection. If the AT LINE option is not included in the directive, the detail line is positioned one line beyond the preceding line. The examples in figure 4-16 illustrate the use of the AT LINE option in the DETAIL directive.

The AT LINE option and the following elements of the DETAIL directive can be stated repeatedly to specify multiple lines in the detail specification. Integer-1 must be increased in each succeeding AT LINE clause. When a relative line number is specified, each detail line appears the designated number of lines beyond the line preceding the detail specification. Examples 2 and 3 in figure 4-16 show DETAIL directives with multiple lines specified. Multiple AT LINE clauses cannot be specified when the PAGE-SIZE directive for the report has specified vertical sections.

The content of a detail line is specified by one or more IS clauses. Multiple IS clauses are used to indicate the horizontal positioning of specific data; the IN option is then included with each IS clause. Example 1 in figure 4-16 illustrates a DETAIL directive with two IS clauses. The expression in each IS clause specifies data that is to appear in the detail line; it identifies one or more data fields, literals, or defined items. With multiple IS clauses, the optional elements are specified for each expression to which they apply.

The ONCE option suppresses placement of a data name value in a detail line after its first appearance until the value changes or a new page begins. This option can be used only when no more than one numbered DETAIL directive is specified in the report format. The ONCE option cannot be used in a DETAIL directive when the AT LINE option specifies an integer that is an absolute number. When a PAGE-SIZE directive has specified more than one section, the ONCE option cannot be specified for a detail line.

The CENTERED option causes the expression to be automatically positioned midway between two horizontal points. If no points are specified by an IN clause, default horizontal points of column 1 and the page width are assumed. A horizontal point specified by the IN clause represents the left point; the right point is the next tab or the page width. Centering is applied only to the expression associated with the CENTERED option.

The IN clause specifies the horizontal positioning of the expression. The option selected specifies either the starting point of the expression or the left point to be used for centering:

NEXT TAB

> The next tabular position after the current position; TAB 1 is assumed if a previous tab has not been specified in the directive.

$$\begin{array}{l}
\underline{\text{DETAIL}}\ [\text{tag-number}]\ \left\{ \left[\text{AT }\underline{\text{LINE}}\text{ integer-1 }\left[\underline{\text{BEYOND}}\right]\right] \right. \\[1em]
\left. \left\{ \text{IS expression } \left[\underline{\text{ONCE}}\right]\left[\underline{\text{CENTERED}}\right] \left[\text{IN} \left\{ \begin{array}{l} \underline{\text{NEXT}}\ \text{TAB} \\ \underline{\text{TAB}}\ \text{integer-2} \\ \underline{\text{COLUMN}}\ \text{integer-3} \end{array} \right\} \right] \right\} \ldots \right\} \ldots
\end{array}$$

Figure 4-15. DETAIL Directive Format

Example 1

```
DETAIL IS NAME IN COLUMN 1 IS ADDRESS IN COLUMN 20
```

This specification produces a single-spaced list of source data with one name and address on each line.

Example 2

```
DETAIL AT LINE 3 IS $1 JULY 1975$ IN COLUMN 40 +
       AT LINE 5 IS NAME IN COLUMN 18 AMOUNT IN COLUMN 45 +
       AT LINE 9 IS COMPANY-NAME IN COLUMN 30
```

This specification assumes a page size of nine lines; the user must establish a page eject with the BREAK directive in order to start a new page for each source record. The literal data of 1 JULY 1975 is placed on the third line of each page starting in the fortieth horizontal position of the line. Source data fields of data-names NAME and AMOUNT are placed on line 5; a source data field of data-name COMPANY-NAME is placed on line 9.

Example 3

```
DETAIL 5 AT LINE 3 BEYOND IS  ...
         AT LINE 5 BEYOND IS  ...
         AT LINE 6 BEYOND IS  ...
```

Each line is placed a relative number of lines beyond the line preceding this detail specification. The first line appears three lines beyond the previous line; the second line is double spaced in relation to the first line; the third line is single spaced in relation to the second line.

Figure 4-16. Sample DETAIL Directives

TAB integer-2

The column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

The print position specified by integer-3.

Line width must be considered if the SECTIONS option of a PAGE-SIZE directive is specified. A section must be wide enough to accommodate the width of a detail line.

DIAGNOSTIC

The DIAGNOSTIC directive provides the means to specify whether or not consecutive duplicate diagnostic messages are to be displayed. If duplicate messages are not displayed, a count of the occurrences is maintained. The format of the DIAGNOSTIC directive is shown in figure 4-17.

$$\underline{\text{DIAG}}\text{NOSTIC} \begin{Bmatrix}\underline{\text{PART}}\\ \underline{\text{FULL}}\end{Bmatrix}$$

Figure 4-17. DIAGNOSTIC Directive Format

The PART option specifies that consecutive duplicate messages are not displayed. Each time the error is encountered, the count of occurrences is incremented by one. If a different diagnostic is encountered, a message is displayed indicating the number of times the diagnostic occurred and the number identifying the diagnostic. For example, error 609 occurs 10 consecutive times, and the following message is displayed after the first 609 diagnostic:

```
THERE WERE 9 MORE 609 DIAGNOSTIC(S).
```

The count of duplicate occurrences continues until a different diagnostic occurs, the count reaches 100, or the transmission ends. In all cases, the message is displayed.

Many diagnostic messages include data names, values, page numbers, and other information identifying the specific error encountered. When duplicate messages are not displayed, this information is lost.

The FULL option specifies that all diagnostic messages are displayed. Specific information related to the error is available to the user.

If the DIAGNOSTIC directive is not specified, DIAGNOSTIC PART is assumed by default.

DIRECTORY

The DIRECTORY directive controls the building of directories for files created by the DISPLAY and EXTRACT directives. The format of the DIRECTORY directive is shown in figure 4-17.1.

```
DIRECTORY   { ON  }  FOR  { DISPLAY }
            { OFF }       { EXTRACT }
```

Figure 4-17.1 DIRECTORY Directive Format

By default, the DISPLAY and EXTRACT directives build a directory for every file they create. Therefore, every dataname put on a DISPLAY or EXTRACT file is known for the life of that file as a described item. The DIRECTORY directive is used to turn this feature off or back on for either directive. When the directory is turned off for a directive, any items on the file created by that directive will not be accessible unless a subsequent DESCRIBE directive is entered for that file. A DIRECTORY directive remains in effect until another DIRECTORY directive is entered or the end of the Query Update session is encountered. It can, however, be overruled for individual DISPLAY and EXTRACT directives by the WITH DIRECTORY clause. (Refer to descriptions of DISPLAY and EXTRACT.)

DISPLAY

The DISPLAY directive retrieves and displays information at a terminal or upon a designated file. The information can be retrieved from the data base, the Query Update temporary storage area, or a source other than the data base (for example, a file established by the DESCRIBE directive). The format of the DISPLAY directive is shown in figure 4-18.

Components of information are retrieved, explicitly stated arithmetic operations are performed, and a listing is prepared at the terminal or upon the named file. The DISPLAY directive can accept literals and retrieve temporary items or register contents. Permanent items are retrieved from the data base according to their record keys, as records are selected by an associated IF directive or relation, or as records are selected in a pass of the entire data base if no IF condition was specified or no KEY option used. Each component of a displayed line is preceded by a space.

When multiple lines of information are sent to a terminal, the output from the DISPLAY directive is organized into pages. The first page, one full screen or 14 lines on a teletypewriter, is transmitted automatically. Query Update displays the following message and waits for the user to respond:

```
MORE . . . ANSWER Y OR N
```

Subsequent pages are transmitted one at a time in response to a user-entered Y (or YES). Any character other than Y (or YES), except the + sign or the = sign, terminates the DISPLAY directive after the page has been transmitted. If a + or = sign is entered, Query Update waits for the user to enter data since these are continuation characters. The paging facility can be disabled by specifying the parameter PL=0 in the Query Update control statement. When the paging facility is disabled, all lines of output are transmitted for display.

The DISPLAY directive in its simplest form consists of the keyword DISPLAY followed by either an expression or the word SAME. The expression can be a list of data names to be displayed, or it can be an arithmetic operation to be performed before displaying the result. The results of floating-point operations are displayed as rounded to the default picture specification. Refer to the DEFINE directive for the specifications. The word SAME is used when the previous DISPLAY directive expression list is to be repeated.

The UPON option directs the output of the DISPLAY directive to a sequential file containing records formatted exactly like those directed to the terminal except that the space preceding each component of a line is omitted when the SEPARATOR ITEM-SIZE directive is in effect. This type of sequential file is required by the FROM and USING options of the MODIFY, REMOVE, and STORE directives, as well as being a source of keys for the KEY IN option of the DISPLAY directive. If the UPON option is omitted, output is directed to the terminal. The file must be positioned by the user with REWIND statements, or by other means prior to use. Any item that is the subject of the DISPLAY UPON directive becomes a described item unless it is renamed with the AS clause.

The WITH DIRECTORY option controls the building of a directory for the file created by the DISPLAY directive. WITH DIRECTORY is valid only when UPON is also specified. It overrides the directory state that is set by default upon entering Query Update or that is set by the DIRECTORY directive. If either WITH DIRECTORY or WITH DIRECTORY ON is specified, a directory is built for the UPON file.

```
DISPLAY [UPON file-name-1 [WITH DIRECTORY [ON ]]]     [KEY  { literal         }]
                                          [OFF]       [     { data-name-1     }]
                                                      [     { IN file-name-2  }]
                                                      [                        ]
                                                      [FROM file-name-3        ]

{ {expression   [AS data-name-2]††}  ... }
{ SAME                                   }

†Not applicable to IMF
```

Figure 4-18. DISPLAY Directive Format

If WITH DIRECTORY OFF is specified, no directory will be built by this directive; therefore, the items displayed upon the file cannot be accessed as described items.

The KEY option is used to indicate the key for a specific record that is to be the source of data for the DISPLAY directive. The KEY option cannot be used when the specified area file is a sequential file. The record key is indicated by one of three options:

literal

Designates the key value for a specific record.

data name

Indicates the temporary data item that contains the key of a specific record.

IN file-name-2

Specifies the sequential file that contains the record keys. Each record in the file contains one record key, left-justified in display format.

If the KEY IN option is specified and Query Update cannot determine the length of the record, Query Update assumes that the record contains the key only and that the record length is the pictured size of the key. Query Update can determine the record length if the file is written during the current session, if the file directory is established by a DESCRIBE directive, or if the record length is specified in a FILE control statement.

A transmission containing the DISPLAY KEY IN file-name-2 directive can contain no other directives that reference area items. When a transmission contains DISPLAY KEY literal or DISPLAY KEY data-name and other directives that reference area items, the record specified by KEY is retrieved. All directives in the transmission are performed using this record. When the DISPLAY KEY directive is used in conjunction with an IF directive, the record can be displayed only when the condition is true.

The FROM option identifies the source to be displayed when it is other than the data base or temporary items from working storage. A directory is required for the designated file; it must be produced by a EXTRACT, DISPLAY UPON, or DESCRIBE directive. For more information see the External File Organizations subsection.

The AS option can be used only when UPON file-name is used. The option renames an item to be displayed. If literals, expressions, or Query Update registers are not renamed with the AS clause, the default name FILLER is assigned and the item is not accessible.

The DISPLAY directive converts elementary data name contents from their storage type (COMPLEX, DOUBLE, and so forth) to a form suitable for display purposes according to the picture specifications. Group data name content is not converted from its storage type: therefore, when group data name content is not in display code, the display is unintelligible. The diagnostic message REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT informs the user when this condition exists. The maximum number of characters that can be displayed for each data name is 4095. If a data name contains more than 4095 characters, only the first 4095 characters are displayed and an informative message is issued.

The DISPLAY directive can be used in retrieving data from a data base relation. All names appearing in a query of a relation must be either temporary data items or data items in records joined in a relation. Query Update issues a diagnostic and rejects the transmission if data items that are not related or not temporary are referenced. If more than one relation contains the data names specified in the directive, Query Update asks the user to specify which relation is to be used. Query Update prompts the user with a list of relation names and the prompting symbols >>, and waits for the user to enter one of the relation names from the list. In batch mode, if an ambiguity exists and Query Update cannot determine which relation to use, Query Update issues a diagnostic and transmission is rejected.

The data from a relation is displayed to the user as a collection of projected record occurrences. The values supplied by a record occurrence are reused for each dependent record. Thus, if record occurrence A from area A has dependent record occurrences B1 and B2 in area B, the record occurrence A is used in displaying A and B1 as the first projected record occurrence and is reused in displaying A and B2 as the second projected record occurrence.

DUPLICATE

The DUPLICATE directive copies information recorded in one Query Update catalog into another Query Update catalog. One catalog is a permanent file that has been attached with the VERSION directive; the other catalog is the default catalog ZZZZZQ2. Before the DUPLICATE directive can be used, the VERSION directive must attach a permanent file. The format of the DUPLICATE directive is shown in figure 4-19.

The UPON DEFAULT option causes information to be duplicated from the permanent file catalog (ZZZZZQ2). When this option is omitted, duplication is from the default catalog (ZZZZZQ2) to a permanent file catalog (attached by the VERSION directive).

A report format is duplicated by specifying REPORT and report-name-1. The AS option causes the duplicated report to be renamed in the receiving catalog. The entire report format is copied into the receiving catalog.

```
DUPLICATE [UPON DEFAULT]

 { REPORT  report-name-1  [AS  report-name-2]                             }
 {                                                                        }
 { SESSION session-id-1  [transmission-id-1 [TO transmission-id-2]]       }
 {                          [AS  session-id-2 [transmission-id-3]]        }
 {                          [BY  increment-size]                          }
```

Figure 4-19. DUPLICATE Directive Format

A session is duplicated into the receiving catalog by specifying SESSION and session-id-1. If no other options are included, the entire session is duplicated as originally recorded. One or more of the following options can be specified to rename a duplicated session or to duplicate a portion of a session.

transmission-id-1

> Causes a single transmission to be duplicated with the same identifying number.

TO transmission-id-2

> Allows a portion of a session to be duplicated in the receiving catalog. The duplicated session transmissions are numbered sequentially starting at transmission-id-1.

AS session-id-2

> Causes a session to be renamed in the new (receiving) catalog. Transmissions are numbered as originally recorded by default.

transmission-id-3

> Causes a session or part of a session to be copied to an existing session. The new transmissions are numbered sequentially starting with transmission-id-3. If the space in session-id-2 is filled before all the transmissions have been duplicated, a diagnostic is issued and the DUPLICATE operation is terminated.

The BY increment size option specifies the value to be added to the transmission id of each duplicated transmission to determine the next transmission id. If transmission-id-3 is not specified, the increment number becomes the first new transmission id. The increment size must be an integer greater than zero and can be up to three digits in length. The default increment size is one. A diagnostic message is issued if the increment size is not in the range 1 through 999.

Execution of the DUPLICATE directive is prevented and diagnostics are issued under the following conditions:

> The report name, session id, or transmission id cannot be found in the sending catalog.
>
> The report name already exists in the receiving catalog.
>
> The transmission length associated with the receiving catalog is less than the transmission length of the sending catalog.

An attempt to overwrite an existing transmission id terminates the execution of a DUPLICATE directive.

END

The END directive terminates Query Update operations and returns control to the operating system. The format of the END directive is shown in figure 4-20.

```
END
```

Figure 4-20. END Directive Format

The END directive closes an open catalog file, extends (if modified) and returns a VERSION attached catalog, and returns the subschema attached by the INVOKE directive. Control is returned to the operating system for subsequent processing at which time default catalog, created area, and report dispositions are determined by the user. Refer to the applicable system reference manual for file disposition procedures.

ERASE

The ERASE directive removes a temporary data name, eliminates a directive from the current catalog, or eliminates a directory created by a DESCRIBE, DISPLAY UPON, or EXTRACT directive. Eliminated directives are displayed after the ERASE directive is executed. The format of the ERASE directive is shown in figure 4-21.

Data names that have been established by DEFINE or SPECIFY directives can be eliminated by the ERASE directive. One or more data names can be specified to be erased. A single item in a matrix cannot be erased. The entire matrix can be erased by specifying the ERASE directive with the subscript ALL. A literal cannot be erased with the ERASE directive.

```
       ( data-name . . .                                                  )
       (                         { ALL              }                     )
       ( REPORT report-name      { layout-directive }                     )
ERASE  (                                                                  )
       (                         { ALL                                   })
       ( SESSION session-id      { transmission-id-1 [TO transmission-id-2]})
       (                                                                  )
       ( DESCRIBE file-name                                               )
```

Figure 4-21. ERASE Directive Format

Report directives in the current catalog can be erased by specifying REPORT. The use of this form of ERASE establishes the specified report as the current report. Any report directives that immediately follow the ERASE REPORT directive are added to the report. The complete report format is erased by including the report name and ALL. Specific directives in the report format are eliminated by specifying the report name and one of the following layout directives:

```
BREAK(n)
DATE
DETAIL(n)
EVALUATE BEFORE (or AFTER)
FOOTING(n)
HEADING(n)
MOVE BEFORE (or AFTER)
PAGE-NUMBER
PAGE-SIZE
PREFACE
RECAP
SELECT(n)
SUMMARY
TABS
TIME
TITLE
```

Erasure of a specific numbered layout directive occurs when the directive and its number (n) are stated; otherwise, all similar directives are removed.

Session directives in the current catalog are erased by specifying SESSION. The entire session is erased by including the session id and ALL. Specific directives in the cataloged session are eliminated by specifying the session id and the transmission id for the particular directive. A range of transmission ids can be specified by including the TO option. All session directives with a transmission id within the designated range are erased.

The use of the ERASE directive is illustrated in the following examples:

ERASE REPORT report-name SELECT 5

> Erases only the SELECT 5 directive from the specified report.

ERASE REPORT report-name SELECT

> Erases all SELECT directives in the named report.

ERASE SESSION session-id 131

> Erases the directive with transmission-id 131 in the specified session.

ERASE SESSION session-id 131 TO 145

> Erases all directives with transmission ids in the range from 131 to and including 145 in the specified session.

A directory established by a DESCRIBE, DISPLAY UPON, or EXTRACT directive can be eliminated by specifying DESCRIBE in the ERASE directive. The file name specified in this option is the logical file name of the file described the directory.

EVALUATE

The EVALUATE directive performs arithmetic operations to determine the value of a temporary data item or a cumulative function. Two different formats are available for this directive. Format 1 is used for general data manipulative operations; format 2 is used for report-oriented operations. The formats of the EVALUATE directive are shown in figure 4-22.

```
Format 1

     EVALUATE { data-name           } ...
              { cumulative-function }

Format 2

                           / REPORT                  \
                          |  TITLE                    |
                          |  RECAP                    |
                          |  ANY SELECT               |
                          |  SELECT tag-number        |
                          |  ANY DETAIL               |
     EVALUATE { BEFORE }  |  DETAIL tag-number        |  { data-name           } ...
              { AFTER  }  |  ANY BREAK                |  { cumulative-function }
                          |  BREAK level-number       |
                          |  ANY HEADING              |
                          |  HEADING level-number     |
                          |  ANY FOOTING              |
                          |  FOOTING level-number     |
                           \ NO SELECT               /
```

Figure 4-22. EVALUATE Directive Format

A temporary data name in the Query Update working storage area can be evaluated with this directive. The method of evaluation must be specified through the VALUE (or =) option in the DEFINE directive. The data name specified must refer to items defined by a DEFINE directive.

A cumulative function, such as SUM or COUNT, must be specified in an EVALUATE directive before its results are available; to display the result, a DISPLAY directive specifying the cumulative function and the data name must be used.

When format 1 is used, conditional evaluation can be designated by using the EVALUATE directive in conjunction with an IF directive.

Format 2 causes evaluation to be performed before or after one of the report production steps. Evaluation can occur before or after the entire report; before or after a title, recap, heading, or footing line; or before or after a SELECT, DETAIL, or BREAK directive is executed. If an EVALUATE directive is used in conjunction with a MOVE directive (both format 2), evaluation is performed before the MOVE. Refer to the SELECT directive for additional information on evaluation in conjunction with the SELECT and DETAIL directives. The BEFORE and AFTER NO SELECT options represent two widely separated time situations:

BEFORE NO SELECT

> When no select criteria testing is performed to generate a detail line, evaluation occurs before each line is generated. This option is used only when one unnumbered DETAIL directive is specified.

AFTER NO SELECT

> When all criteria testing has been performed and no select criteria has been specified, the evaluation is performed. This option is used only when numbered DETAIL directives are specified.

When format 2 is used, any item specified in the expression or condition that establishes the value of a data name specified in the EVALUATE directive must be a temporary item, a literal, or an item from the source file for the report. An item that establishes the value of a cumulative function specified in the EVALUATE directive can only be an item from the source file for the report.

A preceding FORMAT or ALTER directive must be in effect to use format 2 of the EVALUATE directive.

EXECUTE

The EXECUTE directive causes a procedure external to Query Update to be executed. This directive can be specified in conjunction with an IF condition that uses temporary data items only. This directive must not be specified with any other directives in a transmission. The format of the EXECUTE directive is shown in figure 4-23.

Query Update loads and executes the named procedure. The procedure name must be one through seven characters; a colon cannot be included in the name. The EXECUTE directive does not support input/output operations within the external procedure.

The USING option allows a total of 64 data names or literals to be passed to the procedure. A data name must be a temporary item established by a DEFINE directive. A list containing addresses of the parameters is generated after the procedure is loaded. This list is a standard FORTRAN parameter list. Register A1 contains the address of the parameter address list. The entire parameter list is terminated by a word of zeros.

The FROM option specifies the file name to be used as the source of the external procedure; the procedure must be in a user sequential relocatable file. If this option is not included in the EXECUTE directive, the procedure is loaded from the DMSLIB library.

EXHIBIT

The EXHIBIT directive displays information that Query Update uses in performing its operations. Information that can be displayed includes default report specifications, working storage data names, report names with associated report specifications, session ids with associated directives, and relations with associated record names. There are two formats for the EXHIBIT directive, as shown in figure 4-24.

The type of information displayed depends upon the options specified. Figure 4-25 illustrates the EXHIBIT directive under CRM and CDCS data base access modes, with various options specified. When no options are included in the directive, Query Update exhibits the maximum transmission length; the transmission length of the catalog file; the universal character; the current literal delimiter and ITEM-SIZE if it is in effect; the current subschema, subschema library, and area names; and the current or default maximum number of lines,

```
EXECUTE procedure-name [USING {data-name} . . .]  [FROM  file-name]
                              {literal  }
```

Figure 4-23. EXECUTE Directive Format

columns, sections, and images permitted in a report. The subschema, subschema library, and area names are those established by the current INVOKE or CREATE directive. Examples 6 through 10 in figure 4-25 illustrate the EXHIBIT directive under the IMF data base access method. An EXHIBIT of an item that has been described includes the file name for which the item is defined.

CDCS/CRM Format

```
                  || data-name [OF file-name]                                              ||
                  || REPORTS [report-name [layout-directive] ]                             ||
   EXHIBIT        || SESSIONS [session-id [transmission-id-1 [TO transmission-id-2] ] ]    ||
                  || TEMPORARY                                                             ||
                  || RELATION                                                              ||
```

IMF Format

```
                  || data-name [OF file-name]                                              ||
                  || REPORTS [report-name [layout-directive] ]                             ||
                  || SESSIONS [session-id [transmission-id-1 [TO transmission-id-2] ] ]    ||
   EXHIBIT        || TEMPORARY                                                             ||
                  || RELATION                                                              ||
                  || ACCESS [access-path-name]                                             ||
                  || COSETS [coset-name]                                                   ||
```

Figure 4-24. EXHIBIT Directive Format

Example 1

```
--       EXHIBIT
MAXIMUM TRANSMISSION LENGTH    1030
TL OF CATALOG FILE    1030
SEPARATOR $
UNIVERSAL OFF
MAX NUMBER OF LINES 060
MAX NUM. OF COLUMNS 136
MAX NO. OF SECTIONS 010
MAX IMAGES PER PAGE 004
AREA NAME(S):
   QUPOPK
   QUPODH
SUBSCHEMA NAME = QUPO
SUBSCHEMA LIBRARY NAME = QUPOLIB
    ID = QUTEST
```

The subschema, subschema library, and area names from the current INVOKE directive are shown.

Example 2

```
--        EXHIBIT QUPODH
RECORD NAME IS     DAY-HIKES
ALT KEY  T-NAME
KEY IS  DATE-OF-HIKE
AREA PF NAME = QUPODH
    ID = QUTEST
    PW    = *---*
INDEX PF NAME = QUPOWK
    ID = QUTEST
```

The EXHIBIT directive includes the area name QUPODH. The name of the record in the area is shown along with the key, alternate key, and permanent file names with parameters (except passwords) for the area and index file.

Figure 4-25. Sample EXHIBIT Directives (Sheet 1 of 3)

Example 3

```
--       EXHIBIT DAY-HIKES
02 (CRP) TRAIL-NAME
03 (ALT) T-NAME
03 (ELM) MILEAGE
03 (ELM) NO-OF-HOURS
03 (ELM) HIKE-RATING
03 (ELM) HIGHEST-PT
02 (KEY) DATE-OF-HIKE
02 (GRP) GROUP-MEMBERS
03 (ELM) INITIALS
02 (ELM) SCENIC
02 (ELM) DIFFICULTY
```

DAY-HIKES is the data name of a record. The level number and data name for the record key, alternate key, elementary items, and group items are displayed.

Example 4

```
--       EXHIBIT TRAIL-NAME OF DAY-HIKES
TRAIL-NAME OF DAY-HIKES
OCCURS   00003
CONTAINS T-NAME
CONTAINS MILEAGE
CONTAINS NO-OF-HOURS
CONTAINS HIKE-RATING
CONTAINS HIGHEST-PT
```

This example requests the group item TRAIL-NAME to be exhibited. The data name of each elementary item in TRAIL-NAME is displayed.

Example 5

```
--       EXHIBIT RELATION
TRAILROUTES RELATES THE RECORDS:
   BACK-PACK-TRAILS IN QUPOPK
   DAY-HIKES IN QUPODH
   POSSIBLE-HIKES IN QUPOPH
```

The RELATION option lists the name of the relation joining three areas and the associated record names and area names

Example 6

```
EXH
MAXIMUM TRANSMISSION LENGTH      1030
FL OF CATALOG FILE    1030
SEPARATOR  $
UNIVERSAL OFF
MAX NUMBER OF LINES  060
MAX NUM. OF COLUMNS  136
MAX NO. OF SECTIONS  010
MAX IMAGES PER PAGE  004
COBOL    EXTERNAL SCHEMA = ECONOMY
CONCEPTUAL SCHEMA = WORLD
METADB = GLOBAL
 UN/ID = JMD
RECORD NAME(S):
   COUNTRY-SENTENCE
   PRODUCT-SENTENCE
   RIVER-SENTENCE
   PC-LINK
   RC-LINK
```

The external schema, conceptual schema, and record names from the current IMF INVOKE directive.

Figure 4-25. Sample EXHIBIT Directives (Sheet 2 of 3)

Example 7

```
EXH COUNTRY-SENTENCE
PERMISSIONS:
   ALL
ACCESS:
   BY-COUNTRY
ITEMS:
   COUNTRY
   CAPITAL
   CONTINENT
   SIZE
   POPULATION
```

The EXHIBIT directive includes the IMF record name COUNTRY-SENTENCE. The permissions access paths and data item names.

Example 8

```
         EXH ACCESS
ACCESS TO COUNTRY-SENTENCE
   BY-COUNTRY
ACCESS TO PRODUCT-SENTENCE
   BY-PRODUCT
ACCESS TO RIVER-SENTENCE
   BY-RIVER
ACCESS TO PC-LINK
   BY-PRODUCT-LINK
   BY-COUNTRY-LINK-IN-PC
   BY-PC
ACCESS TO RC-LINK
   BY-RIVER-LINK
   BY-COUNTRY-LINK-IN-RC
   BY-RC
```

The ACCESS option lists every record in the IMF external schema and every access path available to the record.

Example 9

```
         EXH ACCESS BY-PC
BY-PC TO RECORD PC-LINK
SEARCH KEYS(S):
   PRODUCT-LINK
   COUNTRY-LINK
```

Record name and search keys are listed when an ACCESS path name is specified.

Example 10

```
         EXH COSET
PRODUCES
   OWNER  COUNTRY-SENTENCE
   MEMBER PC-LINK
COMES
   OWNER  PRODUCT-SENTENCE
   MEMBER PC-LINK
RECEIVES
   OWNER  COUNTRY-SENTENCE
   MEMBER RC-LINK
FLOWS
   OWNER  RIVER-SENTENCE
   MEMBER RC-LINK
```

The COSET option lists all cosets in the IMF external schema and their owner and member record names.

Example 11

```
         EXH COSET PRODUCES
PRODUCES
   OWNER  COUNTRY-SENTENCE
   MEMBER PC-LINK
OWNER MATCHING ITEMS:                   MEMBER MATCHING ITEMS:
   COUNTRY                                COUNTRY-LINK
```

Owner record names, member record names, and matching items are listed when a coset name is specified.

Figure 4-25. Sample EXHIBIT Directives (Sheet 3 of 3)

The data name option causes a display of information related to the type of data name specified. Data names that can be specified and the information displayed for each type are as follows:

Data Name	Display
area name	Record names in the area, primary, alternate, and major keys for each record, index file name, and permanent file parameters for the area and index file with the exception of passwords which are replaced by the symbol *---*.
record name	Level numbers, item names, and group, elementary, primary key, alternate key, or major key indicators.
relation name	Join terms in a relation and the associated record names, in CRM data base access mode. Joined record names, in CDCS data base access mode.
group item name	Number of occurrences and elementary item names within the group.
elementary item name	Size, type, and possible qualifiers up to the record name.
temporary item name	Size and type of the temporary data item.

The REPORTS option lists all report names recorded on the current catalog. If a report name is specified, all layout directives associated with the report name are exhibited. A specific directive can be displayed in its entirety by including a layout directive in this option; only the directive specified is displayed. Layout directives that can be requested are:

BREAK(n)
DATE
DETAIL(n)
EVALUATE BEFORE (or AFTER)
FOOTING(n)
HEADING(n)
MOVE BEFORE (or AFTER)
PAGE-NUMBER
PAGE-SIZE
PREFACE
RECAP
SELECT(n)
SUMMARY
TABS
TIME
TITLE

The SESSIONS option lists session ids that are available in the current catalog. The directives associated with a specific session id are displayed when the session id is included in this option. The transmission-id-1 and TO transmission-id-2 options cause Query Update to locate and list the specifically numbered directives.

The TEMPORARY option produces a listing of all currently known temporary data names from DEFINE directives and condition names from SPECIFY directives.

The RELATION option lists the names of the relations and the records available to the user. The relation name is listed first, followed by the names of the related records and their associated area names.

The ACCESS option lists all access paths in an IMF external schema listed by record name. If an access path name is specified, the record name and search keys are exhibited.

The COSET option lists all cosets in an IMF external schema and their owner and member record names. If a coset name is specified, the owner and member record names and matching items are exhibited.

EXTRACT

The EXTRACT directive creates a subset (a sequential file) of a data base area or a database relation and optionally a directory to the subset. The file is not rewound before or after the EXTRACT directive. The format of the EXTRACT directive is shown in figure 4-26.

The WITH DIRECTORY option controls the building of a directory for the file created by the EXTRACT directive. It overrides the directory state that is set by default upon entering Query Update or that is set by the DIRECTORY directive. If either WITH DIRECTORY or WITH DIRECTORY ON is specified, a directory is built for the UPON file. If WITH DIRECTORY OFF is specified, no directory is built by this directive; therefore, the items displayed upon the file can not be accessed as described items.

Named fields in the expression can be retrieved from records described in an external schema, a DESCRIBE directory, or from temporary data items. If the expression is a record name, the entire record is retrieved from the data base and placed in the specified file. When only the record name has been specified in the EXTRACT directive, the names of the individual data items are not transferred to the temporary directory. Consequently, data item fields cannot be referenced individually. The DESCRIBE directive can be used to generate a new directory for the extracted file,

```
EXTRACT UPON file-name [WITH DIRECTORY [ON ]]  {expression [AS data-name]} ...
                                       [OFF]
```

Figure 4-26. EXTRACT Directive Format

so that the data item fields can again be referenced individually.

Data items from specific records can be selected by using the EXTRACT directive with an IF directive.

The fields (or record) retrieved from the data base records can be renamed in the extracted file by including the AS option. The name entered as data-name-1 is then used to access the field (or record) in the extracted file. If the AS option is not included, literals, registers, and expressions will be given the default name of FILLER, and subsequent directives that specify data-name refer to the extracted data item; the data expression item is not accessible until the directory is eliminated.

Items in a matrix can be extracted individually or as a whole matrix. As long as the matrix items are contiguous in memory, a matrix can be retrieved from the data base by using the EXTRACT directive with the matrix name subscripted with ALL. In the extracted file, the items form a matrix; therefore, any reference to an item in a subsequent directive can specify the matrix name with a subscript. When a matrix item is retrieved individually, it is placed in the specified file as an individual item, not a part of a matrix. If only a single matrix item is extracted and not renamed, a reference to the extracted item in a subsequent directive must specify the matrix name without subscripts. If a number of matrix items are extracted, each must be extracted individually and renamed using the AS data-name-1 option. If one of these renamed items is referenced in a subsequent directive, the item must be specified by the name entered as data-name-1 in the EXTRACT directive.

More than one EXTRACT, DESCRIBE, or DISPLAY UPON directive can be in effect at a given time. Each succeeding EXTRACT, DESCRIBE, or DISPLAY UPON directive generates another temporary directory.

When an EXTRACT directive is used to create a subset of a data base relation, all data items named must be in records joined in one relation. Query Update issues a diagnostic and rejects the transmission if the data items specified are not related. If more than one relation contains the data names specified in the directive, Query Update asks the user to specify which relation is to be used. Query Update prompts the user with a list of relation names and the prompting symbols >>, and waits for the uєer to enter one of the relation names from the list. In batch mode, if an ambiguity exists and Query Update cannot determine which relation to use, Query Update issues a diagnostic, and the transmission is rejected.

FOLLOW

The FOLLOW directive, shown in figure 4-27, applies only to the IMF data base access method. The directive is used to specify a navigation strategy to allow access of more than one record type in a transmission. The records must be related through cosets specified in the external schema.

Navigation begins at the record type of the access path stated and then proceeds to the member or owner record type of the coset(s) stated. A maximum of 13 cosets can be stated; 14 record types are possible in a navigation route. The FOLLOW directive applies only to the DISPLAY and EXTRACT directives. All other Query Update/IMF directives can access only one record type.

The navigation path specified remains in effect until another FOLLOW directive is given, FOLLOW DEFAULT is specified, or the current external schema is no longer in effect.

Refer to the Information Management Facility Version 1 Applications Programming Reference Manual for complete information on the FOLLOW directive.

FOOTING

The FOOTING directive is a report specification that indicates the content and positioning of a footing. A footing is associated with a BREAK directive and appears in a report when the associated break situation occurs. The format of the FOOTING directive is shown in figure 4-28.

```
IMF Format

          { path-name [ { MEMBER } {coset-name} ] ... }
FOLLOW    {             { OWNER  }                    }
          { DEFAULT                                   }
```

Figure 4-27. FOLLOW Directive Format

```
FOOTING level-number { [AT LINE { integer-1 [BEYOND] } ] }
                     {          { data-name          }   }

{ IS expression [CENTERED] [IN { NEXT TAB         } ] } ...}...
{                              { TAB integer-2    }   }
{                              { COLUMN integer-3 }   }
```

Figure 4-28. FOOTING Directive Format

Each footing is assigned a level number. Level number 0 associates a footing with the end of data; it can be used to obtain totals for the entire report. A footing with level number 0 appears in the report after all input data has been processed. Level numbers 1 through 63 link the footing to a BREAK directive with a matching level number. A footing linked to a conditional break appears after the condition is satisfied. A footing linked to a BREAK ON ITEM data name appears when data name content changes. The previous content of data name appears in the footing.

The AT LINE option specifies the vertical position of the footing. An absolute line number (AT LINE integer-1) places the footing on the specified line on the report page. A relative line number (AT LINE integer-1 BEYOND) positions the footing the specified number of lines after the largest line number previously written on the report page.

(The meaning of relative line number differs for footing and detail lines. For more information see the description of the DETAIL directive.) A data name can be specified in place of integer-1; it must be an integer working storage data name previously established by a DEFINE directive. If data name evaluation is involved, the user is responsible for assuring that an EVALUATE directive has been executed for data name before it is required for footing placement. A multiple line footing is specified by repeating the AT LINE option followed by other elements of the directive. If the AT LINE option is not included in the directive, the footing is positioned one line beyond the largest line number previously written on the report page.

The content of a footing line is specified by one or more IS clauses. Multiple IS clauses are used to indicate the horizontal positioning of specific data; the IN option is then included with each IS clause. The expression in an IS clause specifies data that is to appear in the footing line; it identifies one or more data fields, literals, or defined items. With multiple IS clauses, the optional elements are specified for each expression to which they apply. A data name in an expression can contain only that information accumulated up to the point where the footing occurs; if evaluation is involved, the user is responsible for assuring that the EVALUATE directive has been executed.

The CENTERED option causes the expression to be automatically positioned midway between two horizontal points of column 1 and the page width are assumed. A horizontal point specified by the IN clause represents the left point; the right point is the next tab or the page width. Centering is applied only to the expression associated with the CENTERED option.

The IN clause specifies the horizontal positioning of the footing. The option selected specifies either the starting point of the footing or the left point to be used for centering.

NEXT TAB

> The next tabular position after the current position; TAB 1 is assumed if a previous tab has not been specified in the directive.

TAB integer-2

> The column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

> The print position specified by integer-3.

As detail lines are being written, the system considers page size (as specified in the PAGE-SIZE directive) and the space required for any recap or footing lines. When the point on the page is reached where a page overflow condition becomes possible, the system scans ahead to determine the volume of data to be processed without causing a page overflow condition. Data is then formatted and the page ends before an overflow occurs.

FORMAT

The FORMAT directive causes the subsequent report specification directives to be retained in the current catalog under a specified report name. If the current catalog is the default catalog, it is the user's responsibility to ensure retention of the report directives after Query Update is terminated. The FORMAT directive cannot be specified for a catalog file that was created with a smaller transmission length than is currently set. The format of the FORMAT directive is shown in figure 4-29.

```
FORMAT report-name
```

Figure 4-29. FORMAT Directive Format

The report name specified is established as the current report; report name must be unique in the current catalog. All report specification directives (layout directives) that are entered after the FORMAT directive are retained in the current catalog under the report name. When a directive that is not a report specification is entered, Query Update assumes that format specifications have ended and no additional directives are retained under the specified report name. Report specifications can be modified or added to the format by using the ALTER or ERASE directive.

A PREPARE or PREVIEW directive that specifies report name produces a report according to the specifications associated with report name. Disposition of this report, which resides in a local file named report name, is the user's responsibility. Refer to the applicable system reference manual for file disposition procedures.

HEADING

The HEADING directive is a report specification that indicates the content and positioning of a heading. A heading is associated with a BREAK directive and appears in a report when the associated break situation occurs. The format of the HEADING directive is shown in figure 4-30.

Each heading is assigned a level number. Level number 0 associates a heading with the beginning of data; it can be used to initiate a first page prefatory heading. A heading with level number 0 appears in the report before any input data is processed. Level numbers 1 through 63 link the heading to a BREAK directive with a matching level number. A heading linked to a conditional break appears after the condition is satisfied. A heading linked to a BREAK ON ITEM data name always appears on the first page of a report and reappears when data name content changes. The new data name content appears in the heading.

The AT LINE option specifies the vertical position of the heading. An absolute line number (AT LINE integer-1) places the heading on the specified line of the report page. A relative line number (AT LINE integer-1 BEYOND) positions the heading the specified number of lines after the largest line number written on the report page by previous report directives. (The meaning of relative line number differs for heading and detail lines. For more information see the description of the DETAIL directive.) At the beginning of a page, the heading appears the number of lines below the page start or the title line. The ALL option forces a heading to appear on each new page.

A data name can be specified in place of integer-1 in the AT LINE option. The data name must be an integer working storage data name established by a DEFINE directive. If data name evaluation is involved, the user is responsible for assuring that an EVALUATE directive has been executed for data name before it is required for heading placement. A multiple line heading is specified by repeating the AT LINE option followed by other elements of the directive. If the AT LINE option is omitted, the heading is positioned one line beyond the preceding line. If the ON ALL PAGES option is specified, it must be specified for every line number in the HEADING directive.

The content of a heading line is specified by one or more IS clauses. Multiple IS clauses are used to indicate the horizontal positioning of specific data; the IN option is then included with each IS clause. The expression in an IS clause specifies data that is to appear in the heading line; it identifies one or more data fields, literals, or defined items. With multiple IS clauses, the optional elements are specified for each expression to which they apply. A data name in an expression can contain only that information accumulated up to the point where the heading occurs; if evaluation is involved, the user is responsible for assuring that the EVALUATE directive has been executed.

The CENTERED option causes the expression to be automatically positioned midway between two horizontal points. If no points are specified by an IN clause, default horizontal points of column 1 and the page width are assumed. A horizontal point specified by the IN clause represents the left point; the right point is the next tab or the page width.

Centering is applied only to the expression associated with the CENTERED option.

The IN clause specifies the horizontal positioning of the heading. The option selected specifies either the starting point of the heading or the left point to be used for centering.

NEXT TAB

The next tabular position after the current position; TAB 1 is assumed if a previous tab has not been specified in the directive.

TAB integer-2

The column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

The print position specified by integer-3.

```
                      ⎧                                                  
HEADING level-number  ⎨ [AT LINE {integer-1 [BEYOND]} [ON ALL PAGES]]
                      ⎩          {data-name        }

    { IS expression [CENTERED] [IN {NEXT TAB        }] } ... } ...
    {                              {TAB integer-2   }  }
    {                              {COLUMN integer-3}  }
```

Figure 4-30. HEADING Directive Format

HELP

The HELP directive causes Query Update to list the description of a directive or the explanation of a diagnostic message. This directive can also be used to list all available Query Update directives. The format of the HELP directive is shown in figure 4-31.

```
HELP  [directive-name    ]
      [diagnostic-number ]
```

Figure 4-31. HELP Directive Format

The HELP directive with neither option included causes a list of directive names to be printed. The directive names (keywords) of all available Query Update directives are listed.

The directive name option produces a brief statement of the purpose of the specified directive. The diagnostic number option lists an explanation of the diagnostic message indicated by the specified number.

IF

The IF directive is provided as the means to test data for specific conditions before initiating any action on the data. The specified condition must be satisfied or no action takes place. The format of the IF directive is shown in figure 4-32.

```
    { condition      }
IF  { condition-name }
    { SAME           }
```

Figure 4-32. IF Directive Format

The condition to be tested can be specified in the IF directive. A simple condition condition consists of a pair of expressions and a relational operator. Conditions can be compounded by joining two simple conditions with a logical operator. Expressions as well as simple and compound conditions are described in detail in section 3.

A condition name specified in the IF directive refers to a condition that has been previously established with a DESCRIBE or SPECIFY directive. The condition name can reference a simple condition or a compound condition. In addition, the IF directive can be compounded by including a condition name, a logical operator, and another condition. The condition can contain temporary items, described items, or items of the data base.

The SAME option indicates that the condition from the last IF directive is to be repeated. The condition from the most recent IF directive is retested before executing the next directive. A compound condition can be created with SAME and additional conditions. If the SAME option is used again in the next IF directive, the condition tested is the one referred to by the previous SAME plus the additional conditions that were specified. The previous SAME is treated by Query Update as if it is enclosed within parentheses. The SAME option itself must never actually occur inside of parentheses. In the following examples of IF directives,the second and third directives test the data item DIV for the value OPS and the data item DEPT for the value 29.

```
IF DIV EQ $OPS$
IF SAME AND DEPT EQ $29$
IF SAME
```

The condition indicated in the IF directive determines whether or not the associated directives are executed. Associated directives follow the IF directive and include all directives up to another IF directive or the end of the transmission. When the IF directive is the last directive in the transmission, the associated directives are in the next transmission and include all directives up to another IF directive or the end of the transmission. In the following example, the IF directive has three associated directives in the same transmission:

```
—
    IF COST EQ 0 DISPLAY $FREE$ +
    MOVE 0 TO SALES-TAX +
    DISPLAY SALES-TAX
—
```

The two transmissions in the following example show the IF directive in the first transmission and the three associated directives in the following transmission.

```
—
    IF COST NE 0
—
    DISPLAY $EXPENSIVE$ +
    MOVE 0.06 TO SALES-TAX +
    DISPLAY SALES-TAX
—
```

Records can be retrieved from the data base by the action of an IF directive or by a record key literal being included in an associated directive. The search priority used on search keys is primary, major, and alternate. The key with the highest priority will be used if more than one search key has been specified in a transmission. The structure of the IF directive determines the number of records that must be searched during record retrieval. Table 4-2 indicates the number of records that must be searched when the IF directive condition is stated in a simple relational expression as:

```
data name    relational operator    literal
```

In two instances shown in table 4-2, maximum performance is achieved.

The data name is the primary key and the relational operator is EQ (or =).

The data name is an alternate key and the relational operator is EQ (or =).

TABLE 4-2. RECORDS SEARCHED FOR RETRIEVAL

Data-Name	Relational Operator			
	EQ (=)	GT (>) GE (≥)	LT (<) LE (≤)	NE (≠)
Primary or major primary key	Records with primary or major primary key equal to the literal	Records between the literal and the end of the file	Records between the beginning of the file and the literal	All records
Alternate or major alternate key with USAGE IS DISPLAY, COMP-1, COMP-2, INTEGER, or unsigned COMP	Records with alternate or major alternate key equal to the literal	Records with alternate or major alternate key greater than, or greater than or equal to the literal	Records with alternate or major alternate key less than, or less than or equal to the literal	Records with alternate or major alternate key less than the literal or greater than the literal
Alternate or major alternate key with USAGE signed COMP or COMPLEX	Records with alternate or major alternate key equal to the literal	All records in the file	All records in the file	All records in the file
Alternate or major alternate key with USAGE IS LOGICAL or non-key	All records in the file	All records in the file	All records in the file	All records in the file
Numeric alternate key	All records in the file	All records in the file	All records in the file	All records in the file
Non-key	All records in the file	All records in the file	All records in the file	All records in the file

When the IF directive specifies an expression or a permanent data item in place of a literal, or when the literal has a numeric value other than single precision and the key is NUMERIC, INTEGER, FIXED, or FLOATING type, all records in the file are searched.

A temporary data item that contains a literal can be specified in place of a literal in the simple relational expression. The search performance is shown in table 4-2. The value of the temporary data item must not be changed while the search is in process.

When the IF directive condition is stated as a compound relational expression, the number of records to be searched is affected by the logical operator (AND, OR, XOR, or NOT) that joins the relational expressions. The following examples illustrate how search performance is affected by the logical operator. The examples assume that data-name-1 is the primary key, data-name-2 is not a key field, and literal-1 is less than literal-2.

```
IF data-name-1 = literal-1
AND data-name-2 = literal-2
```

One record is searched.

```
IF data-name-1 = literal-1
OR data-name-2 = literal-2
```

All records are searched.

```
IF data-name-1 > literal-1
AND data-name-1 < literal-2
```

All records between literal-1 and literal-2 are searched.

```
IF data-name-1 < literal-1
AND data-name-1 > literal-2
```

No search is made since no record can qualify.

The order of the conditions (relational expressions) in the IF directive can affect retrieval efficiency. Order is not important if either data-name-1 or data-name-2 is a primary key, because the records will be retrieved according to primary key. The order of conditions is important when data-name-1 and data-name-2 both designate keys, one being a major primary key, the other being the alternate or major alternate key. When these keys are involved, records are retrieved according to the key type in the first condition. Therefore, for greater efficiency, the key type causing the smaller number of records to be retrieved should be placed in the first condition.

When permanent item data names or key literals are not specified, no records need to be retrieved from the data base. Conditional testing of temporary items or literals occurs immediately and associated directives are executed when results are true.

An IF directive can have an implied data name when logical operators are used to connect more than one condition. For example, the compound relation IF A EQ 1 OR A OR 2 is interpreted by Query Update as IF A EQ 1 OR A EQ 2.

An IF directive referencing only temporary data items and literals can be used with any directive. In this case, the condition is evaluated immediately. If false, the directives associated with it are skipped without syntax checking. The following example assumes that a temporary data item ANSW has a response from the user which controls whether a report is prepared.

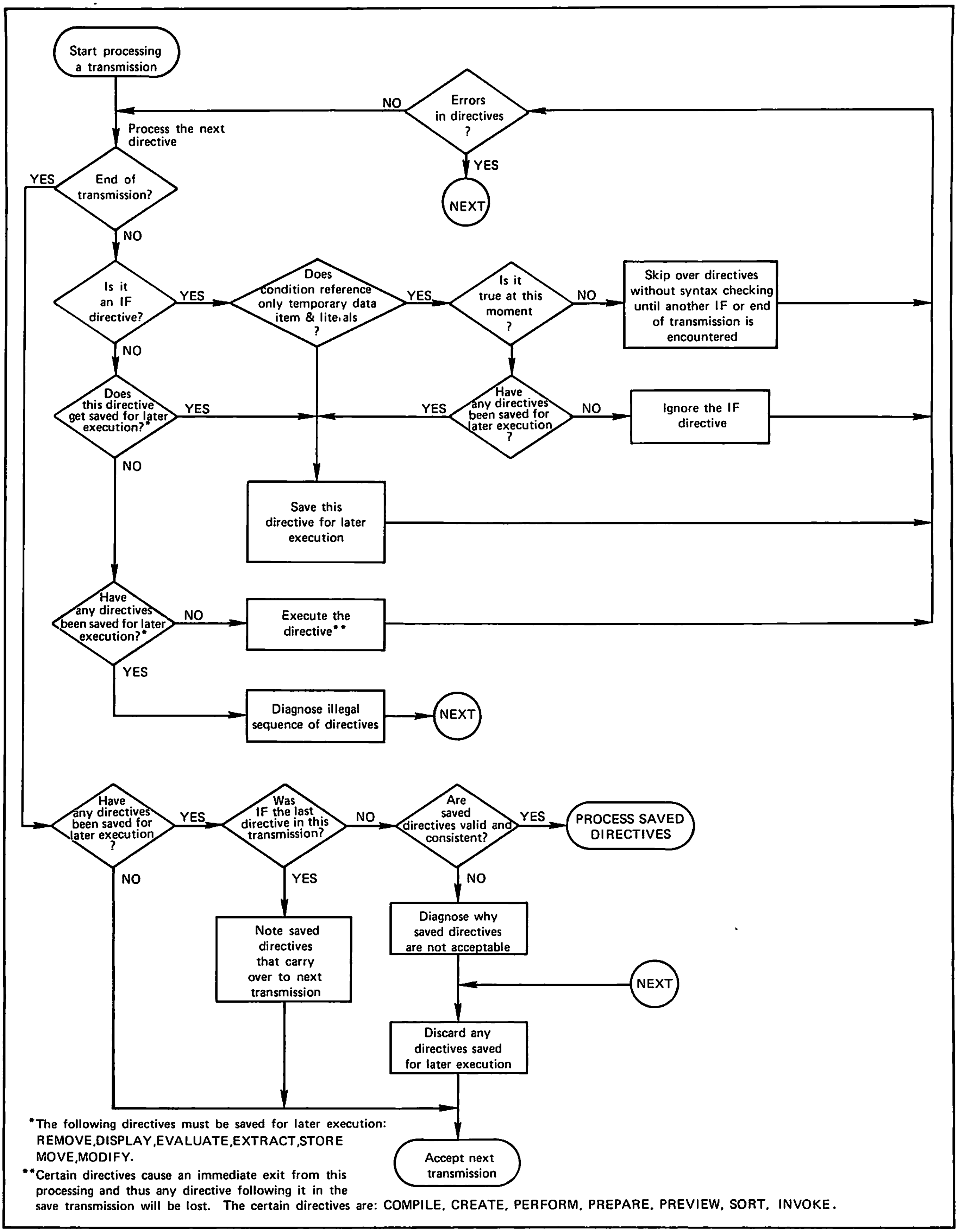
```
IF ANSW EQ $YES$ PREPARE COSTRPT FROM A
```

An IF directive can be used with another directive for selective processing. The following example would display the NAME data item only for those records that contain 123 in the DEPT data item.

```
IF DEPT = $123$ DISPLAY NAME
```

Multiple IF directives can be included in one transmission. When this occurs, Query Update tests the condition in each IF directive. If the condition being tested is true, the associated directives are executed; if the condition is not true, the associated directives are bypassed; the next IF directive is then processed.

Figures 4-33 and 4-34 illustrate the general operations of IF directive processing.


Start processing a transmission
Errors in directives ?
NO
YES
NEXT
Process the next directive
End of transmission?
YES
NO
Is it an IF directive?
YES
Does condition reference only temporary data item & lite, als ?
YES
Is it true at this moment ?
NO
Skip over directives without syntax checking until another IF or end of transmission is encountered
NO
Does this directive get saved for later execution?*
YES
YES
Have any directives been saved for later execution ?
NO
Ignore the IF directive
NO
Save this directive for later execution
Have any directives been saved for later execution?*
NO
Execute the directive**
YES
Diagnose illegal sequence of directives
NEXT
Have any directives been saved for later execution ?
YES
Was IF the last directive in this transmission?
NO
Are saved directives valid and consistent?
YES
PROCESS SAVED DIRECTIVES
NO
YES
NO
Note saved directives that carry over to next transmission
Diagnose why saved directives are not acceptable
NEXT
Discard any directives saved for later execution
Accept next transmission

*The following directives must be saved for later execution: REMOVE,DISPLAY,EVALUATE,EXTRACT,STORE MOVE,MODIFY.

**Certain directives cause an immediate exit from this processing and thus any directive following it in the save transmission will be lost. The certain directives are: COMPILE, CREATE, PERFORM, PREPARE, PREVIEW, SORT, INVOKE.

Figure 4-33. General Flow of Directive Checking and Saving Directives for Later Execution.

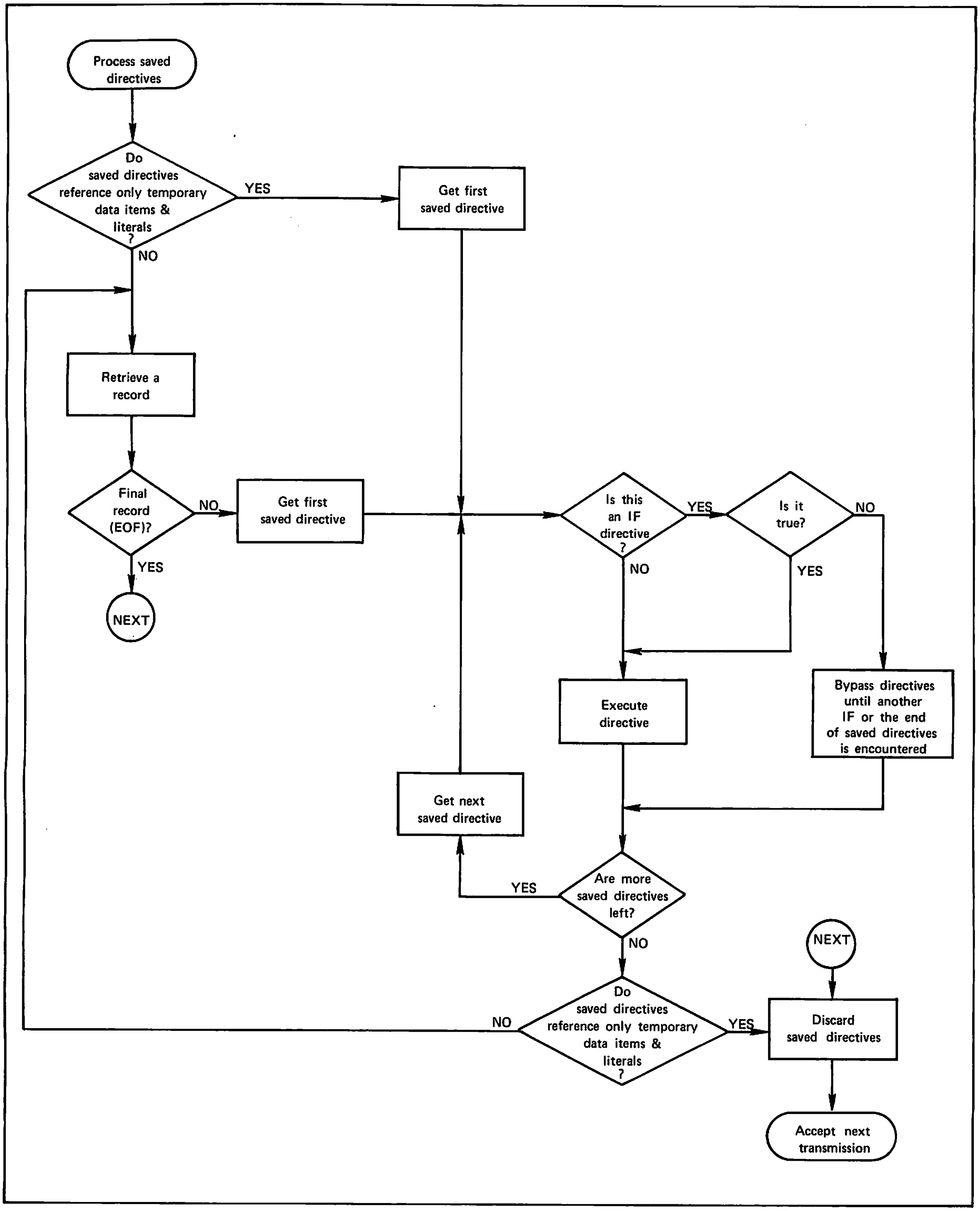

Process saved directives
Do saved directives reference only temporary data items & literals ?
YES
NO
Get first saved directive
Retrieve a record
Final record (EOF)?
NO
YES
NEXT
Get first saved directive
Is this an IF directive ?
YES
NO
Is it true?
NO
YES
Execute directive
Bypass directives until another IF or the end of saved directives is encountered
Get next saved directive
Are more saved directives left?
YES
NO
NEXT
Do saved directives reference only temporary data items & literals ?
NO
YES
Discard saved directives
Accept next transmission

Figure 4-34. General Flow of the Processing of Saved Directives

When an IF directive is used to query a data base relation, it states a condition that the records in the areas in a relation must satisfy to qualify for use in associated directives following the IF directive. The RESTRICT clause in the subschema states a condition to be satisfied at the record level, whereas the IF directive states a condition to apply at the projected record level. The user is generally not aware of the conditions specified in the RESTRICT clause. for the user to receive a record occurrence in a projected record occurrence it must have satisfied the requirements of the RESTRICT clause, if one exists. The projected record occurrence must then satisfy the IF directive entered by the user to contribute to the query.

Relationships between records are established statically by the JOIN clause in the subschema. They cannot be altered or supplemented dynamically by the IF directive. All processing to obtain the record occurrences for a projected record occurrence is performed first and then the IF directive is processed.

The data names specified in the IF directive can be in one or more areas in the relation or can be temporary data items. The data names in the IF directive need not be in the same areas as the data names specified in the DISPLAY or EXTRACT associated directive, but all data names must be contained in the same relation or must be defined as temporary data items.

Data items specified as floating point might be inaccurate by about one part in 10^{14} and, therefore, should not be used in comparisons for equality (equal, not equal).

The examples in figure 4-35 illustrate the IF directive in relational data base queries. The relation in the examples joins three areas: CONTRACTS, PRODUCTS, and EMPLOYEES. Example 1 shows two different queries using the same areas joined in the relation. Refer to section 1 for information about a relational data base.

The efficiency of a retrieval (how many records must be processed to complete the query) depends greatly on the way the query itself is formed. Generally, if the user phrases the IF directive so that the condition applies to the records in the first area of the relation, the retrieval is more efficient. Query Update processing of two additional queries is shown in examples 2 and 3.

INSERT

Although the INSERT directive is available to the Query Update user, the STORE directive performs the same function, and is preferred because it is more flexible. Refer to appendix H for a description of the INSERT directive.

NOTE

Refer to appendix J for recommendations on the use of this feature.

Example 1

```
-- IF DESCRIPTION EQ $PAYROLL$ +
   DISPLAY PROJECT-NO
```

This example tests a data item called DESCRIPTION in the CONTRACTS area and displays the data item PROJECT-NO in the PRODUCTS area.

```
-- IF PROJECT-NO GT $500$ +
   DISPLAY DESCRIPTION
```

The IF directive tests the data item called PROJECT-NO in the PRODUCTS area for a value greater than 500 and the data item DESCRIPTION in the CONTRACTS area is displayed.

Example 2

```
-- IF PRODUCT-NO EQ $A55$  +
   DISPLAY DESCRIPTION
```

PRODUCT-NO is a primary key and one access is required to read the unique record whose key is equal to A55 in the PRODUCTS area. Another access is made to read the unique record in the CONTRACTS area with the primary key CONTRACT-NO. The data item DESCRIPTION in the CONTRACTS area is displayed.

Example 3

```
-- IF CONTRACT-NO GT $400$ +
   AND PROJECT-NO LT $600$ +
   DISPLAY PRODUCT-NO
```

A record is obtained from the CONTRACTS area; then, the PRODUCTS area is read via the alternate key P-CONT-NO and the records retrieved are tested for PROJECT-NO less than 600. The .output is displayed to the user in increasing order of P-CONT-NO values and, within these, in increasing order of PRODUCT-NO values.

Figure 4-35. Relational Data Base Examples for IF Directive

INVOKE

The INVOKE directive initiates processing under CRM, CDCS, or IMF data base access mode. It can also establish areas, relations, and the subschema directory that is used for subsequent directives. When this directive is encountered, the internal data base tables are initialized. In CDCS data base access mode, the INVOKE directive can also identify the data base version. In IMF, the INVOKE directive can specify the name of a previously compiled external schema, identify the conceptual schema from which the external schema is derived, and provide a privacy key to match the privacy lock established in the external schema.

The formats of the INVOKE directive are shown in figure 4-36.

The INVOKE directive can initiate processing under either CRM or CDCS data base access mode. Data base access mode is established by the subschema specified by subschema-name. For example, if subschema-name refers to a CRM subschema, Query Update initiates processing in CRM data base access mode.

Only one INVOKE directive can be in effect at a time. When an INVOKE directive is entered, the effect of any prior CREATE or INVOKE directive is terminated: that includes terminating the effect of CDCS access keys previously entered and releasing the subschema file and tables established to use the subschema specified in the previous CREATE or INVOKE directive. The INVOKE directive can be entered any number of times within a single execution of the QU control statement.

The subschema-name identifies the subschema. IN CDCS data base access mode, Query Update ignores schema-name. Schema-name can be used for documentary purposes and for compatability with future versions of Query Update.

The FROM LIBRARY option identifies the subschema library that contains the subschema directory being used. If the FROM LIBRARY option is omitted, the permanent file name of the subschema library is assumed as follows: for NOS, the first seven characters of the subschema name are used; for NOS/BE, the entire subschema name is used.

When permanent file parameters are required to access the subschema, they must be specified in the INVOKE directive. The PW option can be specified to protect the security of passwords. When Query Update encounters the characters PW alone, terminal input is prompted with the symbols >>>. The user then enters the appropriate passwords. Query Update continues to prompt after each transmission until the user enters *END. The PW option protects the security of passwords since printing of passwords on the output listing or trace file is inhibited.

The FOR DATABASE option, available under CDCS data base access mode, identifies the data base version being used. Version MASTER is assumed when the option is omitted, with one exception: if a preceding VERSION directive that is in effect specified an alternate data base version, that alternate version is assumed. If a preceding VERSION directive is in effect and the FOR DATABASE directive is specified, the same data base must be specified in both the VERSION and INVOKE directives. The version-name must be a one through seven character name of a data base version included in the master directory as a version of the schema being used; MASTER is the name of the default data base version. The data administrator must provide the user with valid alternate version names when data base versions are being used.

```
CDCS Format

   INVOKE subschema-name [OF schema-name]
              [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]
              [FOR DATABASE version-name]†


CRM Format

   INVOKE subschema-name [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]


IMF Format

   INVOKE [FOR REPAIR]† external-schema-name OF schema-name [KEY literal]

              ┌ IN   ┐
              └ FROM ┘   METADB permanent-file-name [user-name] [PW password]


†Not available on NOS 1
```

Figure 4-36. INVOKE Directive Format

The following paragraphs refer to the IMF format in figure 4-36. An example of the INVOKE directive is shown in figure 4-37 The INVOKE directive makes a specific external schema available to a Query Update application. Subsequent query and update operations are made on the record types defined in that external schema. Rules restricting the optional clauses in the IMF INVOKE directive can be found in the IMF Applications Programming reference manual.

```
-- INVOKE QUEXMPL OF QUEXAMP (ID=ABC)
```

This example illustrates the use of a sub-schema named QUEXMPL. All relations and area described in QUEXMPL are made available. The schema name (QUEXAMP) is ignored.

Figure 4-37. INVOKE Directive Example

The FOR REPAIR option allows access of an information base with invalid constraints. While accessing this information base, the application program can delete, modify, and store records to correct the information base. Before the information base can be accessed without the FOR REPAIR option, it must be validated by the Validate utility. Refer to the IMF Schema Definitions reference manual for the description of the Validate utility.

LOOKUP

The LOOKUP directive modifies the order of search between the database and described files. The format of the LOOKUP directive is shown in figure 4-37.1.

```
                    { DATABASE }       { FIRST }
LOOKUP NAMES IN     { FILES    }   FOR { ONLY  }
```

Figure 4-37.1 LOOKUP Directive Format

By default, the database is searched for a dataname after any described files that may exist. The LOOKUP directive is used to reset this order. It is also used to indicate that only one source, either database or files, can be searched for datanames. A LOOKUP directive remains in effect until another LOOKUP directive is entered or the end of the Query Update session is encountered.

MODIFY

The MODIFY directive modifies the values of data items in existing records. Only one area can be modified at a time; each area joined in a relation must be modified separately. A record is selected for modification either by referencing the record key in the MODIFY directive or as a result of an IF directive with an associated MODIFY directive. The format of the MODIFY directive is shown in figure 4-38. An example of the MODIFY directive is shown in figure 4-39.

The MODIFY directive requires permission to modify or extend the area file. This permission is granted according to one of the following:

The passwords specified for the area file with the PW option of the INVOKE directive.

The passwords specified in the subschema in CRM data base access mode.

The keys specified in the ACCESS directive in CDCS data base mode.

Modifying by major or alternate key can result in several records being updated at one time. All records containing the specified major or alternate key value are updated in the same way. When an indexed sequential file contains records with duplicate primary keys, all records having duplicate keys are updated.

```
CDCS/CRM/IMF Format 1

   MODIFY SETTING {data-name} ...


CDCS/CRM/IMF Format 2

                             || SETTING {data-name-1} ... USING key-name ...             ||
                             ||        [SETTING {data-name-2} ...] [FROM file-name]      ||
   MODIFY [record-name]†     ||                                                          ||
                             || MOVE expression-1 TO {data-name-3} ...                   ||
                             ||        [AND expression-2 TO {data-name-4} ...] ...       ||

                 [ VETO ]
                 [ PASS ]

†Not applicable to IMF
```

Figure 4-38. MODIFY Directive Format

```
-- MODIFY EMP-RECORD USING EMP-ID SETTING PHONE

>> 00127    $6024673048$
>> 00291    $7268263409$
   *END
```

Two records named EMP-RECORD are updated in the PHONE field. One record has EMP-ID equal to 00127; the phone number is changed to 6024673048. The second record has EMP-ID equal to 00291; the phone number is changed to 7268263409.

Figure 4-39. MODIFY Directive Example

MODIFY SETTING, shown in format 1, can be used with temporary items. The Query Update user receives a prompt and can enter one line of data to be stored in the temporary item or items.

Record-name, shown in format 2, is the name of the record that is to be modified; if omitted, the record name is inferred from key-name, and the data-name-1 and data-name-2 lists in the MODIFY directive format. Data-name-1 and data-name-2 refer to either data base items or temporary items.

The SETTING/USING option allows the Query Update user to supply the names of the data items to be updated. This option causes Query Update to prompt for input by displaying the symbols >>. The user then enters appropriate data vaules for each record occurrence that is to be modified. Query Update continues to prompt until *END is entered to terminate input. Two or more record occurrences can be modified simultaneously, if the key has duplicate values.

Key-name identifies the primary, major, or alternate key within the record. Key-name is used to retrieve the record to be modified.

The primary key of a record can be changed with a single MODIFY directive, by using both the USING and the SETTING options.

The FROM option supplies the name of a local sequential file that contains a list of key-names and item values. Using the values in the independent file is an alternative to entering the values from the terminal. Information must be referenced in the MODIFY directive in the order in which it is arranged in the independent file. If delimiters are not present for non-numeric literals in the independent file, the SEP ITEM-SIZE directive must be specified before issuing the MODIFY directive. This tells Query Update to use the directory description to determine the size of each item in the independent file.

An IF directive preceding the MODIFY directive can be used to select records for modification; the USING option is not included. No data base-accessing directive, other than IF, can occur in the same transmission as a MODIFY directive containing a USING clause. If the USING clause is present, an IF directive on a temporary item can precede the clause.

The SETTING and MOVE clauses cannot reference area items unless a USING clause or a preceding IF on an area item is specified.

The VETO option causes the record to be displayed before it is modified. The user responds with YES or Y (modify the record), NO or N (do not modify the record), PROCEED or P (negate the current VETO option and modify the record), or EXIT or E (do not modify the record and terminate the transmission).

The PASS option disables the VETO for this transmission when a previous VETO directive specified VETO ON.

The user is responsible for maintaining the integrity of the data base. Updating operations must be consistent with the meaning of the relation defined for the areas (unless constraints between joined record types have been declared in the schema, in CDCS data base access mode). When modifying join terms, the user must consider whether the corresponding data items in other records must be modified as well, and issue any MODIFY directives needed to maintain the relation.

MOVE

The MOVE directive places a value in a data item. Two different formats are available for this directive. Format 1 is used for general data manipulative operations; format 2 is used for report-oriented operations. The formats of the MOVE directive are shown in figure 4-40.

Format 1

```
MOVE {literal-1   } TO data-name-1 . . .  [{literal-2       } TO data-name-2 . . . ] . . .
     {expression-1}                        [{AND expression-2}                     ]
```

Format 2

```
                    ( TITLE                  )
                    ( REPORT                 )
                    ( RECAP                  )
                    ( ANY SELECT             )
                    ( SELECT tag-number      )
                    ( ANY DETAIL             )
MOVE   {BEFORE}     ( DETAIL tag-number      )  expression-1  TO  data-name-1 ...
       {AFTER }     ( ANY BREAK              )    [AND expression-2  TO  data-name-2 ...]...
                    ( BREAK level-number     )
                    ( ANY HEADING            )
                    ( HEADING level-number   )
                    ( ANY FOOTING            )
                    ( FOOTING level-number   )
                    ( NO SELECT              )
```

Figure 4-40. MOVE Directive Formats

The literal-1 option in format 1 specifies a value to be moved to one or more data names. Additional literals can be moved to one or more data names by including the literal-2 option in the MOVE directive.

The expression-1 option moves an expression value to one or more data names. When a working storage DEFINE data name is in the expression, evaluation may be required before the MOVE directive is executed. The user is responsible for assuring EVALUATE performance. Query Update evaluates expressions and stores results after performing conversions required by data name descriptions. When floating-point-to-integer conversions are involved, numeric items are rounded up if the digit following the decimal point is greater than or equal to 5. Items are truncated when the digit is less than 5. Additional expressions can be moved to one or more data names by using the expression-2 option. Each expression-2 option included requires the word AND.

The data-name-1 option in format 1 refers to either a temporary data item (established by a DEFINE directive) or a permanent data item in the record of the area currently being updated. when the receiving field, data-name-1, refers to a permanent data item, the MOVE directive must be preceded by an INSERT or UPDATE directive if the data base is to be modified.

MOVE USER-ID can be used to update the 10-character user identification field. The USER-ID field is used by CDCS for all data base procedures and for log file entries; it can provide an additional privacy check. The field must be updated before the first INVOKE, CREATE, or VERSION directive is used.

The BEFORE and AFTER options in format 2 determine when the move operation is executed in respect to the specified report production step. The move can occur before or after the entire report; before or after a title, recap, heading, or footing line; or before or after a SELECT, DETAIL, or BREAK directive. If an EVALUATE directive is used in conjunction with a MOVE directive (both format 2), evaluation is performed before the move. Refer to the SELECT directive for additional information on the move operation in conjunction with the SELECT and DETAIL directives. The BEFORE and AFTER NO SELECT options represent two widely separated time situations:

BEFORE NO SELECT

> When no select criteria testing is performed to generate a detail line, the move operation occurs before each line is generated. This option is used only when one unnumbered DETAIL directive is specified.

AFTER NO SELECT

> When all criteria testing has been performed and no select criteria has been satisfied, the move operation is performed. This option is used only when numbered DETAIL directives are specified.

A preceding FORMAT or ALTER directive must be in effect to use format 2 of the MOVE directive.

NOTE

The NOTE directive allows user comments to be included in Query Update sessions. The keyword and comments can be a complete transmission, or they can be appended as the last portion of a transmission containing other directives. The format of the NOTE directive is shwon in figure 4-41.

```
NOTE user-comment
```

Figure 4-41. NOTE Directive Format

All characters between NOTE and the end of transmission are accepted as comments. A NOTE directive can be recorded as part of a session and is listed by an EXHIBIT directive for the session; it does not appear in output or as part of a report.

OS

The OS directive allows the user, in interactive mode, to enter operating system control statements during a Query Update session. The format of the OS directive is shown in figure 4-42.

```
OS[,] control-statement-string
```

Figure 4-42. OS Directive Format

The control statement begins at the first character that is not a blank or a comma and ends with the last nonblank character. Query Update appends a period to the control statement and passes it to the operating system for processing. Control is returned to Query Update when the control statement has been processed. The control statement must have fewer than 80 characters.

An OS directive can be recorded as part of a session in interactive mode. A diagnostic is issued if the OS directive is used in batch mode and the transmission is ignored.

PAGE-NUMBER

The PAGE-NUMBER directive is a report specification that indicates the vertical and horizontal positioning of one or more page numbers. The five-digit page number is preceded by the word PAGE and one space. The format of the PAGE-NUMBER directive is shown in figure 4-43.

```
               ┌    ┌ TITLE-LINE     ┐ ┌ TAB integer-2    ┐ ┐
PAGE-NUMBER    │ AT │ LINE integer-1 │ │                  │ │ ...
               │    └ RECAP-LINE     ┘ └ COLUMN integer-3 ┘ │
               └                                            ┘
```

Figure 4-43. PAGE-NUMBER Directive Format

The PAGE-NUMBER directive with no options specified causes the page number to be placed at line 1 in column 100. Vertical positioning of the page number can be specified by one of three options:

TITLE-LINE

Positions the page number in the first line of the title. If there is no title directive, the date is positioned in the first line of each page.

LINE integer-1

Places the page number in the line that is integer-1 lines from the top of the page.

RECAP-LINE

Positions the page number in the first line of the recap. If no RECAP directive was specified, or the BEYOND option was used in the RECAP directive, the date is positioned in the last line of the page.

Horizontal positioning of the page number is specified by either the TAB or COLUMN option:

TAB integer-2

Begins the page number in the column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

Begins the page number in the print position specified by integer-3.

The page number can appear more than one time through repetition of the AT vertical and horizontal positioning options.

PAGE-SIZE

The PAGE-SIZE directive is a report specification that establishes the vertical and horizontal layout of a report page. The maximum number of vertical lines, horizontal columns, sectional page divisions, and multiple copy images can be specified. The format of the PAGE-SIZE directive is shown in figure 4-44.

If the PAGE-SIZE directive is omitted from the report specifications, the defaults provided are page-length of 60 lines, page-width of 136 columns, 1 section, and 1 image.

The LINES option establishes the maximum number of lines for a report page. The maximum page width in columns is determined by the COLUMNS option. The leftmost column (print position) is numbered as 1. Unless modified at installation time, the maximum page width is 136 columns with installation options up to 952 print positions; the maximum page length is 60 lines. When page-size is greater than 136 columns, printed pages are interleaved (one after another) thereby producing the left portion of a report page followed by successive right portions.

The SECTIONS option determines the number of sectional divisions across a page. Only detail lines are affected by the SECTIONS option. A maximum of 10 sections can be specified. The width of a section is the number of columns divided by the number of sections, rounded down. The direction of successive detail line entries is determined by the HORIZONTAL or VERTICAL option: HORIZONTAL places the successive entries across the total number of sections; VERTICAL places the successive entries in one section and then continues to the next section. Figure 4-45 illustrates horizontal and vertical placement of detail line entries. A sample report in section 5 illustrates the VERTICAL option. A DETAIL directive with more than one AT LINE clause excludes the use of the VERTICAL SECTIONS option.

The IMAGES option indicates that the report is to be duplicated the specified number of times across the page. This option can be used for multiple copies when the number of columns (integer-2) multiplied by the number of images (integer-4) is less than or equal to 136. The maximum number of images is 4.

The PARALLEL option is used in printing extra wide reports where a report page is larger than one printer page. If this option is not specified, the system alternates wide report pages as if the first page is separated and placed side-by-side with the next page, effectively becoming the left and right side of a composite page. With the PARALLEL option, all left sides are printed, then all the next sides, and so forth. This option can be used when the page width is greater than 136 print positions.

```
PAGE-SIZE IS [integer-1 LINES] [integer-2 COLUMNS]

[integer-3 { HORIZONTAL } SECTIONS] [integer-4 IMAGES] [PARALLEL]
           { VERTICAL   }
```

Figure 4-44. PAGE-SIZE Directive Format

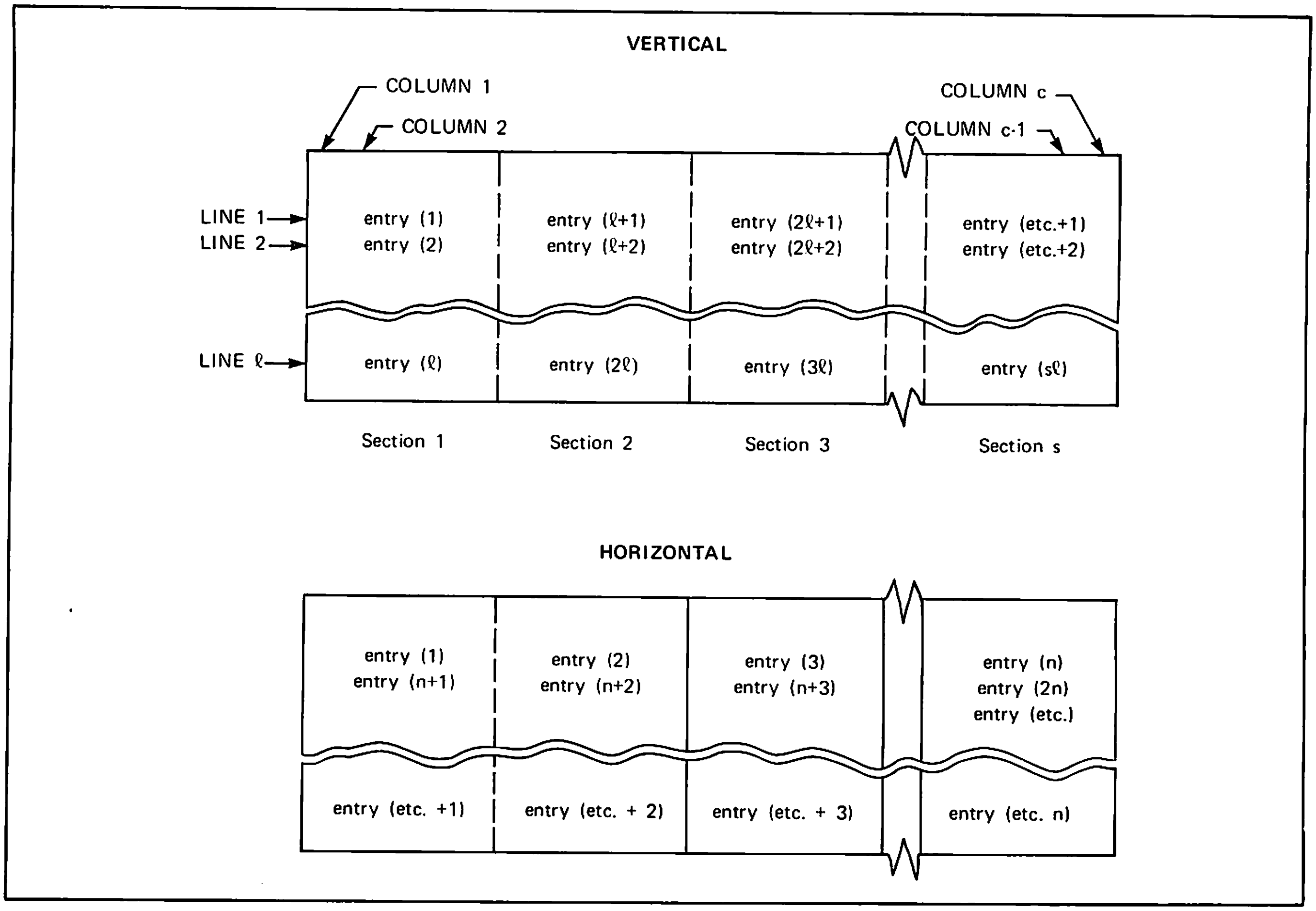


Figure 4-45. Horizontal and Vertical Placement of Detail Lines in Sections

PERFORM

The PERFORM directive retrieves and executes transmissions recorded in the current catalog. The current catalog is either a catalog attached by a VERSION directive or the default catalog (ZZZZZQ2). Transmissions are identified as a group by the session id assigned when the transmissions are originally recorded. If recording, the session id cannot be the same as the session id specified in the preceding RECORDING directive. Specific transmissions are selected by transmission id. Each transmission id within a session is unique. The format of the PERFORM directive is shown in figure 4-46.

Each PERFORM directive causes execution to begin with the first identified transmission. After execution of all the transmissions indicated in the PERFORM directive, control returns to the next transmission following the PERFORM directive. (PERFORM must be the last directive in the transmission.) In interactive mode, control returns to the user for the next transmission.

When no options are included with the PERFORM directive, the entire cataloged session is executed. More than one session can be specified by including the AND option in the directive.

Any portion of a cataloged session can be performed by using an appropriate combination of transmission ids and ranges of transmission ids. The PERFORM directive cannot specify a range of directives where transmission-id-1 is greater than transmission-id-2.

PERFORM directives can be nested; that is, a session being executed because of the specification of a PERFORM directive can, itself, contain another PERFORM directive. The maximum levels of nesting allowed is 15. If the maximum nesting is exceeded, Query Update issues a diagnostic and returns control to the transmission that follows the original PERFORM directive. In interactive mode, control returns to the user for the next transmission.

The REPEAT option causes the PERFORM directive to be executed a number of times. The expression specified in this option is evaluated at the beginning of the perform operation to determine the exact number of times to execute the PERFORM directive. The expression data name or integer number following REPEAT must yield a valid integer value.

The UNTIL option allows the PERFORM directive to be executed until the specified condition is satisfied. The condition is tested at the completion of all transmissions indicated by the PERFORM directive. If the condition has not been satisfied, the directive is performed again.

```
PERFORM  session-id-1

    [transmission-id-1 [TO  transmission-id-2]] . . .

        [AND session-id-2 [transmission-id-3 [TO  transmission-id-4]] . . .] . . .

            [REPEAT  expression]     [VETO]
            [UNTIL  condition  ]     [PASS]
```

Figure 4-46. PERFORM Directive Format

Each cataloged transmission that is retrieved can be displayed before execution by including the VETO option. When the transmission is displayed, the user indicates whether or not it is to be executed.

YES (or Y)

Executes the transmission and retrieves the next transmission.

NO (or N)

Does not execute the transmission; the next transmission is retrieved.

PROCEED (or P)

Disables the VETO mode and executes the remainder of the transmissions automatically.

EXIT (or E)

Terminates the PERFORM directive immediately.

The PASS option disables the VETO mode for the current PERFORM directive. This option is applicable only when the VETO mode has been requested with a previous VETO directive.

PREFACE

The PREFACE directive is a report specification that causes lines of text or another report to precede the current report. The preface is produced immediately preceding the report named in the current FORMAT directive. The PREFACE directive is convenient for identification of banner pages, distribution lists, and similar information covers. The format of the PREFACE directive is shown in figure 4-47.

```
PREFACE IS { TEXT        }  FROM  file-name
           { report-name }
```

Figure 4-47. PREFACE Directive Format

The file name indicated in the FROM specification supplies the source data for the preface. For more information see the External File Organization subsection.

The TEXT option treats each record from the specified file as a single line of the preface. Lines are placed on the page from the top to the assembled maximum line depth and truncated as needed to fit the report page width. Each record is shifted right one character to provide for single line spacing.

The report name option uses records from the specified file to produce a preface in the format established for that report. The preface contains more than one report when the specified report name also contains a PREFACE (or SUMMARY) report name directive. The report name specified in this directive must be in the current catalog.

PREPARE

The PREPARE directive initiates production of a report according to the specifications associated with the report name. The report specification directives are executed to produce the report from a source data file. The format of the PREPARE directive is shown in figure 4-48.

```
PREPARE report-name FROM file-name
```

Figure 4-48. PREPARE Directive Format

The report name specified must exist in the current catalog, which can be either the default catalog or a catalog attached by a VERSION directive. The finished report is placed in a local file named report name. Query Update does not rewind the file before or after preparing the report. Disposition (printing, permanent file retention, dropping, and so forth) of the local file (report name) is by user direction.

The file name specified supplies the source data for the report. An EXTRACT or DISPLAY UPON directive can be used to establish the content of the file. Source data from another source is allowed only if Query Update has been given a DESCRIBE directive in order to locate values for data names. The source data file is automatically rewound by Query Update. The PREPARE directive must be the last directive in a transmission.

General operations involved in report preparation are illustrated by flowcharts in appendix G.

PREVIEW

The PREVIEW directive causes the preparation of a two-page sample report. The report is produced according to the specifications associated with the report name. A directory created through the DESCRIBE, DISPLAY UPON, or EXTRACT directive must exist for the file used in the sample report, whether or not the sample report is created from the actual data in the file. The format of the PREVIEW directive is shown in figure 4-49.

```
PREVIEW report-name [ | FROM file-name          | ]
                    [ | DESCRIBED IN file-name  | ]
```

Figure 4-49. PREVIEW Directive Format

The report name specified must exist in the current catalog, which can be either the default catalog or a catalog attached with a VERSION directive. The finished sample report is placed in a local file named report name. Query Update does not rewind the file before or after preparing the report. Disposition (printing, permanent file retention, dropping, and so forth) is by user direction.

The FROM option is used to supply actual data for the sample report. (See the External File Organization sub-section for more information.) If this option is not included in the PREVIEW directive, dummy data values (alternate fields of X's or Y's for alphabetic information and 8's or 9's for numeric information) are placed in the report.

The DESCRIBED IN option names the directory (named the same as the file it describes) that is to be used to produce the sample report when actual data is not used in the sample report. If only one directory exists for the current session, this option can be omitted. If more than one directory has been created, the DESCRIBED IN option must be specified.

The PREVIEW directive must be the last directive in a transmission. When the PREVIEW directive is followed by the PREPARE directive, in a subsequent transmission, the sample report and the actual report are placed in the local file. The sample report can then be used for forms alignment when the file is output to the printer. If more than two pages are needed for forms alignment, several PREVIEW directives can be specified.

RECAP

The RECAP directive is a report specification that establishes the content and positioning of one or more lines to be generated as each page is completed. When a PAGE-NUMBER, TIME, or DATE directive is specified for the recap line, the information is added to the recap line after centering or positioning. The format of the RECAP directive is shown in figure 4-50.

The AT LINE option specifies the vertical position of the recap line. An absolute line number (AT LINE integer-1) places the recap line on the specified line on each report page. A relative line number (AT LINE integer-1 BEYOND) positions the recap line the specified number of lines after the largest line number previously written on the report page. (The meaning of relative line number differs for recap and detail lines. For more information see the description of the DETAIL directive.) If the AT LINE option is not included in the directive, the recap line is positioned one line beyond the largest line number previously written on the report page. Multiple recap lines are specified by repeating the AT LINE option followed by other elements of the directive.

The content of a recap line is specified by one or more IS clauses. Multiple IS clauses are used to indicate the horizontal positioning of specific data; the IN option is then included with each IS clause. The expression in an IS clause specifies data that is to appear in the recap line; it identifies one or more data fields, literals, or defined items. With multiple IS clauses, the optional elements are specified for each expression to which they apply. A data name in an expression can contain only that information accumulated up to the point where the recap line is generated; if evaluation is involved, the user is responsible for assuring that the EVALUATE directive has been executed.

The CENTERED option causes the expression to be automatically positioned midway between two horizontal points. If no points are specified by an IN clause, default horizontal points of column 1 and the page width are assumed. A horizontal point specified by the IN clause represents the left point; the right point is the next tab or the page width. Centering is applied only to the expression associated with the CENTERED option.

```
RECAP { [AT LINE integer-1 [BEYOND]]

        { IS expression [CENTERED] [IN { NEXT TAB          } ] } ... } ...
                                       { TAB integer-2     }
                                       { COLUMN integer-3  }
```

Figure 4-50. RECAP Directive Format

The IN clause specifies the horizontal positioning of the recap line. The option selected specifies either the starting point of the recap line or the left point to be used for centering.

NEXT TAB

The next tabular position after the current position; TAB 1 is assumed if a previous tab has not been specified in the directive.

TAB integer-2

The column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

The print position specified by integer-3.

The recap line is placed on the page as specified in the RECAP directive; additional information (page number, data, or time) is then added. If the recap line is overwritten by this information, an informative diagnostic is issued and processing continues.

RECORDING

The RECORDING directive initiates or terminates the recording of session transmissions in the current catalog. Transmissions are recorded either as a new session or as a continuation of a previously recorded session. As many different sessions as desired can be recorded on one catalog. The format of the RECORDING directive is shown in figure 4-51.

```
RECORDING { session-id [transmission-id] [BY increment-size] }
          { OFF                                              }
```

Figure 4-51. RECORDING Directive Format

Recording is off when Query Update is entered. If recording is being initiated, a session id of one through six characters is supplied in the directive. Subsequent transmissions are recorded under the specified session id as a new session or as a continuation of a session.

The transmission id option can be used to indicate the transmission id at which the recording of subsequent directives is to begin. If this option is included in the RECORDING directive, transmissions begin with the specified transmission id; if it is not included, transmissions begin with the default value:

For a new session, the increment size is the default value.

For the continuation of a session, the increment size added to the highest previously recorded transmission id is the default value.

When a transmission id is specified, Query Update responds with a message. If the transmission id already exists for the session id, a diagnostic message is issued and recording begins at the default transmission id. If the transmission id does not exist, Query Update responds with a message showing the number of transmissions that can be recorded in the gap using the current increment size.

The increment size option specifies the value to be added to the current transmission id to determine each successive transmission id. The increment-size must be an integer greater than zero and can be up to three digits in length. The default increment size is one. A diagnostic message is issued if the increment size is not in the range 1 through 999.

The recording of a session is terminated by specifying the OFF option. Once a session has been initiated, recording continues until RECORDING OFF is specified.

Query Update responds to a RECORDING directive by supplying a three-digit identifier (transmission id) that is used to identify the next recorded transmission. A transmission has been recorded if the three-digit identifier is incremented. The transmissions are recorded in the current catalog, which is either the default catalog (ZZZZZQ2) or a catalog attached by the VERSION directive. Recording cannot be specified on a catalog file that was created with a transmission length less than the transmission length currently set. Changing catalogs during the recording of a session is not allowed.

Report specification directives for a report format are not recorded under a session id. These directives are retained in the current catalog under a report name established by the FORMAT directive.

During a recording session, some directives are not recorded in the catalog. Table 4-3 shows the action taken for directives specified when a RECORDING directive is in effect. No diagnostics, except for syntax errors, are issued during a recording.

The SEPARATOR directive is executed at the time it is recorded. Consequently, the literal delimiter is available for immediate use.

A PERFORM directive in one cataloged session can be used to execute transmissions in another session. Performs can be nested in this fashion up to 15 levels. Transmissions from the same session cannot be referenced by a PERFORM directive; a diagnostic is issued and the PERFORM is ignored.

TABLE 4-3. ACTION ON DIRECTIVES DURING RECORDING

Action	Recorded	Not Recorded	Retained Under Report Name
Directive not executed	ACCESS COMPILE CREATE DEFINE DELETE DESCRIBE DIAGNOSTIC DISPLAY END ERASE data-name EVALUATE EXECUTE EXTRACT IF INSERT INVOKE MODIFY MOVE NOTE OS PERFORM PREPARE PREVIEW RECOVERY REMOVE RETURN REWIND SORT SPECIFY STOP STORE UNIVERSAL UPDATE USE VERIFY VETO VIA	VERSION (causes a diagnostic)	BREAK(n) DATE DETAIL(n) EVALUATE BEFORE (or AFTER) FOOTING(n) HEADING(n) MOVE BEFORE (or AFTER) PAGE-NUMBER PAGE-SIZE PREFACE RECAP SELECT(n) SUMMARY TABS TIME TITLE
Directive executed	SEPARATOR	ALTER DUPLICATE ERASE REPORT (or SESSION) EXHIBIT FORMAT HELP RECORDING	

RECOVERY

The RECOVERY directive applies only to CDCS data base access mode. The directive establishes a recovery point on the journal log file. CDCS responds to a RECOVERY directive by halting all CDCS activity and suspending all Query Update user activity until the following events occur:

All I/O buffers for data base areas are cleared.

A recovery point log entry is written to the log file for the data base.

The quick recovery file for the data base is emptied.

The RECOVERY directive should be used judiciously, because considerable overhead is involved. The format of the RECOVERY directive is shown in figure 4-52.

```
                                        { literal     }
RECOVERY POINT USING data-name-1       {             }
                                        { data-name-2 }
```

Figure 4-52. RECOVERY Directive Format

Upon execution of the RECOVERY directive, data-name-1 contains a unique recovery point number assigned by CDCS. Data-name-1 must be a defined integer item. Literal or data-name-2 must not exceed 30 characters and must contain a user-supplied explanatory message for the recovery point. The message is written on the journal log file at the same time the recovery point number is established.

Refer to the CDCS 2 reference manual for a complete discussion of logging and recovery procedures.

REMOVE

The REMOVE directive removes specific records from a data base area. Only one area can be modified at a time; each area joined in a relation must be modified separately. The complete record is removed from the data base; the REMOVE directive does not remove part of a record. The format of the REMOVE directive is shown in figure 4-53. An example of the REMOVE directive is shown in figure 4-54.

```
REMOVE [record-name]† [SETTING {data-name-1} ...] USING key-name
                  [SETTING {data-name-2} ...] [FROM file-name]

                  [VETO]
                  [PASS]

†Not applicable to IMF
```

Figure 4-53. REMOVE Directive Format

```
-- REMOVE EMP-RECORD USING EMP-ID

>> 00111
>> 62417
   *END
```

Two records are removed from the EMPLOYE area: one with EMP-ID equal to 00111, and one with EMP-ID equal to 62417. EMP-RECORD can be omitted; if omitted, the record name is inferred from the record key EMP-ID.

Figure 4-54. REMOVE Directive Example

The REMOVE directive requires permission to modify the area file. This permission is granted according to one of the following:

The passwords specified for the area file with the PW option of the INVOKE directive

The passwords specified in the subschema in CRM data base access mode

The keys specified in the ACCESS directive in CDCS data base access mode

Record-name is the name of the record that is to be removed; if omitted, the record name is inferred from key-name in the REMOVE directive format. Data-name-1 and data-name-2 refer to either data base items or temporary items.

The USING option causes Query Update to either prompt the user or read the file file-name to obtain data values. Key-name identifies the primary, major, or alternate key within the record. Each key-name value is used, after appropriate conversion, to retrieve a record occurrence that is to be removed from the data base. More than one record occurrence can be removed if the key has duplicate values.

The SETTING/FROM option can be used if a local sequential file contains items that the user wants ignored. A sequence of data item values is provided from the local file for each record to be deleted. Query Update ignores all items except the items it needs (the key-name fields). The FROM option supplies the name of the local sequential file that contains the item values.

Using the values in the non-data-base file is an alternative to entering the values from the terminal. If delimiters are not present for nonnumeric literals in the non-data-base file, the SEP ITEM-SIZE directive must be specified before issuing the REMOVE directive. This tells Query Update to use the directory description to determine the size of each item in the non-data-base file. A non-data-base file cannot be used if it was created by the EXTRACT directive and if it contains numeric items. The EXTRACT directive does not convert data items to display code, as the DISPLAY directive does.

An IF directive preceding the REMOVE directive can be used to select records for removal; the USING option is not included. No data base-accessing directive, other than IF, can occur in the same transmission as a REMOVE directive containing a USING clause. If the USING clause is present, an IF directive on a temporary item can precede the clause.

The VETO option causes the record to be displayed before it is removed. The user responds with YES or Y (remove the record), NO or N (do not remove the record), PROCEED or P (negate the current VETO option and remove the record), or EXIT or E (do not remove the record and terminate the transmission).

The PASS option disables the VETO for this transmission when a previous VETO directive specified VETO ON.

RETURN

The RETURN directive is used to release files, relations, subschemas, and local files when they are no longer needed during a Query Update session. The files are returned to the operating system; refer to the appropriate system reference manual for information related to a returned file. The format of the RETURN directive is shown in figure 4-55.

```
          ( file-name       )
RETURN    ) area-name       (  ...
          ) relation-name   (
          ( subschema-name  )
```

Figure 4-55. RETURN Directive Format

This directive permits the user to release areas no longer needed while other areas are retained. When the area is released, central memory associated with the area can be freed.

In CRM data base access mode, the RETURN directive accepts area names, relation names, the subschema name, and logical file names. In CDCS data base access mode, the RETURN directive accepts logical file names and subschema names only. An attempt to return individual areas or relations results in a diagnostic. One or more names can appear in one RETURN directive.

In CRM data base access mode, when Query Update returns an area, all internal tables for the area are released; no attempt is made to return the area file. A subsequent RETURN directive issued by the user actually returns the area. By issuing one RETURN directive, the user can release central memory used in the tables without losing the created or temporary file. If a created file has an index file associated with it, the user can return it by issuing a separate RETURN directive.

In CRM data base access mode, an area cannot be returned if it is part of a relation that is still active. Query Update issues a diagnostic to inform the user that the relation is still active. When a relation is returned, only the areas that are not associated with another active relation are returned.

If a user is in CDCS catalog mode, RETURN subschema-name releases all internal descriptions of the areas and relations. The subschema file is not released because it is being used for the version catalog.

If a user is not in CDCS catalog mode, a RETURN subschema-name causes:

> The subschema file to be released.
>
> All internal descriptions of the subschema, areas, and relations to be released.
>
> CDCS to be informed that the current control point is no longer accessing the subschema and areas.
>
> Data base areas to be released by CDCS if no other CDCS user is using the areas.

When a local file is returned, any DESCRIBE directory associated with it is eliminated. When a subschema is returned, all active areas and all active relations are returned. The RETURN directive cannot be used to release an internal Query Update work file with a name beginning with five Z's.

REWIND

The REWIND directive positions a file at the beginning-of-information. This directive is specified whenever it is necessary to ensure that a file is positioned at the beginning-of-information. The format of the REWIND directive is shown in figure 4-56.

```
REWIND file-name . . .
```

Figure 4-56. REWIND Directive Format

One or more file names can appear with one REWIND directive. All specified files are rewound to the beginning-of-information. The REWIND directive should be issued before and after a DISPLAY directive that includes the UPON file name option.

SELECT

The SELECT directive is a report specification that establishes the criteria for selecting a detail line specification for a source record. A condition, or a condition name that has been defined by a SPECIFY directive, determines the select criteria. The format of the SELECT directive is shown in figure 4-57.

```
                        ( condition      )
SELECT tag-number ON  {                  }
                        ( condition-name )
```

Figure 4-57. SELECT Directive Format

The tag number (1 through 63) references the detail specification to be selected when the specified situation occurs. The tag number in this directive must correspond to a DETAIL directive tag number to obtain the desired detail lines. If the tag number does not have a corresponding DETAIL directive tag number, no detail line is generated for that record.

SELECT directive situations are tested in ascending order by tag number during source data processing. When DETAIL directives include tag numbers, SELECT directives are required. An unnumbered DETAIL directive prohibits the use of the SELECT directive. Unless modified when Query Update is installed, a maximum of five tag numbers can be specified.

The specified condition or the condition referenced by the condition name is tested; if the condition is true, the detail line specification that is identified by the same tag number is selected for the record. Each record is read and tested against all the select criteria (in ascending order by tag number) until a condition is satisfied. Once a condition is satisfied, no further testing of select conditions is performed. If no condition is satisfied, the record is ignored.

A SELECT directive determines which DETAIL directive establishes the report line content and positioning for a source record. Consequently, the user can state that an EVALUATE or MOVE directive is to occur either before or after processing of the DETAIL or SELECT directive. The full range of user options concerning an EVALUATE or MOVE directive in relation to a SELECT or DETAIL directive is as follows:

```
                                   ┌ ANY SELECT  ┐
┌ EVALUATE ┐     ┌ BEFORE ┐        │ NO SELECT   │
└ MOVE     ┘     └ AFTER  ┘        ┤ SELECT n    ├
                                   │ ANY DETAIL  │
                                   └ DETAIL n    ┘
```

Figure 4-58 illustrates the flow of processing to select the detail specification for a record; it is assumed that a record is available at the start. In relation to SELECT directive testing, the AFTER option refers to the record having satisfied criteria (the YES paths). One exception occurs when the AFTER NO SELECT option is at point A where the record has failed all SELECT directive criteria and no selection of a detail specification occurs; the record is ignored. The following directive could be used to maintain a count of rejected records:

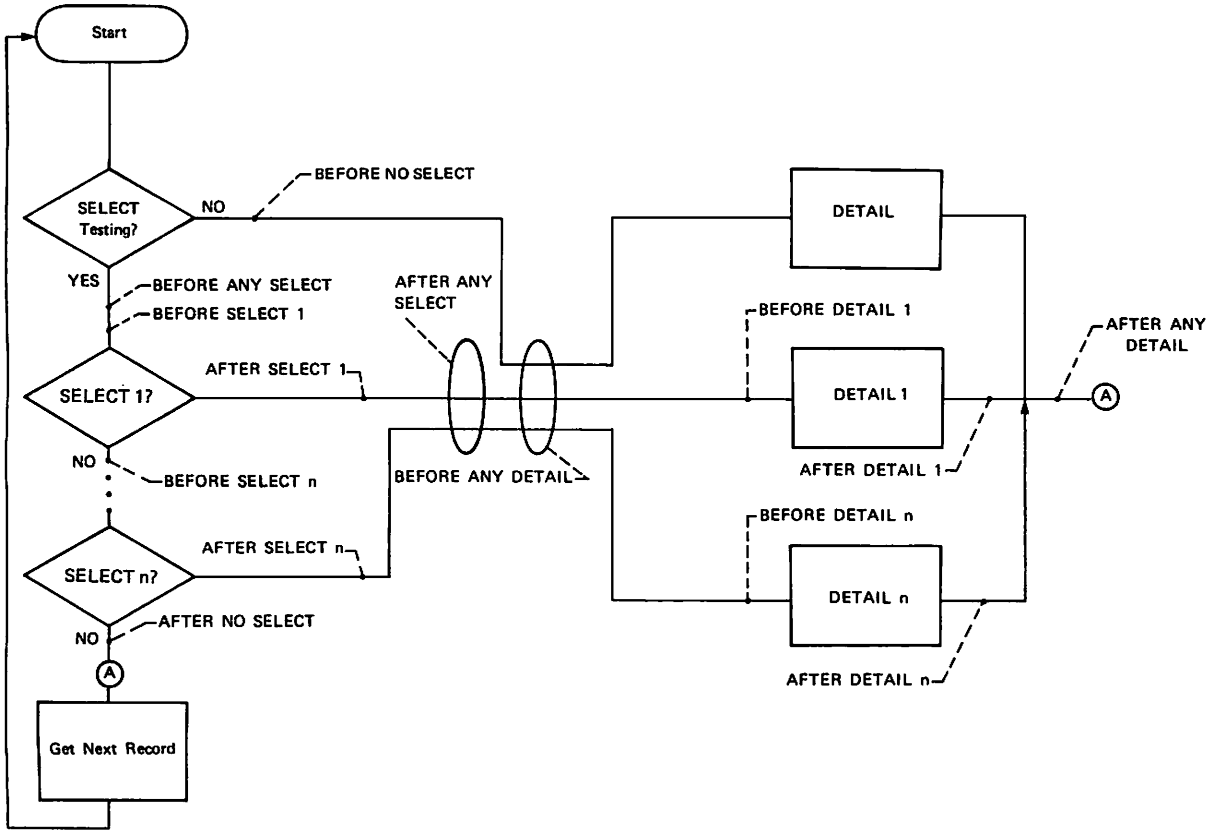
```
MOVE AFTER NO SELECT +
data-name + 1 TO data-name
```

The same processing point can be stated in several ways. However, if a MOVE or EVALUATE directive must be executed before or after a specific SELECT or DETAIL directive, the user should not depend on the ANY option because it could occur at many points.

Figure 4-59 illustrates the flow of look-ahead processing to select the detail specification for a record when the BREAK directive is used. It is assumed that the current record has been read and selected and that the next record has been read at the start. When the BREAK directive is used, the selection of the next record and the determination of the next break takes place before the detail content for the current record is printed. Any MOVE or EVALUATE directive should be related to the SELECT directive, but not to the DETAIL directive, when the MOVE or EVALUATE directive initializes from the current record items that are to be included in the selection of the next record or in the determination of the next break.

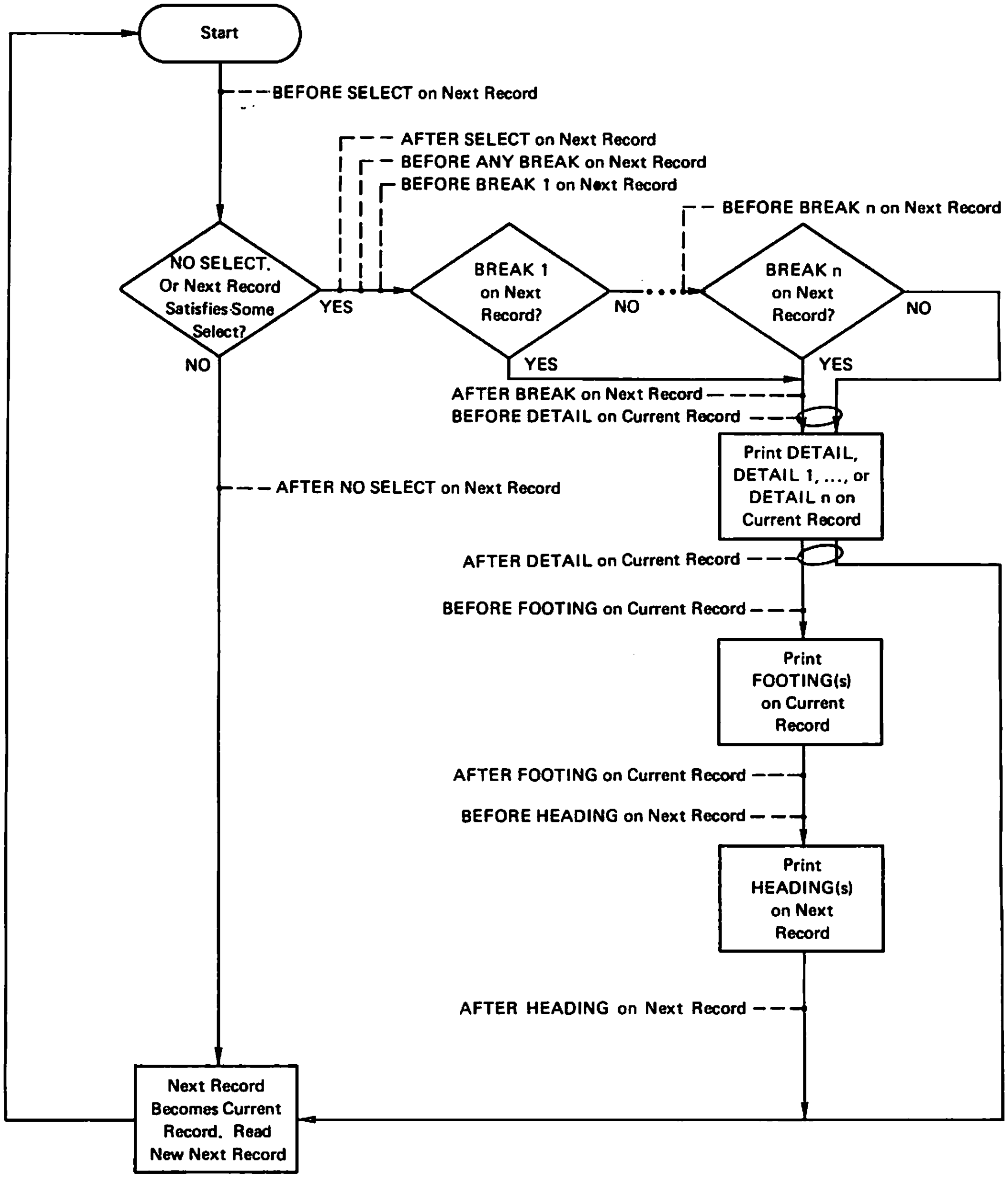
SELECT directive testing occurs in a look-ahead process similar to that of the BREAK directive. Only selected records are used in BREAK directive testing. If a MOVE or EVALUATE directive is used for initialization of items to be included as part of a detail line, the directive should be related to the DETAIL directive and not the SELECT directive. Whether the BREAK directive is or is not used, any combination of the EVALUATE or MOVE BEFORE or AFTER DETAIL directives always uses the same current record that the detail line is prepared from.

For additional information about the flow of processing, see appendix G.



NOTE: A ring around parallel flow lines indicates that the specified processing can occur on each of the encompassed paths.

Figure 4-58. Detail Selection


Start
BEFORE SELECT on Next Record
AFTER SELECT on Next Record
BEFORE ANY BREAK on Next Record
BEFORE BREAK 1 on Next Record
BEFORE BREAK n on Next Record
NO SELECT. Or Next Record Satisfies Some Select?
YES
NO
BREAK 1 on Next Record?
NO
YES
BREAK n on Next Record?
NO
YES
AFTER BREAK on Next Record
BEFORE DETAIL on Current Record
Print DETAIL, DETAIL 1, ..., or DETAIL n on Current Record
AFTER NO SELECT on Next Record
AFTER DETAIL on Current Record
BEFORE FOOTING on Current Record
Print FOOTING(s) on Current Record
AFTER FOOTING on Current Record
BEFORE HEADING on Next Record
Print HEADING(s) on Next Record
AFTER HEADING on Next Record
Next Record Becomes Current Record. Read New Next Record

NOTE: A ring around parallel flow lines indicates that the specified processing can occur on each of the encompassed paths.

Figure 4-59. Detail Selection With Breaks

SEPARATOR

The SEPARATOR directive defines a character for delimiting nonnumeric literals. This directive can also be used to specify that the size of a nonnumeric literal, in certain instances, is determined by the directory description of the item. The format of the SEPARATOR directive is shown in figure 4-60.

```
              ┌ = delimiter ┐
  SEPARATOR   │             │
              └ ITEM-SIZE   ┘
```

Figure 4-60. SEPARATOR Directive Format

The dollar sign ($) is the default delimiter and is in effect until a SEPARATOR directive is used to change the delimiter. A delimiter specified by a SEPARATOR directive remains in effect until it is explicity changed to another delimiter; note that the directive SEPARATOR without any options does not put the default delimiter ($) into effect; it only disables the ITEM-SIZE option. Any of the following characters can be specified as the delimiter:

- * / $ [ ] ≠ → ∨ ∧ ↑ ↓ > < ≥ ≤

The delimiter can be included within a literal by entering the character twice for each desired appearance. For example, $A$$B$ would yield the literal A$B. Beginning and ending delimiters must be separated from other elements or literals in a directive by a space or a comma. An arithmetic or logical operator specified as the delimiter cannot be used in an expression; the equivalent word for the operator must be used in expressions. If a universal character has been established with the UNIVERSAL directive, the same character cannot be specified as the delimiter.

The ITEM-SIZE option specifies that the directory description is used to determine the size of the item. The ITEM-SIZE option does not affect the delimiter. The only effects of the ITEM-SIZE option are:

When data is entered in response to a STORE, MODIFY, or REMOVE directive with the USING/SETTING option specified, delimiters are not required; the size of each input field is taken from the directory description of the associated item.

When data is written to a file in response to a DISPLAY UPON directive, the usual blank preceding each component of the line is eliminated; thus, the displayed fields are concatenated.

The ITEM-SIZE option is not used for report production. The SEPARATOR directive with neither option specified disables the ITEM-SIZE option.

SORT

The SORT directive specifies and initiates the resequencing of records in a file. Before a file can be sorted, it must have a directory produced by an EXTRACT, DESCRIBE, or DISPLAY UPON directive. The format of the SORT directive is shown in figure 4-61.

```
                    ┌─ ┌ UPON file-name-2                                 ┐ ─┐
                    │  │                                                  │  │
                    │  │        ┌ COBOL          ┐                        │  │
                    │  │ USING  │ FORTRAN        │                        │  │
SORT file-name-1    │  │        │ literal-string │                        │  │
                    │  │        └ optional-table ┘                        │  │
                    │  │                                                  │  │
                    │  │ ON {data-name [DESCENDING]}...                   │  │
                    │  │                                                  │  │
                    └─ └ UNIQUE                                           ┘ ─┘
```

Figure 4-61. SORT Directive Format

The records of file-name-1 are resequenced by data name fields using the specified collating sequence. The UPON option specifies that the resequenced records be placed in file-name-2. File-name-1 and file-name-2 can designate the same file. If file-name-1 is different from file-name-2, file-name-2 is created identifical in format to file-name-1 and has the same directory. If a directory already exists for file-name-2, a new one is created. File-name-2 is rewound before the sorted records are written.

The USING option specifies the collating sequence for ordering the file. If this option is omitted from the SORT directive, the COBOL collating sequence is used.

Unique records with duplicate keys retain their original ordering. Collating sequence can be specified by one of the following options:

COBOL

Indicates the COBOL collating sequence.

FORTRAN

Indicates the FORTRAN (display code) collating sequence.

literal-string

Represents a series of 1 to 64 characters enclosed by delimiters as the collating sequence. Unspecified characters are equivalent to the highest value in the collating sequence.

optional-table

Represents the name of a collating sequence defined when Query Update is installed. Query Update is compiled with a variable number of collating tables; COBOL and FORTRAN are two table names. More tables can be defined and named (one to ten alphanumeric characters) during installation.

The ON portion of the directive specifies the data name of a sequencing field. The data name must exist on the directory of the file to be sorted. Up to 25 sequencing fields can be involved in a sort. Sequencing fields must be specified in major to minor order. If the DESCENDING option is included, ordering for the data name is to be from high to low; otherwise, ordering is in ascending sequence. If this option is omitted, the sort key is the whole record treated as an alphanumeric string.

The UNIQUE option eliminates duplicate records from the sorted file. If this option is omitted, duplicate records are retained. The SORT directive must be the last directive in a transmission.

SPECIFY

The SPECIFY directive establishes a name for convenient reference to a condition. Reference to the condition name in subsequent IF and SELECT directives results in testing of the specified condition. The format of the SPECIFY directive is shown in figure 4-62.

```
SPECIFY condition-name AS condition
```

Figure 4-62. SPECIFY Directive Format

A simple condition is a relational expression; that is, an expression joined with another expression by a relation operator.

expression relational operator expression

A compound condition joins relational expressions or expressions with logical operators.

$$\text{relational exp}\quad\text{logical operator}\left\{\begin{array}{l}\text{relational exp}\\ \text{expression}\end{array}\right\}\ldots$$

A complete description of simple and compound conditions appears in section 3. Relational and logical operators are described in detail.

When a condition contains data names, the removal or redefinition of a data name (via a RETURN, ERASE, EXTRACT, INVOKE, or DESCRIBE directive) invalidates the condition. A subsequent attempt to use a condition name that references an invalidated condition would give indeterminate results.

STOP

The STOP directive terminates Query Update operations and returns control to the operating system. STOP is synonymous with END except when using Query Update under the NOS time-sharing system where the END directive must be used for normal termination. It is recommended that the END directive be used; the STOP directive is described only for compatibility with previous releases of Query Update. The format of the STOP directive is shown in figure 4-63.

```
STOP
```

Figure 4-63. STOP Directive Format

Refer to the END directive for details of terminating Query Update.

STORE

The STORE directive creates a new record and places it in an area of the data base; record values are established. Only one area can be modified at a time; each area joined in a relation must be modified separately. Data can be input from the terminal or from a designated file. The format of the STORE directive is shown in figure 4-64. An example of the STORE directive is shown in figure 4-65.

The STORE directive requires permission to modify or extend the area file. This permission is granted according to one of the following:

The passwords specified for the area file with the PW option of the INVOKE directive

The passwords specified in the subschema in CRM data base access mode

The keys specified in the ACCESS directive in CDCS data base access mode

Record-name is the name of the record that is created; if omitted, the record name is inferred from the other item names in the STORE format. Data-name-1, data-name-2, and data-name-3 refer to either data base identifiers within record-name or temporary items declared by a DEFINE directive. The primary key must be included in the data-name-1, data-name-2, or data-name-3 list if data base items are referenced and the file does not have actual key file organization. Record-name, data-name-1, data-name-2, and data-name-3 must all have the same record type.

```
                       ||  SETTING {data-name-1} ... [FROM file-name]           ||
 STORE [record-name]†  ||                                                       ||
                       ||  MOVE expression-1 TO {data-name-2} ...               ||
                       ||       [AND expression-2 TO {data-name-3} ...] ...     ||

          [VETO]
          [PASS]

 †Not applicable to IMF
```

Figure 4-64. STORE Directive Format

```
-- STORE SETTING EMP-ID EMP-NAME PHONE

>> 00111 $GEORGE ROBINSON$  $4027236105$
>> 62417 $REBECCA SWARTZ$   $6037741069$
   *END
```

Two new records are added. The first record has 00111 entered into the EMP-ID field, GEORGE ROBINSON entered into the EMP-NAME field, and 4027236015 entered into the PHONE field. The second record has 62417 entered into the EMP-ID FIELD; the remaining data items are entered into respective fields.

Figure 4-65. STORE Directive Example

The SETTING option causes Query Update to either prompt the user or read the file file-name to obtain data values. These values are placed, after appropriate conversion, into the data base items or temporary items indicated by the name items in the STORE directive format. Each line of user-supplied input or each record in file-name yields one new record in the data base (after the processing of any MOVE options).

The MOVE option causes Query Update to:

Evaluate expression-1.

Move the result to the data-name-2 list.

Repeat the process for each additional AND clause specified.

Expression-1 and expression-2 are arithmetic expressions or logical conditions, containing items within record-name, temporary items, or literals.

If the MOVE option is used without the SETTING option, the STORE directive produces one new record in the data base.

The STORE directive can be preceded by an IF directive on a temporary item. However, no data base-accessing directive, other than IF, can occur in the same transmission as a STORE directive containing the SETTING option.

The VETO option causes the record to be displayed before it is stored. The user responds with YES or Y (store the record), NO or N (do not store the record), PROCEED or P (negate the current VETO option and store the record), or EXIT or E (do not store the record and terminate the transmission).

The PASS option disables the VETO for this transmission when a previous VETO directive specified VETO ON.

If data is input from the terminal, Query Update replies to the STORE directive with two >> symbols and waits for data to be entered. Query Update continues to store records and replies with the >> symbols after each transmission until the user enters *END to terminate the input. If the sequence of data is not to be input from the terminal, the FROM option specifies a file as the alternate source.

SUMMARY

The SUMMARY directive is a report specification that causes lines of text or another report to follow the current report. The summary is produced immediately following the report named in the current FORMAT directive. The SUMMARY directive is convenient for disclaimers, footnotes, bibliographies, or a general summary in a format that differs from the main body of the report. The format of the SUMMARY directive is shown in figure 4-66.

```
SUMMARY IS {TEXT       } FROM file-name
           {report-name}
```

Figure 4-66. SUMMARY Directive Format

The file name indicated in the FROM specification supplies the source data for the summary. For more information see the External File Organizations subsection.

The TEXT option specifies that each record from the specified file be treated as a single line; each record is shifted right one position to provide for single line spacing. Lines are printed from the top of the page to the installation default maximum number of lines.

The content of the specified file can be established by any file production technique that produces display coded records. Each line is truncated as needed to fit the installation default report width.

The report name option uses directives for the specified report to produce a summary report. The summary contains more than one report when the specified report name also contains a SUMMARY (or PREFACE) report name directive. The report name specified in this directive must be in the current catalog.

TABS

The TABS directive is a report specification that establishes tabular positions for a report layout. The designated positions reference horizontal column numbers. The format of the TABS directive is shown in figure 4-67.

```
TABS AT integer-1 [integer-2] . . .
```

Figure 4-67. TABS Directive Format

Report layout tabular positions begin at the print columns specified by the integers. The leftmost print column is numbered 1. The rightmost print column cannot extend beyond the report width. The integers need not be entered in sequence, because the system sorts them in ascending order. Subsequent references to tabular positions (TAB 1, TAB 2, TAB 3 and so forth) are based on the sorted sequence. The integers should be unique; duplicates are ignored.

The CENTERED option for the DETAIL, FOOTING, HEADING, RECAP, and TITLE directives can begin the centering process at a point specified by an IN clause. When a tab setting is specified as the starting point, information is placed between that point and the next tab. If no tab is encountered, the page width defines the rightmost point for the space where centering occurs.

TIME

The TIME directive is a report specification that indicates the vertical and horizontal positioning of the system-supplied time. The digits for hours, minutes, and seconds are expressed as hh.mm.ss; a space appears before and after the time. The format of the TIME directive is shown in figure 4-68.

```
TIME  [ AT  ┌ TITLE-LINE     ┐ ┌ TAB integer-2    ┐ ]  . . .
            │ LINE integer-1 │ │                  │
            └ RECAP-LINE     ┘ └ COLUMN integer-3 ┘
```

Figure 4-68. TIME Directive Format

The TIME directive with no options specified causes the time to appear at line 1 in column 90; if the page size is less than 99 columns, the default placement is the page width minus 29 columns. Vertical positioning of the time can be specified by one of three options:

TITLE-LINE

Positions the time in the first line of the title. If there is no title directive, the date is positioned in the first line of each page.

LINE integer-1

Places the time in the line that is integer-1 lines from the top of the page.

RECAP-LINE

Positions the time in the first line of the recap. If no RECAP directive was specified, or the BEYOND option was used in the RECAP directive, the date is positioned in the last line of the page.

Horizontal positioning of the time is specified by either the TAB or COLUMN option.

TAB integer-2

Begins the time in the column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth).

COLUMN integer-3

Begins the time in the print position specified by integer-3.

The time can appear more than once on a page through repetition of the AT vertical and horizontal positioning specifications. If either specification is omitted, the time is positioned at the default for that specification:

Vertical default

Line 1.

Horizontal default

Column 90; if the page size is less than 99 columns, default is the page width minus 29 columns.

TITLE

The TITLE directive is a report specification that establishes the content and positioning of one or more lines to be placed at the beginning of each report page. When a PAGE-NUMBER, TIME, or DATE directive is specified for the title line, the information is added to the title line after centering or positioning. The format of the TITLE directive is shown in figure 4-69.

The AT LINE option specifies the number of the line on which the title is to appear. Line numbers are counted from the top of the page; the top line is numbered 1. Multiple line titles are specified by repeating the AT LINE option followed by other elements of the directive.

The content of a title line is specified by one or more IS clauses. Multiple IS clauses are used to indicate the horizontal positioning of specific data; the IN option is then included with each IS clause. The expression in an IS clause specifies data that is to appear in the title line; it identifies one or more data fields, literals, or defined items. With multiple IS clauses, the optional elements are specified for each expression to which they apply. A data name in an expression can contain only that information accumulated up to the point where the title line is generated; if evaluation is involved, the user is responsible for assuring that the EVALUATE directive has been executed.

The CENTERED option causes the expression to be automatically positioned midway between two horizontal points. If no points are specified by an IN clause, default horizontal points of column 1 and the page width are assumed. A horizontal point specified by the IN clause represents the left point; the right point is the next tab or the page width. Centering is applied only to the expression associated with the CENTERED option.

The IN clause specifies the horizontal positioning of the title. The option selected specifies either the starting point of the title or the left point to be used for centering.

NEXT TAB

The next tabular position after the current position; TAB 1 is assumed if a previous tab has not been specified in the directive.

TAB integer-2

The column corresponding to the tab setting of the specified tab number (TAB 1, TAB 2, and so forth)

COLUMN integer-3

The print position specified by integer-3.

The title is placed on the page as specified in the TITLE directive; additional information (page number, date, or time) is then added. If the title is overwritten by this information, an informative diagnostic is issued and processing continues.

```
        ( [AT LINE integer-1]
  TITLE <
        (
              (                                 [    ( NEXT TAB         ) ] )   )
              < IS expression [CENTERED]        [ IN < TAB integer-2    > ] > ... > ...
              (                                 [    ( COLUMN integer-3 ) ] )   )
```

Figure 4-69. TITLE Directive Format

UNIVERSAL

The UNIVERSAL directive establishes a character that marks a character position to be ignored during comparison testing. The universal character is recognized only in nonnumeric literals. The format of the UNIVERSAL directive is shown in figure 4-70.

```
UNIVERSAL [IS universal-character]
          [OFF                   ]
```

Figure 4-70. UNIVERSAL Directive Format

When Query Update is entered, no universal character is in effect. The UNIVERSAL directive with neither option included causes the default character to become the universal character. The default character is the ≡ character.

The universal character option specifies one of the following characters as the universal character:

- * / $ [ ] ≠ → ∧ ↑ ↓ < > ≤ ≥ ≡ ∨

The character selected for this option cannot be the current delimiter. (Refer to the SEPARATOR directive.) If the current delimiter is specified as the universal character, a diagnostic message is issued.

The OFF option specifies that no universal character is to be recognized by Query Update. Any previous specification of a universal character is automatically erased. This option is executable whether or not a universal character is currently defined.

The universal character is used in nonnumeric literals that appear within relational expressions. If the default character is in effect, the test of the relational expression NAME EQ $JON S$ would be true when the data item NAME has the value JONAS, JONBS, JONES or JON'S; it would be true no matter what character is in the fourth character position. The relational expressions, either alone or as part of a compound condition, can appear within the IF, BREAK, SELECT, PERFORM, and SPECIFY directives. The universal character is always interpreted as universal when used in this context; in any other context, the universal character is manipulated as a standard character. In a situation where all available universal characters are needed as standard, unique-valued characters within literals, the universal character feature cannot be used.

A character data item can be compared to a nonnumeric literal that is not the same length. When the data item is shorter than the literal, the data item is padded with blanks. When the literal is shorter than the data item, padding occurs as follows. If the literal ends with the universal character, the literal is padded with universal characters up to the length of the data item. Otherwise, the literal is padded with blanks up to the length of the data item.

UPDATE

The UPDATE directive permits an area to be identified for updating before a REMOVE, STORE, or MODIFY directive is entered. When more than one area is being referenced during a session, as is the case in relation processing, Query Update must be able to determine which area is to be updated when an updating directive is issued. Query Update can make this determination from the data names in the directive if no duplicate data names exist in all the areas in use, or if the data names are properly qualified. When UPDATE area-name is used, the area is explicitly identified and the need for qualification is eliminated. The format of the UPDATE directive is shown in figure 4-71.

```
UPDATE area-name
```

Figure 4-71. UPDATE Directive Format

The area name specified must be an area established in the subschema currently in use. All subsequent updating operations are performed on the area specified. The designated area is identified for updating until another UPDATE area-name is issued or the Query Update session is terminated. Figure 4-72 illustrates the use of the directive.

```
--
        UPDATE VENDORS

      VENDORS is the name of the area to be updated.
```

Figure 4-72. Sample UPDATE Directive

Although the UPDATE directive is available to the Query Update user for modifying data items within records in a data base, the MODIFY directive performs the same function, and is preferred because it is more flexible. Refer to appendix H for a description of the UPDATE directive.

NOTE

Refer to appendix J for recommendations on the use of this feature.

USE

Although the USE directive is available to the Query Update user, the INVOKE directive performs the same function, and is preferred because it is more flexible. Refer to appendix H for a description of the USE directive.

NOTE

Refer to appendix J for recommendations on the use of this feature.

VERIFY

The VERIFY directive specifies the names of data items to be displayed at a terminal in response to the VETO option of REMOVE, STORE, and MODIFY directives.

If no VERIFY directive is in effect, or if a VERIFY directive without a parameter is given, Query Update displays the first 40 characters of the record being processed when operating in VETO mode. The format of the VERIFY directive is shown in figure 4-73.

```
VERIFY [data-name] . . .
```

Figure 4-73. VERIFY Directive Example

VERSION

The VERSION directive determines whether Query Update enters either CRM catalog mode or CDCS catalog mode. There are three formats of the VERSION directive, as shown in figure 4-74. Formats 1 and 2 can be used in any data base access mode to enter CRM catalog mode. Format 3 can be used in CDCS data base access mode to enter CDCS catalog mode. Examples of the VERSION directive are shown in figure 4-75.

CRM CATALOG MODE

The current catalog receives directives to be recorded or provides previously recorded directives. Initially, the Query Update default catalog ZZZZZQ2 is available, as a local file, for recording directives or retaining report formats. This catalog can be made permanent at the end of a session with the appropriate operating system commands. (Refer to the applicable system reference manual.) In subsequent sessions, the catalog can be attached with the VERSION directive. Whenever a VERSION directive is specified, the current catalog is closed and extended before the specified catalog is attached. The default catalog (ZZZZZQ2) is reinstated as the current catalog when format 1 of the VERSION directive is used. When format 1 is used, any previous version catalog is no longer available.

When format 2 is used, the specified permanent file is attached and used for all reports and sessions. In addition, the default catalog is available for duplication. When format 2 is used, any prior version catalog is no longer available.

Report specifications are automatically recorded on the current catalog. Other Query Update directives are recorded on the current catalog through the RECORDING directive. If a VERSION directive has not been specified, recording occurs on the default catalog.

```
CDCS/CRM/IMF Format 1

   VERSION IS DEFAULT


CDCS/CRM/IMF Format 2

   VERSION IS catalog-file [( permanent-file-parameters [PW] )]


CDCS Format

   VERSION IS catalog-file OF subschema-name
               [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]
                    [FOR DATABASE version-name]
```

Figure 4-74. VERSION Directive Format

Example 1

```
-- VERSION IS CATALOG-FILE (ID=ABC)
```

CATALOG-FILE is attached and used for all reports and sessions. Any existing current catalog is closed. The user enters CRM catalog mode.

Example 2

```
-- VERSION IS CATALOG-FILE     +
      OF QUEXMPL FROM LIBRARY QULIB (ID=ABC)
```

The subschema QUEXMPL is attached and checked for correct description of CATALOG-FILE. The subschema library name is QULIB. CATALOG-FILE is used for all reports and sessions. The user enters CDCS catalog mode.

Figure 4-75. VERSION Directive Examples

To read, modify, or extend a current catalog, the appropriate password must be specified in the VERSION directive.

If the user is in CDCS catalog mode and not in CDCS data base access mode, prior to entering the format 1 or 2 VERSION directive, the previous subschema is released. CDCS is informed that the previous subschema is no longer in use.

If the user is in CDCS catalog mode and in CDCS data base access mode, when entering format 1 or 2 of the VERSION directive, the previous subschema is still available and the previous catalog is replaced by a new catalog.

The PW option in format 2 of the VERSION directive protects the security of the peruanent file parameters. To use the PW option, the user enters PW without associated values. Query Update responds with the prompt >>>. The user then enters the appropriate permanent file parameters. Query Update continues to prompt until the user enters *END. The security of the permanent file parameters is protected because printing of the parameters is inhibited on the trace file and output listing.

CDCS CATALOG MODE

Under CDCS catalog mode, the specified subschema file is attached (if not already attached). Query Update checks that the catalog file is described correctly within the subschema. The specified catalog file is used for all reports and sessions. In addition, the default catalog is available for duplication.

If the user is in CDCS data base access mode before entering format 3 of the VERSION directive, and if the same subschema is being used, Query Update sets internal descriptions to indicate which file is the catalog file. The effects of the previous INVOKE remain in effect.

Any previous subschema file and all tables describing previous subschema, areas, and catalog files are released under the following conditions:

CDCS data base access mode was in effect with either a different subschema or different data base version. The previous INVOKE is no longer in effect.

CDCS catalog mode was in effect with either a different subschema or data base version.

CRM data base access mode was in effect. (That is, a Query Update/CRM subschema was being used.) The previous INVOKE is no longer in effect.

The catalog-file identifies the catalog being used. Catalog-file must be an area name included in the subschema identified by subschema-name.

The subschema-name identifies the subschema being used. The subschema name must be the name of a Query Update/CDCS subschema.

The FROM LIBRARY option identifies the subschema library that contains the subschema directory being used. If the FROM LIBRARY option is omitted, the permanent file name of the subschema library is assumed as follows: on NOS, the first seven characters of the subschema name are used; on NOS/BE, the entire subschema name is used.

When permanent file parameters are required to access the subschema, they must be specified in the VERSION directive. The PW option can be specified to protect the security of the passwords. When Query Update encounters the characters PW alone, terminal input is prompted with the symbols >>>. The user then enters the appropriate passwords. Query Update continues until the user enters *END. The PW option protects the security of passwords since printing of passwords on the output listing or trace file is inhibited.

VETO

The VETO directive causes data to be displayed at the terminal whenever a record is being deleted, inserted, or updated. This directive is ignored in batch mode. The VETO directive is also used to disable a previous VETO directive. The format of the VETO directive is shown in figure 4-76.

$$\underline{\text{VETO}}\ \left\{\begin{array}{l}\underline{\text{ON}}\\ \underline{\text{OFF}}\end{array}\right\}$$

Figure 4-76. VETO Directive Format

The ON option enables the veto operation for all subsequent DELETE, INSERT, and UPDATE directives. When no data names have been specified in a VERIFY directive, the first 40 characters of the record are displayed at the terminal. Query Update waits for one of the following user responses before performing deletion, insertion, or modification:

YES (or Y)

Allows deletion, insertion, or updating.

NO (or N)

Prevents deletion, insertion, or updating.

PROCEED (or P)

Negates the current VETO option for the transmission; acts as a user response of YES for this record.

EXIT (or E)

Prevents deletion, insertion, or updating; and terminates the transmission.

When the ON option has been specified and a PERFORM directive has been entered, each transmission being performed is displayed at the terminal, and the user must respond Y or N to determine whether or not the transmission is executed. (The P or E responses are also valid.)

The OFF option disables a previous VETO ON directive. At the time Query Update is entered, VETO OFF is in effect.

VIA

The VIA directive is used in relation processing to specify to Query Update which relation should be followed when an ambiguity exists. Ambiguities can exist if the same areas are joined in different relations, and the data items named in DISPLAY and IF directives are from these areas only. The VIA directive need not be entered if only one relation is defined in the subschema, or if no ambiguity exists. The format of the VIA directive is shown in figure 4-77.

```
VIA relation-name
```

Figure 4-77. VIA Directive Format

The relation name specified must be an active relation defined in the subschema currently in use. If the VIA directive is specified and not needed, it is ignored. It can be entered alone or in a transmission preceding or following a query. If more than one VIA directive is entered in a transmission, only the last one specified is used. The relation specified by the VIA directive is accessed whenever an ambiguity exists until a subsequent VIA directive is entered.

If a VIA directive is not included in a transmission in which an ambiguity exists or has not preceded the transmission, Query Update replies with a list of relation names and the prompting symbols >>, and waits for the user to enter one of the relation names from the list. If the user wishes to exit from this mode to study the existing relations, *END or any word that is not a relation name can be entered in response to the prompt.

REPORT GENERATION

The report extractor portion of the Query Update software system is controlled by user-constructed directives for retrieval of information from sequential files and production of printer-formatted reports. A report typically consists of pages containing such information as descriptive titles, columnar divisions, and informative data.

Specifications for a report are established through a series of layout directives. These directives determine the format of the report. Once the report specifications are established, a sample report can be previewed or an actual report can be prepared.

Report specifications that do not change between runs can be compiled in encoded form and stored in a table file. A report is then generated by referencing the report specifications in a control statement that calls the REPORT utility program.

Disposition of printer-formatted output is determined by the user. An output report can be retained as a permanent file, discarded, displayed at a terminal, punched, or printed; normally, a report is produced in a printed form.

Figure 5-1 illustrates the flow of operation for generating a report. Dashed lines indicate alternatives for directive input and output disposition. Figure 5-2 illustrates the flow of operation for the REPORT utility program.

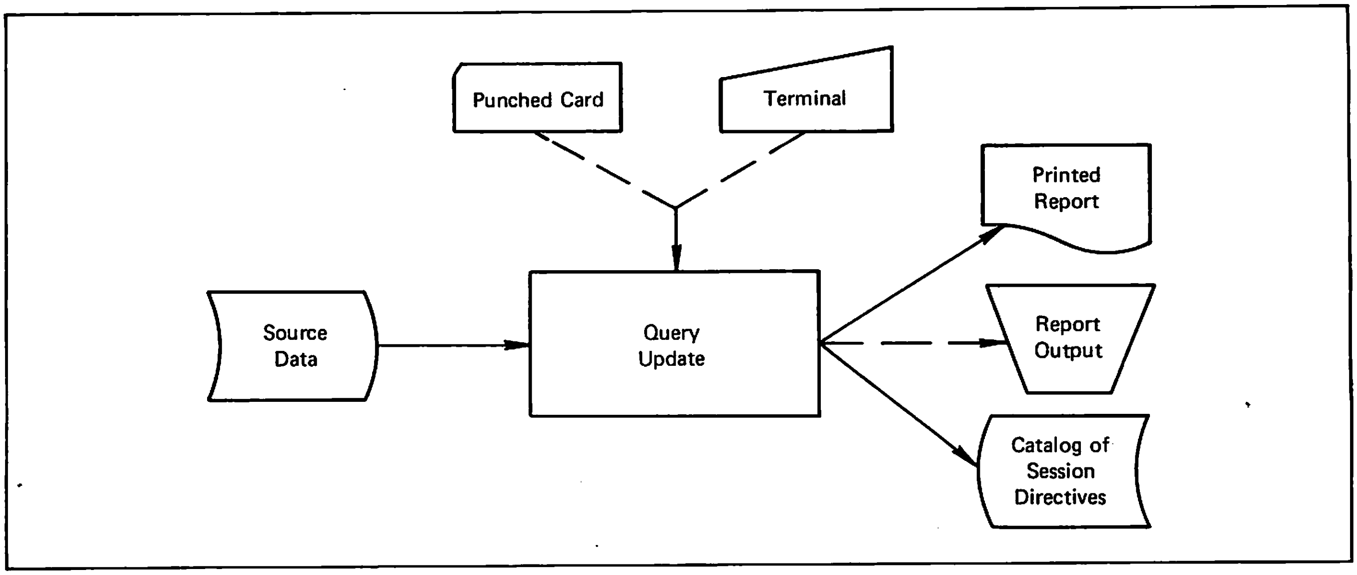


Figure 5-1. Query Update Report Generation

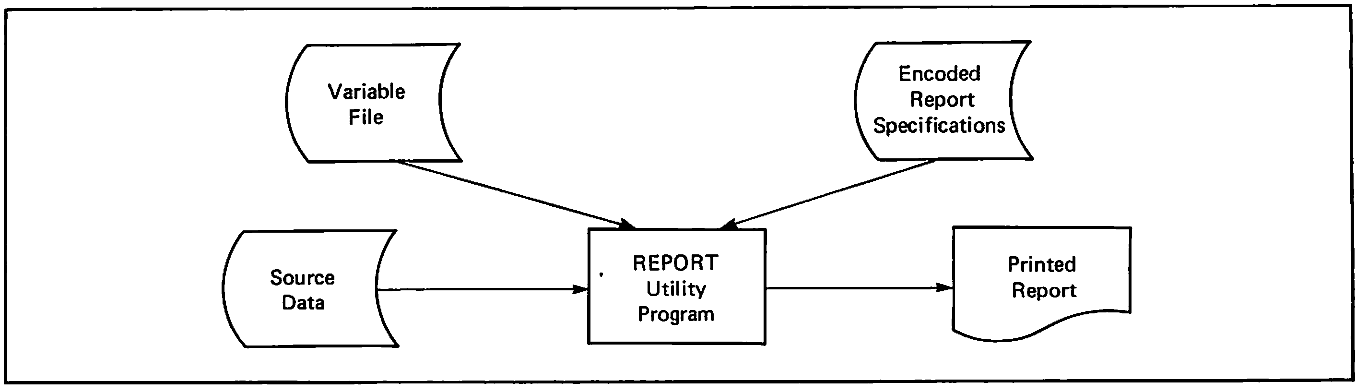


Figure 5-2. REPORT Utility Program Report Generation

REPORT SPECIFICATIONS

The format of a report is supplied through various layout directives. The format is established by first specifying the FORMAT directive to assign a name to identify the complete set of report specifications. The layout directives are then submitted to specify the format to be followed during report preparation.

The layout directives are retained on the current catalog under the report name. If the current catalog is the default catalog, it can be made permanent at the end of the session for use in subsequent sessions.

The most commonly used layout directives are discussed briefly in the following paragraphs and are used to create the sample reports shown in this section. The order in which directives are specified is not significant; however, the directives are presented in a logical order. Complete descriptions of the layout directives are contained in section 4 of this manual. The layout directives for each report can contain a maximum of 200 combined data names, literals, and expressions.

The TABS directive specifies the tab settings for the columns of the report. These tab settings then can be referenced in other layout directives for the placement of data. Reference to a tab setting specifies the relative number of the tab setting; TAB 1 references the tab setting with the lowest column number, TAB 2 references the tab setting with the second lowest column number, and so forth.

The information that is to appear at the top of each report page is specified in a TITLE directive. One or more lines can be described in the directive. The content of a title line is specified by one or more expressions. (Refer to section 3 for a complete description of expressions.) Expressions in a TITLE directive usually contain literals to be placed on specified lines and data names of items to be included in the title lines.

The BREAK directive is used to designate a situation where the body of the report is to be interrupted to print footings and headings. When the specified situation occurs, footings with the same level numbers (or higher level numbers) as the BREAK directive are printed. If a heading has the same level number, the heading and all headings with higher level numbers are printed before detail line printing resumes.

A HEADING or FOOTING directive specifies the information that is to be placed in a heading or footing line. Heading and footing lines are associated with the beginning of data, the end of data, or a BREAK directive. The level number determines when the line is to be printed. Expressions specify the data to be placed in the line. Multiple lines can be described in a HEADING or FOOTING directive.

The DETAIL directive describes one or more detail lines to be placed in the body of the report. The content of a detail line is specified by one or more expressions. When more than one DETAIL directive is required, each directive must be assigned a tag number that corresponds to a SELECT directive.

The SELECT directive specifies the condition under which a DETAIL directive is selected to place a detail line in the body of the report. Each source record is tested against the conditions of the SELECT directives. When a condition is true, the DETAIL directive with the corresponding tag number is selected, and a detail line is placed in the report.

In addition to the directives described in the preceding paragraphs, the following layout directives can be used to specify the format of a report:

DATE

Indicates the vertical and horizontal positioning of the system-supplied date.

EVALUATE

Performs evaluation of a temporary data item before or after a report production step.

MOVE

Places a value in a data item before or after a report production step.

PAGE-NUMBER

Indicates the vertical and horizontal positioning of the system-supplied page number.

PAGE-SIZE

Specifies the maximum number of lines, columns, sections, and multiple copy images on a report page.

PREFACE

Causes lines of text or another report to precede the current report.

RECAP

Specifies a report line to be generated as each page is completed.

SUMMARY

Causes lines of text or another report to follow the current report.

TIME

Indicates the vertical and horizontal positioning of the system-supplied time.

REPORT PREPARATION

A report is produced from a source data file according to the format specifications established through the layout directives. The format specifications must exist in the current catalog (the default catalog or a catalog attached with the VERSION directive).

The PREVIEW directive can be used to produce a two-page sample report. The sample report can be reviewed to ensure that the format specifications produce the desired results. Actual data or dummy data can be used to produce the sample report. When the sample report indicates that changes to the report specifications need to be incorporated, the ALTER directive provides the means to modify the specifications. Subsequent directives remove, replace, or add report specifications.

The actual report is generated by the PREPARE directive. The source records in the designated file are used to create the report in the format specified by the layout directives. The finished report is placed in a local file; the file can then be printed, retained as a permanent file, or disposed in any other manner specified by the user.

SAMPLE REPORTS

The four sample reports that follow are illustrated and described in the following paragraphs. All examples have been prepared as one continuous session.

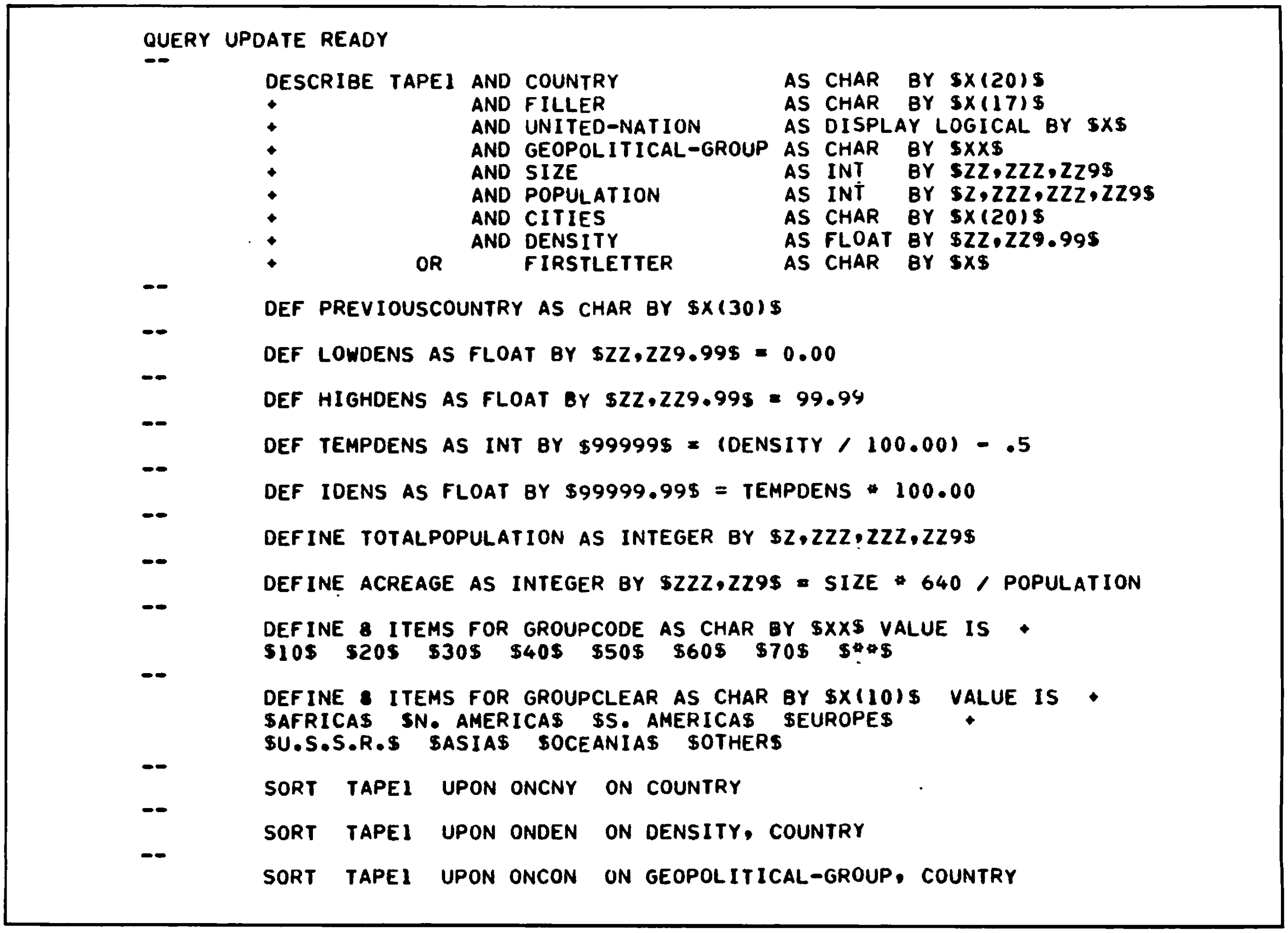
SAMPLE REPORTS DATA DESCRIPTION

The sample reports in this section were prepared using the file and the temporary data items shown in figure 5-3. The DESCRIBE directive defines the file TAPE1. The DEFINE directives establish temporary data items that are used selectively in the sample reports. The SORT directives cause the records to be sorted before the reports are generated.

CYLIST REPORT

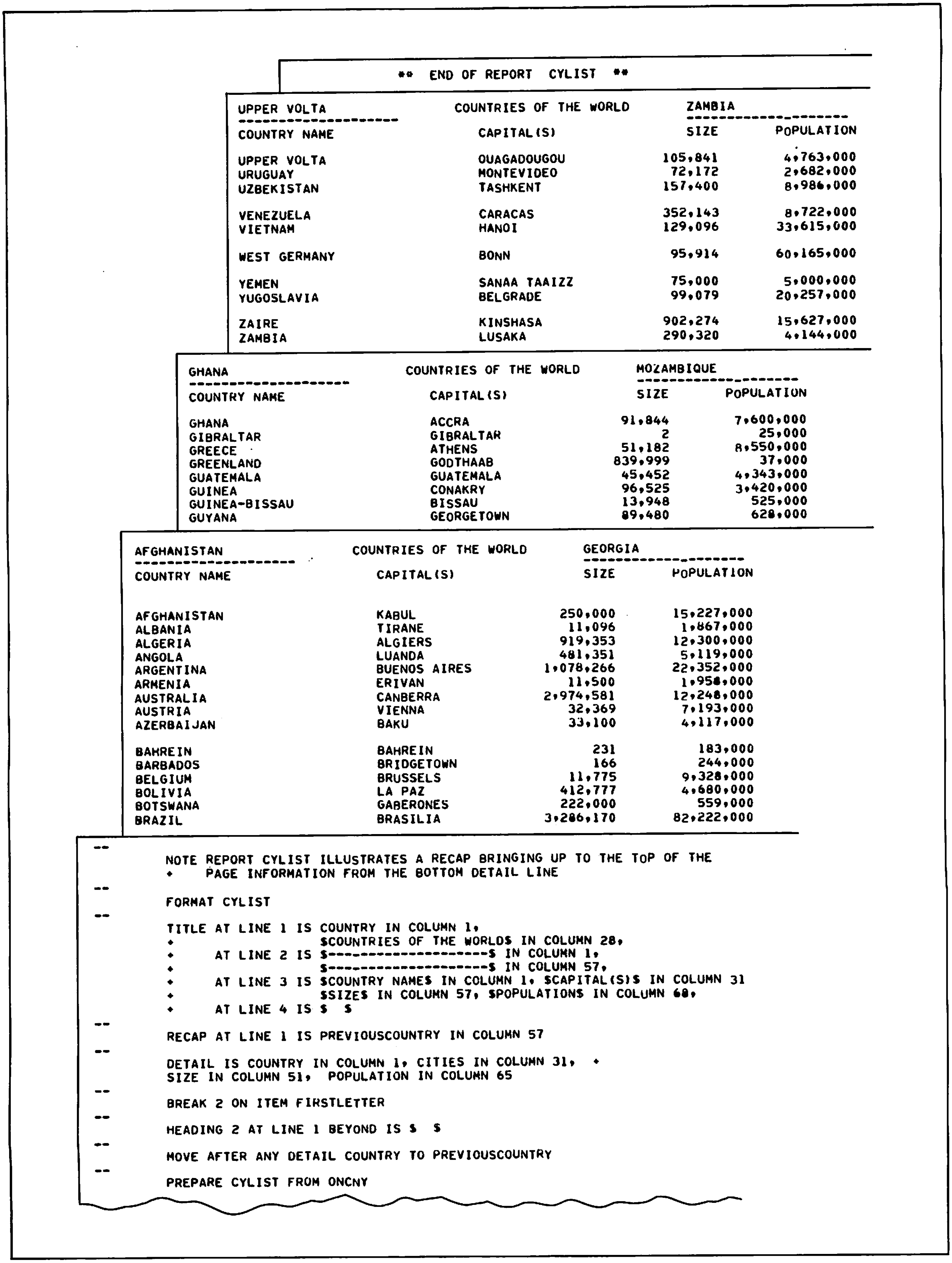
An alphabetic list of the countries (including capitals, size, and population for each) is produced in the CYLIST report. The layout directives for this report include the RECAP directive, which adds to the title line the name of the last country included on the page; the BREAK directive, which defines the condition for interrupting detail lines to insert a heading; and the HEADING directive, which specifies a blank line to be inserted when the BREAK directive condition occurs.

The MOVE directive is executed after a detail line is processed; it places the name of the country into a temporary data item. Figure 5-4 shows the layout directives and sample pages for the CYLIST report.



```
          QUERY UPDATE READY
          --
                   DESCRIBE TAPE1 AND COUNTRY              AS CHAR    BY $X(20)$
                   +              AND FILLER               AS CHAR    BY $X(17)$
                   +              AND UNITED-NATION        AS DISPLAY LOGICAL BY $X$
                   +              AND GEOPOLITICAL-GROUP   AS CHAR    BY $XX$
                   +              AND SIZE                 AS INT     BY $ZZ,ZZZ,ZZ9$
                   +              AND POPULATION           AS INT     BY $Z,ZZZ,ZZZ,ZZ9$
                   +              AND CITIES               AS CHAR    BY $X(20)$
                   +              AND DENSITY              AS FLOAT   BY $ZZ,ZZ9.99$
                   +          OR      FIRSTLETTER          AS CHAR    BY $X$
          --
                   DEF PREVIOUSCOUNTRY AS CHAR BY $X(30)$
          --
                   DEF LOWDENS AS FLOAT BY $ZZ,ZZ9.99$ = 0.00
          --
                   DEF HIGHDENS AS FLOAT BY $ZZ,ZZ9.99$ = 99.99
          --
                   DEF TEMPDENS AS INT BY $99999$ = (DENSITY / 100.00) - .5
          --
                   DEF IDENS AS FLOAT BY $99999.99$ = TEMPDENS * 100.00
          --
                   DEFINE TOTALPOPULATION AS INTEGER BY $Z,ZZZ,ZZZ,ZZ9$
          --
                   DEFINE ACREAGE AS INTEGER BY $ZZZ,ZZ9$ = SIZE * 640 / POPULATION
          --
                   DEFINE 8 ITEMS FOR GROUPCODE AS CHAR BY $XX$ VALUE IS  +
                   $10$  $20$  $30$  $40$  $50$  $60$  $70$  $**$
          --
                   DEFINE 8 ITEMS FOR GROUPCLEAR AS CHAR BY $X(10)$  VALUE IS  +
                   $AFRICA$  $N. AMERICA$  $S. AMERICA$  $EUROPE$      +
                   $U.S.S.R.$  $ASIA$  $OCEANIA$  $OTHER$
          --
                   SORT  TAPE1  UPON ONCNY  ON COUNTRY
          --
                   SORT  TAPE1  UPON ONDEN  ON DENSITY, COUNTRY
          --
                   SORT  TAPE1  UPON ONCON  ON GEOPOLITICAL-GROUP, COUNTRY
```

Figure 5-3. DESCRIBE and DEFINE Directives for Sample Reports



```
                    **  END OF REPORT  CYLIST  **
```

```
UPPER VOLTA                  COUNTRIES OF THE WORLD     ZAMBIA
--------------------                                    --------------------
COUNTRY NAME                    CAPITAL(S)                    SIZE      POPULATION

UPPER VOLTA                     OUAGADOUGOU                105,841       4,763,000
URUGUAY                         MONTEVIDEO                  72,172       2,682,000
UZBEKISTAN                      TASHKENT                   157,400       8,986,000

VENEZUELA                       CARACAS                    352,143       8,722,000
VIETNAM                         HANOI                      129,096      33,615,000

WEST GERMANY                    BONN                        95,914      60,165,000

YEMEN                           SANAA TAAIZZ                75,000       5,000,000
YUGOSLAVIA                      BELGRADE                    99,079      20,257,000

ZAIRE                           KINSHASA                   902,274      15,627,000
ZAMBIA                          LUSAKA                     290,320       4,144,000
```

```
GHANA                        COUNTRIES OF THE WORLD     MOZAMBIQUE
--------------------                                    --------------------
COUNTRY NAME                    CAPITAL(S)                  SIZE      POPULATION

GHANA                           ACCRA                      91,844       7,600,000
GIBRALTAR                       GIBRALTAR                       2          25,000
GREECE                          ATHENS                     51,182       8,550,000
GREENLAND                       GODTHAAB                  839,999          37,000
GUATEMALA                       GUATEMALA                  45,452       4,343,000
GUINEA                          CONAKRY                    96,525       3,420,000
GUINEA-BISSAU                   BISSAU                     13,948         525,000
GUYANA                          GEORGETOWN                 89,480         628,000
```

```
AFGHANISTAN                  COUNTRIES OF THE WORLD     GEORGIA
--------------------                                    --------------------
COUNTRY NAME                    CAPITAL(S)                  SIZE      POPULATION


AFGHANISTAN                     KABUL                     250,000      15,227,000
ALBANIA                         TIRANE                     11,096       1,867,000
ALGERIA                         ALGIERS                   919,353      12,300,000
ANGOLA                          LUANDA                    481,351       5,119,000
ARGENTINA                       BUENOS AIRES            1,078,266      22,352,000
ARMENIA                         ERIVAN                     11,500       1,958,000
AUSTRALIA                       CANBERRA                2,974,581      12,248,000
AUSTRIA                         VIENNA                     32,369       7,193,000
AZERBAIJAN                      BAKU                       33,100       4,117,000

BAHREIN                         BAHREIN                       231         183,000
BARBADOS                        BRIDGETOWN                    166         244,000
BELGIUM                         BRUSSELS                   11,775       9,328,000
BOLIVIA                         LA PAZ                    412,777       4,680,000
BOTSWANA                        GABERONES                 222,000         559,000
BRAZIL                          BRASILIA                3,286,170      82,222,000
```

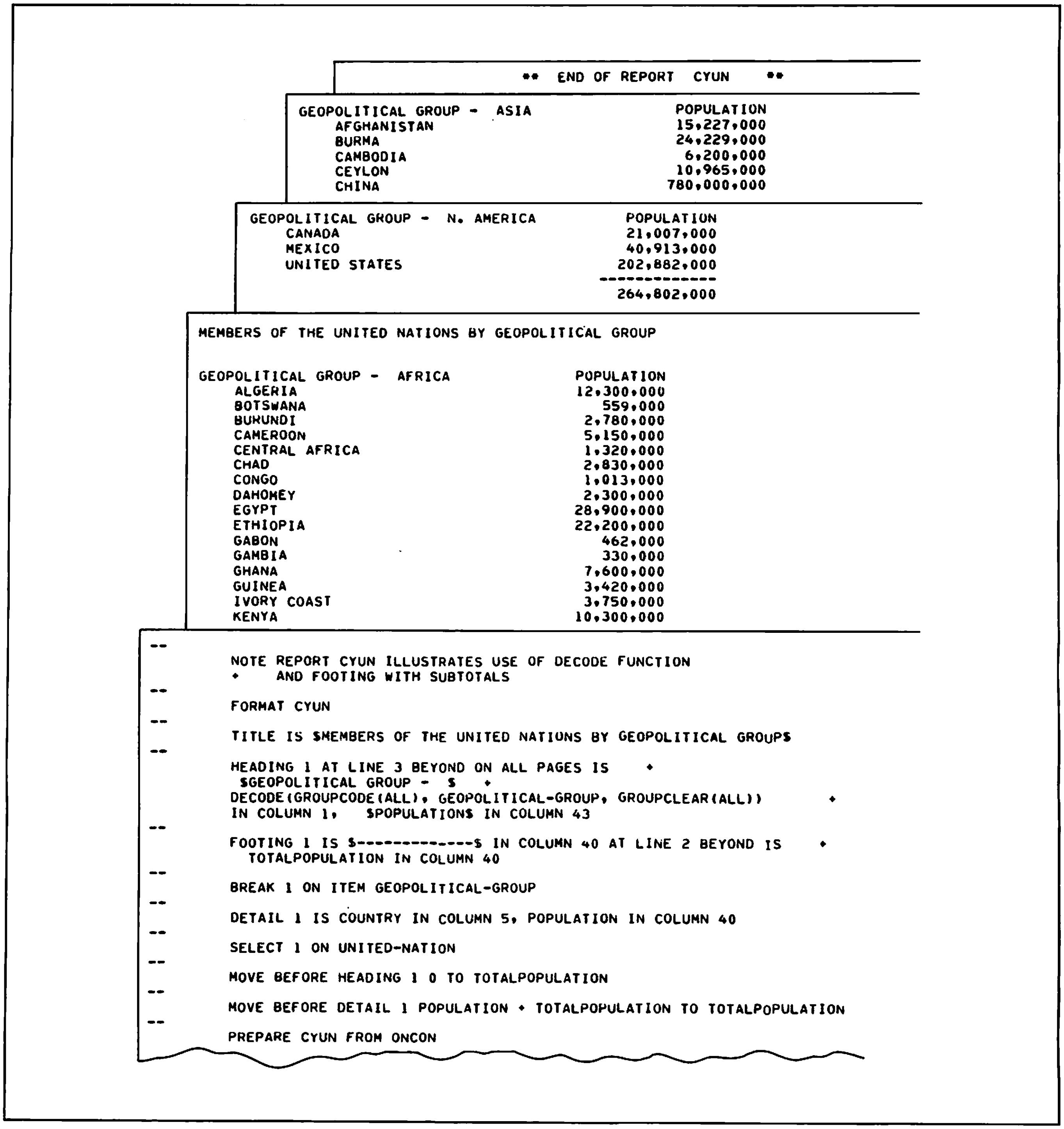
```
--
        NOTE REPORT CYLIST ILLUSTRATES A RECAP BRINGING UP TO THE TOP OF THE
        +    PAGE INFORMATION FROM THE BOTTOM DETAIL LINE
--
        FORMAT CYLIST
--
        TITLE AT LINE 1 IS COUNTRY IN COLUMN 1,
        +                  $COUNTRIES OF THE WORLD$ IN COLUMN 28,
        +     AT LINE 2 IS $--------------------$ IN COLUMN 1,
        +                  $--------------------$ IN COLUMN 57,
        +     AT LINE 3 IS $COUNTRY NAME$ IN COLUMN 1, $CAPITAL(S)$ IN COLUMN 31
        +                  $SIZE$ IN COLUMN 57, $POPULATION$ IN COLUMN 68,
        +     AT LINE 4 IS $  $
--
        RECAP AT LINE 1 IS PREVIOUSCOUNTRY IN COLUMN 57
--
        DETAIL IS COUNTRY IN COLUMN 1, CITIES IN COLUMN 31,  +
        SIZE IN COLUMN 51,  POPULATION IN COLUMN 65
--
        BREAK 2 ON ITEM FIRSTLETTER
--
        HEADING 2 AT LINE 1 BEYOND IS $  $
--
        MOVE AFTER ANY DETAIL COUNTRY TO PREVIOUSCOUNTRY
--
        PREPARE CYLIST FROM ONCNY
```

Figure 5-4. CYLIST Report

CYUN REPORT

The CYUN report lists the countries by geopolitical group; within each group, the countries are listed in alphabetic order. The layout directives and sample pages for this report are illustrated in figure 5-5.

The geopolitical group to which a country belongs is identified by a number in the GEOPOLITICAL-GROUP data item. The records are sorted by this number; all records with the same GEOPOLITICAL-GROUP number are sorted alphabetically. The DECODE function translates the GEOPOLITICAL-GROUP number to the geopolitical group name. A detail line in the report contains the name of the country and its population. The MOVE BEFORE DETAIL 1 directive maintains a running total of the population for each group.



```
                    **  END OF REPORT  CYUN     **
```

```
GEOPOLITICAL GROUP -  ASIA                  POPULATION
    AFGHANISTAN                             15,227,000
    BURMA                                   24,229,000
    CAMBODIA                                 6,200,000
    CEYLON                                  10,965,000
    CHINA                                  780,000,000
```

```
GEOPOLITICAL GROUP -  N. AMERICA            POPULATION
    CANADA                                  21,007,000
    MEXICO                                  40,913,000
    UNITED STATES                          202,882,000
                                         -------------
                                           264,802,000
```

```
MEMBERS OF THE UNITED NATIONS BY GEOPOLITICAL GROUP


GEOPOLITICAL GROUP -  AFRICA                POPULATION
    ALGERIA                                 12,300,000
    BOTSWANA                                   559,000
    BURUNDI                                  2,780,000
    CAMEROON                                 5,150,000
    CENTRAL AFRICA                           1,320,000
    CHAD                                     2,830,000
    CONGO                                    1,013,000
    DAHOMEY                                  2,300,000
    EGYPT                                   28,900,000
    ETHIOPIA                                22,200,000
    GABON                                      462,000
    GAMBIA                                     330,000
    GHANA                                    7,600,000
    GUINEA                                   3,420,000
    IVORY COAST                              3,750,000
    KENYA                                   10,300,000
```

```
--
         NOTE REPORT CYUN ILLUSTRATES USE OF DECODE FUNCTION
         +    AND FOOTING WITH SUBTOTALS
--
         FORMAT CYUN
--
         TITLE IS $MEMBERS OF THE UNITED NATIONS BY GEOPOLITICAL GROUP$
--
         HEADING 1 AT LINE 3 BEYOND ON ALL PAGES IS     +
          $GEOPOLITICAL GROUP -  $   +
         DECODE(GROUPCODE(ALL), GEOPOLITICAL-GROUP, GROUPCLEAR(ALL))       +
         IN COLUMN 1,    $POPULATION$ IN COLUMN 43
--
         FOOTING 1 IS $-------------$ IN COLUMN 40 AT LINE 2 BEYOND IS     +
           TOTALPOPULATION IN COLUMN 40
--
         BREAK 1 ON ITEM GEOPOLITICAL-GROUP
--
         DETAIL 1 IS COUNTRY IN COLUMN 5, POPULATION IN COLUMN 40
--
         SELECT 1 ON UNITED-NATION
--
         MOVE BEFORE HEADING 1 0 TO TOTALPOPULATION
--
         MOVE BEFORE DETAIL 1 POPULATION + TOTALPOPULATION TO TOTALPOPULATION
--
         PREPARE CYUN FROM ONCON
```

Figure 5-5. CYUN Report

A break in the detail lines occurs when the GEOPOLITICAL-GROUP number changes. The footing lines (a dashed line followed by the population total for the group) are then placed in the report. The MOVE BEFORE HEADING 1 directive zeros the population total to begin the next group total. The heading line is placed in the report followed by the detail lines for the group.

CYACRE REPORT

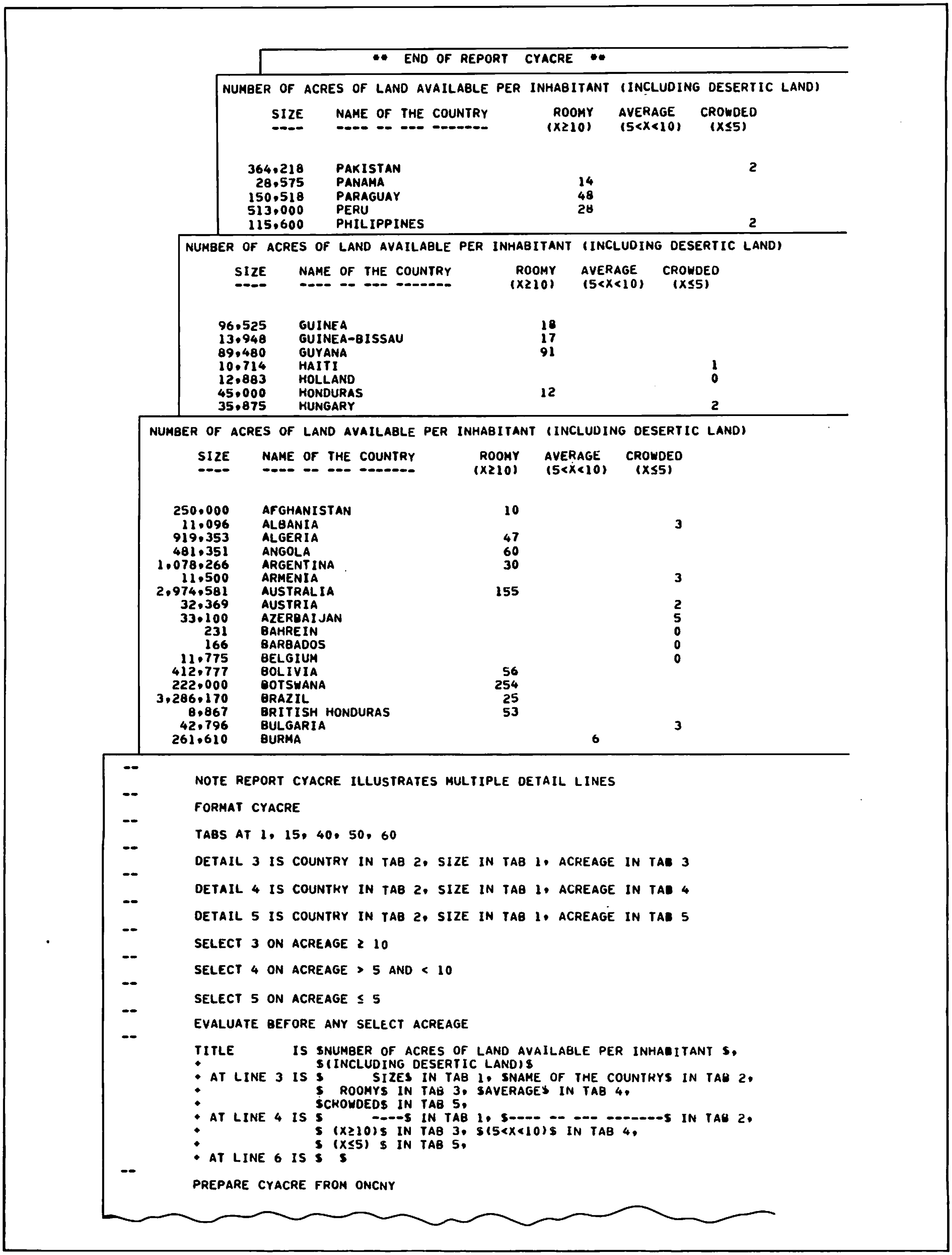
The size of each country and the number of acres per inhabitant are listed in the CYACRE report. In addition, each country is categorized as roomy, average, or crowded, based on the acres per inhabitant. Figure 5-6 shows the layout directives and sample pages for this report.

The acreage per inhabitant (ACREAGE data item) must be evaluated before a detail line can be placed in the report. (Refer to figure 5-3 for the DEFINE ACREAGE directive.

The value of ACREAGE determines the selection of a DETAIL directive to format the detail line. The report is prepared using the sorted file generated for the CYLIST report (figure 5-4).

CYDENS REPORT

The CYDENS report lists the countries by increasing density of population. The source data is sorted into numeric order by the DENSITY data item. The detail lines are grouped into density ranges, which are established by the temporary data items LOWDENS and HIGHDENS. When a record DENSITY item is greater than the HIGHDENS value, a break condition occurs and a new heading is placed in the report. Before the break occurs and the heading is generated, the temporary data items TEMPDENS and IDENS are evaluated, and the appropriate density range is moved to the heading line. (Refer to figure 5-3 for the DEFINE directives used in this report.) The layout directives and sample pages for the CYDENS report are shown in figure 5-7.



```
                    **  END OF REPORT  CYACRE  **

NUMBER OF ACRES OF LAND AVAILABLE PER INHABITANT (INCLUDING DESERTIC LAND)

      SIZE     NAME OF THE COUNTRY        ROOMY   AVERAGE   CROWDED
      ----     ---- -- --- -------        (X≥10)  (5<X<10)   (X≤5)


   364,218     PAKISTAN                                        2
    28,575     PANAMA                       14
   150,518     PARAGUAY                     48
   513,000     PERU                         28
   115,600     PHILIPPINES                                     2
```

```
NUMBER OF ACRES OF LAND AVAILABLE PER INHABITANT (INCLUDING DESERTIC LAND)

      SIZE     NAME OF THE COUNTRY        ROOMY   AVERAGE   CROWDED
      ----     ---- -- --- -------        (X≥10)  (5<X<10)   (X≤5)


    96,525     GUINEA                       18
    13,948     GUINEA-BISSAU                17
    89,480     GUYANA                       91
    10,714     HAITI                                           1
    12,883     HOLLAND                                         0
    45,000     HONDURAS                     12
    35,875     HUNGARY                                         2
```

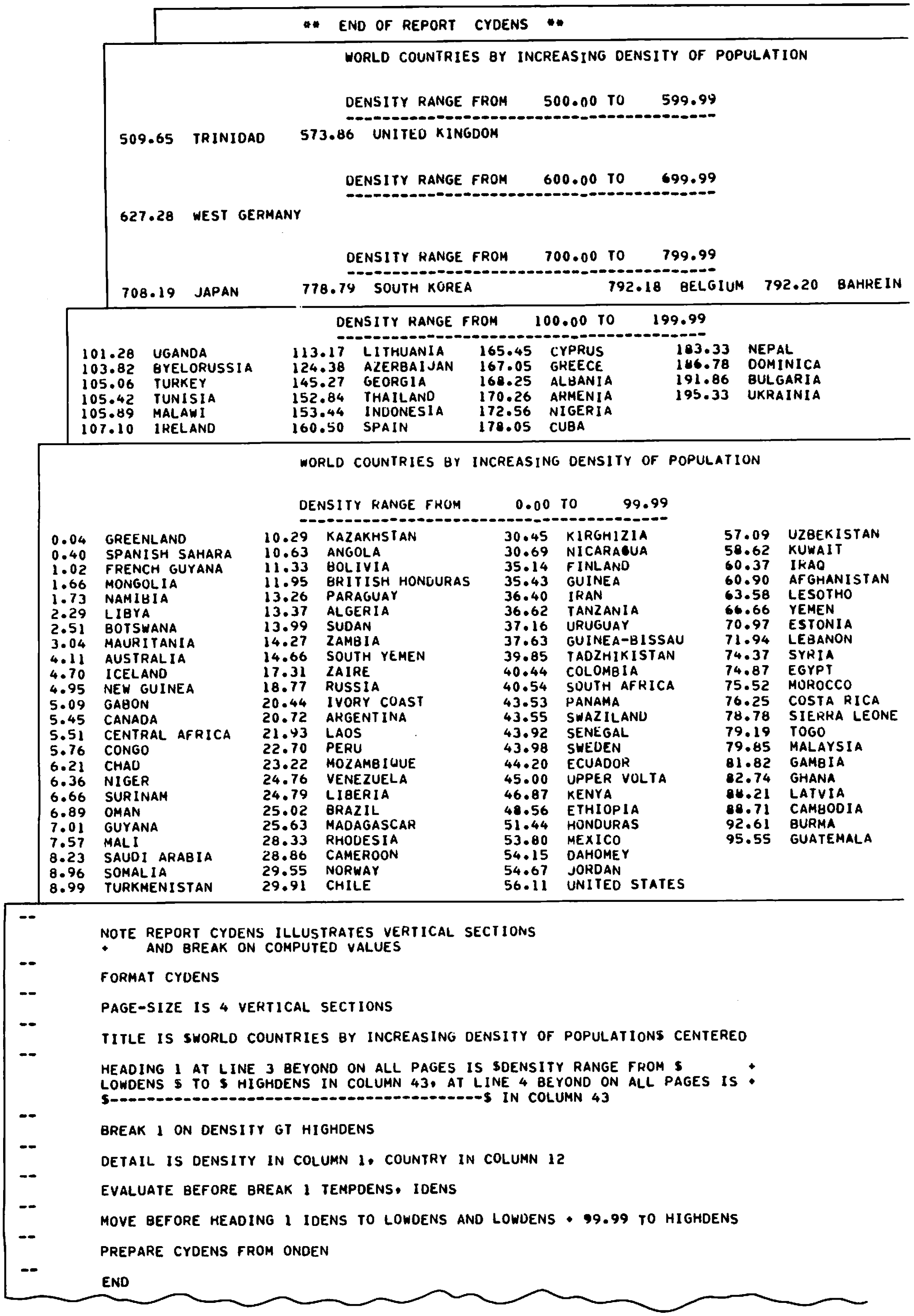
```
NUMBER OF ACRES OF LAND AVAILABLE PER INHABITANT (INCLUDING DESERTIC LAND)

      SIZE     NAME OF THE COUNTRY        ROOMY   AVERAGE   CROWDED
      ----     ---- -- --- -------        (X≥10)  (5<X<10)   (X≤5)


   250,000     AFGHANISTAN                  10
    11,096     ALBANIA                                         3
   919,353     ALGERIA                      47
   481,351     ANGOLA                       60
 1,078,266     ARGENTINA                    30
    11,500     ARMENIA                                         3
 2,974,581     AUSTRALIA                   155
    32,369     AUSTRIA                                         2
    33,100     AZERBAIJAN                                      5
       231     BAHREIN                                         0
       166     BARBADOS                                        0
    11,775     BELGIUM                                         0
   412,777     BOLIVIA                      56
   222,000     BOTSWANA                    254
 3,286,170     BRAZIL                       25
     8,867     BRITISH HONDURAS             53
    42,796     BULGARIA                                        3
   261,610     BURMA                                  6
```

```
--
        NOTE REPORT CYACRE ILLUSTRATES MULTIPLE DETAIL LINES
--
        FORMAT CYACRE
--
        TABS AT 1, 15, 40, 50, 60
--
        DETAIL 3 IS COUNTRY IN TAB 2, SIZE IN TAB 1, ACREAGE IN TAB 3
--
        DETAIL 4 IS COUNTRY IN TAB 2, SIZE IN TAB 1, ACREAGE IN TAB 4
--
        DETAIL 5 IS COUNTRY IN TAB 2, SIZE IN TAB 1, ACREAGE IN TAB 5
--
        SELECT 3 ON ACREAGE ≥ 10
--
        SELECT 4 ON ACREAGE > 5 AND < 10
--
        SELECT 5 ON ACREAGE ≤ 5
--
        EVALUATE BEFORE ANY SELECT ACREAGE
--
        TITLE       IS $NUMBER OF ACRES OF LAND AVAILABLE PER INHABITANT $,
        +              $(INCLUDING DESERTIC LAND)$
        + AT LINE 3 IS $      SIZE$ IN TAB 1, $NAME OF THE COUNTRY$ IN TAB 2,
        +              $  ROOMY$ IN TAB 3, $AVERAGE$ IN TAB 4,
        +              $CROWDED$ IN TAB 5,
        + AT LINE 4 IS $      ----$ IN TAB 1, $---- -- --- -------$ IN TAB 2,
        +              $ (X≥10)$ IN TAB 3, $(5<X<10)$ IN TAB 4,
        +              $ (X≤5) $ IN TAB 5,
        + AT LINE 6 IS $  $
--
        PREPARE CYACRE FROM ONCNY
```

Figure 5-6. CYACRE Report



```
                                   **  END OF REPORT  CYDENS  **

                                   WORLD COUNTRIES BY INCREASING DENSITY OF POPULATION

                                   DENSITY RANGE FROM    500.00 TO    599.99
                                   -----------------------------------------
509.65  TRINIDAD    573.86  UNITED KINGDOM

                                   DENSITY RANGE FROM    600.00 TO    699.99
                                   -----------------------------------------
627.28  WEST GERMANY

                                   DENSITY RANGE FROM    700.00 TO    799.99
                                   -----------------------------------------
708.19  JAPAN       778.79  SOUTH KOREA             792.18  BELGIUM  792.20  BAHREIN
```

```
                                   DENSITY RANGE FROM    100.00 TO    199.99
                                   -----------------------------------------
101.28  UGANDA          113.17  LITHUANIA    165.45  CYPRUS      183.33  NEPAL
103.82  BYELORUSSIA     124.38  AZERBAIJAN   167.05  GREECE      186.78  DOMINICA
105.06  TURKEY          145.27  GEORGIA      168.25  ALBANIA     191.86  BULGARIA
105.42  TUNISIA         152.84  THAILAND     170.26  ARMENIA     195.33  UKRAINIA
105.89  MALAWI          153.44  INDONESIA    172.56  NIGERIA
107.10  IRELAND         160.50  SPAIN        178.05  CUBA
```

```
                              WORLD COUNTRIES BY INCREASING DENSITY OF POPULATION

                              DENSITY RANGE FROM      0.00 TO      99.99
                              -----------------------------------------
 0.04  GREENLAND         10.29  KAZAKHSTAN          30.45  KIRGHIZIA          57.09  UZBEKISTAN
 0.40  SPANISH SAHARA    10.63  ANGOLA              30.69  NICARAGUA          58.62  KUWAIT
 1.02  FRENCH GUYANA     11.33  BOLIVIA             35.14  FINLAND            60.37  IRAQ
 1.66  MONGOLIA          11.95  BRITISH HONDURAS    35.43  GUINEA             60.90  AFGHANISTAN
 1.73  NAMIBIA           13.26  PARAGUAY            36.40  IRAN               63.58  LESOTHO
 2.29  LIBYA             13.37  ALGERIA             36.62  TANZANIA           66.66  YEMEN
 2.51  BOTSWANA          13.99  SUDAN               37.16  URUGUAY            70.97  ESTONIA
 3.04  MAURITANIA        14.27  ZAMBIA              37.63  GUINEA-BISSAU      71.94  LEBANON
 4.11  AUSTRALIA         14.66  SOUTH YEMEN         39.85  TADZHIKISTAN       74.37  SYRIA
 4.70  ICELAND           17.31  ZAIRE               40.44  COLOMBIA           74.87  EGYPT
 4.95  NEW GUINEA        18.77  RUSSIA              40.54  SOUTH AFRICA       75.52  MOROCCO
 5.09  GABON             20.44  IVORY COAST         43.53  PANAMA             76.25  COSTA RICA
 5.45  CANADA            20.72  ARGENTINA           43.55  SWAZILAND          78.78  SIERRA LEONE
 5.51  CENTRAL AFRICA    21.93  LAOS                43.92  SENEGAL            79.19  TOGO
 5.76  CONGO             22.70  PERU                43.98  SWEDEN             79.85  MALAYSIA
 6.21  CHAD              23.22  MOZAMBIQUE          44.20  ECUADOR            81.82  GAMBIA
 6.36  NIGER             24.76  VENEZUELA           45.00  UPPER VOLTA        82.74  GHANA
 6.66  SURINAM           24.79  LIBERIA             46.87  KENYA              88.21  LATVIA
 6.89  OMAN              25.02  BRAZIL              48.56  ETHIOPIA           88.71  CAMBODIA
 7.01  GUYANA            25.63  MADAGASCAR          51.44  HONDURAS           92.61  BURMA
 7.57  MALI              28.33  RHODESIA            53.80  MEXICO             95.55  GUATEMALA
 8.23  SAUDI ARABIA      28.86  CAMEROON            54.15  DAHOMEY
 8.96  SOMALIA           29.55  NORWAY              54.67  JORDAN
 8.99  TURKMENISTAN      29.91  CHILE               56.11  UNITED STATES
```

```
--
          NOTE REPORT CYDENS ILLUSTRATES VERTICAL SECTIONS
          +    AND BREAK ON COMPUTED VALUES
--
          FORMAT CYDENS
--
          PAGE-SIZE IS 4 VERTICAL SECTIONS
--
          TITLE IS $WORLD COUNTRIES BY INCREASING DENSITY OF POPULATION$ CENTERED
--
          HEADING 1 AT LINE 3 BEYOND ON ALL PAGES IS $DENSITY RANGE FROM $      +
          LOWDENS $ TO $ HIGHDENS IN COLUMN 43, AT LINE 4 BEYOND ON ALL PAGES IS +
          $-----------------------------------------$ IN COLUMN 43
--
          BREAK 1 ON DENSITY GT HIGHDENS
--
          DETAIL IS DENSITY IN COLUMN 1, COUNTRY IN COLUMN 12
--
          EVALUATE BEFORE BREAK 1 TEMPDENS, IDENS
--
          MOVE BEFORE HEADING 1 IDENS TO LOWDENS AND LOWDENS + 99.99 TO HIGHDENS
--
          PREPARE CYDENS FROM ONDEN
--
          END
```

Figure 5-7. CYDENS Report

OPERATING SYSTEM INTERFACE 6

Query Update operates under the NOS 2 and NOS/BE operating systems.

The NOS 2 operating system supports all Query Update data base features in both batch and interactive modes.

Under NOS/BE, Query Update/IMF features are not supported; however, Query Update/CDCS and CRM features are supported in both batch and interactive modes.

The QU and REPORT control statements, which are applicable to all operating systems, are described in this section. Other system control statements are described in the respective operating system reference manual.

The INVOKE, CREATE, and VERSION directives require permanent file parameters. These parameters are the same as those required for the ATTACH control statement. Query Update reserved words cannot be used as parameter names. Refer to the appropriate system reference manual for the applicable permanent file parameters.

QU CONTROL STATEMENT

The QU control statement requests Query Update execution in either batch or interactive mode. The parameters in this control statement are the same for the NOS and NOS/BE operating systems. The format of the QU control statement is shown in figure 6-1.

When the RO parameter is specified in the QU control statement, read-only mode is chosen for the duration of the session. To change to another mode, the user must exit from Query Update with the END directive and enter a new QU control statement.

A number of directives and features are disabled or modified in read-only mode. The directives CREATE, REMOVE, STORE, VERIFY, and VETO are disabled. If one of these directives is specified, the user receives diagnostic 307, indicating the particular directive that cannot be used in read-only mode. Destination items in the MOVE and MODIFY directives are limited to temporary data items generated by the DEFINE directive. The user receives diagnostic 308 when improper use of the MOVE and MODIFY directives are diagnosed in read-only mode.

If a data base procedure that calls the CYBER Record Manager interface routine REPLC or DLTE is called for execution when the user has selected read-only mode, diagnostic 346 is issued. The CYBER Record Manager interface routine PUT can be called from a data base procedure only for processing a file that has sequential file organization in read-only mode. Diagnostic 346 is also issued if PUT is used to process a file with other than sequential organization.

Only commas are allowed as separators in the QU control statement; an equals sign links the I, O, T, PL, PW, or TL parameter to a value. The terminator is a period or a right parenthesis. Any other terminator character or separator or an incorrect parameter produces the message, ERROR IN CONTROL CARD, and execution is terminated. Input, output, and transaction files have 12-bit zero byte terminators.

QUMIP CONTROL STATEMENT

The QUMIP control statement is no longer needed, due to the selection of appropriate modules using the FAST DYNAMIC LOADER. The QUMIP control statement is accepted for compatibility with earlier versions and is treated as a QU control statement.

REPORT CONTROL STATEMENT

The REPORT control statement calls the REPORT utility program to produce reports according to specifications in a table file. A table file previously generated by one or more COMPILE directives is required as input to the REPORT utility program. Any table file generated by an earlier version must be regenerated through QU 3.2, if it is to be used with the REPORT utility program. The format of the REPORT control statement is shown in figure 6-2.

The R parameter specifies the report name associated with the format specifications to be used to generate the report. The report is written on a local file with the same name as the report name. Disposition of this file is determined by the user. The specified report name identifies a report format that has been stored on the table file by a COMPILE directive.

The table file that contains the report format is specified by the T parameter. The table file is read sequentially, end-around, from its current position in locating each correct report format. If the report format is not on the specified table file, an error condition is diagnosed, and the report is not generated. If the report format specifies other report formats (in PREFACE or SUMMARY directives), those formats must also be stored on the same table file.

QU [,AF] [I [=file-name]] [,O=file-name] [,T=file-name] [,PL=n] [,PW=n] [,TL=n] [,RO] [,U]

AF — Specifies that any file that has incurred a CRM error 052 on an open is to be accepted and that processing is to continue. This error indicates that the file has not been closed since it was last updated. The error can be the result of concurrent file use (in CRM data base access mode only) by two users or of a system failure during previous use.

Interactive Mode

The default is for Query Update to ask the user whether or not to accept the file, upon encountering a CRM error 052. Specification of the AF parameter overrides this choice; the file is accepted and a message is printed to identify the file.

Batch Mode

The default is for Query Update to close the file that caused the CRM error 052 and to continue processing. If the AF parameter is specified, the file is accepted, a message is printed, and processing continues.

I=file-name — Specifies the directive input file.

Interactive Mode

The default input file is ZZZZZIN, which is connected to the terminal. Specifying the I parameter causes execution to occur as in batch mode; directives are input from the file rather than from the terminal. Specification of the I parameter without a file-name assumes input is from a file named COMPILE.

Batch Mode

The default input file is INPUT. Specification of the I parameter without a file-name assumes input is from a file named COMPILE.

O=file-name — Specifies the output file.

Interactive Mode

The default output file is ZZZZZOU, which is connected to the terminal. Specifying a file-name causes execution to occur as in batch mode. Query Update responses are received by the output file; responses are displayed at the terminal only if the user connects an output file when the O parameter is specified.

Batch Mode

The default output file is OUTPUT.

T=file-name — Specifies the transaction file, which receives a copy of all directive input and system output other than generated reports.

Interactive Mode

The specified file receives a copy of terminal input and Query Update output. At completion of terminal use, the transaction file can be printed to obtain a tangible listing of directives entered at the terminal and system diagnostic messages.

Batch Mode

The transaction file is a duplicate of the output file.

Figure 6-1. QU Control Statement Format (Sheet 1 of 2)

PL=n Specifies the number of lines per page for output.

Interactive Mode

A positive decimal integer must be specified for n. The value must include the number of lines required for the Query Update inquiry for MORE output, which is one line for the NOS operating system or two lines for the NOS/BE operating system. The maximum value allowed is 255. However, a value specified for n that is greater than the value set for the page size of the user's interactive system is overridden by the system. The default page length on the NOS operating system is 16 lines. On the NOS/BE operating system, the default depends on the terminal being used.

Specifying PL=0 disables the paging facility, which is the MORE request, so that all lines of output are transmitted for display.

Batch Mode

The use of this parameter is not applicable. If the parameter is specified; it is ignored.

PW=n Specifies the number of characters per line for output. A positive decimal integer must be specified for n. Zero is not allowed. The value must include two characters per line to be used for carriage control supplied by Query Update. The maximum value allowed is 255. However, a value specified for n that is greater than the value set for the page size of the user's interactive system is overridden by the system.

Interactive Mode

The default page width on the NOS operating system is 72 characters per line; on the NOS/BE operating system, the default page width depends on the terminal being used.

Batch Mode

If the parameter is not specified, the job default page width is used.

TL=n Specifies in characters the maximum length of a transmission. A decimal integer must be specified for n, it cannot exceed six digits. The minimum transmission length allowed is 20 characters

Interactive Mode

A transmission length of 1030 characters is the default.

Batch Mode

A transmission length of 1030 characters is the default.

RO Specifies that update directives are not allowed for interactive or batch mode.

U Specifies that Query Update is not to abort processing upon encountering a fatal diagnostic in batch mode.

Interactive Mode

The use of this parameter is not applicable.

Batch Mode

The default is for Query Update to abort processing upon encountering a fatal diagnostic. The specification of this parameter inhibits Query Update from aborting the processing.

Figure 6-1. QU Control Statement Format (Sheet 2 of 2)

$$\text{REPORT,R=report-name,T=file-name}\left\{\begin{array}{l}\text{P}\\ \text{I=file-name}\\ \text{I=file-name,P}\\ \text{P,I=file-name}\end{array}\right\}\left[\text{,V[=file-name]}\right]$$

R=report-name — Specifies the name of the report to be prepared; this parameter must be specified.

T=file-name — Specifies the table file that contains the encoded report specifications; this parameter must be specified.

P — Requests a two-page sample report with dummy data values.

I=file-name — Requests a report to be prepared from the data in the specified input file.

I=file-name,P or
P,I=file-name — Requests a two-page sample report with data values from the specified input file.

V=file-name — Specifies the file that supplies names and values of temporary data items to be preset before report preparation begins; this parameter is optional. If file-name is omitted from the V parameter, INPUT is assumed by default.

Figure 6-2. REPORT Control Statement Format

The I parameter specifies the file name of the local file that contains the input data for the report to be generated. The input data must be prearranged in the order expected for the break conditions specified in the report format. The input file is automatically rewound by the REPORT utility program before and after the report is generated. The I parameter specified without the P parameter is the same as the PREPARE directive.

The P parameter can be specified alone or with the I parameter. If the P parameter is specified alone, a two-page sample report is prepared according to the report specifications with dummy data values. The report contains alternate fields of X's or Y's for alphabetic information and 8's or 9's for numeric information. When both the P and I parameters are specified, the two-page sample report contains actual data values obtained from the input file. The P parameter specified without the I parameter is the same as the PREVIEW directive without the FROM option; specifying both the P and I parameters is the same as the PREVIEW directive with the FROM option. The REPORT control statement must specify either the I or the P parameter.

The optional V parameter specifies the variable file, which supplies the names and values of temporary data items that are to be preset before the report is generated. These temporary data items were created by DEFINE directives and were included in the report format at the time it was stored in the table file by the COMPILE directive.

The variable file allows the user to supply temporary data item values that were not available or were not initialized at the time the report format was compiled. The format of the variable file is as follows:

```
report-name data-name-1 = literal-1

    [data-name-2 = literal-2]. . .
```

As many data names can be specified as there are temporary data items to be preset for the report. Data names and literals must conform to the rules for Query Update data names and literals. (Refer to section 3.) The following rules also apply when the variable file is input from the card reader (V or V=INPUT):

> Input is free-format and characters can appear in any columns from column 1 through column 72 (unless the card size is changed when Query Update is installed).
>
> A continuation character must be specified when the values for temporary data items are continued from one card to the next.

If the variable file is not input from the card reader (V=file-name), the data names and literals must be entered as a continuous string and must be contained in one record. Each new record begins with report name, which is followed by the variables to be preset for the specified report.

A temporary data item established by a DEFINE directive that includes the VALUE option does not have to be preset through the variable file, unless its initial value had been altered at the time the COMPILE directive was processed.

NOS OPERATING SYSTEM

The following step-by-step procedure for interactive mode shows the control statements to be used. Refer to the NOS reference manuals for the control statement formats.

1. Log in with the appropriate procedure.
2. If an area in the subschema is to be created with the CREATE directive, enter the NOS DEFINE control statement.
3. Enter the QU control statement.
4. Enter the Query Update directives for the session; the last directive is the END directive.
5. If the default catalog was used for the session, enter the NOS DEFINE and COPY control statements to retain the catalog as a permanent file.
6. Log out with the appropriate procedure.

In batch mode, the input card images are specified in the following order:

1. Job statements.
2. DEFINE control statement (if an area in the subschema is to be created with the CREATE directive).
3. QU control statement.
4. DEFINE control statement (if the default catalog is used, and it is to be retained).
5. COPY control statement (if the default catalog is used, and it is to be retained).
6. End-of-system logical record indicator (such as 7/8/9 card).
7. Query Update directives; the last directive is the END directive.
8. End-of-information indicator (such as 6/7/8/9 card).

Any output files generated during the Query Update session are local files. Disposition of these files must occur before control is returned to the operating system.

Permanent file parameters M, PN, PW, R, and UN are accepted. Refer to the NOS reference manual for a more detailed discussion of permissions, passwords, and permanent file parameters.

NOTE

M=RA cannot be used when defining a permanent file for Query Update operations. Query Update modifies, but does not append, files.

NOS/BE OPERATING SYSTEM

The following step-by-step procedure for interactive mode (INTERCOM) shows the NOS/BE control statements to be used. Refer to the NOS/BE reference manual for the control statement formats.

1. Log in with the appropriate INTERCOM procedure.
2. Enter the QU control statement.
3. Enter the Query Update directives for the session; the last directive is the END directive.
4. If an area in the subschema was created with the CREATE directive, enter the CATALOG control statement to retain the area as a permanent file.
5. If the default catalog was used for the session, enter the CATALOG control statement to retain the catalog as a permanent file.
6. Log out with the appropriate INTERCOM procedure.

In batch mode, the input deck contains the following cards in the order specified:

1. Job statement.
2. QU control statement.
3. CATALOG control statement (if an area in the subschema is created with the CREATE directive).
4. CATALOG control statement (if the default catalog is used, and it is to be retained).
5. End-of-system logical record indicator (such as 7/8/9 card).
6. Query Update directives; the last directive is the END directive.
7. End-of-information indicator (such as 6/7/8/9 card).

Any output files generated during the Query Update session are local files. Disposition of these files must occur before control is returned to the operating system.

The NOS/BE operating system has a permanent file manager that allows multiple (concurrent) access to permanent files (the area, the catalog attached by a VERSION directive, and the subschema). Query Update performs internal attaches of the premanent file in a way that allows concurrency; however, the user must set up the permanent files properly with the appropriate CATALOG control statement parameters. It is recommended that password definition be specified at CATALOG time, so that the default permissions of MODIFY, EXTEND, and READ are granted at ATTACH time. In addition, the FO parameter should also be specified at CATALOG time. This allows Query Update optimal multiple access for permanent files being used. Parameters, such as MR or RW, specified in the subschema or during Query Update execution override the default permissions.

Query Update ensures that the subschema has multiple read access, unless the user specifies permanent file parameters in the INVOKE statement (MR=0 or RW=0). These parameters directly inhibit multiple read access. Once the passwords for the Query Update catalog and area files have been specified on the CATALOG control statement, the user includes the applicable passwords in the VERSION and INVOKE directives. If the file is to be updated, passwords should be specified via the appropriate permanent file parameters in the VERSION or INVOKE directive. The appropriate permanent file parameters are those which result in the EXTEND, MODIFY, and READ permissions being granted. If the file is to be read only, passwords should be specified to facilitate the granting of READ permission only. Passwords are included in the VERSION or INVOKE directive by specifying PW=x (or PW=x y).

Permanent file parameters accepted are MR, RW, PW, ID, CY, SN, and LC. Refer to the NOS/BE reference manual for a more detailed discussion of permissions, passwords, and permanent file parameters.

STANDARD CHARACTER SETS

Control Data operating systems offer the following variations of a basic character set:

CDC 64-character set

CDC 63-character set

ASCII 64-character set

ASCII 63-character set

The set in use at a particular installation was specified when the operating system was installed.

Depending on another installation option, the system assumes an input deck has been punched either in 026 or in 029 mode (regardless of the character set in use). Under NOS/BE, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of the job statement or any 7/8/9 card. The specified mode remains in effect through the end of the job unless it is reset by specification of the alternate mode on a subsequent 7/8/9 card.

Under NOS, the alternate mode can be specified by a 26 or 29 punched in columns 79 and 80 of any 6/7/9 card, as described above for a 7/8/9 card. In addition, 026 mode can be specified by a card with 5/7/9 multipunched in column 1, and 029 mode can be specified by a card with 5/7/9 multipunched in column 1 and a 9 punched in column 2.

Graphic character representation appearing at a terminal or printer depends on the installation character set and the terminal type. Characters shown in the CDC Graphic column of the standard character set table (table A-1) are applicable to BCD terminals; ASCII graphic characters are applicable to ASCII-CRT and ASCII-TTY terminals.

Standard collating sequences for the two printer character sets are shown in tables A-2 and A-3.

TABLE A-1. STANDARD CHARACTER SETS

Display Code (octal)	CDC			ASCII		
	Graphic	Hollerith Punch (026)	External BCD Code	Graphic Subset	Punch (029)	Code (octal)
00[†]	: (colon)[††]	8-2	00	: (colon)[††]	8-2	072
01	A	12-1	61	A	12-1	101
02	B	12-2	62	B	12-2	102
03	C	12-3	63	C	12-3	103
04	D	12-4	64	D	12-4	104
05	E	12-5	65	E	12-5	105
06	F	12-6	66	F	12-6	106
07	G	12-7	67	G	12-7	107
10	H	12-8	70	H	12-8	110
11	I	12-9	71	I	12-9	111
12	J	11-1	41	J	11-1	112
13	K	11-2	42	K	11-2	113
14	L	11-3	43	L	11-3	114
15	M	11-4	44	M	11-4	115
16	N	11-5	45	N	11-5	116
17	O	11-6	46	O	11-6	117
20	P	11-7	47	P	11-7	120
21	Q	11-8	50	Q	11-8	121
22	R	11-9	51	R	11-9	122
23	S	0-2	22	S	0-2	123
24	T	0-3	23	T	0-3	124
25	U	0-4	24	U	0-4	125
26	V	0-5	25	V	0-5	126
27	W	0-6	26	W	0-6	127
30	X	0-7	27	X	0-7	130
31	Y	0-8	30	Y	0-8	131
32	Z	0-9	31	Z	0-9	132
33	0	0	12	0	0	060
34	1	1	01	1	1	061
35	2	2	02	2	2	062
36	3	3	03	3	3	063
37	4	4	04	4	4	064
40	5	5	05	5	5	065
41	6	6	06	6	6	066
42	7	7	07	7	7	067
43	8	8	10	8	8	070
44	9	9	11	9	9	071
45	+	12	60	+	12-8-6	053
46	-	11	40	-	11	055
47	*	11-8-4	54	*	11-8-4	052
50	/	0-1	21	/	0-1	057
51	(	0-8-4	34	(	12-8-5	050
52	)	12-8-4	74	)	11-8-5	051
53	$	11-8-3	53	$	11-8-3	044
54	=	8-3	13	=	8-6	075
55	blank	no punch	20	blank	no punch	040
56	, (comma)	0-8-3	33	, (comma)	0-8-3	054
57	. (period)	12-8-3	73	. (period)	12-8-3	056
60	≡	0-8-6	36	#	8-3	043
61	[	8-7	17	[	12-8-2	133
62	]	0-8-2	32	]	11-8-2	135
63	%[††]	8-6	16	%[††]	0-8-4	045
64	≠	8-4	14	" (quote)	8-7	042
65	↱	0-8-5	35	_ (underline)	0-8-5	137
66	∨	11-0	52	!	12-8-7	041
67	∧	0-8-7	37	&	12	046
70	↑	11-8-5	55	' (apostrophe)	8-5	047
71	↓	11-8-6	56	?	0-8-7	077
72	<	12-0	72	<	12-8-4	074
73	>	11-8-7	57	>	0-8-6	076
74	≤	8-5	15	@	8-4	100
75	≥	12-8-5	75	\	0-8-2	134
76	¬	12-8-6	76	¬ (circumflex)	11-8-7	136
77	; (semicolon)	12-8-7	77	; (semicolon)	11-8-6	073

[†] Twelve zero bits at the end of a 60-bit word in a zero byte record are an end-of-record mark rather than two colons.

[††] In installations using a 63-graphic set, display code 00 has no associated graphic or card code; display code 63 is the colon (8-2 punch). The % graphic and related card codes do not exist and translations yield a blank (55_8).

TABLE A-2. CDC CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal	CDC Graphic	Display Code	External BCD	Collating Sequence Decimal/Octal	CDC Graphic	Display Code	External BCD
00 00	blank	55	20	32 40	H	10	70
01 01	≤	74	15	33 41	I	11	71
02 02	%	63 †	16 †	34 42	∨	66	52
03 03	[	61	17	35 43	J	12	41
04 04	→	65	35	36 44	K	13	42
05 05	≡	60	36	37 45	L	14	43
06 06	∧	67	37	38 46	M	15	44
07 07	↑	70	55	39 47	N	16	45
08 10	↓	71	56	40 50	O	17	46
09 11	>	73	57	41 51	P	20	47
10 12	≥	75	75	42 52	Q	21	50
11 13	¬	76	76	43 53	R	22	51
12 14	.	57	73	44 54	]	62	32
13 15	)	52	74	45 55	S	23	22
14 16	;	77	77	46 56	T	24	23
15 17	+	45	60	47 57	U	25	24
16 20	$	53	53	48 60	V	26	25
17 21	*	47	54	49 61	W	27	26
18 22	−	46	40	50 62	X	30	27
19 23	/	50	21	51 63	Y	31	30
20 24	,	56	33	52 64	Z	32	31
21 25	(	51	34	53 65	:	00 †	none†
22 26	=	54	13	54 66	0	33	12
23 27	≠	64	14	55 67	1	34	01
24 30	<	72	72	56 70	2	35	02
25 31	A	01	61	57 71	3	36	03
26 32	B	02	62	58 72	4	37	04
27 33	C	03	63	59 73	5	40	05
28 34	D	04	64	60 74	6	41	06
29 35	E	05	65	61 75	7	42	07
30 36	F	06	66	62 76	8	43	10
31 37	G	07	67	63 77	9	44	11

†In installations using the 63-graphic set, the % graphic does not exist. The : graphic is display code 63, External BCD code 16.

This collating sequence is the default COBOL sequence in SORT directive.

TABLE A-3. ASCII CHARACTER SET COLLATING SEQUENCE

Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code	Collating Sequence Decimal/Octal		ASCII Graphic Subset	Display Code	ASCII Code
00	00	blank	55	20	32	40	@	74	40
01	01	!	66	21	33	41	A	01	41
02	02	"	64	22	34	42	B	02	42
03	03	#	60	23	35	43	C	03	43
04	04	$	53	24	36	44	D	04	44
05	05	%	63†	25	37	45	E	05	45
06	06	&	67	26	38	46	F	06	46
07	07	'	70	27	39	47	G	07	47
08	10	(	51	28	40	50	H	10	48
09	11	)	52	29	41	51	I	11	49
10	12	*	47	2A	42	52	J	12	4A
11	13	+	45	2B	43	53	K	13	4B
12	14	,	56	2C	44	54	L	14	4C
13	15	–	46	2D	45	55	M	15	4D
14	16	.	57	2E	46	56	N	16	4E
15	17	/	50	2F	47	57	O	17	4F
16	20	0	33	30	48	60	P	20	50
17	21	1	34	31	49	61	Q	21	51
18	22	2	35	32	50	62	R	22	52
19	23	3	36	33	51	63	S	23	53
20	24	4	37	34	52	64	T	24	54
21	25	5	40	35	53	65	U	25	55
22	26	6	41	36	54	66	V	26	56
23	27	7	42	37	55	67	W	27	57
24	30	8	43	38	56	70	X	30	58
25	31	9	44	39	57	71	Y	31	59
26	32	:	00†	3A	58	72	Z	32	5A
27	33	;	77	3B	59	73	[	61	5B
28	34	<	72	3C	60	74	\	75	5C
29	35	=	54	3D	61	75	]	62	5D
30	36	>	73	3E	62	76	^	76	5E
31	37	?	71	3F	63	77	_	65	5F

†In installations using a 63–graphic set, the % graphic does not exist. The : graphic is display code 63.

DIAGNOSTICS

Query Update analyzes the syntax of a directive after it is transmitted. Upon detection of errors (such as a missing delimiter, a literal that is too long, or a missing or misplaced keyword), the system responds with a diagnostic message and can reject the erroneous directive. When a transmission contains more than one directive, rejection of any directive causes the remainder of the transmission to be ignored. When any portion of the directive is incorrect, the ent!re directive must be reentered.

Each Query Update diagnostic message is preceded by a number in parentheses. The message explains the diagnosed error and, if appropriate, includes the erroneous portion of the directive. If the error occurs while performing a session, the error message includes the session and transmission identification.

When the user requires more information than that conveyed by the diagnostic message, a HELP directive that includes the diagnostic number produces an explanatory system response to suggest remedial actions.

In addition to syntax diagnosed errors, the system produces diagnostic messages during directive execution when such errors occur as record not found, file not available, input/output not possible.

Additionally, some facilities used by Query Update can produce their own diagnostic messages. Processors such as CYBER Record Manager, Loader, Sort/Merge, or the operating system generate diagnostics that are easily distinguished from a Query Update message. These messages can be brought to the attention of a system analyst for interpretation.

All diagnostic messages that can be issued during a Query Update session are listed in table B-1. Informative messages, which are not numbered, are listed at the end of the table. The Action column gives the applicable corrective action; reentering the entire directive applies to each action. Lowercase letters within diagnostic messages are replaced with information related to the specific directive that caused the diagnostic to be issued. The following letters are used in the diagnostic messages:

c or n	Current syntax word
a	Alphabetic name
o	Octal number
d	decimal number

TABLE B-1. DIAGNOSTIC MESSAGES

Error Code	Message	Significance	Action
1	c INVALID CHARACTER MASK	A valid mask can contain the letters Y and N enclosed in separator characters and preceded by the letter M. Up to 255 letters Y and N are allowed.	Change the character mask to the correct format.
2	c INVALID OCTAL LITERAL	A valid octal literal can contain the digits 0 through 7 enclosed in separator characters and preceded by the letter O. Up to 20 octal digits are allowed.	Change the octal literal to the correct format.
3	c DELIMITER MISSING FOLLOWING n	A delimiter, which is defined as either a right parenthesis or a terminating separator for a literal or a character mask, has not been found.	Reenter the literal or character mask, including a delimiter.
4	MATCHING DELIMITER MISSING n	A terminating delimiter has not been found for an alphanumeric literal.	Reenter the literal, including a terminating delimiter.
5	n FIRST CHARACTER INVALID	The first character for a source word is a nonalphanumeric character that is neither a delimiter for a literal nor a sign for a numeric literal.	Reenter the source word using either a delimiter for a literal, a sign for a numeric literal, or an alphanumeric character as the first character.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
6	INVALID CHARACTER IN n	An alphabetic character has been found in an integer or numeric literal or a nonalphanumeric character has been found in a name.	Reenter the integer or name using allowed characters.
7	INVALID CHARACTER IN NAME OR KEYWORD n	A nonalphanumeric character has been found in a data name or in a keyword.	Reenter the data name or the keyword using alphanumeric characters only.
8	n IS INVALID NAME	A nonalphanumeric character has been found in a data name or in a keyword.	Reenter the data name or the keyword using alphanumeric characters only.
9	n MISPLACED DECIMAL POINT	Either a decimal point was not expected, or it was in the wrong place according to the picture specification.	Reenter the data.
10	n HAS DUPLICATE DECIMAL POINT	The numeric literal contains more than one decimal point.	Reenter the numeric literal with only one decimal point.
11	n IS INVALID FIXED POINT LITERAL	A valid fixed point literal can contain only numeric characters, one sign (+ or -) as the leftmost character, and one decimal point in any position except as the righmost character.	Reenter the fixed point literal according to required format.
12	n IS INVALID FLOATING POINT LITERAL	A valid floating point literal consists of a fixed point literal, the character E, and an integer literal.	Reenter the floating point literal according to required format.
13	n NEEDS ALPHA CHARACTER	A data name must include at least one alphabetic character.	Reenter the data name, including at least one alphabetic character.
14	n COMPLEX NUMBER MISSING IMAGINARY PORTION	A complex number consists of both real and imaginary portions; the imaginary portion is the numeric literal following the character I.	Reenter the complex number, including both real and imaginary portions.
15	n INVALID COMPLEX NUMBER	A valid complex number consists of two numeric literals separated by the character I.	Reenter the complex number according to the required format.
16	n INCOMPLETE COMPLEX NUMBER	A valid complex number consists of two numeric literals separated by the character I.	Reenter the complex number according to the required format.
17	n IS INCOMPLETE FLOATING POINT LITERAL	A valid floating point literal consists of a fixed point literal, the character E, and an integer literal.	Reenter the floating point literal according to the required format.
18	ILLEGAL DATA TYPE	The following formats describe and define a data item: INTEGER, FLOATING, DOUBLE, COMPLEX, LOGICAL, and CHARACTER. The DESCRIBE directive also uses FIXED and NUMERIC data items.	Reenter the data item using an accepted format.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
19	SYNTAX TABLE OVERFLOW	This is a system error in Query Update and should never be issued.	Notify the system analyst for corrective action.
20	-c- IS NOT A VALID PERMANENT FILE NAME	A valid permanent file name consists of 1 to 40 alphabetic or alphanumeric characters.	Reenter a valid permanent file name.
21	c -MOVE DESTINATION- MAY NOT BE LITERAL OR DESCRIBED ITEM	The data item to which a literal or expression is moved must be a temporary data item (established by a DEFINE directive) or a permanent data item in the record of the area being updated.	Reenter the MOVE directive with the correct data item.
22	c n INVALID SYNTAX	Keywords are out of order or an expected keyword cannot be found.	Reenter the transmission.
23	c TAG-NUMBER OUT OF RANGE OR SELECT MISSING	The tag number for a DETAIL directive is too large, or no SELECT directive has the corresponding tag number.	Reenter the transmission.
24	OCTAL LITERAL TOO BIG	The maximum size allowed for an octal literal is 20 digits.	Reenter the octal literal, not exceeding 20 digits.
25	CHARACTER MASK TOO BIG	The maximum size allowed for a character mask is 255 digits.	Reenter the character mask, not exceeding 255 digits.
26	DECIMAL LITERAL TOO BIG	The maximum size of a decimal literal is 14 digits.	Reenter the decimal literal, not exceeding 14 digits.
27	PERFORM IS ONLY DIRECTIVE EXECUTED FOR THIS TRANSMISSION	The PERFORM directive is followed by another directive in the same transmission; directives following the PERFORM directive are not executed.	Reenter the directive in a separate transmission.
28	DIRECTIVE KEYWORD MISSING AFTER c	A keyword expected in sequence for a directive cannot be found.	Reenter the directive, including the necessary keyword.
29	ERRONEOUS DIRECTIVE AND REST OF TRANSMISSION IGNORED	A syntax error has caused syntax scanning of the transmission to be terminated; the rest of the transmission is not executed.	Reenter the directive.
30	INVALID CHARACTER IN SOURCE-WORD	A nonalphanumeric character appeared in a source word.	Reenter the source word using only alphanumeric characters.
31	INVALID USE OF SIGN	A sign character (+ or -) can occur only as the leftmost character in a fixed point, floating point, or integer literal.	Reenter the entry, using a sign character only as the leftmost character in a fixed point, floating, point or integer literal.
32	ERROR IN IMAGINARY PORTION OF COMPLEX NUMBER	The imaginary portion of a complex number must be a floating point literal.	Reenter the complex number using the correct format for the imaginary literal.
33	MAXIMUM SOURCE-WORD LENGTH EXCEEDED	The maximum source word length allowed is 255 characters.	Reenter the entry.
35	VALID DELIMITER NOT FOUND	A terminating delimiter for a character literal cannot be found.	Reenter the character literal, using a terminating delimiter.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
36	UNRECOGNIZABLE STATEMENT	The initial word of the directive is not a keyword for a Query Update directive.	Reenter the directive with a keyword as the initial word of the directive.
37	n DOUBLE PRECISION INVALID IN COMPLEX NUMBER	A complex number cannot consist of double precision components.	Reenter the complex number, eliminating the double precision components.
38	c TRANSM-ID ALREADY EXISTS	While recording or duplicating a session, an attempt was made to overwrite an existing transmission id.	Reenter the DUPLICATE directive; no action for RECORDING.
39	-a- IS AN UNKNOWN SESSION NAME AND/OR DIRECTIVE	Either the transmission id or the session name is unknown in the current catalog.	Reenter the appropriate transmission id or session name.
40	c INVALID SUBSCRIPT	A subscript must be an integer literal, a data name containing an integer, or one of the figurative subscripts (ALL, ANY, LAST, or NEXT).	Reenter the entry using the appropriate subscript.
41	INVALID USE OF LAST/NEXT	The figurative subscripts LAST and NEXT can be used only with occurring items that have been defined with a DEPENDING ON clause.	Reenter the entry.
42	ERROR IN SUBSCRIPT DIMENSIONING	The subscript format specified by the user is not compatible with the OCCURS clause recorded in the subschema.	Reenter the subscript format.
43	SUBSCRIPT BEYOND RANGE	An attempt was made to reference a matrix at a point beyond the maximum value of the subscript.	Reenter the transaction using the correct subscript.
44	c UNDEFINED DATA-NAME	The designated data name is not defined either in the temporary working storage of Query Update or as part of the area file definition.	Define the data name either in temporary working storage or as part of the area file definition.
45	c DATA-NAME BEING USED	An attempt was made to define a data name that has already been defined.	Check to make sure correct data name is being used, and reenter the transaction.
46	c PRECEDING FIELD WAS VARIABLE LENGTH	The variable length field has to be the last item of a record.	Reorganize the order of the fields, or omit this entry.
47	c SHOULD PRECEDE -AS-	The designated data name must be followed by AS in the SPECIFY directive.	Reenter the SPECIFY directive, including AS following the data name.
48	SPECIFY MISSING CONDITIONAL EXPRESSION	A valid conditional expression must follow AS in the SPECIFY directive.	Reenter the SPECIFY directive.
49	-c- SESSION-ID OR LEVEL NUMBER TOO LONG	A session name cannot exceed six characters; a report level number cannot exceed two digits.	Reenter the session name or the report level number.
50	SESSION REACHED DIRECTIVE LIMIT	The directive id limit of 999 has been reached.	Create another session to continue, or use smaller increments.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
51	n ILLEGAL NUMERIC ITEM FOLLOWING c	An expected numeric literal contains a nonnumeric character.	Reenter the numeric literal.
52	ERROR IN DIRECTORY DESCRIPTION OF ITEM	The directory description of an item is incorrect; this description is generated by the DDL definition of the area file.	Consult the data administrator.
53	CONVERSION FUNCTION NOT PROVIDED	The following conversions are not allowed: between logical and numeric items, between character and numeric items, and between logical and character items.	Reenter the transmission.
54	DATA TRUNCATED DURING CONVERSION	Data was truncated when a value was moved to a location field too small to hold the entire value.	Move the data to a larger location field or take no action.
55	DATA INCOMPATIBLE WITH DATA TYPE	During the conversion of one data type to another, Query Update encountered characters in the source field that are illegal in the destination field.	Reenter the data.
56	c UNSUPPORTED DATA TYPE	This is a system error that should not occur in normal usage.	Follow site-defined procedures for reporting software problems.
58	a - LOGICAL OPERATOR REQUIRES LOGICAL TYPE OPERAND	One of the logical operators (AND, OR, or XOR) is neither followed nor preceded by an expression that yields a logical value of true or false.	Reenter with a logical type operand.
59	LOGICAL USAGE IS INVALID ARITHMETIC OPERAND	An arithmetic expression cannot contain an item that is data type LOGICAL.	Reenter the arithmetic expression without a logical item.
60	TWO SUCCESSIVE OPERATORS USED	Two arithmetic or logical operators occurred without an intervening operand.	Reenter the entry including an intervening operand.
61	c ILLEGAL USE OF OPERATOR	An arithmetic or logical operator cannot be used outside an arithmetic, conditional, or logical expression.	Reenter the entry, eliminating the arithmetic or logical operator.
62	LEFT/RIGHT PARENTHESIS NOT BALANCED	The number of left parentheses does not equal the number of right parentheses.	Reenter the entry with the number of left parentheses equaling the number of right parentheses.
63	c MUST PRECEDE OPERAND/ OPERATOR	The designated item must be followed by an operand or an operator.	Reenter the entry in the correct format.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
64	c ILLEGAL DATA-NAME	A valid data name cannot exceed 30 alphanumeric characters and embedded hyphens.	Input a data name in the correct format.
65	OPERATOR MUST SEPARATE OPERANDS	Two operands are adjacent to each other; an operator is required between the operands.	Reenter the entry with an operator between the two operands.
66	TOO MANY RIGHT PARENTHESES	The number of right parentheses exceeds the number required.	Reenter the entry with the number of right parentheses equaling the number of left parentheses.
67	c FUNCTION REQUIRES PARENTHESIZED PARAMETER LIST	The designated function must be specified with parameters enclosed in parentheses.	Reenter the entry with the parameters enclosed in parentheses.
68	c IS OR HAS INVALID FUNCTION PARAMETER	The designated function contains an invalid parameter; valid parameters must be legal names or literals.	Input a legal name or literal.
69	TRANSMISSION NOT RECORDED - IT HAS UNRECORDABLE DIRECTIVE	The transmission contains a directive that cannot be recorded. Refer to RECORDING directive.	Reenter the entire entry, deleting the directive that cannot be recorded.
70	PARENTHETICAL NESTING EXCEEDS LIMIT NEAR c n	The limit for nested parentheses is 25.	Reorganize and reenter the entry, not exceeding the limit for nested parentheses.
71	RELATIONAL OPERATOR MUST FOLLOW c	The designated source word must be followed by a relational operator.	Reenter the entry, following the source word with a relational operator.
72	CHARACTER USAGE IS INVALID ARITHMETIC OPERAND	A character literal or a data item defined as character usage has been included in an arithmetic expression.	Reenter the arithmetic expression in the correct format.
73	c DIRECTIVE-ID MUST BE NUMERIC, 3 DIGITS LONG	A valid transmission id must be in the range 1 through 999.	Input a valid transmission id.
74	ILLEGAL FILE-NAME AFTER -FROM-	The reserved word FROM must be followed by a file name that is 1 to 7 alphanumeric characters; the first character must be alphabetic.	Input a valid file name.
75	DIRECTIVE RANGE IS NEGATIVE	The PERFORM directive specifies a range of directives where transmission-id-1 is greater than transmission-id-2.	Reenter the transmission.
76	PERFORM TABLE OVERFLOW	This error occurs when the level of nested PERFORMS exceeds the limit allowed by Query Update. The PERFORM was terminated.	Redefine the session structure.
77	c INVALID QU DIRECTIVE	The first word of the directive does not identify a valid Query Update directive.	Reenter the directive with a keyword as the initial word of the directive.
78	TL=d. THIS TRANSMISSION WAS d. EXCESS IGNORED	The input record (including continuation lines) has exceeded the maximum transmission length.	Reformat and reenter the entry or utilize the TL control card option.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
79	ONCE ILLEGAL WITH ABSOLUTE LINE NUMBER	The reserved word ONCE is not legal in a DETAIL directive that specifies AT LINE x, where x is an absolute number.	Reenter the DETAIL directive, deleting the reserved word ONCE.
80	c INVALID PICTURE	The PICTURE clause does not conform to the rules governing the picture specification.	Enter the correct picture clause; refer to appendix F.
81	MURAL SIZE EXCEEDS MAX	Every edited item has a mural describing the picture. The maximum size of a mural is 31 words. Depending on the number of insertion characters (the maximum lengths of an edited item is approximately 1600 characters).	Define two or more items whose lengths are small enough so that their murals do not exceed the maximum.
82	c INSERTION CHARACTERS EXCEED 63	The picture specification cannot use more than 63 insertion characters.	Reformat and reenter the picture specification.
83	c - ZERO REPEAT COUNT TREATED AS ONE	In a picture specification, a repetition of 0 is treated as 1.	No action.
84	c - REPEAT COUNT MUST BE NUMERIC	In a picture specification with a REPEAT clause, the literal specified as the repetition count must be numeric.	Change the repetition count to be a numeric literal.
85	n INVALID PICTURE - MUST BE CHARACTER LITERAL	The picture specification must specify a character literal.	Change the picture specification to a character literal.
86	NO READ PERMISSION	An attempt has been made to read the catalog when the password on a VERSION directive did not give read permission.	Add the read password to the VERSION directive.
87	PRIVACY VIOLATION. WRITING ON CATALOG NOT ALLOWED	The VERSION directive specified to attach the current catalog did not give the correct passwords to modify and extend the catalog.	Enter the VERSION directive with the correct passwords.
88	FORMAT/ALTER DIRECTIVE MISSING	A FORMAT or ALTER directive must be specified before a report-oriented directive (ERASE, DETAIL, TITLE, etc.) can be specified. The FORMAT or ALTER directive establishes the name of the current report.	Enter a FORMAT or ALTER directive.
89	n - SESSION-ID OR KEYWORD OFF MISSING	In the RECORDING directive, the keyword RECORDING must be immediately followed by either a session id or the reserved word OFF.	Reenter the RECORDING directive using the correct format.
90	n...MISSING -AND- IN DESCRIBE	The DESCRIBE directive must contain a file name followed by the reserved word AND.	Reenter the DESCRIBE directive using the correct format.
91	n...MISSING -AS- IN DESCRIBE	Immediately after a data name in the DESCRIBE directive, the reserved word AS must be specified.	Reenter the DESCRIBE directive using the correct format.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
92	n...MISSING -BY- IN DESCRIBE	In the DESCRIBE directive, the usage of the data item must be followed by the reserved word BY.	Reenter the DESCRIBE directive using the correct format.
93	n ILLEGAL DATA TYPE	An illegal data type has been specified in a DEFINE directive.	Reenter the DEFINE clause with a legal data type.
94	c OCCURRING POINTER NAME NOT PREVIOUSLY DESCRIBED	In a DEFINE directive, the data name specified in the DEPENDING ON clause has not been previously defined.	Define the data name, and reenter the DEFINE directive.
95	n PICTURE SPECIFICATION ERROR	The picture specification is not valid.	Enter the correct picture clause; refer to appendix F.
96	-ALL- NEEDED TO ERASE ENTIRE REPORT OR SESSION	An entire report or session is erased by specifying the ERASE directive with either the report name or the session id followed by the reserved word ALL.	Reenter the ERASE directive including the reserved word ALL.
97	NON-NUMERIC EDIT INPUT FOR a	Edited data items can contain only numeric characters.	Reenter the transmission with only numeric characters.
98	DEFAULT RECAP LINE - NUMBER USED	A report specification used the AT RECAP LINE clause but no RECAP directive was specified in the report format. This is not an error; the default recap line is one line beyond the last line printed on the page.	No action.
99	DEFAULT TITLE LINE-NUMBER USED	A report specification used the AT TITLE LINE clause but no TITLE directive was specified in the report format. This is not an error; the default title line is line 1.	No action.
100	A SINGLE ITEM IN A MATRIX CANNOT BE ERASED	A single matrix item (or occurring item) cannot be specified in an ERASE directive; the erasure is not performed.	Reenter the ERASE directive, specifying the subscript ALL to erase the entire matrix.
101	c ILLEGAL FILE NAME	The file name must be 1 to 7 alphanumeric characters, and the first character must be alphabetic.	Reenter the transmission with a legal file name.
102	c DUPLICATE-REPORT NAME	The designated report name already exists in the current catalog.	Reenter the transmission with a unique report name.
103	c UNKNOWN REPORT NAME	The designated report name does not exist on the current catalog.	Reenter the transmission with a valid report name.
104	c IS NOT AN ERASABLE ITEM	A literal cannot be erased by the ERASE directive. Defined and specified items are the only temporary items that can be erased.	If a described item is to be erased, an entire new DESCRIBE directive must be given, referring to the appropriate item as filler. To erase a defined array, include the figurative subscript ALL in parentheses following the data name.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
105	CATALOG FILE ERROR o ON a	The CYBER Record Manager error number listed has occurred on the specified file.	Refer to the CYBER Record Manager reference manual for an explanation of the diagnostic.
106	LINE NUMBER a NOT IN RANGE 1 TO d	The number of lines specified for a report page exceeds the maximum number of 60 lines per page.	Reenter the directive with a proper line number.
107	COLUMN NUMBER a NOT IN RANGE 1 TO d	The number of columns specified for a report page exceeds the maximum number of columns per page, 136.	Reenter the directive with a proper column number.
108	ILLEGAL LEVEL NUMBER a OR A -BREAK- IS MISSING	Report level numbers cannot be negative or exceed 63; the level number specified is not valid. Or, a BREAK directive using the indicated level number requires a heading or footing with the same level number.	Reenter the transmission.
109	c ILLEGAL DATA-NAME	A valid data name contains 1 to 30 alphanumeric characters; the first character must be alphabetic.	Reenter the transmission with a valid data name.
110	-FROM- MISSING AFTER c	A PREPARE, PREVIEW, or SUMMARY directive requires the reserved word FROM followed by a file name.	Reenter the directive with the correct format.
111	LEVEL NUMBER MISSING AFTER c	A HEADING, FOOTING, or BREAK directive requires a level number after the directive keyword.	Reenter the directive with the correct format.
112	n INVALID INTEGER AFTER -TO-	In a DUPLICATE, ERASE, or EXHIBIT directive, the integer following the reserved word TO can contain up to three digits.	Reenter the directive with the correct format.
113	INVALID ERASE DIRECTIVE	A syntax error has been found in the ERASE directive.	Reenter the ERASE directive.
114	SESSION-ID MISSING AFTER c	In an ERASE SESSION directive, a session id of 1 to 6 alphanumeric characters must be specified; the first character must be alphabetic.	Reenter the ERASE SESSION directive with a valid session id.
115	n INVALID TAB NUMBER	The tab number specified indicates a column that is beyond the maximum column range for the report.	Reenter the directive with a valid tab number.
116	TAB c NOT SPECIFIED	The tab number specified (in a report specification) has not been defined in a TAB directive.	Reenter the TAB directive defining the needed tab number.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
117	DUPLICATE SIGN IN COMPLEX NUMBER	A complex number consists of a real part and an imaginary part; each part is a floating point literal. The floating point literals can contain only two signs: (1) the leftmost character in a field and (2) immediately following the character E for the sign of the exponent.	Reenter the directive with the correct format for the complex number.
118	-n- IS NOT A VALID ERASE OPTION OR DATA-NAME	The ERASE directive can specify only described, report, session, or temporary data names to be erased.	No action. Not all items are erasable; consult the data administrator.
119	UNKNOWN SESSION IDENTIFIER	The session name cannot be found in the current catalog.	Check to see if the intended session name was input.
120	c UNDEFINED USER CATALOG	The requested user catalog has not been defined by a VERSION directive.	Enter the VERSION directive defining the catalog, and then reenter the transaction.
121	DUPLICATE ENTRY IN SESSION	Each transmission id within a session must be unique.	Input a unique transmission id.
122	REPORT SPECIFICATION LIST FULL	The Query Update report table is full. A layout directive for each report can contain, at most, 200 combined data names, literals, and expressions.	A report specification that exceeds the table size can be specified only if the system analyst installs Query Update with a larger table.
123	-SECTIONS- MISSING AFTER -HORIZ- OR -VERT-	In the PAGE-SIZE directive, the reserved word HORIZONTAL or VERTICAL must be followed by the reserved word SECTIONS.	Reenter the PAGE-SIZE directive with the correct format.
124	n UNRECOGNIZABLE PAGE-SIZE KEYWORD	In the PAGE-SIZE directive, only the following words can be specified: LINES, COLUMNS, HORIZONTAL, VERTICAL, SECTIONS, IMAGES, and PARALLEL.	Reenter the PAGE-SIZE directive using one of the valid reserved words.
125	ERROR IN PAGE-NUMBER DIRECTIVE	A syntax error has occurred in a PAGE-NUMBER directive.	Reenter the PAGE-NUMBER directive.
126	ERROR IN DATE DIRECTIVE	A syntax error has occurred in the DATE directive.	Reenter the DATE directive.
127	ERROR IN TIME DIRECTIVE	A syntax error has occurred in the TIME directive.	Reenter the TIME directive.
128	NUMBER OF IMAGES NOT WITHIN ALLOWABLE RANGE	The number of images specified in the PAGE-SIZE directive is greater than the maximum number of four.	Reenter the PAGE-SIZE directive without exceeding the maximum number of images allowed at your installation (sections for images).
129	NUMBER OF SECTIONS NOT WITHIN ALLOWABLE RANGE	The number of sections specified in the PAGE-SIZE directive is greater than the maximum number of 10.	Reenter the PAGE-SIZE directive without exceeding the maximum number of images allowed at your installation (sections for images).

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
130	c UNDEFINED SYNTAX TABLE OR PROCEDURE	A reserved word has been interpreted as a directive keyword within a Query Update directive. This error normally occurs after previous syntax errors.	Reenter those directives having syntax errors.
131	c TOO MANY OPTIONS IN DIRECTIVE, TABLE OVERFLOW	A Query Update internal table has overflowed because of too many options in an EXHIBIT or ERASE directive.	Input an additional EXHIBIT or ERASE directive to obtain the desired results.
132	DUPLICATE REPORT ENTRY	The report format being constructed already has an entry for this option.	No action.
133	UNKNOWN REPORT ENTRY IN a	The report format being referenced is not defined in the current catalog.	Check for correct name of report and reenter the transmission.
134	n SPECIFIED COLLATING SEQ MUST BE DATA-NAME/LITERAL	The SORT directive must specify a data name or literal after the reserved word USING to define the collating sequence.	Reenter the SORT directive with the format described.
135	FIRST KEY SPECIFICATION MUST FOLLOW -ON-	In the SORT directive, the specification of the first key for the sort operation must be made after the reserved word ON.	Reenter the SORT directive with the correct format.
136	n KEY SPECIFICATION NOT VALID DATA-NAME	One of the key specifications following the reserved word ON in the SORT directive is not a valid data name.	Reenter the SORT directive with a valid data name.
137	KEYWORD UPON MUST PRECEDE OUTPUT FILE SPECIFICATION	In the SORT directive, the name of the output file for the sort operation must follow the reserved word UPON.	Reenter the SORT directive with the correct format.
138	c SPECIFICATION FOR COLLATING SEQ NOT DEFINED	The word used to specify the collating sequence for the SORT directive is not one of the words defined by the installation as valid collating sequence table names.	Reenter the SORT directive with a valid collating sequence table name.
139	INSERT, DELETE, AND UPDATE REQUIRE MODIFY PERMISSION	The INSERT, DELETE, and UPDATE directives require modify permission on the area file. This permission is granted according to the passwords specified for the area file in the USE directive.	Input the USE directive specifying modify permission for the area file.
140	c INVALID KEY	The key specified is not valid.	Reenter the entry with a valid key.
141	DATA-NAME OR -SAME- MUST FOLLOW THE KEYWORD -USING-	In the INSERT, UPDATE, or DELETE directive, either a data name or the reserved word SAME must follow the reserved word USING.	Reenter the directive using the correct format.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
142	PRIMARY KEY OR ONLY ONE ALTERNATE KEY SHOULD BE SPECIFIED	In the INSERT USING directive, the primary key must be specified. In the UPDATE USING or DELETE USING directive, only one alternate key can be specified if the primary key is not specified.	If the error occurred on INSERT USING, reenter the directive with the primary key specified. If the error occurred on the UPDATE USING or DELETE USING, reenter the directive with only one alternate key specified (or with the primary key specified along with the alternate keys).
143	c LINE-NUMBER IS INTEGER OR INTEGER ITEM ONLY	The line number specified for a report layout directive must be an integer or a data name that is type integer.	Reenter the transmission using TYPE integer for the line number.
144	c ONLY DEFINED ITEMS MAY BE EVALUATED	In the EVALUATE directive, the data names specified must refer to items defined by a DEFINE directive.	Input the data names with a DEFINE directive, and reenter the EVALUATE directive.
145	-c- ILLEGAL PASSWORD	One of the passwords specified for a permanent file is illegal.	Reenter the entry with legal passwords.
146	n UNKNOWN P.F. PARAMETER	One of the parameters specified for a permanent file is not legal.	Refer to the appropriate operating system reference manual for valid parameters.
147	ONCE ILLEGAL WITH MULTIPLE SECTIONS	The reserved word ONCE cannot be specified for a detail line when a PAGE-SIZE directive has specified more than one section.	Reissue the transmission.
148	ONCE WILL BE IGNORED	The reserved word ONCE has been illegally specified in a DETAIL directive; it is ignored.	No action.
149	VALID EXPRESSION MISSING	A valid arithmetic or conditional expression has not been specified.	Reenter the entry with a valid expression.
150	c IS INVALID SEPARATOR	The character specified as the delimiter in the SEPARATOR directive is not a valid delimiter.	Reenter the SEPARATOR directive with a valid delimiter.
151	ONE CHARACTER DELIMITER ONLY	Only one character can be used as a delimiter.	Reenter the entry.
152	WARNING - LAST KEY NOT SORTED. 25 LIMIT EXCEEDED	In the SORT directive, a maximum of 25 keys can be specified for the sort operation. The last key specified was not sorted because it exceeded the limit.	No action.
153	c GREATER THAN PAGE-SIZE WHICH IS d LINES	The line indicated exceeds the number of lines specified for the report page in the PAGE-SIZE directive.	Change the number of lines for the report page by reentering the PAGE-SIZE directive.
155	-AT LINE DATA-NAME- FOR FOOTINGS AND HEADINGS ONLY	Use of a data name to indicate a line number is valid only for HEADING and FOOTING directives. All other directives must use an integer number.	Reenter the directive using an integer number to indicate a line number.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
156	TAG NUMBER REQUIRED WHEN SELECT IS SPECIFIED	When a report format contains SELECT directives, all DETAIL directives must have a tag number.	Reenter DETAIL directives with tag numbers.
157	SELECT TAG MISSING	An integer tag number must be specified in the SELECT directive.	Reenter the SELECT directive.
158	TOO MANY SELECTS - TABLE OVERFLOW	The number of SELECT directives specified for the report exceeds the maximum number of five.	Check the installation options.
159	TOO MANY BREAKS - TABLE OVERFLOW	The number of BREAK directives specified for the report exceeds the maximum number of five.	Check the installation options.
160	-LINE I BEYOND- IS INVALID FOR TITLE	The use of the reserved word BEYOND is illegal in the TITLE directive.	Reenter the TITLE directive.
161	-ON ALL PAGES- CANNOT BE USED WITH PART OF THE HEADING	If the ON ALL PAGES clause is specified in a HEADING directive, it must be specified for every line number in the HEADING directive.	Reenter the HEADING directive.
162	DETAIL AT ABS. LINE-NUMBER INVALID WITH VERTICAL SECTIONS	The DETAIL directive cannot specify an absolute line number when the PAGE-SIZE directive for the report has specified vertical sections.	Reenter the DETAIL directive without an absolute line number.
163	c INVALID AFTER -AT LINE DATA-NAME-	The reserved word BEYOND cannot be specified after the AT LINE data-name clause.	Reenter the entry.
164	-NO- FOR SELECT ONLY	In EVALUATE and MOVE directives, the reserved word NO can refer only to the SELECT directive.	Reenter the transmission.
165	NUMBER OF FUNCTION PARAMETERS IN ERROR	The number of parameters specified for the function is not correct.	Correct the number of parameters and reenter.
166	CONVERSION ERROR, c IGNORED	The specified word is ignored due to a conversion error.	Consult the data administrator.
167	THIS REPORT CONTAINS NO EXECUTABLE STATEMENT	The report being prepared has no statement that leads to generation of a line.	Recreate the report.
168	BREAK DIRECTIVE MISSING -ON-	The reserved word ON, followed either by the word ITEM or OVERFLOW or by a condition, must follow the level number for a BREAK directive.	Reissue the transmission.
169	CHARACTER TO NUMERIC (c) CONVERSION ILLEGAL	Conversion from character to any numeric data type is illegal.	Do not attempt the conversion.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
170	DATA TRUNCATED DURING CONVERSION FOR - - - c	During the required conversion for the specified value, data was truncated because the source field was shorter than the original field.	No action; or consult the data administrator.
171	-LINE 0- ILLEGAL LINE-NUMBER	Since the first line in a report page is numbered line 1, specification of line 0 for report directives is illegal.	Reenter a valid line number.
172	-VERSION- CANNOT BE RECORDED	Changing catalogs during the recording of a report is not allowed.	No action.
173	ILLEGAL CONDITION AFTER UNTIL, PERFORM IGNORED	In a PERFORM directive that contains the UNTIL condition clause, the condition following UNTIL is not a valid conditional expression. The entire PERFORM directive is ignored.	Reenter the PERFORM directive eliminating the conditional expression following UNTIL.
174	REPEAT MUST BE FOLLOWED BY INTEGER-VALUE ARITH EXPRESSION	When the word REPEAT is specified in the PERFORM directive; the expression, data name, or integer number following REPEAT must yield a valid integer value.	Reenter the PERFORM directive with a valid integer value.
175	ON/OFF MUST FOLLOW VETO	If the reserved word VETO is specified in the PERFORM directive, it must be followed by the reserved word ON or OFF.	Reenter the PERFORM directive with a valid combination of reserved words.
176	n ARRAY PRESET ERROR, PRESET UNPREDICTABLE	A matrix (array) established by the DEFINE directive can only be preset to a single expression for all entries.	Reenter the transmission.
177	SUBSCRIPT OUT OF BOUNDS	A subscript specified for a matrix (or an occurring item) is greater than the limits of that matrix.	Reenter the entry using a subscript within the limits of that matrix.
178	SUBSCRIPT ERROR	A subscript error has been encountered.	Reenter the entry with a correct subscript.
179	UPPER BOUND OF ARRAY DEFINITION MISSING	A matrix (array) specified either with a DEFINE directive or through a DDL subschema description for a file is lacking definition of an upper boundary.	Include the definition of an upper boundary in either the DDL subschema or in the DEFINE directive.
180	FIELD(S) TOO LONG FOR CENTERING	Items to be centered are too long for the designated space.	Reorganize the data and reenter the request.
181	SUBSCRIPT NAME UNDEFINED OR TYPE INVALID	A data name specified as a subscript for a matrix item is not a defined data name or integer type data name, or is scaled.	Reenter the transmission.
182	-FILLER- INVALID DATA-NAME	The word FILLER is a reserved word in Query Update and can be used only in the DESCRIBE directive.	Reenter the entry without the word FILLER.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
183	MAX QUALIFICATION LEVEL EXCEEDED	The maximum number of five qualifications for a data name has been exceeded.	Check the installation options; consult the data administrator.
184	c NOT FULLY QUALIFIED	The specified item needs further qualification to distinguish it from other items having the same name.	Reenter the entry giving further qualification for the item specified.
185	-c- IS NOT A VALID PERMANENT FILE ID	The specified word is not a valid identifier for a permanent file.	Refer to the operating system reference manual for rules governing permanent file ids.
186	-c- IS NOT A VALID PERMANENT FILE CYCLE NUMBER	The specified number is not valid to indicate a permanent file cycle number. Cycle numbers must be in the range of 1 through 999.	Reenter a valid cycle number.
187	-c- PASSWORD TOO LONG	Passwords are limited to a maximum of nine characters.	Reenter a valid password.
188	-c-, -n- ... P.F. PARAMETERS NOT ALLOWED	The specified words are illegal when given as permanent file parameters.	Reenter the entry using valid permanent file parameters.
189	NO -SAME- LIST TO REFER TO	The reserved word SAME was specified in a DISPLAY, IF, INSERT, UPDATE, or DELETE directive, and a previous list does not exist for the directive.	Reenter the directive creating a list.
190	KEY MUST BE LITERAL CONSTANT	DELETE, INSERT, or UPDATE directives not utilizing the KEY or USING options must be followed by a literal which specifies the value of the key; either a character literal or a numeric literal is valid.	Reenter the directive using the correct format.
191	ILLEGAL RECORD NAME SPECIFIED	The data names following the reserved word USING or SETTING in an INSERT, UPDATE, DELETE, MODIFY, STORE, or REMOVE directive are not all defined in the same record description for the area file.	Reenter the directive by using only data names defined in the same record description for the area file.
192	TOO FEW PARAMETERS FOR UPDATE, INSERT, OR DELETE	An INSERT or DELETE directive has been specified without parameters, or an UPDATE directive with no preceding IF directive has been specified without parameters.	Reenter the directive with valid parameters.
193	ILLEGAL SEQUENCE OF DIRECTIVES, NO FILE UPDATE DONE	When a transmission contains multiple directives and one of the directives requires an update on the area file, all directives in that transmission must also refer to this area file. The valid directives that can be specified to refer to the area file are IF, INSERT, UPDATE, DELETE, MOVE, EVALUATE, DISPLAY, and EXTRACT.	Reenter the transmission with a sequence of valid directives.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
194	PERMANENT CATALOG IS NOT THE CURRENT CATALOG	The DUPLICATE directive cannot be specified when no VERSION directive is in effect.	Input the VERSION directive, followed by the DUPLICATE directive.
195	THIS TRANSMISSION ALREADY EXISTS	A DUPLICATE or a RECORDING directive attempted to write on the catalog a transmission id that already exists for the session.	No action.
196	DATA-NAME IN USING MUST REFERENCE AREA ITEM	A data name specified after the reserved word USING in an INSERT, UPDATE, or DELETE directive must be a data name defined as existing on the area file.	Reenter the transmission.
197	KEY VALUE MUST BE STORED IN TEMPORARY ITEM	The data name following the reserved word KEY in an INSERT, UPDATE, or DELETE directive must be defined as a temporary item.	Define the data name as a temporary item using the DEFINE directive, and reenter the INSERT, UPDATE, or DELETE directive.
198	ILLEGAL KEY SPECIFIED	The data name following the reserved word KEY in an INSERT, UPDATE, or DELETE directive must be defined as a temporary item.	Define the data name as a temporary item using the DEFINE directive, and reenter the INSERT, UPDATE, or DELETE directive.
199	NEXT IS INVALID FOR DISPLAY DIRECTIVE	The subscript NEXT cannot be used in a DISPLAY directive to reference an item in a matrix or an occurring item.	Reenter the DISPLAY directive with the correct format.
200	NON-RECOGNIZABLE DATA TYPE	The valid data types for Query Update are FIXED, INTEGER, FLOATING, DOUBLE, COMPLEX, LOGICAL, CHARACTER, and NUMERIC.	Reenter a valid data type.
201	ONLY CONSTANTS ARE VALID FOR ARRAY PRESET	Constants are the only legal preset values for a matrix (array) being preset by the VALUE IS (or =) clause in the DEFINE directive.	Reenter the DEFINE directive using only constants as preset values.
202	-UPDATE-, -DELETE-, AND -DISPLAY KEY- INVALID ON SEQ. FILE	An UPDATE, DELETE, or DISPLAY KEY directive is illegal when the area file specified through the USE directive is a sequential file.	No action.
203	ONLY ONE KEY MAY BE SPECIFIED	In a DISPLAY directive, the reserved word KEY followed by a literal can be specified only once.	Reenter the DISPLAY directive using the reserved word KEY followed by a literal only once.
204	TRANSMISSION MAY CONTAIN ONLY ONE KEY REFERENCE	In a transmission containing multiple directives, only one directive can specify a key.	Reenter the transmission specifying a key only once.
205	DIRECTIVE-ID TOO LONG	A directive id (transmission id) must be numeric and in the range of 1 through 999.	Reenter the directive id in the correct format.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
206	c UNKNOWN TRANSMISSION-ID	The transmission id specified in the DUPLICATE or RECORDING directive does not exist for the session on the catalog.	Reenter the directive with a valid transmission id.
207	FROM OPTION INVALID WITH GIVEN DIRECTIVES OR DATA-NAMES	In a DISPLAY directive containing the FROM file name option, all the data names to be displayed must refer to a file that has been described either through the DESCRIBE directive or through the EXTRACT directive.	Reenter the DISPLAY directive, with the correct data names.
208	DESTINATION OVERLAPS KEY FIELD	An attempt was made to move data into an area file item that had the same position in the file record as the key.	Use DELETE or INSERT to perform a modification of the key field.
209	REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT	This is an informative diagnostic only. It indicates that the data to be displayed might not be in display code format and, therefore, could be some wild combination of special characters. This could be caused by integer data being displayed without conversion to display code.	No action.
210	DATA-NAMES REFERENCE > 1 RECORD NAME	The data names following the reserved word USING in an INSERT, UPDATE, DELETE, MODIFY, STORE, or REMOVE directive are not all defined in the same record description for the area file.	Reenter the directive using only data names defined in the same record description for the area file.
211	INVALID SYNTAX FOR EXECUTE DIRECTIVE	The value following the reserved word USING in the EXECUTE directive is not a legal data name or literal.	Reenter the EXECUTE directive with a legal data name or literal.
212	-n- IS NOT VALID FOLLOWING HELP	The only valid HELP directive parameter is either a directive name or a diagnostic number.	Reenter the HELP directive with either a directive name or a diagnostic number.
213	USE REQUIRES PARENTHESIZED PARAMETER LIST	The USE directive requires that the permanent file parameters for the area file and the subschema file be enclosed in parentheses.	Reenter the USE directive with the permanent file parameters for the area file and the subschema file in parentheses.
214	PERFORM SESSION NAME SAME AS RECORDING SESSION NAME	The session name specified in the PERFORM directive cannot be the same as the session name specified in the preceding RECORDING directive. This is to minimize the chances of getting into perpetual loops while performing sessions.	Reenter the PERFORM directive with a valid session name.
215	EMBEDDED HYPHENS NOT ALLOWED IN -a- PARAMETER	The specified permanent file parameter is not allowed to contain hyphens. It can consist of letters and numbers only.	Reenter the directive without embedded hyphens in the permanent file parameters.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
216	EOF BEFORE COMPLETION OF TRANSMISSION	While reading from the input file, and end-of-file was detected before a transmission containing continuation lines was completed.	Check the input file.
217	ILLEGAL NUMERIC DATA, DISPLAY SUPPRESSED	A numeric field contains a nonnumeric character. The display for this record was suppressed.	Reenter the numeric literal with only numeric characters.
218	TRANSMISSION-ID a ALREADY EXISTS	The specified transmission id already exists.	Input the EXHIBIT SESSIONS directive to help in finding a suitable gap in which to record.
219	c NOT IN RANGE OF ALLOWED INCREMENTS	The increment size must be in the range of 1 through 999.	Reenter the entry with a valid increment size.
220	n INVALID INTEGER AFTER -BY-	The reserved word BY in a DUPLICATE or RECORDING directive must be followed by an integer specifying the increment size.	Reenter the DUPLICATE or RECORDING directive with integer specifying the increment size following the reserved word BY.
221	DUPLICATION OF a-a WOULD WRITE OVER OR PAST a-a	Duplication terminates when the end of a gap is reached. The transmission id that would go past the end of the gap and the transmission id that marks the end of the gap are specified.	Duplication has been partially completed. Reenter or complete the session as appropriate.
222	DUP OF a-a WOULD REQUIRE TRANSMISSION-ID a	Duplication terminates when the upper limit (999) is reached. The specified transmission id requires that limit to be exceeded.	Duplication has been partially completed. Reenter or complete the session as appropriate.
223	KEY SHOULD NOT BE SPECIFIED FOR AK FILE	In the INSERT directive, a key should not be specified for an area with actual key organization.	Reenter the INSERT directive without specifying a key for the area with actual key organization.
224	NO READ PERMISSION FOR INDEX FILE, USE REJECTED	The index file specified in the USE directive does not have read permission. The USE directive is not executed.	Add the read password to the index file for the USE directive.
225	MODIFY/EXTEND PERM FOR AREA BUT NOT INDEX	When modify/extend permission is specified for an area, it must also be specified for the index file.	Specify modify/extend passwords for the index file, and reenter the transmission.
226	AREA NOT DECLARED AS MULTIPLY INDEXED	The USE or CREATE directive specifies an index file, but the subschema does not designate multiple keys for the area.	Either change the subschema to include multiple keys for the area, or eliminate the index file from the USE or CREATE directive.
227	-DESCRIBE- D ITEMS NOT ALLOWED IN AN -IF- CONDITION	The condition specified in an IF directive can contain only temporary items or items of the data base. An item established with the DESCRIBE directive is outside the data base.	Reenter the IF directive with only temporary items or data base items.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
228	FIELD OVERLAP IN PAGE HEADING(S)	One or more of the page number, date, and time fields overlap existing data. This could have been caused by using the default location of these fields.	Check to see if there is any overlap with the default parameters. Correct or redefine overlap data.
229	a MULTIPLY DEFINED ON CONTROL CARD. LAST WILL BE USED	In the REPORT control statement, the specified parameter (I, P, R, T, or V) appears more than once. The last occurrence of the parameter takes precedence.	No action.
230	a PARAMETER ON CONTROL CARD MISSING. FATAL ERROR	In the REPORT control statement, the specified parameter (R or T) has not been included. This is a fatal error.	Reenter the REPORT control statement with either the R or T parameter.
231	I OR P PARAMETER MUST BE PRESENT ON CONTROL CARD	The REPORT control statement must specify either an input file (I parameter) or the preview option (P parameter). This is a fatal error.	Reenter the REPORT control statement with either the I or the P parameter.
232	UNRECOGNIZED CONTROL CARD OPTION a	The specified option is not valid in the REPORT control statement; valid parameters are I, R, P, T, and V.	Reenter the REPORT control statement with a valid parameter.
233	INVALID DELIMITER ON CONTROL CARD FOLLOWING -a-	The REPORT control statement contains a delimiter that is not valid; valid delimiters are the equals sign, comma, right parenthesis, and period.	Reenter the REPORT control statement with a valid separator.
234	CONTROL CARD ERROR -- a MUST BE FOLLOWED BY =LFN	In the REPORT control statement, the T or I parameter must specify the file name of the table file or the input file.	Reenter the REPORT control statement with either the T or I parameter specifying the file name of the table file or the input file.
236	PREPARE/PREVIEW/COMPILE CANT BE FOLLOWED BY ANY DIRECTIVE	PREPARE/PREVIEW/COMPILE directives must appear as the last directive in a transmission. If any directive follows, it is ignored. The occurrence of this diagnostic does not alter report preparation.	Split the directives into separate transmissions.
238	MISSING TABLE a	The report name specified in the REPORT control statement cannot be found on the table file.	Reenter the REPORT control statement with a valid report name.
239	VALUE SUPPLIED HAS WRONG TYPE -- VARIABLE a	The type (character, numeric, logical, etc.) of the literal supplied for the temporary data item is not compatible with the type of the data item.	Reenter the type of the literal for the temporary data item so that it is compatible with the type of the data item.
240	ILLEGAL FILE NAME a	The specified file name is not valid. A valid file name can contain 1 to 7 alphanumeric characters; the first character must be alphabetic.	Reenter the directive with a valid file name.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
241	PARAMETER a NOT FOUND IN THE DEFINE OR SPECIFY LIST	The specified parameter has not been established by either a DEFINE or SPECIFY directive.	Correct the parameter in the variable file, or recreate the table file for that report including definition of the missing parameter.
242	ENTRY FOR VARIABLE a HAS ILLEGAL TYPE VALUE	The specified variable does not have a legal type code. This is an internal error.	Consult the data administrator.
243	ILLEGAL LOGICAL VALUE a	The logical value is not TRUE or FALSE.	Reenter the logical value as TRUE or FALSE.
244	NO TRANSMISSIONS ALLOWED, THE FIRST ID WOULD BE a	The default starting transmission id exceeds 999. The specified transmission id is the result of adding the current increment size to the highest transmission id of directives in the session.	Define a lower transmission id of increment size for RECORDING.
245	UNIVERSAL AND SEPARATOR CHARS CANNOT BE THE SAME	The universal character and the delimiter cannot be the same value.	Reenter the SEPARATOR and/or UNIVERSAL directives using valid characters.
246	n ILLEGAL UNIVERSAL CHARACTER	The character specified for the universal character is either the same as the delimiter or an invalid universal character.	Reenter the UNIVERSAL directive using a valid character.
247	c - USER NUMBER TOO LONG	The user number (UN) is greater than seven characters.	Reenter the user number with seven or fewer characters.
248	c - PASSWORD TOO LONG	The specified password (PW) is greater than seven characters.	Reenter the password with seven or fewer characters.
249	-c- IS AN ILLEGAL PERMANENT FILE NAME	A valid permanent file name contains 1 to 40 alphanumeric characters; the first character must be alphabetic.	Reenter the directive with a valid permanent file name.
250	-c- SETNAME TOO LONG OR NULL	The SN parameter is limited to a maximum of seven characters.	Reenter the directive with seven or fewer characters in the SN parameter.
251	ATTEMPT TO USE OLD REPORT TABLE FILE. TERMINATED	An attempt was made to use a table file incompatible with this version of Query Update. The table file for this report must be recreated by using the Query Update COMPILE directive.	Recompile the report using the COMPILE directive.
253	EXTEND/APPEND PERMISSION NEEDED ON LOGFILE	The area requires a logfile that can be extended; this permission is not available.	Consult the data administrator for corrective action.
254	CREATE AND USE MUST SPECIFY A DIRECTORY WITH -OF-	The CREATE and USE directives require that a subschema be specified with the OF clause.	Reenter the CREATE or USE directive specifying the subschema with the OF clause.
255	CANNOT FIND THE NEEDED LOGFILE	The area requires a logfile that exists as a permanent file; the needed file does not exist.	Consult the data administrator for corrective action.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
256	P.F. ERROR o ON -a-	The indicated permanent file manager error has occurred from a permanent file request on the specified file.	Refer to the user return code section of the permanent file macro descriptions in the operating system reference manual for a full list of return codes.
257	-a- NOT FOUND. CHECK PF PARAMETERS	The specified file was not found by the permanent file manager. The password, pack name, or user number may be incorrect, the requested mode may exceed your authorization, or the file may not exist.	Check the permanent file parameters, and reenter the directive. If the problem persists, check with your data administrator.
258	-a- BUSY. TRY AGAIN LATER	The specified file is attached to some job or terminal with a mode or permission that does not allow concurrent access with your mode or permission.	Consult the data administrator for corrective action.
259	LFN -a- ALREADY IN USE	Query Update attempted to attach a permanent file using the specified logical file name. The attach failed because that lfn is already in use.	Return the file with that lfn, and reenter the directive.
260	P.F. -a- NOT KNOWN BY THE SYSTEM. CHECK PARAMETERS	The permanent file manager cannot find the specified file.	Check the ID or SN parameter, the cycle number (CY), the spelling of the permanent file name, or perform an AUDIT to see if the file exists.
261	-a- UNAVAILABLE. FILE IS ATTACHED ELSEWHERE	The specified file is unavailable. It is attached to another terminal or job such that concurrent access is not allowed.	No action.
262	-a- ALREADY ATTACHED	The specified file is already attached to this job/terminal. It must be returned before Query Update can process this directive.	Return the file, and reenter the directive.
263	USER -SEARCH- EXIT REQUIRES -OPEN- AND -CLOSE- EXITS	An OPEN and a CLOSE data base procedure must be supplied when a SEARCH data base procedure is specified.	Specify an OPEN and a CLOSE data base procedure in the DDL subschema.
264	-FILE- CARD TRUNCATED RECORDS FROM d CHARS to d CHARS	A FILE card set the fixed-length of the record to be less than the amount needed for all the extracted items.	Correct the fixed-length of the record on the FILE card.
265	TAG OR LEVEL NUMBER MISSING	The report production step specified in the MOVE or EVALUATE directive requires a tag or level number when not preceded by ANY or NO.	Reenter the MOVE or EVALUATE directive with a tag or level number.
266	SYSTEM ABORT DUE TO CP TIME LIMIT	The operating system aborted Query Update because the central processor time limit for the job was exceeded.	Increase the time limit.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
267	SYSTEM ABORT DUE TO ARITHMETIC MODE ERROR	This is a Query Update system error.	Follow site defined procedures for reporting software problems.
268	SYSTEM ABORT DUE TO A PP REQUESTED ABORT	This is a Query Update system error.	Follow site defined procedures for reporting software problems.
269	SYSTEM ABORT DUE TO AN EXPLICIT ABORT REQUEST BY -QU-	This is a Query Update system error.	Follow site defined procedures for reporting software problems.
270	SYSTEM ABORT DUE TO A PP CALL ERROR	This is a Query Update system error.	Follow site defined procedures for reporting software problems.
271	SYSTEM ABORT DUE TO AN OPERATOR DROP	The console operator dropped the Query Update run.	Consult the operations staff to determine the cause.
272	SYSTEM ABORT DUE TO AN OPERATOR KILL	The console operator killed the Query Update run.	Consult the operations staff to determine the cause.
273	SYSTEM ABORT DUE TO AN OPERATOR RERUN	The console operator reran the Query Update run.	Consult the operations staff to determine the cause.
274	SYSTEM ABORT DUE TO AN ECS PARITY ERROR	This is a hardware error.	Consult the system analyst.
275	SYSTEM ABORT DUE TO A REQUIRED BUT MISSING RECALL STATUS	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
276	SYSTEM ABORT DUE TO -HUNG IN AUTO RECALL-	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
277	SYSTEM ABORT DUE TO EXCEEDED MASS STORAGE LIMIT	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
278	SYSTEM ABORT DUE TO AN EXCEEDED I/O TIME LIMIT	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
279	SYSTEM ABORT DUE TO A PROGRAM STOP	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
280	SYSTEM ABORT DUE TO AN EXCEEDED FILE LIMIT	The file limit for the user was exceeded.	Return any local files that are not needed. If the files are needed, the file limit should be increased.
281	SYSTEM ABORT DUE TO UNKNOWN CAUSE	The cause of this error is unknown.	Consult the system analyst.
282	-n- MAY NOT FOLLOW -ANY-	The report production directives that can be specified after ANY are BREAK, DETAIL, FOOTING, HEADING, and SELECT.	Reenter the transmission.
283	-n- MAY NOT FOLLOW -NO-	The only report production directive that can be specified after NO is SELECT.	Input the SELECT directive.
284	-n- INVALID REPORT PRODUCTION STEP	The report production steps that can be specified in the MOVE and EVALUATE report directives are BREAK, DETAIL, FOOTING, HEADING, SELECT, RECAP, REPORT, and TITLE.	Reenter the directive with a valid report production step.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
285	SYSTEM ABORT DUE TO MASS STORAGE TRACK LIMIT	This is an operating system error.	Consult the operations staff.
286	SYSTEM ABORT DUE TO ERROR-FLAG SETTING -- o	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
287	-PART- OR -FULL- MUST BE SPECIFIED FOR THE -DIAG- DIRECTIVE	The DIAGNOSTIC directive requires that PART or FULL be specified.	Reenter the DIAGNOSTIC directive with PART or FULL specified.
288	-a- HAS BAD -AT LINE N- VALUE	The specified report production contains an -AT LINE DATA-NAME- clause. The value of < DATA-NAME > specified a line number which does not exist in the current pages (less than 1 or greater than page size).	If the < DATA NAME > value is in error, correct it. Otherwise, increase PAGE-SIZE to exceed the value of < DATA-NAME >.
289	INVALID PICTURE SPECIFICATION FOR CHARACTER ITEM	The item being displayed was defined with an invalid picture and cannot be displayed or viewed.	If this is a temporary defined item, the contents can be moved to another temporary defined item, and can be displayed there.
290	THE PARAMETER VALUE FOR a MUST BE AN INTEGER	The specified permanent file parameter must be given an integer value.	Correct the directive after consulting the appropriate operating system manuals for permanent file parameter definitions.
291	-c- IS AN INVALID PACK NAME	The specified pack name does not conform to the pack naming conventions.	Reenter with a valid pack name. Refer to the NOS reference manual for rules on naming packs.
292	-c- IS AN INVALID DEVICE SPECIFICATION	The specified value does not conform to the format of device specifications.	Refer to the NOS reference manual for legal device specifications. Reenter the directive with a valid device specification.
293	INTERNAL ERROR - MOVEC CALLED WITH NEGATIVE LENGTH FROM o	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
294	THE -OS- DIRECTIVE IS NOT SUPPORTED IN BATCH MODE. IGNORED	The OS directive is for use by the interactive user. The control statements should not be preceded by the OS directive.	Resubmit without the OS directive.
295	THE -OS- DIRECTIVE REQUIRES A CONTROL STATEMENT	A control statement must follow the directive keyword OS.	Reenter the OS directive with a control statement.
296	CONTROL STATEMENT LENGTH EXCEEDS 79 CHARS. REQUEST IGNORED	The control statement entered with the OS directive must have fewer than 80 characters.	Reenter the OS directive with a valid number of characters, or issue the control statement outside of Query Update.
297	NO SPACE AVAILABLE. -OS- DIRECTIVE IGNORED	Query Update cannot process the OS directive because of internal space limitations.	Space can be released by erasing unneeded temporary data names or by issuing appropriate return directives.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
298	FIELD IN LINE EXTENDS BEYOND PAGE OR SECTION WIDTH	A field in the entered report directive would extend beyond the page or section width.	The field can be shortened or repositioned, or the page or section width can be changed.
299	-n- IS NOT A VALID FILE NAME	An invalid name was used in the USE, CREATE, or RETURN directive.	Check to see whether the subschema name, area name, relation name, or file name is valid.
300	KEYWORD -OF- MUST FOLLOW -c-	The CREATE and USE directives require the keyword OF.	Reenter the CREATE or USE directive with the keyword OF.
301	THE SUB-SCHEMA IS REQUIRED TO HAVE PERMANENT FILE PARAMETERS	The USE and CREATE directives require permanent file parameters for the subschema specified. They must be enclosed in parentheses.	Reenter the CREATE or USE directive with permanent file parameters for the subschema specified.
302	USE/CREATE/INVOKE MUST BE LAST DIRECTIVE IN THE TRANSMISSION	A USE, CREATE, or INVOKE directive must be the last directive specified in a transmission. It cannot be followed by another directive in the same transmission.	Reenter the transmission.
303	-c- IS NOT VALID AS A PERMANENT FILE PARAMETER	The value specified is not valid for the keyword preceding it.	Reenter the transmission with a valid value for the keyword preceding it.
304	-c- IS A DUPLICATE PERMANENT FILE PARAMETER	The indicated permanent file parameter can be specified only once in a permanent file parameter list. The PW parameter under the NOS/BE operating system is the only parameter for which more than one specification is valid.	Reenter the transmission.
305	QU ABORT DUE TO FATAL DIAGNOSTIC	Query Update encountered a fatal diagnostic and aborted processing the batch session.	The condition that caused the fatal diagnostic should be corrected, and the entire batch should be resubmitted.
306	-a- CANNOT BE CALLED WITH THE -OS- DIRECTIVE	Certain control statements cannot be entered with the OS directive because of restrictions imposed by the operating system.	No action.
307	READ-ONLY MODE DISABLES c	The RO parameter in the Query Update control statement disables the indicated directive. The directives CREATE, DELETE, INSERT, VERIFY, UPDATE, STORE, MODIFY, and REMOVE cannot be used in read-only mode.	Reenter the Query Update control statement without the RO parameter, or take no action.
308	READ-ONLY MODE LIMITS TARGETS TO TEMPORARY ITEMS	The RO parameter in the Query Update control statement limits the scope of the UPDATE, MODIFY, and MOVE directives. Destination items in these directives must be temporary data items generated by the DEFINE directive.	Exit READ ONLY mode if update is required. If necessary, check permanent file parameters.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
309	INCOMPATIBLE SUB-SCHEMA. PLEASE RECOMPILE	This subschema was compiled with an old version of DDL which is incompatible with the current version of Query Update. The subschema should be recompiled with the correct DDL.	Recompile the subschema with the correct DDL.
310	NO ROOM FOR HANDLING THIS CATALOG FILE	Query Update cannot reserve the needed space to handle the catalog file. The file was returned.	No action.
311	CANNOT DUPLICATE TO A CATALOG OF SMALLER TL	The transmission length for the duplication catalog is smaller than the transmission length for the source catalog. Duplication can only be done from a smaller transmission length to a larger transmission length, or if the transmission lengths are the same.	No action.
312	CANNOT RECORD ON A CATALOG OF SMALLER TL	Recording is not allowed on a catalog file that was created with a smaller transmission length than is currently set.	Type EXHIBIT to find the transmission length of the catalog file. Reenter Query Update setting a transmission length comparable to the transmission length of the catalog file.
313	CANNOT FORMAT/ALTER REPORTS ON CATALOG OF SMALLER TL	The FORMAT or ALTER directive cannot be specified for a catalog file that was created with a smaller transmission length than is currently set.	Type EXHIBIT to find the transmission length of the catalog file. Reenter Query Update setting a transmission length comparable to the transmission length of the catalog file.
314	SPECIFIED -FROM- FILE CANNOT BE FOUND	The local file specified in this directive is not available.	Check to see if the file name has been entered correctly.
315	ENTRY POINT -a- NOT FOUND FOR AREA - a	Data base procedures required for the USE or CREATE directive issued could not be loaded by Query Update.	Contact the data administrator for corrective action.
316	THE RELATION NAME c FOLLOWING VIA IS NOT DEFINED	The directive VIA must be followed by a relation name defined for the subschema currently in use. The VIA directive need not be entered if only one relation is defined.	Reenter the VIA directive with the relation name.
317	INTERNAL ERROR - DBP ENTRY POINT LIST OVERFLOW	The block of memory allocated by Query Update for the data base procedure entry point list is not large enough. This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
318	INTERNAL ERROR -- NO AREA TABLE	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
319	ENTRY POINT -a- NOT FOUND	The entry point for the external procedure specified in the EXECUTE directive could not be found. The procedure must be in a user file specified with the FROM option or in the SYSMISC library if the FROM option is omitted.	The procedure must be in a user file specified with the FROM option or in the SYSMISC library if the FROM option is omitted.
320	NO DESCRIBE ENTRY FOR THE FILE c	The file specified in the ERASE directive must have been described in a DESCRIBE or EXTRACT directive.	Check to see if the correct file is being used.
321	INTERNAL ERROR - DESCOUNT ENTRY NEGATIVE	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
322	INTERNAL ERROR - MISSING RANK ENTRY	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
323	NO RELATION CONNECTS THE AREAS WHICH ARE REFERENCED	When more than one area is referenced in a transmission, the areas must be joined in a relation.	Check the subschema and transmission to determine whether the correct areas have been used.
324	NO USE ENTRY FOR THE FILE n	The indicated area file was not specified in the currently active USE directive, either directly through the area specification or indirectly through the specification of the subschema only or the relation only.	Specify the file, then reenter the transmission.
325	LFN -c- HAS NOT BEEN DESCRIBED	The logical file name indicated names a file that has not been described with the DESCRIBE or EXTRACT directive or specified as the output file in a SORT directive. It cannot be used as a source file unless so described.	Reenter the transmission, specifying a described file.
326	AREA -a- DOES NOT HAVE ANY PERMANENT FILE PARAMETERS	The permanent file parameters for the area specified in a USE or CREATE directive are missing.	Reenter the CREATE or USE directive, specifying permanent file parameters.
327	-a- IS NOT AN ACTIVE AREA NAME. PLEASE RE-ENTER	The area name that has been entered in the prompting sequence following a USE directive in which a relation name and the MODIFIED option have been specified is not an area joined in the relation indicated.	Check the subschema for areas joined in the relation. Reenter the transmission with a valid area name.
328	-a- IS NOT A VALID P.F. PARAMETER	The valid permanent file parameters can be entered under the NOS/BE operating system are ID, PW, SN, CY, LC, MR, and RW; under the NOS operating system, UN, PW, PN, R, and M.	Reenter the transmission with valid permanent file parameters.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
329	TOO MANY PASSWORDS FOR -a-	Using the USE directive with the MODIFIED or PW option under the NOS/BE operating system, the user attempted to specify more than five passwords for an area or index file. A maximum of five passwords is allowed.	Reenter one or more directives to specify the desired passwords. A suggested sequence is to enter, after a prompt for permanent file parameters, the area name or index file name followed by the PW parameter specified alone and the PW parameter specified with the desired passwords. The first specification of PW causes Query Update to ignore the existing passwords; the second enters the desired ones.
330	DEPEND ON ITEM LESS THAN 1 FOR ARRAY SUBSCRIPTED BY LAST	The -DEPENDING ON- item for an array subscripted with the figurative subscript -LAST- must be equal to or greater than 1 if the array is to be the source field for a move or display.	Reenter the transmission.
331	SYSTEM ABORT DUE TO A USER ABORT OR AN OPERATOR DROP	The user aborted Query Update with a %A in NOS/BE or S in STOP in NOS, or the operator dropped the Query Update run.	If an operator drop, contact the operations staff to determine the cause.
332	-c- PARAMETERS MUST BE OF CONSISTENT DATA TYPE	In any one use of the functions, MAX, MIN, and MASK, the given parameters must have the same data type with one exception: mixed integer and octal parameters are allowed in the use of the MASK function. In the use of the DECODE function, the parameter types cannot be mixed if one or more of the parameters is CHARACTER or LOGICAL.	Reenter the transmission with correct function parameters.
333	ITEM SIZE EXCEEDS 2047 CHARACTERS	The size of an item is restricted to a maximum of 2047 characters.	Define two or more items whose lengths are each less than or equal to 2047 characters.
334	NUMBER OF ITEMS IN ARRAY OUT OF RANGE	The number of items in an array must be within the range of 1 through 4095.	Reenter the transmission.
336	EMPTY - EXECUTE FROM - FILE	The user file specified in the FROM option of the EXECUTE directive was empty and did not contain the expected relocatable external procedure.	Provide necessary routines required by the EXECUTE request.
337	INTERNAL ERROR - NO TABLES FOR THE EXECUTE DIRECTIVE	The procedure specified in the EXECUTE directive could not be called. This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
338	UPDATING IS ALLOWED AGAINST A SINGLE AREA ONLY	The fields specified in the UPDATE, INSERT, or DELETE directive are in more than one area. Updating can be performed only on one area at a time.	Qualify the fields to reference only one area at a time.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
339	AREA TO BE UPDATED IS NOT THE CURRENT TARGET AREA	The area to be updated must be the same as that given in a previous UPDATE directive in which the area name option is used.	Enter a new UPDATE area name directive to update another area.
340	CANNOT FIND SPECIFIED RELATION NAME	The relation name that was entered does not name a relation in the current subschema specified.	Check the applicable relation names for the correct one, or verify the correct spelling of the name.
341	THE AREA NAME SPECIFIED IS NOT VALID	The area name specified in the UPDATE directive must be an active area.	Check the applicable area names for the correct one, or verify the correct spelling of the name.
342	-a- USED BY RELATION, CANNOT BE RETURNED	The area name specified in the RETURN directive cannot be returned because it is a part of a relation that is still active.	No action.
343	COULD NOT ATTACH LIBRARY OF DATABASE PRODEDURES	Query Update could not attach the data base procedure library. Further processing of the USE or CREATE directive cannot proceed.	Consult the data administrator.
344	n CANNOT FOLLOW - UPDATE AREANAME -	Format 2 of the UPDATE directive identifies the area that is to be updated with updating directives to follow. Only area name should be specified following the word UPDATE.	Reenter the UPDATE directive with only the area name following the word UPDATE.
345	THE AREA TO BE USED CANNOT BE ISOLATED	Query Update cannot determine which area to use to process the current directive.	Enter either the UPDATE directive or the USE directive to identify the area.
346	CALL TO -a- TRAPPED WHILE EXECUTING -a-	The indicated data base procedure referenced an entry point that does not exist withing the current Query Update overlay. This is a nonfatal system error.	Consult the data administrator concerning the procedure so that the procedure can be checked and the reference removed, if possible.
347	DUPLICATE ALTERNATE KEY ON INSERT	The record is not inserted. The area being updated has at least one alternate key defined which cannot have duplicate values. The record to be inserted would create a duplicate alternate key situation for this area.	No action.
348	MISSING -ON UPDATE-PROCEDURE. TRANSMISSION TERMINATED	The area being updated has a SEARCH data base procedure that indicates that all input/output is to be performed by data base procedures. Any updating of this area requires an UPDATE data base procedure to perform the necessary input/output.	Consult the data administrator.
349	DISPLAY KEY REFERENCES MORE THAN ONE AREA	The display directive specifying the KEY option must contain a key that is unique to one area only. If an ambiguity exists, Query Update issues this message.	Enter a USE directive specifying the single area.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
350	ONLY ONE OUTPUT LFN CAN BE SPECIFIED WITH -UPON-	The UPON option used to specify the output file for the SORT directive was used more than once.	Specify only one output file.
351	ONLY ONE COLLATING SEQUENCE CAN BE SPECIFIED WITH -USING-	The USING option in the SORT directive specifies a collating sequence for ordering the file.	Specify only one collating sequence.
352	REDUNDANT USE OF -UNIQUE- IGNORED	The UNIQUE option in the SORT directive was specified more than once. Query Update ignores the additional specification.	No action.
353	ONLY ONE SET OF KEYS MAY BE SPECIFIED WITH -ON-	The ON option in the SORT directive was specified more than once. Only one set of up to 25 sequencing fields can be specified.	Specify only one set of up to 25 sequencing fields.
354	-n- INVALID IN SORT DIRECTIVE SYNTAX	The indicated syntax word is not valid for the SORT directive.	Reenter the SORT directive with the correct syntax.
355	CANNOT FIND DESCRIPTION OF SOURCE FILE	The DESCRIBED IN option, used to name the directory for the file from which a report is produced, was omitted from the PREVIEW or the COMPILE directive. This message might also diagnose the case of no directory or multiple directories for a file or the case of several files with different file structures for which a single directory exists.	Reenter the PREVIEW or COMPILE directive with the DESCRIBED IN option.
356	FILE NAME MISSING	A file name is required for the directive.	Reenter the directive with the required file name.
357	INDEFINITE OR INFINITE OPERAND ENCOUNTERED	An attempt was made to perform an operation on an indefinite or infinite operand. Possible cause is leaving out the DISPLAY option on an AS clause in a DESCRIBE directive.	Add the DISPLAY option or make the appropriate change to to eliminate the bad operand.
358	ORIGINAL MRL DOES NOT MATCH CURRENT MRL FOR SUBSCHEMA -a-	The indicated file was created with a different maximum record length (MRL) than the maximum record length in the current subschema. This message is informative; Query Update allocates larger buffers to accommodate files created with a larger maximum record length than that associated with the fields described in the subschema.	No action; or consult the data administrator to correct the subschema description to make it compatible with its file.
359	-a- NOT FIRST SAVED DIRECTIVE	The directive indicated must be the first saved directive in a transmission. The directive can be preceded by an IF directive that names a temporary data item because temporary data items are evaluated immediately and not saved.	Reenter the transmission with the indicated directive as the first saved directive.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
360	ORIGINAL KL DOES NOT MATCH KL FOR SUBSCHEMA -a-	The indicated file was created with a different key length (KL) than the KL in the current subschema. The file was not opened.	Consult the data administrator to correct the subschema description to make it compatible with the data base file in use.
361	VALID ERASE OPTION OR DATA NAME OMITTED	The ERASE directive must specify DESCRIBE, REPORT, SESSION, or temporary data names to be erased.	Reenter the ERASE directive with a valid ERASE option or data name.
362	-c- MUST BE LITERAL OR DEFINED ITEM	Parameters passed to a procedure must be either literals or items established in a DEFINE directive.	Reenter the EXECUTE directive with valid parameters.
363	-c- NOT ON FILE TO BE SORTED	The SORT keys must be data items on the file to be sorted.	Reenter the SORT directive with valid keys.
364	AMBIGUOUS RELATIONS IN BATCH MODE - COMMAND IGNORED	Ambiguous relations in batch mode will not be resolved unless the -VIA- directive specifies which relation to use.	Input the VIA directive, then reenter the command that was ignored.
365	-OF- ILLEGAL FOR DEFINED, DESCRIBED, OR SPECIFIED ITEMS	The data name in this directive must be a defined, described, or specified item; -OF- cannot be used with defined, described, or specified items.	Reenter the transmission without OF file name.
366	c MUST BE WORD ALIGNED	The indicated data item is described as integer, fixed, floating, double, complex, or logical without the display option and must be word aligned.	Reenter the transmission with FILLER in the DESCRIBE directive to word align the data items.
367	-n- IS INVALID OPTION OF -EXHIBIT-	The indicated syntax word is not a valid option of the EXHIBIT directive.	Reenter the EXHIBIT directive with a valid EXHIBIT option.
368	VERIFY LIST DID NOT REFERENCE EXACTLY ONE AREA	The -VERIFY- dataname list did not reference an area, or referenced more than one area.	Reenter the VERIFY directive with data names from only one area.
369	VERIFY LIST CLEARED DUE TO DATANAME ERROR	The -VERIFY- dataname list contains an error. The previous -VERIFY- list, if any, was cleared; the erroneous list was ignored.	Reenter the dataname list with the VERIFY directive.
370	CANNOT HAVE MULTIPLE DISPLAY -FROM- OR -KEY IN- FILES	A transmission can contain only one DISPLAY FROM file or one DISPLAY KEY IN file.	Reenter each DISPLAY FROM file and DISPLAY KEY IN file as a separate transmission.
371	-c- TRUNCATED TO 4095 CHARACTERS	The data name indicated is longer than 4095 characters and is truncated to 4095 characters.	Reenter the directive specifying data items shorter than 4095 characters.
372	MORE THAN 64 AREAS IN USE	A maximum of 64 areas can be specified at one time.	Reenter the USE directive specifying fewer areas.
373	MORE THAN 59 RELATIONS IN USE	A maximum of 59 relations can be specified at one time.	Reenter the USE directive specifying fewer relations.
374	-c- IS NOT AN AREA ITEM	An item specified in the EXTRACT directive can only be an item that is contained within an area.	Reenter the EXTRACT directive using only area items.
375	FILENAME - <N.> IS AN INVALID MOVE DESTINATION	Indicated source word is not valid as a destination for a MOVE directive.	Reenter the MOVE directive with a valid name.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
376	REMAINDER OF ENTRY IGNORED	After the PW or MODIFIED option in a USE directive was specified, there was an error in a parameter subsequently entered. Query Update ignores any parameter entered after the one in error.	Reenter appropriate directives to include the parameter that was in error and any that were ignored.
377	CDCS DATA-NAMES REFERENCE > 1 RECORD	The data names in a transmission containing an INSERT, UPDATE, DELETE, STORE, MODIFY, REMOVE, or DISPLAY KEY directive in CDCS data base access mode are not all defined in the same record description for the area file.	Reenter the transmission with data names defined in the same record description for the area file.
378	IF DIRECTIVE REFERENCES KEYS FROM > 1 RECORD NAME	In CDCS data base access mode, a full pass through a file can take place, because primary or alternate keys from a record description different from the key in the first IF directive are treated as non-key items.	Reenter the IF directive with keys from one record.
379	-c- INVALID ACCESS KEY	The access key must be a 1- to 30-character literal or defined item of type character; it cannot be subscripted.	Reenter the ACCESS directive with a valid literal or defined item.
380	AREA-NAME NOT SUPPLIED FOR ACCESS KEY	An area name must accompany the FOR AREA clause in the ACCESS directive.	Reenter the ACCESS directive with an area name.
381	AREA -c- IS NOT IN USE	The ACCESS directive specified an area that is not known by the system.	Reenter the ACCESS directive with a correct area name.
382	-c- IS NOT THE CURRENT CATALOG	The ACCESS directive specified a catalog that is not the same as the current catalog.	Reenter the ACCESS directive with the current catalog name.
383	NO AREA OPTION SPECIFIED IN -FOR- CLAUSE	The AREA, CATALOG, or AREAS option must accompany the FOR clause in an ACCESS directive.	Reenter the ACCESS directive with the AREA, CATALOG, or AREAS option.
384	ACCESS KEY ALREADY EXISTS FOR FILE -a-. NEW ONE IGNORED	The indicated file already has a key for the input/output options given by the current ACCESS directive. It is illegal to overwrite the existing key; the current directive is ignored.	No action.
385	NOT IN CDCS CATALOG MODE. NO ACCESS KEY ALLOWED	An access key is allowed only for CDCS catalog files and not for CRM catalog files.	Do not specify an access key.
386	NOT IN CDCS DATABASE MODE. NO ACCESS KEY ALLOWED	An access key is allowed only for CDCS data base access files and not for CRM data base access files.	Do not specify an access key.
387	FIRST PARAMETER MUST BE A DEFINED INTEGER	The first parameter must be defined as an integer because it will contain the unique recovery point number assigned by CDCS.	Define the first parameter as an integer.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
388	SECOND PARAMETER MUST BE 1 TO 30 ALPHANUMERIC CHARACTERS	The second parameter must be a 1- to 30-character alphanumeric literal or data name because it will contain a user-supplied explanatory message for the recovery point.	Define the second parameter as a 1- to 30-character alphanumeric literal or data name.
389	DATANAME MUST FOLLOW THE KEYWORD -SETTING-	In the STORE, MODIFY, or REMOVE directive, at least one data name must follow the reserved word SETTING.	Reenter the directive with at least one data name following SETTING.
390	DATANAME MUST FOLLOW THE KEYWORD -USING-	When used in the MODIFY or REMOVE directive, a legal data name must follow the reserved word USING.	Reenter the directive with a legal data name following USING.
391	ONLY ONE -MOVE- CLAUSE ALLOWED	In any STORE or MODIFY directive, the MOVE clause can only be specified once.	Reenter the directive with only one MOVE clause.
392	-TO- MUST SEPARATE EXPRESSION AND DESTINATION IN MOVE	When a MOVE clause occurs in a STORE or MODIFY directive, the keyword TO must separate the expression to be moved from the data name list specifying the MOVE destinations.	Reenter the directive with the keyword TO.
393	-USING- MUST SPECIFY A PRIMARY, ALTERNATE, OR MAJOR KEY	The USING clause in a MODIFY or REMOVE directive must specify an area item that is a primary, alternate, primary major, or alternate major key for the area in use.	Reenter the directive with a valid key.
394	SEARCH KEY MISSING	A primary, alternate, or major key must be specified for the MODIFY or REMOVE directive through the USING clause, or a primary key for the STORE directive through the SETTING or MOVE clause. If a key is not specified, an IF directive on an area item must precede the MODIFY or REMOVE directive to select the record to be updated.	Reenter the directive with a key specified, or enter an IF on an area item.
395	CDCS MUST BE INVOKED IN ORDER TO EXECUTE THE CURRENT DIRECTIVE	The current directive requires a subschema in CDCS data base access mode or a CDCS catalog file to be invoked during execution.	Execute a USE, CREATE, INVOKE, or VERSION directive.
396	-c- FROM RESTRICT CLAUSE HAS NOT BEEN DEFINED	The indicated syntax word is used as an identifier in a RESTRICT clause for the relation in use, but it has not been defined as a temporary item.	Define the indicated syntax word as a temporary item.
397	c AND AREA ITEM SPECIFICATIONS DO NOT MATCH FOR RESTRICT	The indicated syntax word is used as an identifier in a RESTRICT clause but its data specifications do not match those of the area item to which it is compared.	Redefine the indicated syntax word with the correct data specifications.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
398	INSUFFICIENT PARAMETERS FOR -STORE- OR -MODIFY-	The STORE and MODIFY directives require at least a USING or MOVE clause to be legal.	Reenter the directive with a USING or MOVE clause.
399	-MODIFY- OR -REMOVE- BY KEY ILLEGAL ON SEQUENTIAL FILE	A MODIFY or REMOVE directive that specifies a search key is illegal when the area in use is a sequential file.	Reenter the directive without specifying a key.
400	AREA OR RELATION IGNORED ON -USE- FOR CDCS SUBSCHEMA	In CDCS data base access mode, the USE directive makes all the areas and relations in the subschema available for use; the specified area or relation is ignored.	No action
401	KEYWORD -LIBRARY- MUST APPEAR AFTER THE -FROM-	When the optional word FROM is included in a CREATE, INVOKE, or VERSION directive, the word LIBRARY must follow the FROM.	Reenter the directive with the word LIBRARY.
402	AREA OR RELATION CANNOT BE RETURNED WHEN CDCS IS INVOKED	Query Update does not allow areas and relations to be returned in CDCS data base access mode; only subschemas can be returned.	Reenter the RETURN directive with the subschema name.
403	-a- UNAVAILABLE. FILE IS ARCHIVED	The indicated file must be returned to mass storage before Query Update can read or write it.	Consult the operations staff for installation procedures regarding archived files.
404	FATAL CDCS ERROR; SUBSCHEMA RETURNED	A fatal error caused CDCS to terminate the subschema; the subschema and all central memory associated with it have been returned.	Consult the data administrator.
405	CANNOT RUN INTERACTIVE ON NOS	CDCS can only be used with jobs of batch origin on the NOS operating system.	Submit the job as a batch job.
406	SUBSCHEMA -a- NOT THE SAME AS CDCS CATALOG FILE	A subschema in CRM data base access mode is illegal when in CDCS data base access mode because the CREATE, INVOKE, or USE directive must specify the same subschema that describes the catalog file.	Specify the correct subschema.
407	AREA -a- IS PART OF A RELATION -VERSION- REJECTED	If a VERSION directive specifies a CDCS area file, the file must not be part of any relation.	Reenter the directive with the name of the correct file.
408	SUBSCHEMA -a- MUST BE A CDCS SUBSCHEMA	To enter CDCS catalog mode, the VERSION directive must specify a subschema in CDCS data base access mode.	Reenter the directive and specify a subschema in CDCS data base access mode.
409	AREA -a- NOT IN SUBSCHEMA -a-	The VERSION directive specified an area that is not part of the indicated subschema.	Reenter the VERSION directive with a correct area name.
410	AREA -a- IS NOT SEQUENCE DISPLAY	The SEQUENCE clause in the schema must specify DISPLAY when an area is to be used as a CDCS catalog file.	Consult the data administrator.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
411	AREA -a- MUST HAVE FO = IS, AND RT = W	An area being used as the CDCS catalog file must have a file organization of indexed sequential (IS) and control word type records (W).	Specify the correct permanent file parameters.
412	AREA -a- IMPROPER RECORD FORMAT FOR -VERSION-	The indicated area, which is being used as the CDCS catalog file, must be of the following format: The area must contain only one record. The area must contain only two items. The first item must be the primary key and be 10 alphanumeric characters in length. The second item must be TL (transmission length) alphanumeric characters in length.	Specify the proper record format for the area.
413	FILE -a- DOES NOT EXIST	During modification of the permanent file parameters, a permanent file name that does not match the permanent file name of the subschema or any of the areas involved was specified.	Specify the correct file name.
414	FILE -a- MAY NOT HAVE PF PARAMS MODIFIED	If the CREATE, USE, INVOKE, or VERSION directive names a subschema in CDCS data base access mode, permanent file parameter modification is allowed only for the subschema file because CDCS obtains permanent file parameters for the area and index files from the master directory; the area and index file parameters cannot be modified by Query Update.	No action.
415	ILLEGAL NAME FOR DATABASE VERSION	The name specified in the FOR DATABASE clause of the CREATE, INVOKE, or VERSION directive does not neet the requirements of 1-7 characters long: A-Z, 0-9.	Change the name specified in the FOR DATABASE clause to conform to the requirements.
416	VERSION NAME MUST MATCH THAT OF CDCS CATALOG	If a VERSION directive has specified a database version name for the catalog, that version name remains in effect as long as the catalog does. Any attempt to specify a new version name by a CREATE or INVOKE directive is in error.	Specify a new version in a VERSION directive.
417	AMBIGUITY EXISTS. PLEASE QUALIFY -c-	A described item has been referenced which exists for more than one file. Its file name must also be supplied.	Describe the item further by specifying a file name in an OF file-name clause.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
418	-c- IS A DESCRIBED ITEM AND CANNOT BE RESET	A described item may not be the subject of a SETTING clause in the STORE, MODIFY, or REMOVE directives nor the destination of a MOVE.	Use a data base data name as the subject of the SETTING clause.
419	-AS- CLAUSE ILLEGAL WITHOUT -UPON- OPTION	On the DISPLAY directive, the use of the AS clause to rename an expression is not allowed unless an UPON file and its directory are being created.	Include an UPON file-name clause to qualify the displayed item.
420	KEYWORD -DATABASE- MUST APPEAR AFTER THE -FOR-	On a CREATE, INVOKE, or VERSION the FOR is optional, but if present, the keyword DATABASE must follow.	Either include the keyword DATABASE in the directive, or omit the word FOR.
421	FAILURE DURING SORT	SORT returned a fatal or catastrophic error. See dayfile for SORT error number.	See the dayfile for the SORT error number, then refer to the SORT/MERGE reference manual for an explanation and action.
423	CANNOT PRESET MORE THAN THREE LEVELS OF NESTING	Data is nested for more than three levels in the subschema.	Ask the data administrator to either remove levels of nesting in subschema or to insert values in the data base. Error is not fatal and causes no problem unless a display is requested of an item that has no value assigned.
424	LFN IS SAME AS <A> FILE NAME	The local file name matched the name of the permanent file attached by the system.	Change the local file name.
425	CANNOT USE FILE <c> AS BOTH SOURCE AND TARGET FILE	The same file cannot be used for both reading and writing. If the source file was not explicitly stated, it might have been assigned by Query Update as the result of a previous DISPLAY UPON directive.	Either return the file or qualify the data names.
426	-DIRECTORY- COMMAND NEEDS -ON-/OFF- OPTION SPECIFIED	The DIRECTORY command requires that either ON or OFF be specified.	Reenter the command including the ON/OFF option.
427	-DIRECTORY- COMMAND NEEDS THE -DISPLAY-/-EXTRACT- OPTION	The DIRECTORY command requires that either DISPLAY or EXTRACT, be specified.	Reenter the command including the directive option.
428	-LOOKUP- COMMAND NEEDS -DATABASE-/-FILES- OPTION SPECIFIED	The LOOKUP command requires that either DATABASE or FILES be specified.	Reenter the command including the data source option.
429	-LOOKUP- COMMAND NEEDS -FIRST-/-ONLY OPTION SPECIFIED	The LOOKUP command requires that either FIRST or ONLY be specified.	Reenter the command including the search order option.
430	KEYWORD MISSING IN DISPLAY/ EXTRACT -WITH DIRECTORY- CLAUSE	The WITH DIRECTORY clause of the DISPLAY and EXTRACT commands must contain at least the keywords WITH DIRECTORY.	Reenter the DISPLAY or EXTRACT command with proper syntax.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
503	EXTERNAL SCHEMA INITIALIZED BY INCOMPATIBLE VERSION OF IMF	The version of IMF used to initialize the external schema is not compatible with this version of Query Update.	Initialize the external schema using the compatible version of IMF. Reenter the INVOKE directive.
504	c n - I EXPECTED A PASSWORD LITERAL FOLLOWING c	The INVOKE directive contains an invalid specification for the PW or KEY option.	Correct the INVOKE directive to specify the correct password literal for the PW or KEY option.
505	INTERNAL ERROR a TRYING TO OPERN THE EXTERNAL SCHEMA	An internal error occurred during processing of this directive.	Follow site-defined procedures for reporting software errors or operational problems.
506	I DO NOT FIND ANY a BY THE NAME a	The directive specifies a data name, record name, access path name, or coset name that is not contained in the external schema.	Correct the directive to specify a valid external schema data name, record name, access path name, or coset name.
509	-c- IS ONE ENTRY TOO MANY.	The PRINT directive or the MOVE option of the MODIFY or STORE directive contains more than 20 names; a maximum of 20 names can be specified. The VIA option of the PRINT or WRITE directive contains more than 13 coset names; a maximum of 13 coset names can be specified.	Correct the directive to specify no more than 20 names or 13 coset names.
510	IMF ERROR CODE o, a a, TRYING TO a YOUR RECORD	Query Update is returning the information that IMF returned in the ERRCODE field of the IMF communication zone. The description of the error code, as represented in the ERROR-TYPE field of the communication zone of a COBOL program, is also given.	Consult section 3 of this manual for a detailed explanation of the error or status code returned in the ERRCODE field of the communication zone.
511	ERROR NUMBER o WHILE PROCESSING ENTRY NUMBER o	A conversion error has occurred while processing this directive.	Consult the Query Update reference manual for an explanation of the indicated error code and take appropriate corrective action.
512	INTERNAL ERROR TRYING TO PROCESS DIRECTIVE a	An internal error occurred during processing of this directive.	Follow site-defined procedures for reporting software errors or operational problems.
513	NAME EXPECTED AFTER c INSTEAD OF n	The PRINT or WRITE directive contains a invalid coset name in the VIA option, or the WRITE directive contains an invalid name in the AS option.	Correct the directive to specify a valid name.
514	DATA-NAMES OR EXPRESSIONS MUST FOLLOW THE DIRECTIVE	The initial keyword of the directive is followed immediately by a carriage return.	Correct the directive syntax and reenter the directive.
519	KEYWORD -LIBRARY- OR -METADB- MUST FOLLOW -IN- OR -FROM-	The INVOKE directive contains a word other than METADB after the keyword IN or FROM in the METADB specification.	Correct the directive syntax and reenter the directive.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
520	METABASE PERMANENT FILE CANNOT BE ATTACHED	The INVOKE directive contains an invalid permanent file name for the metabase in the METADB specification; or the file containing the metabase is missing, has an incorrect password, or is already attached.	Check the syntax of the INVOKE directive. If the syntax is incorrect, correct it and reenter the directive. If the syntax is correct, consult the information base administrator.
521	THE CONCEPTUAL SCHEMA HAS NOT BEEN CORRECTLY INITIALIZED	The conceptual schema specified in the INVOKE directive has not been initialized properly.	Inform the information base administrator that the conceptual schema has not been initialized properly.
522	THE EXTERNAL SCHEMA HAS NOT BEEN CORRECTLY INITIALIZED	The external schema specified in the INVOKE directive has not been initialized properly.	Inform the application administrator that the external schema has not been initialized properly.
523	THE CONCEPTUAL SCHEMA CANNOT BE FOUND	The INVOKE directive does not specify a valid conceptual schema name.	Correct the INVOKE directive to specify a valid conceptual schema name.
524	THE EXTERNAL SCHEMA CANNOT BE FOUND	The INVOKE directive does not specify a valid external schema name.	Correct the INVOKE directive to specify a valid external schema name.
525	THE PRIVACY KEY YOU SUPPLIED IS NOT THE CORRECT ONE	The literal specified in the KEY option of the INVOKE directive does not match the use privacy lock of the external schema.	Correct the INVOKE directive to specify the correct privacy key.
526	THE METABASE DOES NOT APPEAR TO BE CORRECT	The file address or the metabase table length is not valid.	Follow site-defined procedures for reporting software errors or operational problems.
527	I COULD NOT a YOUR RECORD BECAUSE OF CONSTRAINT o	The attempted update operation (store, modify, or remove) could not be executed because a constraint has been imposed in the conceptual schema on the record type to be updated.	Adjust the logic of the application to avoid the constraint violation.
528	THE NAVIGATION ROUTE LOOPS AT RECORD a	The same record type is accessed more than once by the specified navigation route.	Adjust the navigation route to avoid accessing the same record type more than once.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
529	THE NAVIGATION ROUTE DOES NOT BATCH RECORD a	The indicated record type cannot be accessed by the specified navigation route.	Adjust the navigation route to include the desired record type.
530	THE NAVIGATION ROUTE HAS A GAP AT RECORD a	The specified navigation route cannot access all desired record types because the route is broken at the indicated record type. The starting point of a particular coset has not been reached.	Adjust the navigation route to make it an unbroken linear path.
531	APPROPRIATE PERMISSION TO PROCESS RECORD a NOT GRANTED	The external schema record description for the desired record type does not allow the type of access required by this directive. The ordinal a refers to the relative position of the record type within the external schema. Ordinals are assigned in sequential order beginning with the value 1 for the first record type.	Ask the application administrator to change the external schema to allow the desired type to access, or change the application.
532	THE a DIRECTIVE REFERENCES MORE THAN ONE RECORD TYPE	The MODIFY, REMOVE, or STORE direcive contains names of data items from more than one record type. An update operation can be performed on a single record type only.	Correct the directive to specify names of data items from a single record type only.
533	THE SEARCH KEY c IS BEYOND THE 60TH ITEM	The search key named in the USING specification of the MODIFY or REMOVE directive is a data item declared for the record type.	Follow site-defined procedures for reporting software errors or operational problems.
535	THE NAMES AFTER -USING- DO NOT MAKE UP AN ACCESS PATH	The USING specification of the MODIFY or REMOVE directive does not contain all the search keys for the access path.	Correct the directive to specify all the search key names for the access path.
536	IMF MUST BE INVOKED IN ORDER TO EXECUTE CURRENT DIRECTIVE	The INVOKE directive, which must be specified before the current directive, has not been specified.	Enter the INVOKE directive and reenter the desired IMF directive.
537	-KEY- OPTION ILLEGAL WHEN ENTERING CRM/CDCS DATA BASE MODE	The INVOKE directive contains the LIBRARY option, which implies that the user is attempting to invoke CRM, or CDCS, and the KEY option, which implies that the user is attempting to invoke IMF.	Correct the directive to remove the LIBRARY option or the KEY option.
538	THE QU/IMF INTERFACE IS NOT AVAILABLE	An IMF directive has been entered, but the Query Update interface to IMF is not available.	Follow site-defined procedures for reporting software errors or operational problems. Query Update must be rebuilt with the symbol -IMF- defined.
539	THIS DIRECTIVE NOT SUPPORTED FOR IMF INFORMATION BASE ITEMS	This directive cannot be specified when using IMF. (The DELETE, EXTRACT, and INSERT directives are not allowed; the DISPLAY, EVALUATE, IF, MOVE, and UPDATE directives can be used only with temporary items.)	Use a valid IMF directive to perform the desired function.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
540	INVOKE DISALLOWED, INCOMPATIBLE VERSION OF IMF.	The version of IMF with which this version of Query Update was built is not at a level compatible with this level of Query Update. The INVOKE directive was disallowed because the correctness of the information base might have been jeopardized.	Follow site-defined proceudres for reporting software errors or operational problems. Query Update must be rebuilt with a compatible version of IMF.
	NO ACCESS PATHS EXIST	The EXHIBIT directive contains the ACCESS option, but IMF has not been properly invoked.	Enter the INVOKE directive and reenter the EXHIBIT directive.
	NO COSETS EXIST	The EXHIBIT directive contains the COSET option, but IMF has not been properly invoked.	Enter the INVOKE directive and reenter the EXHIBIT directive.
542	INFOBASE IS INVALID DUE TO INVALID CONSTRAINTS	The information base referenced in the INVOKE directive is not valid because it contains invalid constraints.	To invoke information base with invalid constraints, use the INVOKE FOR REPAIR directive.
543	REPAIR MODE INVALID FOR VALID INFOBASE	The information base referenced in the INVOKE FOR REPAIR directive is valid and may not be accessed in repair mode.	To invoke an information base with valid constraints, use the INVOKE directive.
544	-FOR REPAIR- IS INVALID FOR CRM/CDCS -INVOKE-	The INVOKE FOR REPAIR directive may only be used when accessing an IMF information base with invalid constraints.	To invoke a data base in either CRM or CDCS data base access methods, use the INVOKE directive.
545	ON AN -INVOKE-, THE KEYWORD -REPAIR- MUST FOLLOW -FOR-	The only valid syntax involving FOR immediately following INVOKE, is INVOKE FOR REPAIR. Both FOR and REPAIR are required with this form.	Include the word REPAIR after INVOKE FOR.
600	SOURCE DATA CONVERSION ERROR d FROM a PAGE d LINE d	The specified conversion error occurred while converting data into display code format during execution of the specified directive. Examine the directive and contents of the report and refer to specified conversion error for more information.	Examine the directive and contents of the report and refer to action for specified conversion error code.
601	CONVERSION ERROR EXCEEDS LIMIT	The number of errors found while generating a report has exceeded 61; generation of the report terminates.	Consult the data administrator.
602	SOURCE FILE NOT CLOSED BEFORE REPORT GENERATION	The file that contains the data for report generation was not closed before the PREPARE directive was executed.	Consult the data administrator.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
603	FIELD OVERLAP IN PAGE BUILD, INDICATED BY STARS	Report directives indicate that two or more sets of data are to be placed on the same location on the page. The overlapping field has been set to asterisks in the report page.	Reenter the report directives causing the overlap in the report.
604	OVERLAPPING FIELD WILL BE TRUNCATED IN DETAIL d	The indicated DETAIL directive contains specifications for a data item to be transferred to a report page in a location that is not long enough to contain that data item; the data is truncated.	Reenter the report directives specifying the layout of the report, or take no action if the truncated field is adequate.
605	OVERLAPPING FIELD WILL BE TRUNCATED IN HEADING d	The indicated HEADING directive contains specifications for a data item to be moved to a report page in a location that is not long enough for the data item; the data is truncated.	Reenter the report directives that caused the truncation in the report; or take no action if the truncated field is adequate.
606	OVERLAPPING FIELD WILL BE TRUNCATED IN FOOTING d	The indicated FOOTING directive contains specifications for a data item to be moved to a report page in a location that is not long enough for the data item; the data is truncated.	Reenter the report directives that caused the truncation in the report; or take no action if the truncated field is adequate.
607	OVERLAPPING FIELD WILL BE TRUNCATED IN TITLE	The TITLE directive for this report indicates that a data item is to be moved to a report page in a location that is not long enough for the data item; the data is truncated.	Reenter the report directives that caused the truncation in the report; or take no action if the truncated field is adequate.
608	OVERLAPPING FIELD WILL BE TRUNCATED IN RECAP	The RECAP directive for this report indicates that a data item is to be moved to a report page in a location that is not long enough for the data item; the data is truncated.	Reenter the report directives that caused the truncation in the report; or take no action if the truncated field is adequate.
609	OVERLAPPING FIELD WILL BE TRUNCATED IN DETAIL	The DETAIL directive for this report indicates that a data item is to be moved to a report page in a location that is not long enough for the data item; the data is truncated.	Reenter the report directives that caused the truncation in the report; or take no action if the truncated field is adequate.
610	INTEGER ITEM SUBSCRIPT OUT OF BOUND	Retrieval of data from a matrix or an occurring item during report generation has led to the condition where the subscript for the matrix or occurring item is out of bounds.	Consult the system analyst or data administrator. Correct the usage of the integer item to maintain a valid index.
611	SYSTEM ERR-DTLBIT TBL OVERFLOW IN PAGE	This is an internal error. It is generated by specification of a large number of vertical sections for a block on a report page that contains many lines between a heading and footing.	Follow site-defined procedures for reporting software problems.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
612	WARNING - MORE THAN 1 DETAIL FORMAT IN BLOCK WITH ONCE	In generation of a report, a detail block contained more than one DETAIL directive, and one of these directives specified the ONCE option.	Revise the DETAIL directive so the ONCE option is not used, or revise the report specifications to avoid mixing other detail formats within that block of detail line occurrences (perhaps by the inclusion of a HEADING or a FOOTING to end one detail block and start another).
613	DATA-NAME (ALL) EXCEEDS SECTION WIDTH. LINE TRUNCATED	Specification of the figurative subscript ALL in a report directive has led to a condition where the data to be written on the report page exceeds the number of columns that are available in the section; the line is truncated.	Reenter the report directive that used the figurative subscript ALL.
614	SPECIFIED a CAUSES FIELD TO EXTEND BEYOND PAGE SIZE	The column defined for the tab number or the column specified causes the field to extend beyond the last column of the page.	Reenter the directive specifying a column that allows printing of the entire field or redefine the tab value.
615	PAGE SIZE TOO NARROW FOR DEFAULT POSITION OF a IGNORED	The current page size for the report is not able to accommodate the default position of the report directive.	Adjust the page size to accommodate the field.
616	a EXTENDS BEYOND PAGE LENGTH	A combination of headings is too large to fit on a page; the specified heading is ignored.	Shorten the headings to use fewer lines or specify a larger PAGE-SIZE.
617	-c- INVOLVES AREA-ITEM	A defined item involving an area item cannot be evaluated during report preparation.	Erase and redefine the item without using area-items in its value expression. Then reenter the PREPARE, COMPILE, or PREVIEW directive.
618	DETAIL WITH MULTIPLE -AT LINE- INVALID WITH VERTICAL SECTIONS	The DETAIL directive cannot specify more than one AT LINE clause when the PAGE-SIZE directive specifies vertical sections.	Reenter the DETAIL directive with only one AT LINE clause or reenter the PAGE-SIZE directive without specifying vertical sections.
619	CONDITION IN FOLLOWING RPT DIRECTIVE INVOLVES AN AREA ITEM	All data items referenced in a report directive must be previously described, extracted from the source data file, or defined. If the condition for the indicated directive contains a SPECIFY data name, all items referenced in the SPECIFY condition must be previously described, extracted from the source data file, or defined.	Erase the report directive; reenter the directive and use only data items that have been previously described, extracted from the source file, or defined. Reenter the PREPARE, PREVIEW, or COMPILE directive.
620	TOO MANY NESTED PREFACE-SUMMARY ENTRIES	A maximum of 10 PREFACE/SUMMARY entries is allowed. The name printed at the end of the message caused the table to overflow.	Describe fewer PREFACE or SUMMARY entries for the report.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
800	DUPLICATE KEY ON INSERT	In the INSERT or STORE directive, the key of the record that is to be inserted already exists on the area file.	Check to see if the intended key was input.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
801	FROM FILE ALLOWED ONLY WITH USING CLAUSE	The USING clause in an INSERT, UPDATE, or DELETE directive must be specified if the FROM clause is specified.	Reenter the directive including the USING clause if the FROM clause is specified.
802	UNKNOWN KEY ON DELETE	In the DELETE or REMOVE directive, the key of the record that is to be deleted does not exist on the area file.	Check to see if the intended key was input.
803	SUBSCHEMA a NOT IN FILE a	The indicated subschema does not exist in the indicated subschema library.	Specify a valid subschema name.
804	ERROR o TRYING TO READ SUBSCHEMA a	The indicated error status was returned by one of the subschema directory access routines.	Consult the data administrator.
805	ERROR FROM KEY CONVERSION	Conversion of the value to be used as the key for an INSERT, UPDATE, DELETE, or DISPLAY KEY directive has led to a conversion error.	Consult the data administrator.
806	QU ERROR - ENTERED FILE ACCESS OVERLAY WITH NOTHING TO DO	This is an internal Query Update error.	Follow site-defined procedures for reporting software problems.
807	FATAL ERROR - CANT SEARCH FILE AND HAVE -USING- PHRASE	In a transmission containing directives that apply to the area file, one UPDATE or DELETE directive contains the USING data-names clause. Another directive (such as a MOVE or an IF directive) indicates that the entire area file must be searched to satisfy the directive. It is illegal to have these two types of processing in one transmission, the transmission is ignored.	Reenter only one of these directives.
808	CONVERSION ERROR FIELD d	A conversion error has occurred for the specified field while processing an INSERT, UPDATE, or DELETE directive with the USING clause. Conversion errors normally result from conversion of any type variable to logical or from conversion of character literals to numeric variables.	Check the input data for valid content.
809	USING TABLE OVERFLOW - EXCESS DATA	Data items entered while processing the USING clause of an INSERT, UPDATE, or DELETE directive exceed the data list in the directive.	Reenter the INSERT, UPDATE, or DELETE directive with fewer data items in the USING clause.
810	LITERAL TOO LONG FOR STORE	A literal entered while processing the USING clause of an INSERT, UPDATE, or DELETE directive is too long for Query Update to store it. A maximum of 255 characters is allowed.	Reenter the directive using a valid number of characters for the literal.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
811	UNBALANCED SEPARATOR	An odd number of delimiters has been found while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive.	Reenter the directive with an even number of delimiters.
812	REMAINING INPUT DISCARDED	An error occurred while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive. The input after the error is ignored.	Reenter the directive.
813	FIELD d CANNOT BE ENCLOSED IN SEPARATORS	Delimiters enclosed a field that should not have been enclosed in separators. This error occurs while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive.	Reenter the directive with delimiters appropriately placed.
814	FIELD d MUST BE ENCLOSED IN SEPARATORS	Delimiters did not enclose a a field that must be enclosed in separators. This error occurs while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive.	Reenter the directive with delimiters appropriately placed.
815	UPDATE/INSERT (W/O USING) SHOULD PRECEDE MOVE TO AREA-ITEM	When the destination field of the MOVE directive is a data name which is defined in an area, precede the MOVE directive with an UPDATE directive or an INSERT directive (either alone or with the KEY or literal option); otherwise, the MOVE occurs but the area file is not updated.	Enter either an UPDATE or an INSERT directive; then reenter the MOVE directive.
816	UNKNOWN KEY	The key specified for directive does not exist on the area file.	Check to see if the intended key was input.
817	KEY LITERAL MISSING	The literal that corresponds to the key data name cannot be found. This area occurs while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive.	Add missing keyword and reissue transmission.
818	INPUT LESS THAN ITEM SIZE	The data entered contains fewer characters than the number of characters allocated for the data item. This error occurs while processing input for the USING clause of an INSERT, UPDATE, or DELETE directive.	Reenter the directive with the expected number of characters for the data item.
819	ERROR o TRYING TO OPEN FILE a	An attempt has been made to open a specified file, and CYBER Record Manager returned the indicated error code.	Refer to the appropriate CYBER Record Manager reference manual for further information.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
820	IF -KEY- REQUESTED AREA ITEM MUST BE REFERENCED	At least one area item must be referenced when the KEY option is specified in the directive, or all of the items are temporary items.	Reenter the directive referencing at least one area item.
821	OTHER USER UPDATING -a- OR NOT CLOSED SINCE LAST UPDATE	This error can be caused by either of the following: Attempted concurrent use of the indicated file while another user is updating the file. The file not being closed since the last update.	Refer to the CYBER Record Manager reference manual for information on error 52B.
822	FILE -a- NOT CLOSED SINCE LAST UPDATE	The indicated file has not been closed since the last update.	Refer to the CYBER Record Manager reference manual for information on error 52B.
830	EMPTY SOURCE FILE OR NO RECORDS QUALIFIED	The source file for the report contains either no records or no records that satisfy selection criteria.	Check to see if the file name is valid.
850	USE/INVOKE/CREATE DIRECTIVE CANNOT BE PROCESSED	Due to input/output errors or subschema problems, processing of the USE, INVOKE, or CREATE directive is terminated.	Reenter the USE, INVOKE, or CREATE directive.
851	ENTRY POINT a NOT FOUND	The subschema for the area file specified in the USE directive indicates that data base procedures are to be loaded for the area file. The specified entry point cannot be found.	Consult the data base administrator; check to see if data base procedures are provided.
852	INDEX FILE NOT SPECIFIED	The USE or CREATE directive and the subschema do not specify an index file for an area that has multiple keys. A USE or CREATE directive must be issued in which the INDEX option is used to specify the name of the index file and the associated permanent file parameters.	Add index file to USE or CREATE directive, or to the AREA-NAME clause in the subschema if specifying subschema or relation.
855	WAITING d SECONDS	Query Update waits the specified number of seconds before reattempting the operation that failed.	No action.
861	INCORRECT SUBSCHEMA NAME	The name that is on the subschema file is not the same as the permanent file name for the subschema. This indicates an error in the subschema.	Check that the correct subschema name was used, or perform the DDL run to generate the subschema.
862	-a- NOT FOUND IN THIS SUBSCHEMA	The USE directive specified an area name or a relation name that is not described in the subschema.	Check to see if the intended area name or relation name was input.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
863	CURRENT SUBSCHEMA CONTAINS INCONSISTENT DATA	An error exists in the subschema.	Perform a new DDL run to regenerate the subschema.
869	USE/INVOKE/CREATE DIRECTIVE REJECTED	The USE, INVOKE, or CREATE directive processing is terminated because of errors in the processing of the subschema or the area file.	Consult the data administrator. Reinput the USE, INVOKE, or CREATE directive.
872	NO READ PERMISSION FOR AREA FILE, USE REJECTED	The area file does not have read permission. The permanent file password that grants read permission for the area file must be specified in the USE directive.	Input the USE directive specifying the permanent file password that grants read permission for the area file.
873	NO MODIFY PERMISSION FOR AREA	In the USE directive for this area, the permanent file password that grants modify permission has not been specified.	Input the USE directive specifying the permanent file password that grants modify permission.
874	NO MODIFY/EXTEND PERMISSION WAS GRANTED FOR THIS AREA	In the USE directive for this area, the permanent file passwords that grant modify and extend permission have not been specified.	Reenter the USE directive specifying the permanent file passwords that grant modify and extend permission.
875	FILE a CONTAINS NO DATA AND CANNOT BE OPENED	The indicated area contains no data records and cannot be opened for read-only mode.	No action.
900	REPORT TOO LARGE TO COMPILE	The report tables for this report exceeded the buffer allocated by Query Update. Erasing unnecessary items or describe lists can permit this report to compile. The PREPARE directive can be used to generate reports without using the COMPILE directive and REPORT utility.	Eliminate unnecessary items or lists and reenter the COMPILE directive, or use the PREPARE directive to generate the report.
901	JOB FL OR QU INTERNAL STORAGE EXCEEDED	The maximum field length for Query Update is not large enough to execute the requested directives.	Increase the maximum allowable field length.
902	P.F.M. ERROR o ON FILE a	The indicated permanent file manager error has occurred from a permanent file request on the specified file.	Refer to the operating system reference manual for further information.
903	I/O ERROR o FILE a	Certain input/output errors cannot be attributed by the system to some syntax error or some error in the logic of the user request. The system displays the error code returned by CYBER Record Manager.	Refer to the appropriate CYBER Record Manager reference manual for a detailed explanation of a specific error number.
904	CRM/CDCS ERROR o FILE/RELATION a FUNCTION a	Query Update is reporting the information that CDCS returned in the data base status block.	Refer to the CDCS reference manual for a detailed explanation of the indicated error number.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
905	INTERNAL ERROR DETECTED BY a	This is a Query Update system error.	Follow site-defined procedures for reporting software problems.
906	FUNCTION PARAMETER FOR JULIAN IN ERROR	The parameter for the JULIAN function must have at least five characters.	Reenter the directive with the parameter for the JULIAN function having at least five characters.
907	NUMBER OF EVALUATED ITEMS EXCEEDS LIMIT	The number of items specified in the list to be evaluated exceeds the limit allowed in Query Update.	Use a second EVALUATE directive for the excess items.
908	CDCS ERROR o FILE a FUNCTION a ITEM d	Query Update is reporting the information that CDCS returned in the data base status block.	Refer to the CDCS reference manual for a detailed explanation of the indicated error number.
909	SYSTEM ABORT DUE TO A TERMINAL INTERRUPT	A terminal interrupt was encountered. Query Update must terminate its operations.	Your files should be intact when the system recovers. Reenter QU when the system recovers.
913	INVALID DATA ENCOUNTERED DURING CONVERSION	Invalid data (such as a character item in a numeric field, or numeric or other items in a logical field) has been encountered during the conversion process.	Check data for validity.
914	USER CATALOG UNUSABLE	The catalog specified in the VERSION directive is unusable.	Consult the data administrator.
917	DATA BASE PROCEDURE RETURN CODE OUT OF RANGE	The value for a data base procedure return code can only be 0, 1, 2, or 3.	Correct the data base procedure so that it returns only 0, 1, 2, or 3.
918	CURRENT DIRECTIVE TERMINATED BY DATA BASE PROCEDURE	Processing of a user data base procedure for the area file generated a return code indicating that the current directive should be terminated.	No action.
919	AREA CLOSED BY DATA BASE PROCEDURE	Processing of a user data base procedure for the area file generated a return code indicating that the area should be closed.	No action.
920	RECORD IGNORED BY DATA BASE PROCEDURE	Processing of a user data base procedure for an update generated a return code indicating that the record should be ignored.	No action.
924	SESSION TERMINATED BY DATA BASE PROCEDURE	Processing of a user data base procedure for the area file generated a return code indicating that the entire Query Update session should be terminated; in effect, an END directive is issued.	No action.
933	SINGLE ITEM CANNOT BE SUBSCRIPTED	Only matrix items can be subscripted.	Reenter the item without the subscript.
934	-NEXT- IS FOR DEPEND ON ITEM ONLY	The figurative subscript NEXT can be used only for a matrix whose upper bounds depend on a data name.	Reenter the transmission with a valid subscript.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
935	-LAST- IS FOR DEPEND ON ITEM ONLY	The figurative subscript LAST can be used only for a matrix whose upper bounds depend on a data name.	Reenter the transmission with a valid subscript.
936	-ANY- IS FOR CONDITION ONLY	The figurative subscript ANY can be used only in conditional expressions.	Reenter the transmission with a valid subscript.
937	ONE FIGURATIVE SUBSCRIPT PER CONDITION	Only one of the figurative subscripts (LAST, ANY, NEXT, or ALL) can be used in a conditional expression.	Reenter the transmission with a valid subscript.
938	-NEXT- IS NOT FOR CONDITION OR EXPRESSION	The figurative subscript NEXT cannot be used in a conditional or arithmetic expression.	Reenter the expression without the subscript NEXT.
940	NO FIGURATIVE SUBSCRIPT AFTER -ANY- OR -ALL-	No figurative subscript can follow the subscript ANY or ALL in an expression.	Reenter the expression without the extra figurative subscripts.
941	SUBSCRIPT OUT OF BOUNDS	The subscript for an item in a matrix was less than one, or has exceeded the limits declared for the matrix.	Reenter the transmission with a valid subscript.
942	-NEXT- EXITS UPPER BOUND	The subscript generated by use of of the figurative subscript NEXT has exceeded the limits declared for the matrix.	Reenter the transmission with a valid subscript.
943	-NEXT- IS ONLY FOR DESTINATION ITEM	The figurative subscript NEXT can be used only for the destination item in a MOVE directive.	Reenter the transmission with a valid subscript.
944	INVALID USE OF FIGURATIVE SUBSCRIPT	The only legal subscripts are an integer numeric literal, an integer type item, or one of the figurative subscripts (NEXT, LAST, ANY, or ALL).	Reenter the transmission with a valid subscript.
945	DIVISION BY 0 ILLEGAL	An expression that results in a division by zero is illegal.	Reenter the expression with an appropriate divisor.
946	SYSTEM ABORT - MEMORY REQUEST ERROR	This is an internal Query Update error.	Follow site-defined procedures for reporting software problems.
947	INSUFFICIENT FL - TRANSMISSION IGNORED	The last directive request was not performed because of insufficient field length.	Reduce the current field length by using the RETURN and ERASE directives to dispose of unneeded resources.
948	ILLEGAL ARGUMENT FOR EXPONENTIATION	An expression of the form X raised Y is illegal if X is a negative floating number, a negative double precision number, or a complex number.	The expression must be modified to eliminate the exponentiation or make the base a nonnegative floating or double precision number.
949	a CAUSED ERROR AFTER LINE d PAGE d	This message follows any diagnostic encountered during report generation in order to specify which directive caused the error after the specified line was printed.	Examine directive and contents of report, and refer to the action indicated for the preceding error code.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
950	-c- PARAMETERS MUST BE OF CONSISTENT DATA LENGTH	In any one use of the MASK function, the parameters must have identical lengths. The length of a parameter of type CHARACTER or NUMERIC is equal to the number of characters specified in its picture specification or assumed by default. The length of a parameter of type DOUBLE or COMPLEX is two computer words. The length of a parameter of any other data type is one computer word.	Reenter the transmission with correct function parameters.
951	DUPLICATE TABULAR POSITIONS ARE IGNORED	The integers entered in a TABS directive should be unique; duplicates are ignored.	No action.
none	NO MORE TRANSMISSIONS ALLOWED	The number of transmissions that can be recorded in the current session with the current increment size has been met. The recording of that session is terminated. The next transmission will be rejected.	Examine the increment used for a possible problem.
none	END OF SESSION a	This informative message acknowledges the end of the indicated recording session after RECORDING OFF has been specified.	No action.
none	d TRANSMISSION(S) ALLOWED	When the user adds to an existing catalog with the RECORDING directive, Query Update calculates the number of transmissions that can be recorded using the current increment size, and issues this message.	No action.
none	d FORCED d REJECTED	This message is issued following data base procedure processing of ON MATCH and ON MISMATCH conditions. In the case of ON MATCH, it follows the message d ACCESSES d HITS d IO-S and gives the number of accepted records that did not meet IF requirements and the number of records rejected by the procedures even though an IF was satisfied.	No action.
none	CANT GET -a- SHALL WE TRY AGAIN	If an area, index file, subschema, or independent sequential file cannot be attached because it is in use, Query Update issues this message followed by the next message below, which requests a yes or no answer.	No action.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
none	ANSWER Y or N	Query Update prompts the user for a yes or no answer in these circumstances: a file requested cannot be attached, and the message listed above is output; and after the first screen of output has been displayed following a DISPLAY directive, when more data can be displayed.	Input a Y or N.
none	d ACCESSES, d HITS, d IO-S	When a query accesses a single area, the number of ACCESSES indicates the number of single records. When a query accesses areas through a relation, the number of ACCESSES indicates the number of projected records. The number of HITS indicates the number of accesses that satisfied the query or update criteria. The number of IO-S indicates the number of logical records read from or updated in an area file.	No action.
none	INDEX FILE NOT SPECIFIED	The USE or CREATE directive does not specify an index file for an area that has multiple keys. For the USE directive, records can be read but cannot be updated. For the CREATE directive, an index file must be specified; the directive is not processed.	No action.
none	CURRENT DIRECTIVE TERMINATED	Processing of the current directive is terminated due to problems encountered while attempting to attach the index file.	Reenter the directive
none	NO RECORD QUALIFIED	The condition specified in the IF directive involves alternate keys or relation processing; no record satisfies the condition.	No action.
none	NO RELATION EXISTS	Areas must be joined in a relation if more than one area is used in a single query.	Consult the system analyst to determine if a relation should be set up.
none	MORE THAN ONE RELATION EXISTS FOR THIS QUERY PLEASE SPECIFY WHICH OF THE FOLLOWING SHOULD BE USED	When an ambiguity exists in a relation query, Query Update issues two messages followed by a list of the relation names in the current subschema. The user must enter one of the relation names from the list following the symbols prompted by Query Update.	Enter one of the relation names from the list, following the prompt symbols.
none	DURING PROCESSING OF DATABASE PROCEDURE a	This message is used to identify the cause of an abort or abnormal condition. Otherwise, the user would interpret the condition as an error in Query Update.	Consult the data administrator.

TABLE B-1. DIAGNOSTIC MESSAGES (Contd)

Error Code	Message	Significance	Action
none	d FATAL DIAGNOSTIC(S)	The number of fatal diagnostics issued during the preceding Query Update session.	Correct the fatal errors, and resubmit the Query Update session.
none	WILL YOU ACCEPT THE FILE - ANSWER Y OR N	A file has been opened that has not been closed since the last update. File contents can be inaccurate; the interactive user has the option of reading the file.	Enter Y to accept the file and continue processing; enter N to immediately close the file.
none	FILE WAS ACCEPTED AND PROCESSING WILL CONTINUE	The user has elected to continue processing on a file that was not closed since the last update.	No action.
none	SHALL WE RETRY THE CDCS REQUEST - ANSWER Y OR N	CDCS is not available at a system control point, CDCS has locked the requested record, or another job has exclusive access to the requested area or data base procedure library.	Enter Y to retry the CDCS request; enter N to cancel the request.
none	NO ACCESS PATHS EXIST	The EXHIBIT directive contains the ACCESS option, but IMF has not been properly invoked.	Enter the INVOKE directive and reenter the EXHIBIT directive.
none	NO COSETS EXIST	The EXHIBIT directive contains the COSET option, but IMF has not been properly invoked.	Enter the INVOKE directive and reenter the EXHIBIT directive.

Alphabetic Type -
Description of a data item that restricts its contents to a value represented by alphabetic characters.

Alphanumeric Type -
Description of a data item that restricts its contents to a combination of a value represented by characters 0 to 9 and alphabetic characters.

Alternate Key -
A key, other than the primary key, by which an indexed sequential, direct access, or actual key file can be accessed.

Area -
Uniquely named data base subdivision that contains data records; a file.

Area-Name -
A name assigned to a mass storage area used by Query Update; file name.

Beginning-of-Information (BOI) -
CYBER Record Manager defines beginning-of-information as the start of the first user record in a file. System-supplied information, such as an index block or control word, does not affect beginning-of- information. Any label on a tape exists prior to beginning-of-information.

Blocks -
The term block has several meanings depending on context. On tape, a block is information between interrecord gaps on tape. CYBER Record Manager defines several blocks depending on organization. (See Table C-1).

TABLE C-1. BLOCK TYPES

Organization	Blocks
Indexed sequential	Data block; index block
Direct access	Home block; overflow block
Actual key	Data block
Sequential	Block type I, C, K, E

Break (Page Break) -
The point during preparation of a report page when headings and/or footings are to be inserted.

Catalog -
A file containing a collection of directives that can be used as specifications for Query Update operations.

Concurrency -
Simultaneous access to the same data in a data base by two or more applications programs during a given span of time.

Condition -
One of a set of specified values for which a data item can be tested.

Condition-Name -
Name assigned to a specific value, set of values, or range of values, for which a data item can be tested.

Control Word -
A system-supplied word that precedes each W type record in storage.

CYBER Database Control System (CDCS) -
The DMS-170 controlling module that provides the interface between an applications program and a data base.

CYBER Record Manager -
A generic term relating the common products AAM and BAM, which run under the NOS and NOS/BE operating systems and which allow a variety of record types, blocking types, and file organizations to be created and accessed. The execution time input/output of COBOL, FORTRAN, Sort/Merge 5, ALGOL, and the DMS-170 products is implemented through CYBER Record Manager. Neither the input/output of the NOS and NOS/BE operating systems nor any of the system utilities such as COPY or SKIPF are implemented through CYBER Record Manager. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

Data Administrator -
A person who defines the format and organization of the data base through the specification of the sub-schema.

Data Base -
A systematically organized, central pool of information; organization is described by a specific subschema.

Data Base Procedure -
A special-purpose routine that performs a predefined operation; its use is specified in the subschema.

Data Name -
Name assigned to a data item; can contain up to 30 letters, digits, or embedded hyphens, but must contain at least one letter.

Deadlock -
A situation that arises in concurrent data base access when two or more applications programs are contending for a resource that is locked by one of the other ones, and none of the programs can proceed without that resource.

Delimiter -
A special character (the separator character) which is used to bound nonnumeric literals (character strings, octal literals, mask literals).

Direct Access File -
In the context of CYBER Record Manager, a direct access file is one of the five file organizations. It is characterized by the system hashing of the unique key within each file record to distribute records randomly in blocks called home blocks of the file.

In the context of NOS permanent files, a direct access file is a file that is accessed and modified directly, as contrasted with an indirect access permanent file.

Directive -
A Query Update command that describes an operation to be performed; consists of the reserved words of the Query Update language in combination with user-supplied elements.

End-of-Information (EOI) -
CYBER Record Manager defines end-of-information in terms of the file organization and file residence, as shown in table C-2.

TABLE C-2. END-OF-INFORMATION BOUNDARIES

File Organization	File Residence	Physical Position
Sequential	Mass storage	After the last user record.
	Labeled tape in SI, I, S, or L format	After the last user record and before any file trailer labels.
	Unlabeled tape in SI or I format	After the last user record and before any file trailer labels.
	Unlabeled tape in S or L format	Undefined.
Word Addressable	Mass storage	After the last word allocated to the file, which might be beyond the last user record.
Indexed Sequential, Actual Key	Mass storage	After the record with the highest key value.
Direct Access	Mass storage	After the last record in the most recently created overflow block or home block with the highest relative address.

Figurative Subscript -
A word that indicates a relative element in a matrix.

File -
A collection of records treated as a unit.

File-Name -
Name assigned to an input or output file; can contain one to seven letters and digits, the first of which must be alphabetic.

Floating-point Literal -
A string of digits with a decimal point and an optional exponent.

Footing -
Lines of print that comprise a caption and occur after a break.

Function -
Procedure that yields a result.

Heading -
Lines of print that comprise a caption and occur after a break.

Information Management Facility (IMF) -
A data modeling tool that supports preliminary analysis of information and uses a three-schema approach.

Item -
A character or set of contiguous characters stored as a data unit within a record.

Join Terms -
The identifiers (data items) that are used to join two files in a relation.

Key -
One or more data items, the contents of which identify the type or location of a record, or the sequence of data.

Layout -
Arrangement and structure of information on a printed page, record, or file.

Literal -
A constant completely defined by its own identity.

Logging -
The facility through which historical records are kept of operations performed by users on data base areas. Logging information is used in data base recovery and restoration operations.

Logical Operator -
Word defining the logical connections between relational operators.

Major Alternate Key -
The leading portion of an alternate key field.

Major Key -
Leading characters of a symbolic key in an indexed sequential file.

Major Primary Key -
The leading portion of a primary key field.

Mass Storage -
Storage of a large amount of data on a device having a large capacity, such as a magnetic disk; data is readily available to the central processing unit of a computer.

Numeric Type -
Description of a data item that restricts its contents to a value represented by characters from 0 to 9. The item described can also contain a leading or trailing operational sign. Query Update recognizes NUMERIC, COMPLEX, DOUBLE (double-precision), FLOATING (floating-point), and INTEGER numeric types.

Permanent File -
Mass storage file cataloged by the system so that its location and identification are always known to the system. Permanent files cannot be destroyed accidentally during normal system operation. Controls over file access and mode of use are provided to define various degrees of access.

Picture -
Description of the general characteristics and editing requirements of a data item.

Primary Key -
Key that must be defined for indexed sequential, direct access, or actual key file when the file is first created.

Random File -
In the context of CYBER Record Manager, a file with word-addressable, indexed-sequential, direct-access, or actual-key organization in which individual records can be accessed by the values of their keys.

In the context of the NOS or NOS/BE operating systems, a file with the random bit set in the file environment table in which individual records are accessed by their relative PRU numbers.

Record -
CYBER Record Manager defines a record as a group of related characters. A record or a portion thereof is the smallest collection of information passed between CYBER Record Manager and a user program. Eight different record types exist, as defined by the RT field of the file information table.

Other parts of the operating systems and their products might have an additional or different definition of record.

Record Type -
The term record type can have one of several meanings, depending on the context. CYBER Record Manager defines eight record types established by an RT field in the file information table. Tables output by the loader are classified as record types (such as text, relocatable, or absolute), depending on the first few words of the tables.

Recovery -
A process that makes a data base useful after some type of software or hardware failure has occurred.

Relation -
The logical structure formed by the joining of records based on common identifiers.

Relational Data Base -
A data base of files joined in relations through data item identifiers.

Schema -
A detailed description of the internal structure of a data base.

Session -
Series of transmissions sent by a user between the sign-on and the sign-off messages sent by Query Update. Each session or portions thereof can be recorded at the user's option, so that the directives can be re-executed at a later time without being reentered.

Session ID -
The 6-character session identifier assigned by the user.

Source Data -
Data records used for Query Update operations.

Source Word -
One of the recognized elements of Query Update syntax: names, literals, keywords, function names, special characters.

Subschema -
Plan or outline described with Data Description Language statements regarding names and characteristics of data items, records, areas, and relationships that must be maintained among data base elements.

Subschema Library -
A permanent file containing one or more subschemas.

Subscripting -
Use of an integer or variable to identify a particular element in a table.

Syntax -
The structure of directives in Query Update.

Temporary Items -
Non-contiguous data items that are developed and processed internally and are not part of external files, but reside in working storage.

Transmission -
One or more directives submitted as a unit; can be continued from one line to the next (interactive mode) or from one card to the next (batch mode) by entering a continuation character.

Transmission ID -
The 3-digit system-supplied identifier assigned to one or more directives in a session catalog.

W Type Record -
One of the eight record types supported by CYBER Record Manager. Such records appear in storage preceded by a system-supplied control word. The existence of the control word allows files with sequential organization to have both partition and section boundaries.

Working Storage -
Describes noncontiguous data items that are not part of external files, but are developed and processed internally; temporary storage.

Zero-Byte Terminator -
Twelve bits of zero in the low order position of a word that marks the end of line to be displayed at a terminal or printed on a line printer. The image of cards input through the card reader or terminal also has such a terminator.

SUMMARY OF QUERY UPDATE DIRECTIVES D

ACCESS

```
                                                 ┌     ┌                                 ┐         ┐
                ┌             ┐ ┌     ║INPUT ║ ┐ │     │ AREA area-name . . .          │         │
ACCESS KEY      │ IS {literal   } │ ON  ║I-O   ║ │ │ FOR { CATALOG [catalog name]     } │ . . .   │ . . .     4-1
                │    {data-name } │     ║OUTPUT║ │ │     │ AREAS                         │         │
                └             ┘ └              ┘ └     └                                 ┘         ┘
```

ALTER

```
ALTER report-name                                                                                    4-2
```

BREAK

```
                                      ( OVERFLOW        )
BREAK level-number [AND EJECT] ON    { ITEM data-name    }                                          4-2
                                      ( condition       )
```

COMPILE

```
COMPILE report-name [DESCRIBED IN file-name] UPON table-file                                         4-3
```

CREATE

CDCS Format

```
    CREATE area-name OF subschema-name [FROM LIBRARY permanent-file-name]                            4-4
                   [( permanent-file-parameters [PW] )] [FOR DATABASE version-name]
```

CRM Format

```
    CREATE area-name [INDEX IS index-file-name] OF subschema-name
                   [FROM LIBRARY permanent-file-name] [( permanent-file-parameters  [PW] )]
```

DATE

```
       ┌      ┌              ┐  ┌                 ┐ ┐
       │      │ TITLE-LINE   │  │                 │ │
DATE   │ AT   │ LINE integer-1│  │ TAB integer-2   │ │ . . .                                          4-5
       │      │ RECAP-LINE   │  │ COLUMN integer-3│ │
       └      └              ┘  └                 ┘ ┘
```

DEFINE

```
DEFINE [integer ITEMS [DEPENDING ON data-name-1] FOR]                                                4-5

                  ┌     ( INTEGER    ) ┐
                  │     │ FLOATING   │ │
                  │     │ DOUBLE     │ │
    data-name-2   │ AS {  COMPLEX     }│ [BY picture-specification] [{ VALUE IS } expression]
                  │     │ LOGICAL    │ │                             [{ =        }           ]
                  │     │ CHARACTER  │ │
                  └     ( NUMERIC    ) ┘
```

DESCRIBE

```
                        (                                                                      4-7
DESCRIBE file-name <  { AND } [integer-l ITEMS [DEPENDING ON data-name-l] FOR]
                        { OR  }
                        (
                                 ( FIXED     )
                                 | INTEGER   |
                                 | FLOATING  |                          )
      {FILLER     }              | DOUBLE    |                          |
      {data-name-2} AS [DISPLAY] < COMPLEX   > BY picture-specification > . . .
                                 | LOGICAL   |                          |
                                 | CHARACTER |                          )
                                 ( NUMERIC   )
```

DETAIL

```
                        (                                                              4-9
DETAIL [tag-number]   <  [AT LINE integer-1 [BEYOND] ]
                        (
                                                                     )
     { IS expression [ONCE] [CENTERED] [ IN  { NEXT TAB         } ] } . . .  > . . .
     {                                 [     { TAB integer-2    } ] }        |
                                       [     { COLUMN integer-3 } ] }        )
```

DIAGNOSTIC

```
DIAGNOSTIC { PART }                                                            4-10
           { FULL }
```

DIRECTORY

```
DIRECTORY { ON  }  FOR  { DISPLAY }                                            4-11
          { OFF }       { EXTRACT }
```

DISPLAY

```
                                                      [        ( literal         ) ]
DISPLAY [ UPON file-name-1 [WITH DIRECTORY] [ ON  ] ] [ KEY†  < data-name-1      > ]   4-11
                                            [ OFF ]   [        ( IN file-name-2  ) ]
                                                      [                            ]
                                                      [ FROM file-name-3           ]

    { {expression [AS data-name-2]} . . . }
    { SAME                                }
```

†Not applicable to IMF

DUPLICATE

```
DUPLICATE UPON DEFAULT                                                         4-12

    ( REPORT report-name-1 [AS report-name-2]                                  )
    |                                                                          |
    | SESSION session-id-1 [transmission-id-1 [TO transmission-id-2] ]         |
    |                     [AS session-id-2 [transmission-id-3] ]               |
    (                     [BY increment-size]                                  )
```

END

```
END                                                                       4-13
```

ERASE

```
        ( data-name . . .                                              )
ERASE   ( REPORT report-name {ALL             }                        )  4-13
        (                    {layout-directive}                        )
        ( SESSION session-id {ALL                                     }  )
        (                    {transmission-id-1 [TO transmission-id-2]}  )
        ( DESCRIBE file-name                                           )
```

EVALUATE

```
EVALUATE {data-name          } . . .                                      4-14
         {cumulative-function}
```

```
                   ( REPORT                )
                   ( TITLE                 )
                   ( RECAP                 )
                   ( ANY SELECT            )
                   ( SELECT tag-number     )
                   ( ANY DETAIL            )
EVALUATE {BEFORE}  ( DETAIL tag-number     )  {data-name          } . . .  4-14
         {AFTER }  ( ANY BREAK             )  {cumulative-function}
                   ( BREAK level-number    )
                   ( ANY HEADING           )
                   ( HEADING level-number  )
                   ( ANY FOOTING           )
                   ( FOOTING level-number  )
                   ( NO SELECT             )
```

EXECUTE

```
EXECUTE procedure-name [ USING {data-name} . . . ] [FROM file-name]     4-15
                               {literal  }
```

EXHIBIT

CDCS/CRM Format

```
           [ data-name [OF file-name]                                       ]  4-15
           [ REPORTS [report-name [layout-directive] ]                      ]
  EXHIBIT  [ SESSIONS [session-id [transmission-id-1 [TO transmission-id-2] ] ] ]
           [ TEMPORARY                                                      ]
           [ RELATION                                                       ]
```

IMF Format

```
           [ data-name [OF file-name]                                       ]
           [ REPORTS [report-name [layout-directive] ]                      ]
  EXHIBIT  [ SESSIONS [session-id [transmission-id-1 [TO transmission-id-2] ] ] ]
           [ TEMPORARY                                                      ]
           [ RELATION                                                       ]
           [ ACCESS [access-path-name]                                      ]
           [ COSETS [coset-name]                                            ]
```

EXTRACT

```
                                            [ON ]
EXTRACT UPON file-name [WITH DIRECTORY [OFF]] {expression [AS data-name]} . . .        4-19
```

FOLLOW

```
IMF Format                                                                             4-20

       ( path-name [ {MEMBER} {coset-name} ] . . . )
FOLLOW (             {OWNER }                      )
       ( DEFAULT                                   )
```

FOOTING

```
                        (   [        {integer-1 [BEYOND]} ]
FOOTING level-number    (   [AT LINE {data-name         } ]                            4-20
                        (

   {                              [    {NEXT TAB       } ] }       )
   {IS expression [CENTERED]      [ IN {TAB integer-2  } ] } . . . ) . . .
   {                              [    {COLUMN integer-3} ] }       )
```

FORMAT

```
FORMAT report-name                                                                     4-21
```

HEADING

```
                        (   [        {integer-1 [BEYOND]}                 ]
HEADING level-number    (   [AT LINE {data-name         } [ON ALL PAGES]  ]            4-22
                        (

   {                              [    {NEXT TAB       } ] }       )
   {IS expression [CENTERED]      [ IN {TAB integer-2  } ] } . . . ) . . .
   {                              [    {COLUMN integer-3} ] }       )
```

HELP

```
HELP [directive-name    ]                                                              4-23
     [diagnostic-number ]
```

IF

```
   {condition      }
IF {condition-name }                                                                   4-23
   {SAME           }
```

INVOKE

CDCS Format

```
    INVOKE subschema-name [OF schema-name]                                          4-28
                 [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]
                 [FOR DATABASE version-name]
```

CRM Format

```
    INVOKE subschema-name [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]
```

IMF Format

```
    INVOKE [FOR REPAIR] external-schema-name OF schema-name [KEY literal]

                [ IN   ]
                [ FROM ] METADB permanent-file-name [user-name] [PW password]
```

LOOKUP

```
LOOKUP NAMES IN { DATABASE } FOR { FIRST }                                          4-29
                { FILES    }     { ONLY  }
```

MODIFY

CDCS/CRM/IMF Format 1 4-29

```
    MODIFY SETTING {data-name} . . .
```

CDCS/CRM/IMF Format 2

```
                                 || SETTING {data-name-1} . . . USING key-name . . .      ||
                                 ||        [SETTING {data-name-2} . . .] [FROM file-name] ||
    MODIFY [record-name]†        ||                                                        ||
                                 || MOVE expression-1 TO {data-name-3} . . .               ||
                                 ||        [AND expression-2 TO {data-name-4} . . .] . . . ||

                  [ VETO ]
                  [ PASS ]
```

†Not applicable to IMF

MOVE

```
MOVE {literal-1   } TO data-name-1 . . . [{literal-2       } TO data-name-2 . . . ] . . .      4-30
     {expression-1}                       {AND expression-2}
```

```
MOVE {BEFORE}                                                                                  4-30
     {AFTER }

    {TITLE                    }
    {REPORT                   }
    {RECAP                    }
    {ANY SELECT               }
    {SELECT tag-number        }
    {ANY DETAIL               }
    {DETAIL tag-number        }  expression-1 TO data-name-1 . . .
    {ANY BREAK                }     [AND expression-2 TO data-name-2 . . . ] . . .
    {BREAK level-number       }
    {ANY HEADING              }
    {HEADING level-number     }
    {ANY FOOTING              }
    {FOOTING level-number     }
    {NO SELECT                }
```

NOTE

```
NOTE user-comment                                                                              4-31
```

OS

```
OS [,] control-statement-string                                                                4-31
```

PAGE-NUMBER

```
PAGE-NUMBER [ AT [TITLE-LINE       ] [TAB integer-2   ] ] . . .                                4-31
            [    [LINE integer-1   ] [COLUMN integer-3] ]
            [    [RECAP-LINE       ]                    ]
```

PAGE-SIZE

```
PAGE-SIZE IS [integer-1 LINES] [integer-2 COLUMNS]                                             4-32

    [integer-3 {HORIZONTAL} SECTIONS] [integer-4 IMAGES] [PARALLEL]
    [          {VERTICAL  }         ]
```

PERFORM

```
PERFORM session-id-1                                                                           4-33

    [transmission-id-1 [TO transmission-id-2] ] . . .

        [AND session-id-2 [transmission-id-3 [TO transmission-id-4] ] . . . ] . . .

            [REPEAT expression]      [VETO]
            [UNTIL condition  ]      [PASS]
```

PREFACE

```
PREFACE IS { TEXT        } FROM file-name                                        4-34
           { report-name }
```

PREPARE

```
PREPARE report-name FROM file-name                                               4-34
```

PREVIEW

```
PREVIEW report-name [|| FROM file-name             ||]                           4-35
                    [|| DESCRIBED IN file-name     ||]
```

RECAP

```
RECAP { [AT LINE integer-1 [BEYOND] ]                                            4-35
      {
  {                                                  {  NEXT TAB     } ] }       }
  { IS expression [CENTERED]  [  IN                  {  TAB integer-2 } ] } . . . { . . .
  {                                                  {  COLUMN integer-3} ] }     }
```

RECORDING

```
RECORDING { session-id [transmission-id] [BY increment-size] }                   4-36
          { OFF                                              }
```

RECOVERY

```
RECOVERY POINT USING data-name-1 { literal     }                                 4-37
                                 { data-name-2 }
```

REMOVE

```
REMOVE [record-name]†  [SETTING {data-name-1} . . .] USING key-name              4-37
             [SETTING {data-name-2} . . .] [FROM file-name]

             [ VETO ]
             [ PASS ]
```

†Not applicable to IMF

RETURN

```
RETURN { file-name         }                                                     4-38
       { area-name         }  . . .
       { relation-name     }
       { subschema-name    }
```

REWIND

```
REWIND file-name . . .                                                           4-39
```

SELECT

```
SELECT tag-number ON {condition     }                                      4-39
                     {condition-name}
```

SEPARATOR

```
SEPARATOR [= delimiter]                                                    4-42
          [ITEM-SIZE  ]
```

SORT

```
                  [|| UPON file-name-2                        ||]
                  [||                                         ||]
                  [||         COBOL                           ||]
                  [||         FORTRAN                         ||]
                  [|| USING   literal-string                  ||]
SORT file-name-1  [||         optional-table                  ||]          4-42
                  [||                                         ||]
                  [|| ON {data-name DESCENDING} . . .         ||]
                  [||                                         ||]
                  [|| UNIQUE                                  ||]
```

SPECIFY

```
SPECIFY condition-name AS condition                                        4-43
```

STOP

```
STOP                                                                       4-43
```

STORE

```
                      || SETTING {data-name-1} . . . [FROM file-name]          ||   4-43
STORE [record-name]†  ||                                                       ||
                      || MOVE expression-1 TO {data-name-2} . . .              ||
                      ||     [AND expression-2 TO {data-name-3} . . . ] . . .  ||

            [VETO]
            [PASS]
```

†Not applicable to IMF

SUMMARY

```
SUMMARY IS {TEXT       } FROM file-name                                    4-44
           {report-name}
```

TABS

```
TABS AT integer-1 [integer-2] . . .                                        4-45
```

TIME

```
       [    [ TITLE-LINE     ]  [ TAB integer-2    ] ]
TIME   [ AT [ LINE integer-1 ]  [ COLUMN integer-3 ] ] . . .                4-45
       [    [ RECAP-LINE     ]                       ]
```

TITLE

```
TITLE  { AT LINE integer-1                                                  4-46
       {
     {                                      { NEXT TAB          }   }       }
     { IS expression [CENTERED]  [  IN      { TAB integer-2     } ] } . . . } . . .
     {                                      { COLUMN integer-3  }   }       }
```

UNIVERSAL

```
UNIVERSAL [ IS universal-character ]                                        4-46
          [ OFF                    ]
```

UPDATE

```
UPDATE area-name                                                            4-47
```

VERIFY

```
VERIFY data-name . . .                                                      4-47
```

VERSION

CDCS/CRM/IMF Format 1 4-48

```
    VERSION IS DEFAULT
```

CDCS/CRM/IMF Format 2

```
    VERSION IS catalog-file [( permanent-file-parameters [PW] )]
```

CDCS Format

```
    VERSION IS catalog-file OF subschema-name
                  [FROM LIBRARY permanent-file-name] [( permanent-file-parameters [PW] )]
                       [FOR DATABASE version-name]
```

VETO

```
VETO { ON  }                                                                4-49
     { OFF }
```

VIA

```
VIA relation-name                                                           4-50
```

EXPRESSIONS, RELATIONS, CONDITIONS 3-9

The notation to describe an expression is shown below. Arithmetic operators are any of the following:

Word	Symbol	Arithmetic Operation
PLUS	+	Addition
MINUS	-	Subtraction
TIMES	*	Multiplication
OVER	/	Division
RAISED	**	Exponentiation

Expression Notation 3-9

$$\left\{\begin{matrix}\text{data name-1}\\ \text{literal-1}\\ \text{function-1}\end{matrix}\right\}\left[\left\{\begin{matrix}\text{arithmetic}\\ \text{operator}\end{matrix}\right\}\left\{\begin{matrix}\text{data name-2}\\ \text{literal-2}\\ \text{function-2}\end{matrix}\right\}\right]\ldots$$

Relational Operators

Relational operators are any of the following abbreviations or symbols: 3-10

Abbreviation	Symbol	Meaning
EQ	=	equals
NE	≠	not equal
GT	>	greater than
LT	<	less than
GE	≥	greater than or equal
LE	≤	less than or equal

Relational Expression Notation 3-10

expression-1 {relational operator} expression-2

Compound relational expressions are of the form:

(relational-exp-1) {logical operator} (relational-exp-2)

Logical Operators 3-10

Compound relational expressions or conditions are of the form:

Word	Symbol	Meaning
AND	^	Both conditions must be satisfied
OR	∨	Either or both conditions must be met
XOR	None	One or the other but not both conditions must be met
NOT	¬	May be used to reverse a relational expression

Condition Notation 3-10

$$\left[\begin{matrix}\neg\\ \underline{\text{NOT}}\end{matrix}\right]\text{ relational-exp-1}\left[\left\{\begin{matrix}\wedge\\ \underline{\text{AND}}\\ \underline{\text{OR}}\\ \underline{\text{XOR}}\\ \underline{\vee}\end{matrix}\right\}\left[\begin{matrix}\neg\\ \underline{\text{NOT}}\end{matrix}\right]\left\{\begin{matrix}\text{relational-exp-2}\\ \left[\begin{matrix}\neq\\ \cdot\\ \cdot\\ \cdot\\ \leq\end{matrix}\right]\text{ expression}\end{matrix}\right\}\right]\ldots$$

Precedence Order 3-10

Expressions defined by arithmetic, relational, and logical operators are evaluated according to the following precedence hierarchy:

**	(exponentiation)
/*	(division or multiplication)
+-	(addition or subtraction)
GT GE LT LE EQ NE	(relational operators)
NOT	(logical operator)
AND	(logical operator)
OR XOR	(logical operators)

FUNCTIONS

Independent Functions

$\underline{\text{ABS}}\ \underline{(}\text{data-name}\underline{)}$ 3-6

$\underline{\text{DECODE}}\ \underline{(}\text{data-name-1 }\underline{\text{(ALL)}}, \left\{\begin{array}{l}\text{literal}\\ \text{data-name-2}\end{array}\right\}, \text{ data-name-3 }\underline{\text{(ALL))}}$ 3-6

$\underline{\text{MIN}}\ \underline{(}\left\{\begin{array}{l}\text{data-name}\\ \text{literal}\end{array}\right\}, \ldots \underline{)}$ 3-6

$\underline{\text{MAX}}\ \underline{(}\left\{\begin{array}{l}\text{data-name}\\ \text{literal}\end{array}\right\}, \ldots \underline{)}$ 3-6

$\underline{\text{JULIAN}}\ \underline{(}\left\{\begin{array}{l}\text{data-name}\\ \text{literal}\end{array}\right\} \underline{)}$ 3-6

$\underline{\text{GREG}}\ \underline{(}\left\{\begin{array}{l}\text{data-name}\\ \text{literal}\end{array}\right\} \underline{)}$ 3-6

$\underline{\text{MASK}}\ \underline{(}\left\{\begin{array}{l}\text{data-name-1}\\ \text{literal-1}\end{array}\right\}, \left\{\begin{array}{l}\text{data-name-2}\\ \text{literal-2}\end{array}\right\}, \{\text{literal-3}\} \underline{)}$ 3-6

$\underline{\text{SCAN}}\ \underline{(}\left\{\begin{array}{l}\text{data-name-1}\\ \text{literal-1}\end{array}\right\}, \left\{\begin{array}{l}\text{data-name-2}\\ \text{literal-2}\end{array}\right\}, \left[\begin{array}{l}\text{data-name-3}\\ \text{literal-3}\end{array}\right] \ldots \underline{)}$ 3-8

Cumulative Functions

SUM (data-name)	3-8
COUNT (data-name)	3-8
MEAN (data-name)	3-9
MINS (data-name)	3-9
MAXS (data-name)	3-9

RESERVED WORD LIST E

The following words and their underlined abbreviations can be used only as shown in the format specifications. They cannot be used as a user-defined name, such as a data name, file name, report name, session name, or permanent file parameter. Any part word including the underlined part up to the full word is a valid abbreviation. For example, CRE and CREA are valid abbreviation of CREATE. Both abbreviations are reserved words.

ABS
ACCESS
AFTER
ALL
ALTER
AND
ANY
AREA
AREAS
AS
AT

BEFORE
BEYOND
BREAK
BY

CATALOG
CENTERED
CHARACTER
COLUMNS
COMPILE
COMPLEX
COSETS
COUNT
CREATE
CURRENT-ANY
CURRENT-DATE
CURRENT-LINE
CURRENT-PAGE
CURRENT-REPORT
CURRENT-SESSION
CURRENT-TIME
CY

DATABASE
DATE
DECODE
DEFAULT
DEFINE
DELETE
DEPENDING
DESCENDING
DESCRIBE
DESCRIBED
DETAIL
DIAGNOSTIC
DIRECTORY
DISPLAY
DOUBLE
DUPLICATE

EJECT
END
EQ
ERASE
EVALUATE
EXECUTE
EXHIBIT
EXTRACT

FILES
FIRST
FIXED
FLOATING
FOLLOW
FOOTING
FOR
FORMAT
FROM
FULL

GE
GREG
GT

HEADING
HELP
HORIZONTAL

ID
IF
IMAGES
IN
INDEX
INPUT
INSERT
INTEGER
INVOKE
IS
ITEM-SIZE
ITEMS
I-O

JULIAN

KEY

LAST
LC
LE

LIBRARY
LINE
LINES
LOGICAL
LOOKUP
LT

M
MASK
MAX
MAXS
MEAN
MEMBER
METADB
MIN
MINS
MINUS
MODIFIED
MODIFY
MOVE
MR

NAMES
NE
NEXT
NO
NOT
NOTE
NUMERIC

OF
OFF
ON
ONCE
ONLY
OR
OS
OUTPUT
OVER
OVERFLOW
OWNER

PAGE-NUMBER
PAGE-SIZE
PAGES
PARALLEL
PART
PASS
PERFORM
PLUS
PN
POINT

PREFACE
PREPARE
PREVIEW
PRINT
PW

R
RAISED
RECAP
RECAP-LINE
RECORDING
RECOVERY
RELATION
REMOVE
REPAIR
REPEAT
REPORTS
RETURN
REWIND
RW

SAME
SCAN
SECTIONS
SELECT
SEPARATOR
SESSIONS
SETTING
SN
SORT
SPECIFY
STOP
STORE
SUM
SUMMARY

TABS
TEMPORARY
TEXT
TIME
TIMES
TITLE
TITLE-LINE
TO

UN
UNIQUE
UNIVERSAL
UNTIL

UPDATE
UPON
USE
USER-ID
USING

VALUE
VERIFY
VERSION
VERTICAL
VETO
VIA

WHERE
WITH

XOR

+
-
*
/
(
)
=
≠
<
>
≤
≥
∧
∨
**
¬
↱
$
↑
↓
[
]
:
;
≡

PICTURE SPECIFICATION CLAUSE

The DEFINE and DESCRIBE directives include the BY picture-specification clause, which is a method for specifying the size of an item. It can also be used to indicate the location of an operational sign or an assumed decimal point. As an editing clause, BY describes the desired format to be used in display of an item.

Item type is determined by the type of characters in the picture specification; 9, S, V, and P describe a numeric item. An alphabetic item can contain only A; an alphanumeric item is defined by X or a combination of 9's and A's. If a picture specification contains characters other than 9's, it must be enclosed by a pair of delimiters.

The CHARACTER type is manipulated as is the COBOL display usage class (all moves are alphanumeric). The difference between A and X pictures is documentary only.

The size of a nonedited item is determined by the number of 9, X, or A characters in the picture specification. The characters S, V, and P are not counted in determining the size; S can appear only once in a picture specification. Size can also be specified by a digit in parentheses to indicate the number of times the character is repeated. For instance, 999999 and 9(6) are equivalent; XXXXXX and X(6) are equivalent; each indicates an item with six character positions.

A picture specification can contain a maximum of 30 characters including parentheses; however, the item pictured can be larger. To illustrate, a picture specification containing 75 A's is too long, but A(75), containing only 5 characters, can be used to describe an item with 75 alphabetic characters. Alphabetic and alphanumeric items cannot exceed 2047 characters. A numeric item can contain 18 digits (only 14 digits are significant); additional leading or trailing zeros for decimal point alignment can be specified up to a total of 30 characters. When floating-point-to-integer conversions are involved, a numeric item is rounded up if the digit following the decimal point is greater than or equal to 5. The item is truncated when the digit is less than 5. No blanks are allowed within a picture specification.

NUMERIC ITEMS

A numeric item is a data item specified in a DEFINE or DESCRIBE directive as type INTEGER, FLOATING, DOUBLE, COMPLEX, or NUMERIC. The characters used in describing numeric items are 9, S, V, and P. The use of these characters is described in the following list:

9 Indicates a numeric character in the corresponding position in the item. Each occurrence of 9 must be counted in determining item size.

S Designates that an operational sign is specified by a sign overpunch in the rightmost character in the data item. The S must be the leftmost character in the picture specification. It is allowed only for a type NUMERIC item. A type NUMERIC item can contain a negative value only if S or an insertion character +, -, CR, or DB is specified. (See the Insertion Characters subsection.) When S is specified, the sign does not occupy a character position, nor is it counted in the size. When an item described with an S is displayed, no sign will be displayed.

Other types of numeric items (INTEGER, FLOATING, DOUBLE, and COMPLEX) can contain negative values even though S is not specified.

V Indicates the position of an assumed decimal point. An actual decimal point can never occur in a numeric item. An assumed decimal point does not occupy a character position, and it is not counted in determining size. The character V is specified between the symbols representing characters on either side of the assumed point location. It can appear only once in a picture specification.

For example, assume a data item of 2567 that is pictures as 9V999; the item size is four characters, and the value is 2.567. It would be displayed as 2567 because the decimal point character is not actually present.

P Specifies the location of an assumed decimal point; one P is specified for each implied position between the rightmost character and the assumed decimal point, or between the leftmost character and the assumed decimal point. The item is treated as if a zero were substituted for each P and a decimal point placed before or after the last P. The character P is never considered in determining size. The character P is not allowed in the picture specification clause for temporary items described as type INTEGER.

For example, an item composed of digits 2567 is treated as 256700. in computation if the picture specification in the BY clause is 9999PP or as .002567 if it is PP9999.

Figure F-1 illustrates some numeric item pictures and the representation of some sample data values.

Picture-Specification	Data Value	Display Code Storage
999	123	$\boxed{1}\boxed{2}\boxed{3}$
99V999	12345	$\boxed{1}\boxed{2}{\uparrow}\boxed{3}\boxed{4}\boxed{5}$
S99V99	+1234	$\boxed{1}\boxed{2}{\uparrow}\boxed{3}\boxed{\overset{+}{4}}$
PPP9999	.0001234	${\uparrow}0\ 0\ 0\ \boxed{1}\boxed{2}\boxed{3}\boxed{4}$
SPPP9999	-.0001234	${\uparrow}0\ 0\ 0\ \boxed{1}\boxed{2}\boxed{3}\boxed{\bar{4}}$
S999PPP	-123000.	$\boxed{1}\boxed{2}\boxed{\bar{3}}\boxed{0\ 0\ 0}{\uparrow}$

Figure F-1. Numeric Items

ALPHANUMERIC ITEMS

Alphanumeric items that do not specify editing can contain only the characters X, 9, and A. In alphanumeric items, the use of 9 and A is the same as for numeric and alphabetic items; X indicates that the corresponding character position in the item can contain any character in the character set. Some alphanumeric pictures and the representation of source data are shown in figure F-2.

Picture-Specification	Data Value	Display Code Storage
AAAAA or A(5)	COSTS	$\boxed{C}\boxed{O}\boxed{S}\boxed{T}\boxed{S}$
AAAA or A(4)	WXYZ	$\boxed{W}\boxed{X}\boxed{Y}\boxed{Z}$
XXXXXXXX or X(8)	ABCD-***	$\boxed{A}\boxed{B}\boxed{C}\boxed{D}\boxed{-}\boxed{*}\boxed{*}\boxed{*}$
XXXXXXXX or X(8)	123.4567	$\boxed{1}\boxed{2}\boxed{3}\boxed{.}\boxed{4}\boxed{5}\boxed{6}\boxed{7}$
AAAA999	ABCD123	$\boxed{A}\boxed{B}\boxed{C}\boxed{D}\boxed{1}\boxed{2}\boxed{3}$
A(4)9(3)	ABCD123	$\boxed{A}\boxed{B}\boxed{C}\boxed{D}\boxed{1}\boxed{2}\boxed{3}$

Figure F-2. Alphanumeric Items

EDITING

Editing alters the format and punctuation of data when it is displayed.

NUMERIC EDITING

The picture specification for a numeric edited item can contain combinations of the following symbols: B, P, V, Z, 0 (zero), 9, comma, period, asterisk, plus sign, minus sign, CR, DB, slash, and a currency sign. Permissible combinations are determined according to the order of precedence of symbols and editing rules.

Numeric editing is accomplished by insertion or replacement of characters when displaying the numeric item. The actual item stored does not have insertion or replacement characters included.

Under NOS, the 029 keypunch cannot be used to make the Hollerith punch patterns that represent +0 or -0.

Insertion Characters

An insertion character actually appears in the edited data item and is counted in the displayed item size. A maximum of 63 insertion characters can be specified in any one string. The insertion characters are described as follows:

$ A single dollar sign as the leftmost symbol in the picture specification appears as the leftmost character in the edited data item. It is counted in the size of the item.

+ A plus sign as the first or last symbol of the picture specification is inserted as indicated in the edited data item, provided the data is positive or unsigned. If the data is negative, however, a minus sign is inserted. The sign is counted in the size of the item.

- A minus sign as the first or last symbol of˜the picture specification is inserted as indicated in the edited data item, provided the data is negative (contains a negative operational sign). Otherwise, a blank is inserted in the indicated character position. The sign or blank is counted in the size of the item.

. A period in the picture specification represents an actual decimal point. It appears as indicated in the edited data item and is counted in the size. The picture specification for an item can never contain more than one decimal point, either actual or assumed.

, or / A comma or slash in the picture specification is inserted as indicated in the edited item and is counted in the size of the item.

0 A zero in the picture specification is inserted as indicated in the edited item and is counted in the size of the item.

B The character B in the picture specification indicates a blank is to be inserted in the corresponding position in the edited data item. It is counted in the size of the item.

CR This symbol, which represents credit, can be specified only as the last two characters and positions to the right. It appears in the last two character positions of the edited data item, provided the value of the data is negative. If the data item is positive or unsigned, the last two character positions are set to blanks. This symbol is counted as two characters in the size of the item.

DB This symbol, which represents debit, is specified in the same way and has the same results as the credit symbol.

Figure F-3 illustrates the results of numeric editing with insertion characters.

Picture-Specification	Data Value	Displayed Data
$99	48	$ 4 8
$99.99	4834 ↑	$ 4 8 . 3 4
9,999	4834	4 , 8 3 4
+999	292	+ 2 9 2
+999	$29\overset{+}{2}$	+ 2 9 2
+999	$29\bar{2}$	- 2 9 2
999-	$29\bar{2}$	2 9 2 -
999-	$29\overset{+}{2}$	2 9 2 Δ
999-	292	2 9 2 Δ
$BB999.99	24321 ↑	$ Δ Δ 2 4 3 . 2 1
$00999.99	24321 ↑	$ 0 0 2 4 3 . 2 1
99.99CR	$113\bar{4}$ ↑	1 1 . 3 4 C R
99.99CR	1134 ↑	1 1 . 3 4 Δ Δ
99.99DB	2376 ↑	2 3 . 7 6 D B
99.99DB	2376 ↑	2 3 . 7 6 Δ Δ
99/99/99	011269	0 1 / 1 2 / 6 9

Figure F-3. Numeric Editing by Insertion

Replacement Characters

A replacement character in the picture specification suppresses leading zeros in the data and replaces them with the specified character or a blank. Replacement characters are described below.

Only one replacement character, repeated as needed, can be used in a picture specification. The replacement characters are as follows:

Z At the left of the picture specification, one Z is specified for each leading zero to be suppress and replaced by blanks. Z's can be preceded by one of the insertion characters $, +, or - and interspersed with decimal point, commas, zeros, or B's.

Zeros are not suppressed to the right of the first nonzero digit nor to the right of an assumed or actual decimal point unless the value of the data is zero, and all character positions in the item are described by a Z. In such case, even the actual decimal point is suppressed, and the edited item is all blanks.

If $, +, or - precedes the Z's, it is inserted in the far left character position of the item even if succeeding zeros in the item are suppressed. Where the value of the data is zero and all succeeding character positions are Z's, the $, +, or - is replaced by blanks.

If a comma, zero, or B is encountered before zero suppression terminates, the character is not inserted in the edited data item; it is suppressed and a blank inserted in its place.

* The asterisk causes leading zeros to be replaced by an asterisk instead of a blank. It is specified in the same way as the editing character Z and follows the same rules, except that an actual decimal point is not replaced by an asterisk when the value of the data is zero.

$ The dollar sign as a replacement character suppresses leading zeros; it acts as a floating sign and is inserted directly preceding the first non-suppressed character. One more dollar sign must be specified than the number of zeros to be suppressed. This dollar sign is always present in the edited data whether or not any zero suppression occurs. The remaining dollar signs have the same effect as Z's by suppressing leading zeros. No editing character can precede the initial dollar sign, and each dollar sign is counted in the size of the item.

+ A plus sign used as a replacement character is a floating sign. It is specified one more time that the number of leading zeros to be suppressed. It functions in the same way as the floating dollar sign; a plus sign is placed directly preceding the first nonsuppressed character if the edited data is positive or unsigned, and a minus sign is placed in this position if the edited data is negative.

- A minus sign used as a replacement character is a floating sign. It is specified in the same way as the floating plus sign. It is specified in the same way as the floating plus sign. It functions in the same way as the floating plus sign, except a blank is placed directly preceding the first nonsuppressed character, if the edited data is positive or unsigned.

Figure F-4 illustrates the results of numeric editing with replacement characters.

Picture-Specification	Data Value	Displayed Data
ZZ999	00923	Δ Δ 9 2 3
ZZZ99	00923	Δ Δ 9 2 3
ZZZZ.ZZ	000000	Δ Δ Δ Δ Δ Δ Δ
****.**	000000	* * * * . * *
$***.99	00923 ↑	$ * * 9 . 2 3
$$$9.99	000824 ↑	Δ Δ $ 8 . 2 4
---9.99	0052$\bar{6}$ ↑	Δ Δ - 5 . 2 6
+++9.99	03456 ↑	Δ + 3 4 . 5 6
$$$.99	3265 ↑	$ 3 2 . 6 5
ZZ9999	123456	1 2 3 4 5 6
ZZ9999	000005	Δ Δ 0 0 0 5
ZZ9999	100000	1 0 0 0 0 0
ZZZZZ	000010	Δ Δ Δ Δ 1 0
$9999	1234	$ 1 2 3 4
$9999	0012	$ 0 0 1 2
$ZZ99	0023	$ Δ Δ 2 3
$ZZ99	0005	$ Δ Δ 0 5
$$$$9	1234	$ 1 2 3 4
$$$$9	0123	Δ $ 1 2 3
$$$,$ZZ.99	000001	Illegal Picture

Figure F-4. Numeric Editing By Replacement

ALPHANUMERIC EDITING

The picture specification of an alphanumeric edited item is restricted to certain combinations of A, X, 9, B, and 0 (zero). The picture specification must contain at least one B or one zero and at least one

X or one A. Alphanumeric editing inserts a zero or blank in the designated positions. None of the original data is lost. Each character is counted in determining displayed item size. Figure F-5 illustrates some examples of alphanumeric editing.

SUMMARY OF EDITING RULES

The rules of editing can be summarized as follows:

Only one character of the set Z, $, *, S, +, and - can be used as a replacement character; the character can be specified more than once.

If the replacement character Z or * is used with one of the insertion characters $, +, or -, the + or - can be specified as either the leftmost or rightmost character.

The characters + and - cannot be included in the same picture specification.

A 9 cannot be specified to the left of a replacement character.

Symbols that can appear only once are V, S, decimal point, CR, and DB.

The decimal point cannot be the rightmost character in a picture specification.

Picture-Specification	Data Value	Displayed Data
AABBAA	ABCD	A B Δ Δ C D
AA00AA	WXYZ	W X 0 0 Y Z
XXBBXX	A1B1	A 1 Δ Δ B 1
X0X0X0	ZXY	Z 0 X 0 Y 0
XB(3)X0(4)X	N15	N Δ Δ Δ 1 0 0 0 0 5

Figure F-5. Alphanumeric Editing

REPORT EXTRACTION LOGIC FLOWCHARTS G

This appendix presents single page flowcharts where each page represents generalized operations as one or more subroutines. A lettered connector symbol begins each flowchart as a subroutine entrance. A connector labeled RETURN represents the subroutine exit from each flowchart. RETURN indicates that continued reference should restart on another page at the point where the logic path shows execution of a specific subroutine.

The flowcharts on the following pages provide generalized descriptions of Query Update operations and should not be interpreted as absolute depictions of the software system. Similarly, nonstandard subroutine symbols are used to emphasize user direction to another flowchart. Capitalized words represent key portions of directives and underlining is used to call attention to significant parts of directives.

The lettered connectors and flowchart titles are:

A	Figure G-1.	Report Control
B	Figure G-2.	Record and Detail Selection
C	Figure G-3.	Initialize Page
D	Figure G-4.	Headings
E	Figure G-5.	Breaks
F	Figure G-6.	Detail
G	Figure G-7.	Footings
H	Figure G-8.	Close Page
I	Figure G-9.	Print Detail with No Breaks

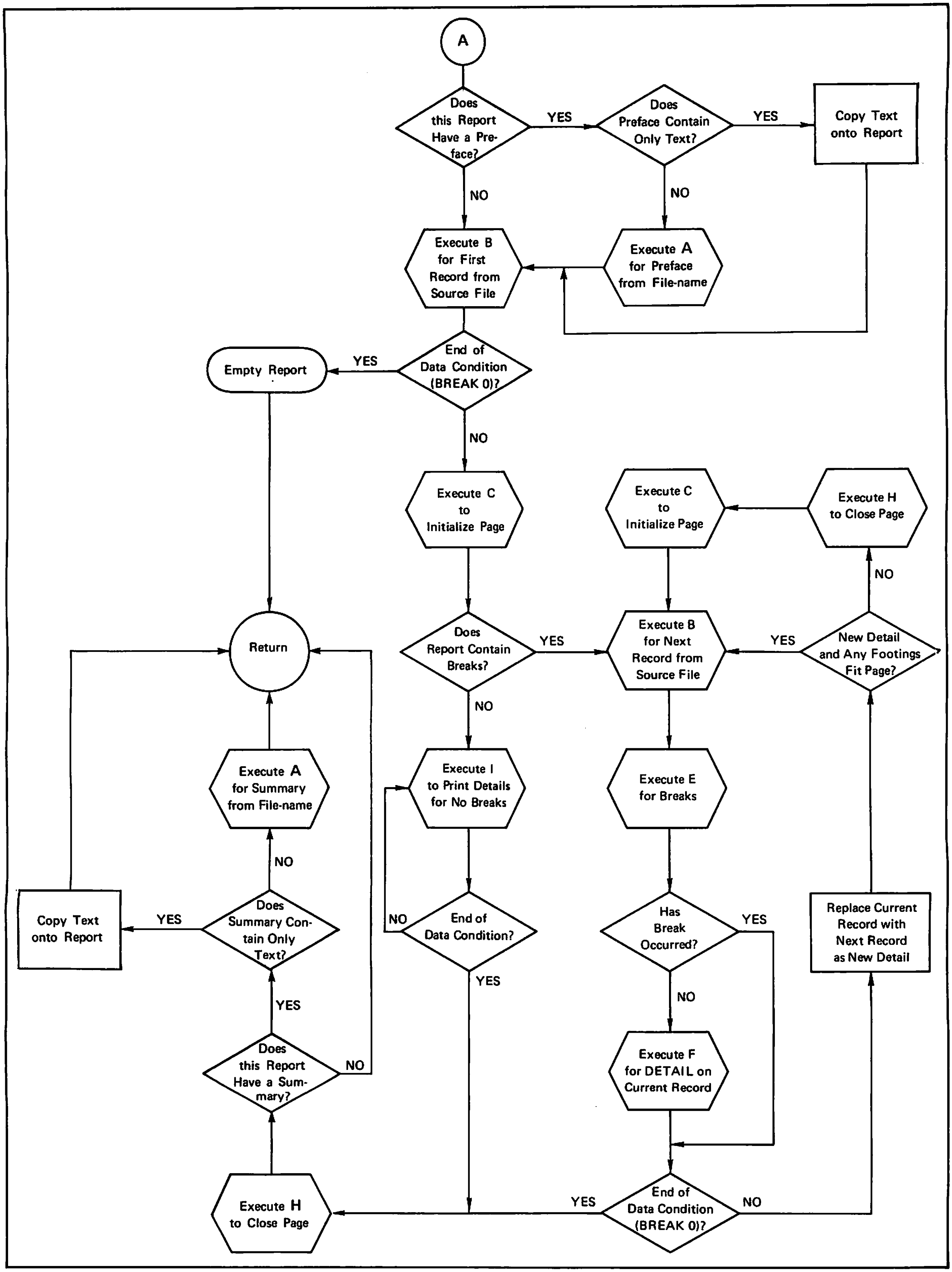

A
Does this Report Have a Preface?
YES
Does Preface Contain Only Text?
YES
Copy Text onto Report
NO
NO
Execute B for First Record from Source File
Execute A for Preface from File-name
End of Data Condition (BREAK 0)?
YES
Empty Report
NO
Execute C to Initialize Page
Execute C to Initialize Page
Execute H to Close Page
NO
Return
Does Report Contain Breaks?
YES
Execute B for Next Record from Source File
YES
New Detail and Any Footings Fit Page?
NO
Execute I to Print Details for No Breaks
Execute E for Breaks
Execute A for Summary from File-name
NO
Copy Text onto Report
YES
Does Summary Contain Only Text?
NO
End of Data Condition?
Has Break Occurred?
YES
Replace Current Record with Next Record as New Detail
YES
YES
NO
Does this Report Have a Summary?
NO
Execute F for DETAIL on Current Record
Execute H to Close Page
YES
End of Data Condition (BREAK 0)?
NO

Figure G-1. Report Control

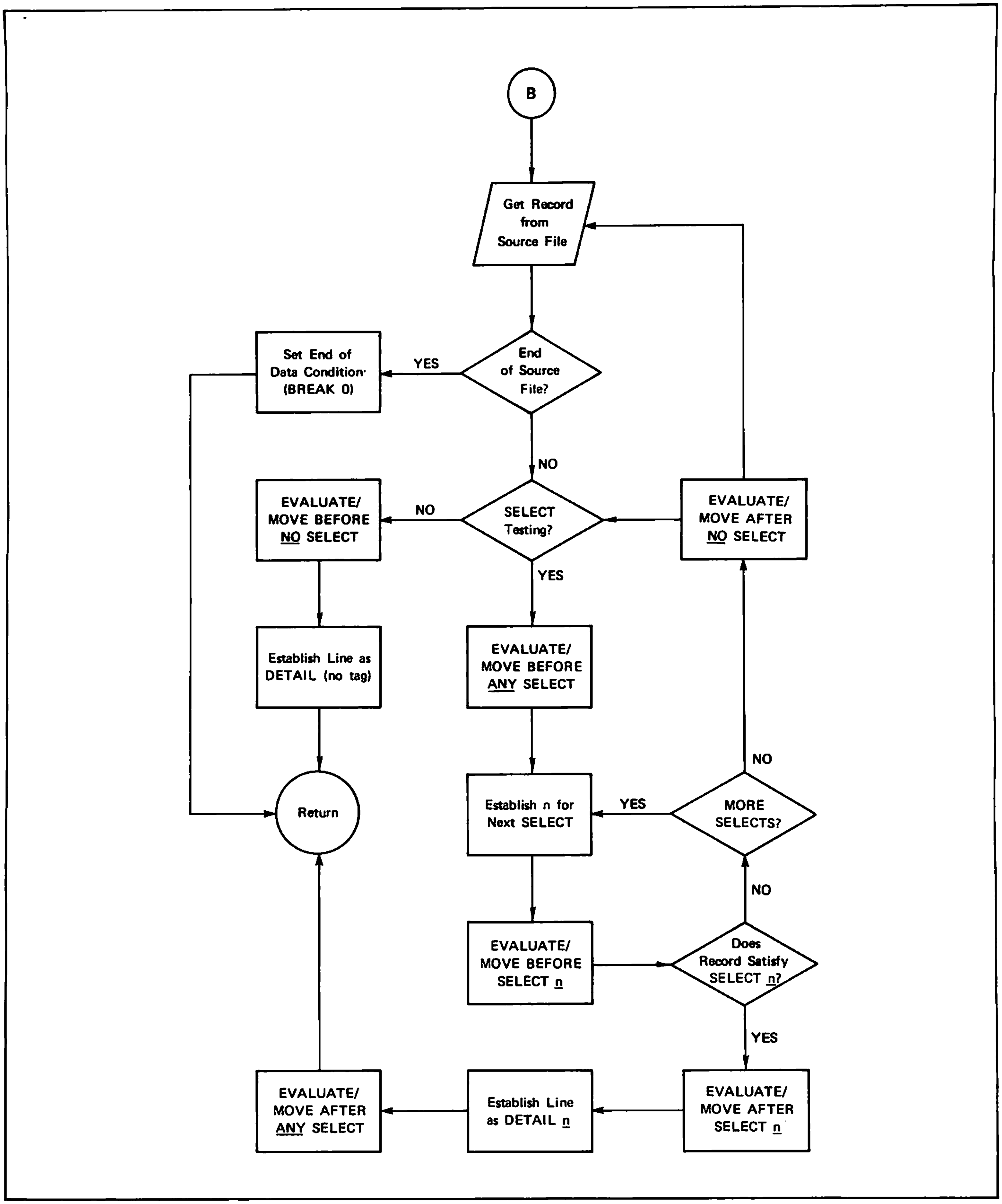

B
Get Record from Source File
End of Source File?
YES
Set End of Data Condition (BREAK 0)
NO
SELECT Testing?
NO
EVALUATE/ MOVE BEFORE NO SELECT
EVALUATE/ MOVE AFTER NO SELECT
YES
Establish Line as DETAIL (no tag)
EVALUATE/ MOVE BEFORE ANY SELECT
NO
Return
Establish n for Next SELECT
YES
MORE SELECTS?
NO
EVALUATE/ MOVE BEFORE SELECT n
Does Record Satisfy SELECT n?
YES
EVALUATE/ MOVE AFTER ANY SELECT
Establish Line as DETAIL n
EVALUATE/ MOVE AFTER SELECT n

Figure G-2. Record and Detail Selection

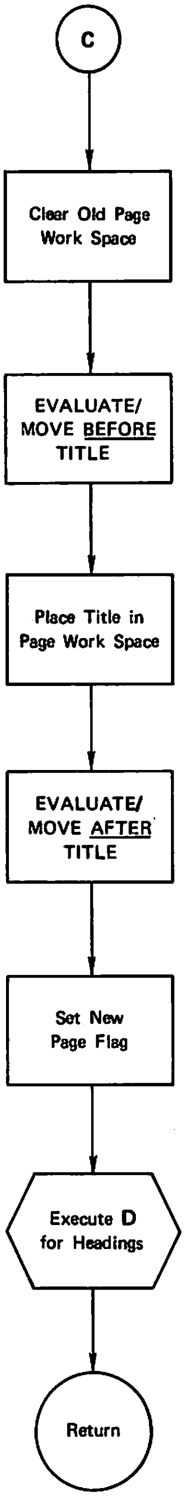

C
Clear Old Page Work Space
EVALUATE/ MOVE BEFORE TITLE
Place Title in Page Work Space
EVALUATE/ MOVE AFTER TITLE
Set New Page Flag
Execute D for Headings
Return

Figure G-3. Initialize Page

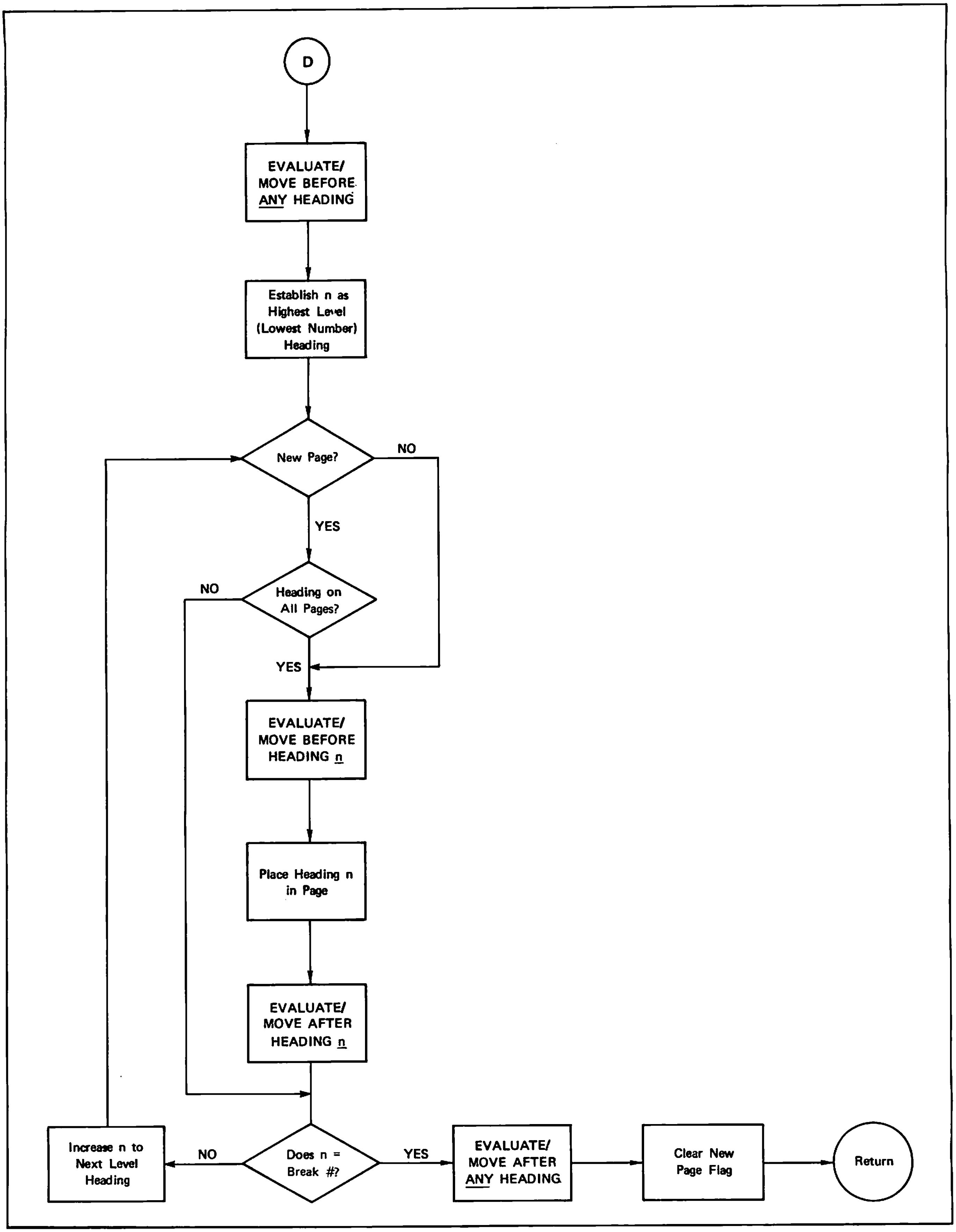

D
EVALUATE/ MOVE BEFORE ANY HEADING
Establish n as Highest Level (Lowest Number) Heading
New Page?
NO
YES
Heading on All Pages?
NO
YES
EVALUATE/ MOVE BEFORE HEADING n
Place Heading n in Page
EVALUATE/ MOVE AFTER HEADING n
Increase n to Next Level Heading
NO
Does n = Break #?
YES
EVALUATE/ MOVE AFTER ANY HEADING
Clear New Page Flag
Return

Figure G-4. Headings

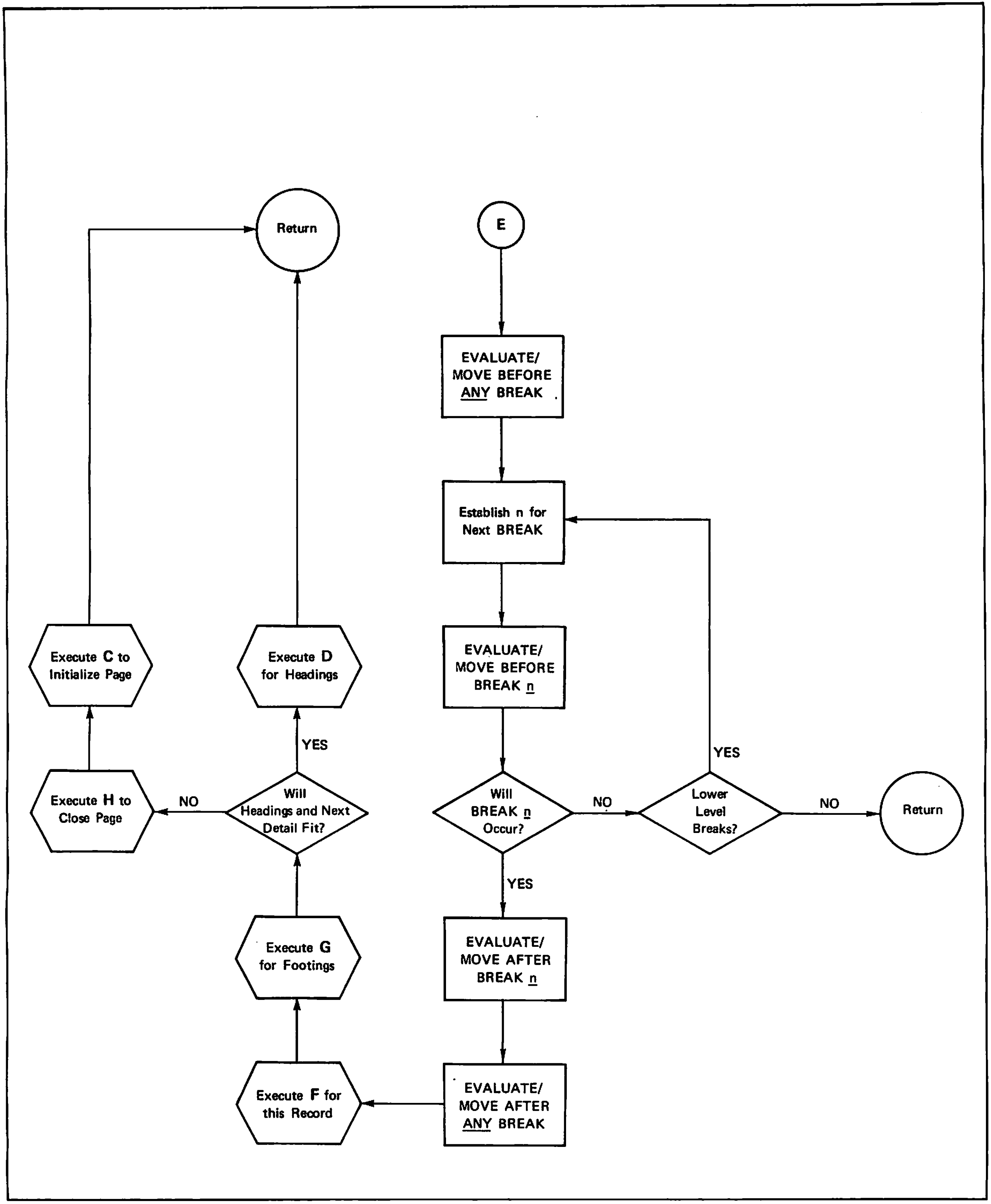

Return
E
EVALUATE/ MOVE BEFORE ANY BREAK
Establish n for Next BREAK
EVALUATE/ MOVE BEFORE BREAK n
Execute C to Initialize Page
Execute D for Headings
YES
YES
Execute H to Close Page
NO
Will Headings and Next Detail Fit?
Will BREAK n Occur?
NO
Lower Level Breaks?
NO
Return
YES
Execute G for Footings
EVALUATE/ MOVE AFTER BREAK n
Execute F for this Record
EVALUATE/ MOVE AFTER ANY BREAK

Figure G-5. Breaks

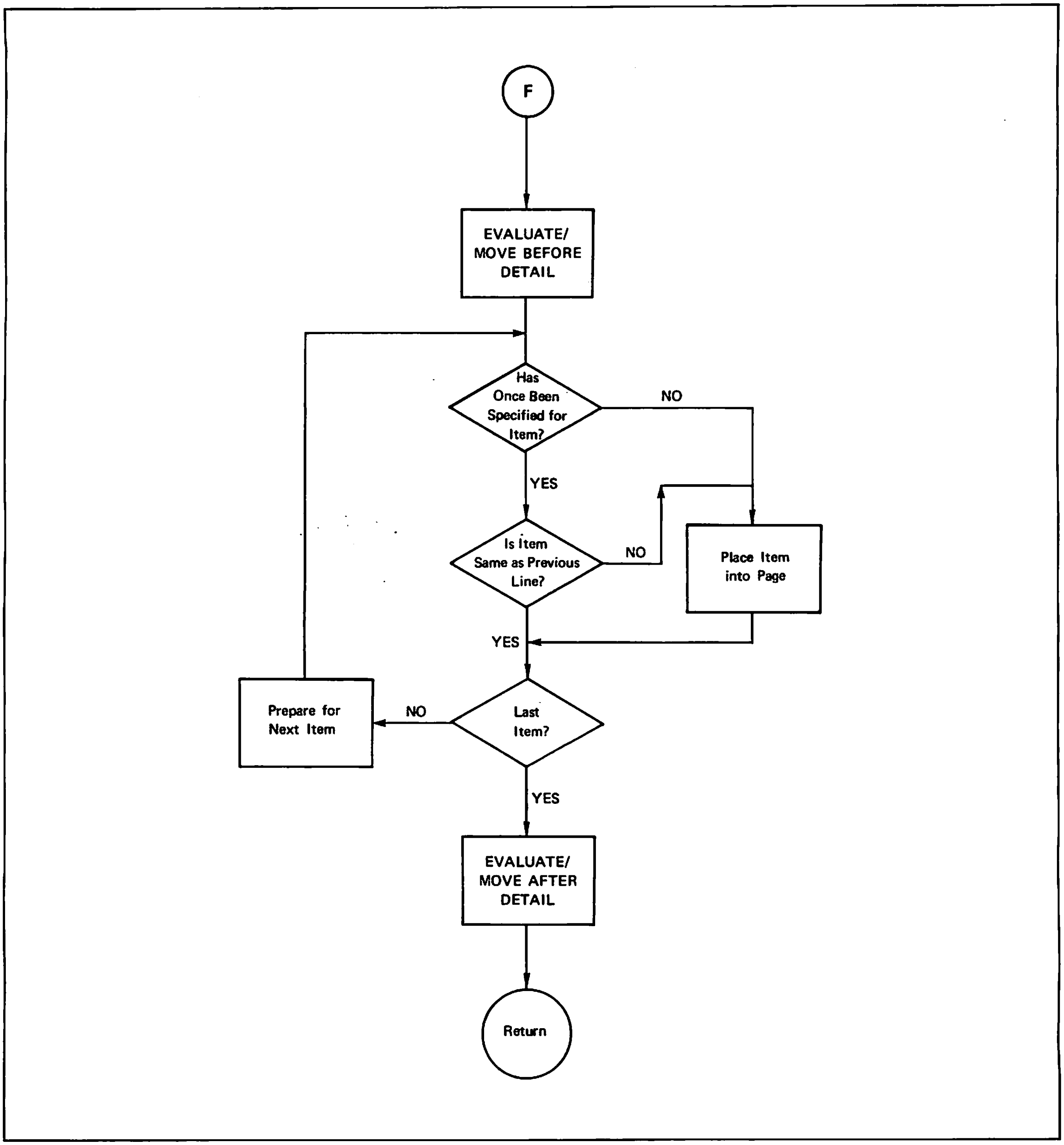

F
EVALUATE/ MOVE BEFORE DETAIL
Has Once Been Specified for Item?
NO
YES
Is Item Same as Previous Line?
NO
Place Item into Page
YES
Prepare for Next Item
NO
Last Item?
YES
EVALUATE/ MOVE AFTER DETAIL
Return

Figure G-6. Detail

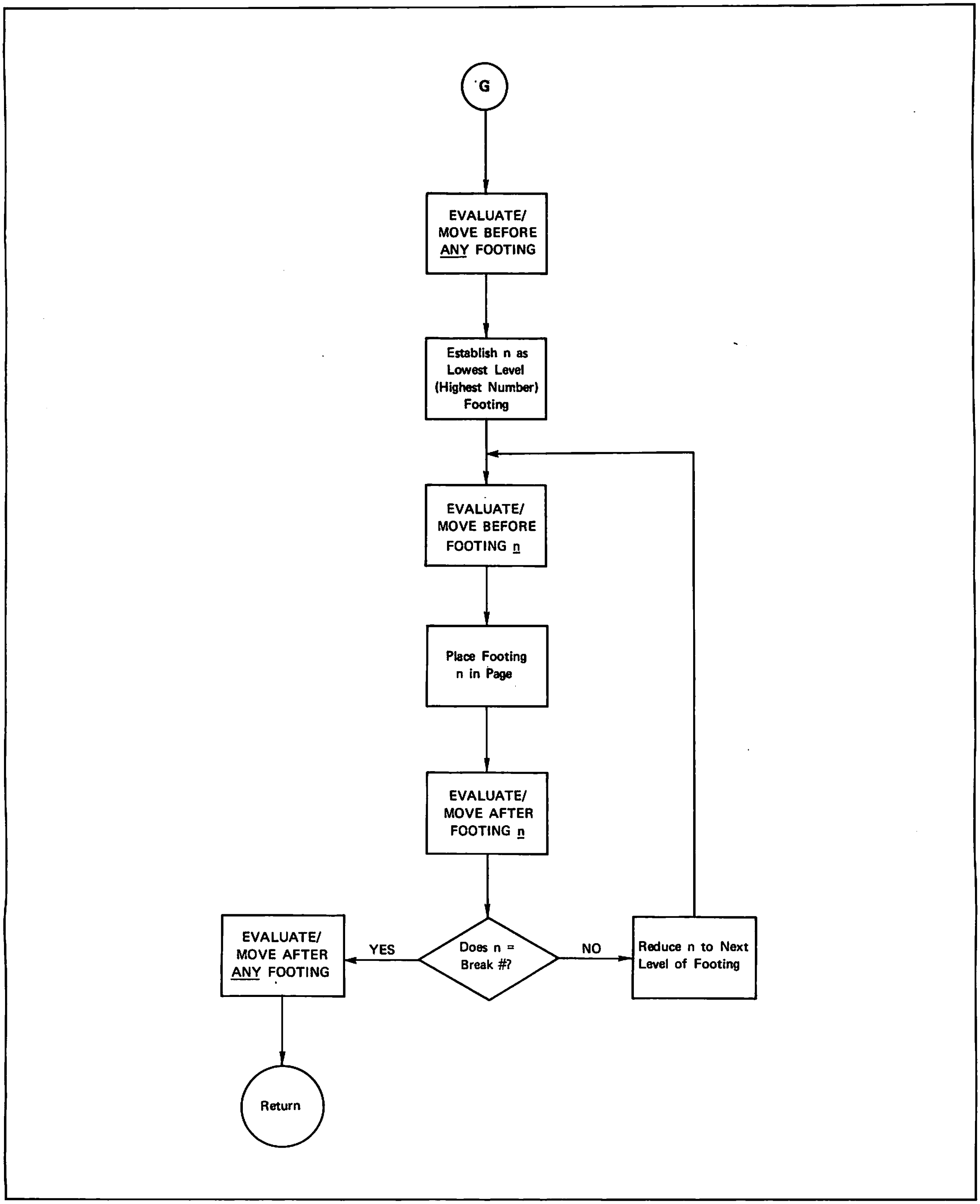

G
EVALUATE/ MOVE BEFORE ANY FOOTING
Establish n as Lowest Level (Highest Number) Footing
EVALUATE/ MOVE BEFORE FOOTING n
Place Footing n in Page
EVALUATE/ MOVE AFTER FOOTING n
Does n = Break #?
YES
NO
Reduce n to Next Level of Footing
EVALUATE/ MOVE AFTER ANY FOOTING
Return

Figure G-7. Footings

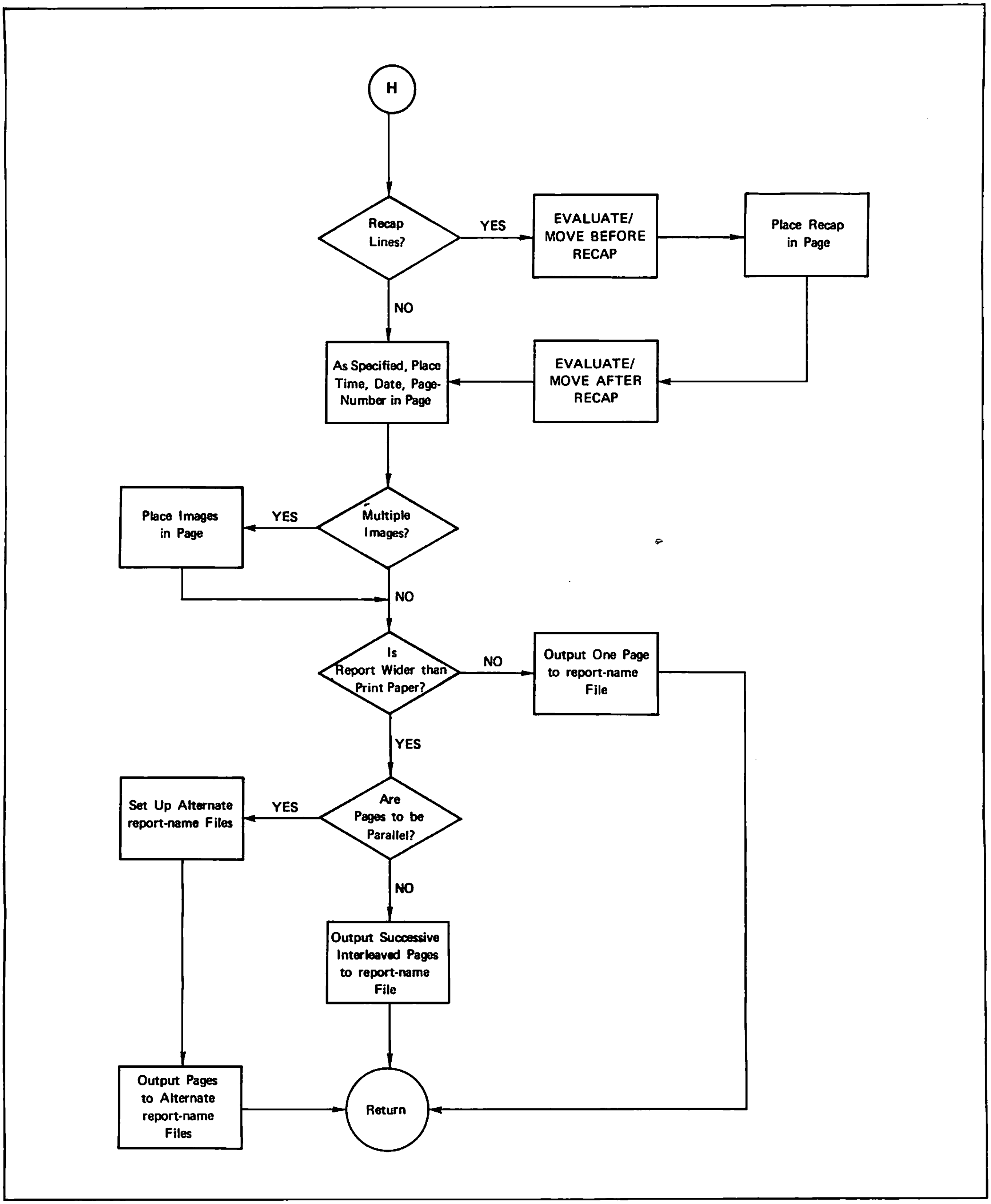

H
Recap Lines?
YES
EVALUATE/ MOVE BEFORE RECAP
Place Recap in Page
NO
As Specified, Place Time, Date, Page-Number in Page
EVALUATE/ MOVE AFTER RECAP
Place Images in Page
YES
Multiple Images?
NO
Is Report Wider than Print Paper?
NO
Output One Page to report-name File
YES
Set Up Alternate report-name Files
YES
Are Pages to be Parallel?
NO
Output Successive Interleaved Pages to report-name File
Output Pages to Alternate report-name Files
Return

Figure G-8. Close Page

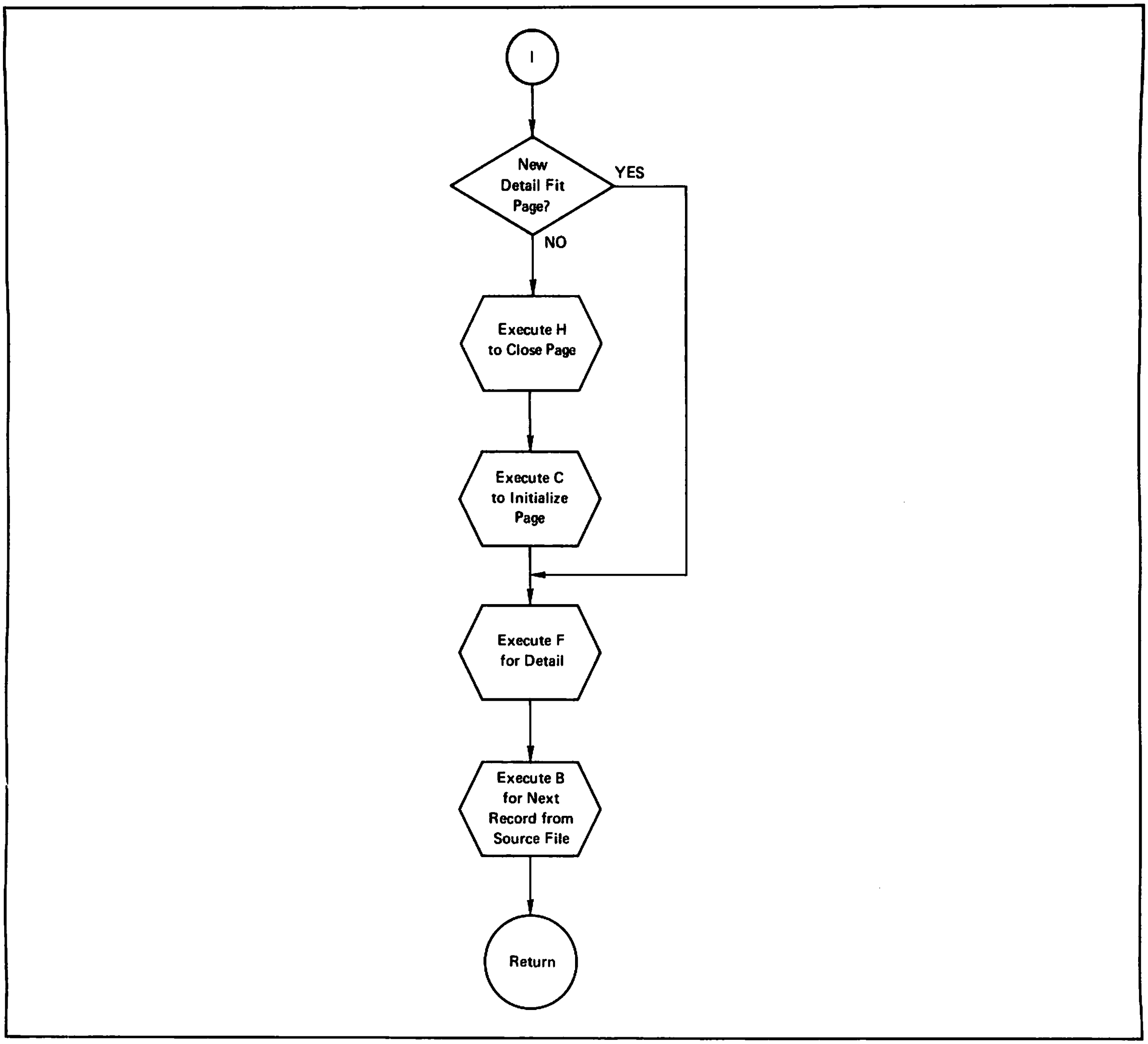


Figure G-9. Print Detail with No Breaks

ALTERNATIVE QUERY UPDATE DIRECTIVES H

Several Query Update directives can be used as alternatives to the directives described in section 4. The directives in section 4 are the recommended directives. The directives described in this appendix are provided to maintain compatibility with earlier releases of Query Update. Table H-1 shows directives that can be used as alternatives to one another. The old and the new directives perform the same function.

TABLE H-1. ALTERNATIVE DIRECTIVES

Old Directive	New Directive
DELETE	REMOVE
INSERT	STORE
UPDATE	MODIFY
USE	INVOKE

DELETE

The DELETE directive removes specific records from a data base area. Only one area can be modified at a time; each area joined in a relation must be modified separately. Records are selected for deletion either by referencing record key literals in the DELETE directive or as the result of an IF directive associated with the DELETE directive. The complete record is removed from the data base; the DELETE directive does not remove part of a record. The format of the DELETE directive is shown in figure H-1.

The DELETE directive requires permission to modify or extend the area file. This permission is granted according to the passwords specified for the area file in the USE directive.

When a delete operation is associated with an IF directive, the DELETE directive can be specified in its shortest form as only the keyword DELETE. Records that satisfy the specified condition are deleted from the data base.

The following example shows an IF directive with an associated DELETE directive. ALL EMPLOYEES records that have the letter E for the CODE data item and PUBLICATIONS for the SECTION data item are deleted from the data base.

```
IF CODE OF EMPLOYEES EQ $E$ AND +
SECTION = $PUBLICATIONS$ DELETE
```

The KEY option specifies a temporary item that contains a key to select the record for deletion. The literal option identifies the key of the specific record to be deleted. When the literal option is used in conjunction with an IF directive, the record is retrieved before the condition is tested; deletion occurs when the condition is true.

The USING option indicates that a sequence of data name values is provided for each record to be deleted. The data names must be defined in the same record description for the area file. If the FROM option is included, the values are provided from the specified file. For more information see the External File Organizations subsection. If the FROM option is not included, the values are entered through the terminal. One of the values must provide the record key; all others are ignored. If the values are nonumeric literals, the delimiter must be used to separate the literal unless SEPARATOR ITEM-SIZE has been specified. For each record to be deleted, the values entered correspond to the data names specified in the USING option. A DELETE directive with the USING option cannot be associated with an IF directive that references an area item.

When the records to be deleted contain primary and alternate keys, the data names in the USING option must conform to one of the following rules:

If records are to be selected for deletion by primary key, the primary key must be specified; additional data names can reference alternate keys or any other fields in the record.

If records are to be selected for deletion by alternate key, the desired alternate key must be specified; additional data names must not reference the primary key or any other alternate keys in the record.

```
         ┌                                                  ┐
         │ KEY data-name-1                                  │
DELETE   │ literal                                          │ ┌ VETO ┐
         │ [FROM file-name] USING ┌ data-name-2 ... ┐       │ └ PASS ┘
         │                        └ SAME            ┘       │
         └                                                  ┘
```

Figure H-1. DELETE Directive Format

Deletion by alternate key can result in several records being deleted for one alternate key value. When an alternate key value is entered (at the terminal or from the designated file), all records containing that alternate key value are deleted. In the following example, all records with the value ACCOUNTING in the alternate key field DEPT are deleted.

```
    - - DELETE USING DEPT

    >>  $ACCOUNTING$
```

When an indexed sequential file contains records with duplicate primary keys, all records having duplicate keys are deleted when the DELETE directive is specified for the particular key.

An *END or an end of file terminates the delete operation. When the FROM option is included, the contents of file name must be in the order specified by the data names in the USING option. SAME indicates that the previously specified sequence of data names is to be used. However, if any directive (such as USE or RETURN) that alters or eliminates any of the data names in the USING sequence is entered between the specification of the USING list and the DELETE USING SAME directive, the results are unpredictable.

The VETO option in a DELETE directive causes the record to be displayed before it is deleted. Either the first 40 characters of the record or the data items specified in a previous VERIFY directive are displayed at the terminal. The user then responds with one of the following entries which references the DELETE request:

YES (or Y)

Deletes the record.

NO (or N)

Does not delete the record.

PROCEED (or P)

Negates the current VETO option for the transmission; acts as a user response of YES for this record.

EXIT (or E)

Prevents deletion and terminates the transmission.

The PASS option disables the VETO for this transmission when a previous VETO directive specified VETO ON.

When the current area is described in the subschema as having actual key file organization, the primary key must be known before it can be specified in the DELETE directive with the literal option. In actual key file organization, primary key values are assigned by the system. The user must display records in the file or otherwise dump the contents of the file to obtain the key values assigned by the system before records can be deleted.

A relation joining several areas can be updated by modifying each area separately. When more than one area is in use at one time, Query Update must be able to identify which area is to be updated. The identification of an area to be updated can be derived by Query Update from the contents of the DELETE directive, if no duplicate data names exist in all the areas in use or if the data names are properly qualified. The user has the option of using format 2 of the UPDATE directive to identify the area from which records are to be deleted. Refer to the description of the UPDATE directive for further details.

Query directives for a relation can be interspersed with updating directives for one area, but they cannot be combined in the same transmission.

The user is responsible for maintaining the integrity of the data base and must ensure that the updating operation performed is consistent with the meaning of the relation defined for the areas. When deleting a record, the user must determine whether or not other related records must be deleted as well, and perform additional required delete operations.

The DELETE directive cannot be used when the area file specified through the USE directive is a sequential file.

INSERT

The INSERT directive creates a record entry and places it in an area of the data base. Only one area can be modified at a time; each area joined in a relation must be modified separately. Data can be input from the terminal or from a designated file. The format of the INSERT directive is shown in figure H-2. Figure H-3 illustrates the INSERT directive.

The INSERT directive requires permission to modify or extend the area file. This permission is granted according to the passwords specified for the area file in the USE directive.

```
          ┌ KEY data-name-1 [OF record-name-1]                  ┐
          │ literal [OF record-name-2]                          │  ┌VETO┐
INSERT    │                                   ┌ data-name-2 ... ┐ │  └PASS┘
          └ [FROM file-name]  USING           └ SAME          ┘ ┘
```

Figure H-2. INSERT Directive Format

Example 1

```
--        INSERT KEY NEW-NAME
```

A record is inserted with the value stored in the temporary data item NEW-NAME entered in the record key.

Example 2

```
--        INSERT $ADAMS JOHN$ OF EMPLOYEES
```

An EMPLOYEES record is inserted with the literal ADAMS JOHN entered in the record key. The rest of the record items are preset to zeros or blanks, consistent with the layout of the EMPLOYEES record.

Example 3

```
--        INSERT USING NAMES OF EMPLOYEES, PHONE, ROOM-NO, DEPT-NO
>>        $ADAMS JOHN$ $7343$ $110D$ $2110$
>>        $SMITH JOSEPH$ $7926$ $259C$ $6539$
>>        *END
```

Two EMPLOYEES records are inserted: the first record has ADAMS JOHN entered in the record key and the second record has SMITH JOSEPH entered in the record key; the remaining data items are entered in the respective fields.

Example 4

```
--        UPDATE CONTRACTS
--        INSERT USING CONTRACT-NO, DESCRIPTION
>>        $A8451$ $ACCOUNTING ANALYSIS$
>>        $C9508$ $MARKETING ANALYSIS$
>>        *END
```

The UPDATE directive identifies the CONTRACTS area as the area to be modified. Two records are inserted with A8451 and C9508 entered as record keys.

Figure H-3. Sample INSERT Directives

The KEY option causes the value of data-name-1 to be entered as the key in the record being created. Data-name-1 can be qualified if the area contains more than one record type. Data-name-1 must be defined as a temporary item, and must contain either a character value or numeric value. A key cannot be specified for an area with actual key organization. The INSERT KEY directive must be the first saved directive in a transmission. (See figures 4-29 and 4-30 in section 4 for information about saved directives.) The INSERT KEY directive can be preceded by an IF directive on temporary items since that IF directive is evaluated immediately and not saved. All directives within the transmission with INSERT KEY use the record set up by the INSERT KEY directive.

The literal option inserts a record with the specified literal entered as the key; the remaining fields of the record are initialized to their null values (usually blanks or zeros) as appropriate. The literal is used to find the position for an insertion in an area. Nonnumeric literals must be enclosed by delimiters. The literal option should be restricted to areas with only one record type. When an area has more than one record type, the literal must be qualified by OF and the name of the record; otherwise, record initialization occurs for the first record described by the directory. If more than one area was specified by the USE directive, the literal option must be qualified or format 2 of the UPDATE directive must be used to isolate the area to be updated.

The INSERT directive with the KEY or literal option specified inserts a record that contains data only in the record key field. This directive can be followed by a MOVE directive to place data in other fields of the record.

The USING option provides the means to enter several data items in a series of records to be inserted. The data items to be entered are identified by specifying the data names in the USING option. The data names must be defined in the same record description for the area file. One of the data names must be the name of the primary key, unless the area receiving the new record has an actual key file organization. The SAME option indicates that the previously specified sequence of data names is to be used. However, if any directive (such as USE or RETURN) that alters or eliminates any of the data names in the USING sequence is entered between the specification of the USING list and the INSERT USING SAME directive, the results are unpredictable.

If the data is input from the terminal, Query Update replies to the INSERT directive with two > symbols and waits for the data to be entered. Query Update continues to make insertions and replies with the awaiting data symbols after each transmission until the user enters *END to terminate the input. If the sequence of data is not to be input from the terminal, the FROM option specifies a file as the alternate source. (For more information see the External File Organization subsection.)

The data items entered for each record must be in the order specified by the data names in the USING option. A data item within the string of data items can be omitted by entering two consecutive delimiters ($$) for the data item. If the string of data items is terminated before values are entered for all data names, the remaining data items are automatically omitted. Each omited data item is initialized to its null value (blank or zero) in the record being inserted. When a nonnumeric literal is entered for a data item, the delimiter must be used to delimit the literal unless the SEPARATOR ITEM-SIZE directive has been specified. Values must be properly sized if the SEPARATOR ITEM-SIZE directive is in effect.

The VETO option causes the record to be displayed before it is inserted. The user responds with YES or Y (insert the record), NO or N (do not insert the record), PROCEED or P (negate the current VETO option and insert the record), or EXIT or E (do not insert the record and terminate the transmission).

The PASS option disables the VETO for this transmission when a previous VETO directive specified VETO ON.

When the current area is described in the subschema as having actual key file organization, the primary key cannot be specified in the INSERT directive; neither the data name of the primary key nor a primary key value can be specified. In actual key file organization, primary key values are assigned by the system. An INSERT directive with no options included should be followed immediately in the same transmission by a MOVE directive. The MOVE directive places data in the record to be inserted; if a MOVE directive is not specified, a blank record is inserted in the area. Alternate key fields (specified in the subschema) can be included in the INSERT (or MOVE) directive; the record can be subsequently retrieved by alternate key.

A relation joining several areas can be updated by modifying each area separately. When more than one area is in use at one time, Query Update must be able to identify which area is to be updated. The identification of an area to be updated can be derived by Query Update from the data names in the INSERT directive, if there are no duplicate data names in all the areas in use or if the data names are properly qualified.

If an INSERT directive is used to create a new record where only the value of its primary key is given by the user and the other items are set to null values by Query Update, there is no way to identify the area because no data name accompanies the directive. Format 2 of the UPDATE directive permits the area to be identified before the INSERT directive is entered; in addition, it reduces the amount of qualification needed for other types of updates. Refer to the description of the UPDATE directive for further details.

Query directive for a relation can be interspersed with updating directives for one area, but they cannot be combined in the same transmission.

The user is responsible for maintaining the integrity of the data base and must ensure that the updating operation performed is consistent with the meaning of the relation defined for the areas.

UPDATE

The UPDATE directive has two different functions: to obtain records from the data base for modification and to identify the area for which subsequent updating directives are to apply. Format 1 of the UPDATE directive retrieves a record from the data base, places it in a work area where modifications can be made, and returns the record to the data base. Only one area can be updated at a time; each area joined in a relation must be updated separately. A record is selected for updating either by referencing the record key literal in the UPDATE directive or as the result of an IF directive with an associated UPDATE directive. Format 2 of the UPDATE directive is entered before the user begins updating an area to identify the area to be updated. The formats of the UPDATE directive are shown in figure H-4.

```
Format 1

        ┌KEY data-name                              ┐
        │literal                                    │ ┌VETO┐
UPDATE  │                                           │ │PASS│
        │[FROM file-name] USING ┌data-name ...┐     │ └    ┘
        │                       └SAME         ┘     │
        └                                           ┘
Format 2

UPDATE area-name
```

Figure H-4. UPDATE Directive Formats

The UPDATE directive requires permission to modify or extend the area file. This permission is granted according to the passwords specified for the area file in the USE directive.

The KEY option in format 1 specifies a temporary data item that contains the key to select the record to be updated. The literal option identifies the record to be updated by specifying the value for the record key. When the literal option is used in conjunction with an IF directive, the record is retrieved before the condition is tested; the record can be updated only when the condition is true. The data name must contain either a character value or numeric value.

The USING option indicates that a sequence of data name values is provided for each record to be updated. The data names must be defined in the same record description for the area file. If the FROM option is included, the values are provided from the specified file. For more information see the External File Organizations subsection. If the FROM option is not included, the values are entered through subsequent transmissions. One of the data names must specify the record key; all others identify data items to be updated. For each record to be updated, the values entered correspond respectively to the data names specified in the USING option. An UPDATE directive with the USING option cannot be associated with an IF directive that references an area item.

UPDATE USING can be used to update the 10-character user identification field, USER-ID. The USER-ID field is used by CDCS for all data base procedures and for log file entries; it can provide an additional privacy check. The field must be updated before the first USE, INVOKE, CREATE, or VERSION directive.

When the records to be updated contain primary and alternate keys, the data names specified in the USING option must conform to one of the following rules:

> If records are to be selected for updating by primary key, the primary key must be specified; additional data names can reference alternate keys or any other fields in the reocrd.
>
> If records are to be selected for updating by alternate key, the desired alternate key must be specified; additional data names must not reference the primary key or any other alternate keys in the record.

Updating by alternate key can result in several records being updated at one time. All records containing the specified alternate key value are updated in the same way. When an indexed sequential file contains records with duplicate primary keys, all records having duplicate keys are updated.

An *END or an end-of-file terminates the update operation. SAME indicates that the previously specified sequence of data names is to be used. However, if any directive (such as USE or RETURN) that alters or eliminates any of the data names in the USING sequence is entered between the specification of the USING list and the UPDATE USING SAME directive, the results are unpredictable.

The data items entered for each record must be in the order specified by the data names in the USING option. A data item within the string of data items can be omitted by entering two consecutive delimiters ($$) for the data item. If the string of data items is terminated before values are entered for all data names, the value of the remaining data items is contingent on the SEPARATOR ITEM-SIZE option. If the option is not in effect, the remaining data items are omitted from the update operation and retain the value stored in the record. If the SEPARATOR ITEM-SIZE option is in effect, the remaining data items are filled with enough blanks or zeros to complete the field.

The VETO option in an UPDATE directive causes the record to be displayed before it is updated. Either the first 40 characters of the record or the data items specified in a previous VERIFY directive are displayed at the terminal. The user then responds with one of the following entries:

YES (or Y)

> Updates the record.

NO (or N)

> Does not update the record.

PROCEED (or P)

> Negates the current VETO option for the transmission; acts as a user response of YES for this record.

EXIT (or E)

> Prevents update and terminates the transmission.

The PASS option disables the VETO for the transmission when a previous VETO directive specified VETO ON.

Format 1 of the UPDATE directive can be used two different ways. The basic difference between the two usages is the manner in which record selection and modifications are expressed.

> The UPDATE directive selects the record to be updated when the KEY or literal option is specified or as the result of a preceding IF directive. Items to be modified, as well as the modifications, are specified in a MOVE directive following the UPDATE directive. Examples 1 and 2 in figure H-5 illustrate the separation of record selection and modification. If the data name in the IF directive in example 2 is not the record key, the entire file is searched.
>
> The UPDATE directive combines record selection and modification when the USING option is specified. This usage of the UPDATE directive is effective when a number of records require modification of the same data name. Only one directive is required to select and modify one or more records. Examples 3 and 4 in figure H-5 illustrate combining record selection and modification.

Temporary items can also be updated with the UPDATE directive. The USING option specifies the data name of the temporary item. The data to be stored in the temporary item is then entered by the user. Only one user response is accepted by Query Update when the USING option specifies only temporary items. If the FROM option is also used, only one set of values is accepted.

Example 1

```
--
        UPDATE 312654 MOVE $PLASTIC ALLOYS CORP$ TO VENDOR-NAME
```

The record with 312654 in the key data item is retrieved for modification. The MOVE directive specifies the data to be stored in the data item VENDOR-NAME.

Example 2

```
--
        IF VENDOR-NO EQ 312654 UPDATE +
        MOVE $PLASTIC ALLOYS CORP$ TO VENDOR-NAME
```

If VENDOR-NO is not the record key, each record in the file is checked for a VENDOR-NO data item that is equal to 312654. If the condition is true, the record is modified as specified in the MOVE directive.

Example 3

```
--
        UPDATE USING VENDOR-NO STATE-OF-ADDRESS
>>
        312654 $CAL$
>>
        312660 $CAL$
>>
        *END
```

VENDOR-NO is the data-name of the record key used to select records for modification. STATE-OF-ADDRESS is the data-name of a data item to be modified. Each response by the user causes a record to be retrieved and modified.

Example 4

```
--
        UPDATE FROM VNDFILE USING VENDOR-NO VENDOR-NAME
```

VNDFILE contains a series of keys (VENDOR-NO) and names (VENDOR-NAME) terminated by end-of-information.

Example 5

```
--
        UPDATE VENDORS
```

VENDORS is the name of the area to be updated.

Figure H-5. Sample UPDATE Directives

When the current area is described in the subschema as having actual key file organization, the primary key must be known before it can be specified in the UPDATE directive with the literal option. In actual key file organization, primary key values are assigned by the system. The user must display records in the file or otherwise dump the contents of the file to obtain the key values assigned by the system before records can be updated.

Format 2 of the UPDATE directive permits an area to be identified for updating before a DELETE, INSERT, or format 1 UPDATE directive is entered. When more than one area is being referenced during a session, as is the case in relation processing, Query Update must be able to determine which area is to be updated when an updating directive is issued. Query Update can make this determination from the data names in the directive if no duplicate data names exist in all the areas in use, or if the data names are properly qualified. When format 2 of the UPDATE directive is used, the area is explicity identified and the need for qualification is eliminated.

The area name specified must be an area established in the subschema currently in use. All subsequent updating operations are performed on the area specified. The designated area is identified for updating until another format 2 UPDATE directive is issued or the Query Update session is terminated. Example 5 in figure H-5 illustrates the use of the format.

Query directives for a relation can be interspersed with updating directives for one area, but they cannot be combined in the same transmission. When updating an area, all directives in the transmission must also refer to that area.

The user is responsible for maintaining the integrity of the data base and must ensure that the updating operation performed is consistent with the meaning of the relation defined for the areas. When modifying join terms, the user must consider whether the corresponding data items in other records must be modified as well, and issue any UPDATE directives needed to maintain the relation.

The UPDATE directive cannot be used when the area file specified through the USE directive is a sequential file.

USE

The USE directive determines either CRM data base access mode or CDCS data base access mode for subsequent directives. The USE directive establishes the areas relations, and the subschema directory that Query Update uses for subsequent directives. When this directive is encountered, Query Update initializes its internal data base access tables. If used in a transmission with other directives, USE must be the last directive in the transmission. The formats of the USE directive are shown in figure H-6.

Only one USE directive can be in effect at a time. Up to 64 areas or 59 relations can be made available by a USE directive. When a USE directive is entered, the subschema and all areas and index files that were made available by a previous USE directive are released. The first attach and open of any area and index file are delayed until the processing of the first directive that actually uses the area.

The USE directive determines whether Query Update enters CRM data base access mode or CDCS data base access mode. The USE directive specifies a subschema. If that subschema is a Query Update/CRM subschema, Query Update enters CRM data base access mode. If the subschema is a Query Update/CDCS subschema, Query Update enters CDCS data base access mode.

CRM DATA BASE ACCESS MODE

Format 1 of the USE directive makes available to the Query Update user all relations and all areas described in the subschema referenced by the subschema name. When this directive is specified, the user can proceed immediately with queries referencing any relation or area and can update any area in the specified subschema. The relevant areas and index files are attached by Query Update using the permanent file parameters recorded in the subschema directory. Refer to the Operating System section for permanent file parameters.

The PW option is PW with no associated value specified in the USE directive. When the PW option is specified, Query Update responds in insert mode to allow the user to replace or supplement permanent file parameters obtained from the subschema directory. To exercise the PW option, the user enters PW with no associated value as if it were the last permanent file parameter entered following the subschema name in the USE directive.

In this format and in the following formats of the USE directive, by specifying the PW or MODIFIED option the user indicates that Query Update is to use the values entered after the USE directive to supplement or replace permanent file parameters obtained from the subschema directory. The use of the PW or MODIFIED option does not alter any value stored in the subschema directory; Query Update uses the values entered for the current session or until another USE directive is entered. If the PW or MODIFIED option is not exercised, the area and index file permanent file parameters are obtained from the subschema directory or from the permanent file parameters entered in the USE directive. If the PW or MODIFIED option is exercised, Query Update recognizes the option, prompts the user for permanent file parameters with the characters >>>, and waits for the values to be entered. When Query Update prompts with the characters >>>, Query Update is in insert mode.

In insert mode, the format of the permanent file parameter information entered by the user is the area name or index file name followed by permanent file keywords and values. An optional equals sign can link the parameter keyword to its value. A separate transmission is required for each area or index file that requires permanent file parameter modification. To terminate the operation, the user enters an *END in the first column following the prompt; Query Update then terminates insert mode.

```
Format 1

    USE sub-schema-name[(permanent-file-parameters [PW])]

Format 2

    USE relation-name ... [MODIFIED]

        OF sub-schema-name[(permanent-file-parameters [PW])]

Format 3

    USE area-name [(permanent-file-parameters [PW])]

        [INDEX IS index-file-name [(permanent-file-parameters[PW])]]

        OF sub-schema-name[(permanent-file-parameters [PW])]
```

Figure H-6. USE Directive Formats

In insert mode, all applicable permanent file keywords and their associated values can be entered in one transmission. Each keyword must be followed by only one value with the exception of PW under the NOS/BE operating system, which can be followed by no value or up to five values. With the exception of PW under the NOS/BE operating system, the parameters specified are used instead of those already recorded in the subschema directory, or they are used in addition to those recorded if no previous parameters exist. Under the NOS/BE operating system, when the PW parameter is entered (in insert mode) with no values specified, all previous passwords are ignored; when the PW parameter is entered with values specified, the values are used in addition to those already recorded in the subschema directory. If there is an error in a transmission, Query Update ignores the portion of the transmission following the error.

In the use of the PW and MODIFIED options, the privacy of the permanent file passwords is maintained as the passwords are replaced by the symbol *---* in the trace or output areas.

Format 1, although most convenient, can carry an expensive overhead if the subschema contains many area descriptions. Central memory must be allocated for each relation and each area because it is not known in advance which ones are to be accessed by the user. More computer time is consumed whenever a data name is supplied by the user because Query Update must search all area descriptions to detect duplicate names. Format 1 is recommended for use when a subschema describes very few areas and relations, all of which are usually used together in a Query Update session.

Format 2 is used to specify only the relations that are to be available to the Query Update user. The relation names specified must name relations established in the subschema referenced by the subschema name. Permanent file parameters required to access the subschema must also be included.

In both format 1 and format 2, privacy can be maintained with a permanent file read password which is required to access the subschema.

The MODIFIED and PW options, as described in format 1, are available if the user intends to replace or supplement permanent file parameters obtained from the subschema directory with values entered after the USE directive. To exercise the MODIFIED option, the user enters MODIFIED after the relation name in the USE directive. To exercise the PW option, the user enters PW with no associated value as if it were the last permanent file parameter following the subschema name in the USE directive.

Example 1 in figure H-7 adds a password for each area and index file in the relation RELATION1 to supplement the ID parameter already stored in the subschema SUBSC.

Example 1.

```
-- USE RELATION1 MODIFIED OF SUBSC +
   (ID XYZ)

>>> PRODUCTS PW PDC2

>>> XPRODUCTS PW PDC2

>>> EMPLOYEES PW CLK2

>>> CONTRACTS PW HFR2

>>> *END
```

Example 2.

```
-- USE AREAL (ID=QUFIG PW) OF QUSUB +
   (ID=QUFIG)

>>> AREAL PW PW A B C ID=NEW CY=4

>>> *END
```

Figure H-7. USE Directive Examples

Format 3 is used to establish a single area and a subschema directory that Query Update uses for subsequent directives. It can also be used to override the permanent file parameters of an independent area that is not part of a relation.

The area name specified must be an area established in the subschema referenced by the subscheme name. Permanent file parameters required to access the subschema must also be specified. The area premanent file parameters are specified unless the subschema designates the area as TEMPORARY.

Under the NOS operating system, an area that is used for delete or update operations cannot be accessed by another user. Under the NOS/BE operating system, an area that is used for delete or update operations can be accessed by another user with read permission only. This remains in effect for the length of the transmission only.

The INDEX option is required when the area contains primary and alternate keys. The index file name identifies the indes file that was generated when the area was created. If permanent file parameters are specified for the area, permanent file parameters must also be specified for the index file. Permanent file parameters can be omitted from the INDEX option only if the area permanent file parameters are omitted. In this situation, Query Update expects both the area and the index file to be local files that are available when the USE directive is processed; the area must be designated TEMPORARY in the subschema.

The PW option, as described in format 1, is available if the user intends to replace or supplement permanent file parameters obtained from the subschema directory with values entered after the USE directive. To exercise the PW option, the user enters PW with no associated value as if it were the last permanent file parameter following the area name, the index file name, or the subschema name in the USE directive.

Example 2 of figure H-7 illustrates the use of the PW option to replace and supplement permanent file parameters for the area AREA1. The PW entered in the USE directive specifies this option. The first PW entered after the area name causes Query Update to ignore the area passwords recorded in the subschema directory. The second PW entered after the area name causes Query Update to use the subsequent passwords. The specification of the ID and CY parameter keywords causes Query Update to use the value entered after the keyword instead of the value recorded in the subschema directory.

A general distinction between the USE and CREATE directives concerns the files they reference. The USE directive attaches a permanent file; CREATE initializes a local file. If a file has been designated as TEMPORARY in the subschema directory, the USE directive also references a local file. The permanent file parameters option is not included for a temporary file.

CDCS DATA BASE ACCESS MODE

If the USE directive specifies a Query Update/CDCS subschema, it is treated as a format 1 USE and makes all area and relations available for single-area queries, relational queries, and single-area updates. Usage of formats 2 or 3 to specify a Query Update/CDCS subschema results in an informative diagnostic message. Query Update ignores such options as MODIFIED or INDEX, and processes the directive as if format 1 were specified. In either case, CDCS attaches all areas referenced by the subschema.

The USE directive cannot be used if the subschema name differs from the subschema library name, because the syntax does not provide for specification of both names. The INVOKE directive can be used instead.

The USE directive makes available to the Query Update user all relations and all areas described in the subschema referenced by the subschema name. After the USE directive is specified, the user can proceed with queries referencing any relation or area and can update any area in the specified subschema. The relevant areas and index files are attached by Query Update, using the permanent file parameters recorded in the subschema directory. Refer to the Operating System section for permanent file parameters.

If the USE directive makes available a relation that contains a RESTRICT on a data name that is not within the record, this data name must be defined with the DEFINE directive prior to any query using the relation. The data name must be defined with the same type and length as the subschema identifier used for comparison.

The USE directive can specify any permanent file parameters that are required to attach the subschema library file. If PW is specified, but not followed by a parameter, permanent file parameters are not listed on the output or trace listing. Query Update then responds with >>> and the user enters:

```
    file-name, PW=param-1 [param-2 . . .]
```

This input is listed on the output or trace listing with the parameter values replaced by *----*; thus, permanent file parameter security is provided. Query Update responds with >>> until the user enters *END.

When CDCS catalog mode is active, the USE directive must specify a Query Update/CDCS subschema and this subschema must be the same one specified by the VERSION directive. If CDCS catalog mode is not active, the USE directive causes the previous subschema to be released. All internal descriptions of the previous subschema, areas, and relation are released. If Query Update is in CDCS data base access mode, CDCS is informed that the previous subschema is no longer in use.

SUMMARY OF DATA DEFINITION IN DMS-170 I

This summary includes all the clauses or statements that can be used in defining data items. Table I-1 shows the schema definition required for data items of each schema data class and the subschema definitions that correspond to each schema data class.

For Query Update access to schema-defined data base files in CYBER Record Manager (CRM) data base access mode, the data base must be defined in the Query Update subschema exactly as the data base is defined in the schema. Therefore, every data item must be defined to correspond in size and class to the schema definition of the item.

For COBOL, FORTRAN, and Query Update access to schema-defined files in CYBER Database Control System (CDCS) data base access mode, the definition of data items in the subschemas does not have to correspond exactly to the schema definition of data items. Through mapping, CDCS can generate a record image conforming to the subschema format from a record in schema format, or can perform the conversion of data from subschema format to schema format. Detailed information about conversions allowed is included in the CDCS 2 Data Administration reference manual or in the CDCS 2, the DDL 3 volume 1, and the DDL 3 volume 2 reference manuals.

TABLE I-1. DATA DEFINITION IN DMS-170

SCHEMA					COBOL Subschema		FORTRAN 5 Sub-schema Type Statement	Query Update Sub-schema in CDCS Data Base Access Mode		Query Update Sub-schema in CRM Data Base Access Mode	
Data Class No.	Data Class Name	PICTURE Clause	TYPE Clause	Internal Representation	PICTURE Clause	USAGE Clause		PICTURE Clause	USAGE Clause	PICTURE Clause	USAGE Clause
0	Display alpha-numeric	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	CHARACTER	Display code, alphanumeric	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)	CHARACTER	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)	Alpha-numeric (A X 9; not all As or 9s; mixed specification used as all Xs)	DISPLAY (or none)
1	Display alpha-betic	Alpha-betic (A)	None	Display code, alphabetic	Alpha-betic (A)	DISPLAY (or none)	CHARACTER	Alpha-betic (A)	DISPLAY (or none)	Alpha-betic (A) (or none)	DISPLAY (or none)
3	Display integer	Numeric (9 T)	None	Display code numeric, can have sign overpunch in last character position	Numeric (9 S)	DISPLAY COMP (or none)	None†	Numeric (9 S and insertion and re-placement characters)	DISPLAY COMP (or none)	Numeric (9 S and insertion and re-placement characters)	DISPLAY COMP (or none)
4	Display fixed	Numeric (9 P T V)	None	Display code numeric plus implicit or explicit deci-mal or scaling position	Numeric (9 S V P)	DISPLAY COMP (or none)	None†	Numeric (9 S V P) and inser-tion and replacement characters)	DISPLAY COMP (or none)	Numeric (9 S V P and inser-tion and replacement characters)	DISPLAY COMP (or none)
10	Coded binary integer	None	FIXED integer-1 integer-2 (where the inte-ger value is 1 through 18)	Binary integer	Numeric (9 S V P)	COMP-1 INDEX ††	INTEGER LOGICAL BOOLEAN	Numeric (9 S V P and inser-tion and replacement characters)	COMP-1 INDEX †† LOGICAL	Numeric (9 S V P and inser-tion and replacement characters)	COMP-1 INTEGER LOGICAL

TABLE I-1. DATA DEFINITION IN DMS-170 (Contd)

SCHEMA					COBOL Subschema		FORTRAN 5 Sub-schema Type Statement	Query Update Sub-schema in CDCS Data Base Access Mode		Query Update Sub-schema in CRM Data Base Access Mode	
Data Class No.	Data Class Name	PICTURE Clause	TYPE Clause	Internal Representation	PICTURE Clause	USAGE Clause		PICTURE Clause	USAGE Clause	PICTURE Clause	USAGE Clause
13	Coded floating point normalized	None	FLOAT integer-1 (where the integer value is 1 through 14)	Signed, normalized floating point (1 word)	Numeric (9 S V P)	COMP-2	REAL BOOLEAN	Numeric (9 S V P and insertion and replacement characters)	COMP-2	Numeric (9 S V P and insertion and replacement characters)	COMP-2
14	Coded double precision	None	FLOAT integer-1 (where the integer value is 15 through 29)	Signed, normalized floating point (2 words)	None†	None†	DOUBLE PRECISION	Numeric (9 S V P and insertion and replacement characters)	DOUBLE	Numeric (9 S V P and insertion and replacement characters)	DOUBLE
15	Coded complex	None	COMPLEX	Floating point with real part and imaginary part (2 words)	None†	None†	COMPLEX	Numeric (9 S V P and insertion and replacement characters)	COMPLEX	Numeric (9 S V P and insertion and replacement characters)	COMPLEX

†No corresponding subschema type. Valid conversion shown in the CDCS 2 Data Administration reference manual or in the DDL3 volume 2 or FORTRAN Data Base Facility reference manuals.

††No picture clause allowed.

FUTURE SYSTEM MIGRATION GUIDELINES J

This appendix contains programming practices recommended by CDC for users of the software described in this manual. When possible, application programs based on this software should be designed and coded in conformance with these recommendations.

Two forms of guidelines are given. The general guidelines minimize application program dependence on the specific characteristics of a hardware system. The feature use guidelines ensure the easiest migration of an application program to future hardware or software systems.

GENERAL GUIDELINES

Good programming techniques always include the following practices to avoid hardware dependency:

> Avoid programming with hardcoded constants. Manipulation of data should never depend on the occurrence of a type of data in a fixed multiple such as 6, 10, or 60.
>
> Do not manipulate data based on the binary representation of that data. Characters should be manipulated as characters, rather than as octal display-coded values or as 6-bit binary digits. Numbers should be manipulated as numeric data of a known type, rather than as binary patterns within a central memory word.
>
> Do not identify or classify information based on the location of a specific value within a specific set of central memory word bits.
>
> Avoid using COMPASS in application programs. COMPASS and other machine-dependent languages can complicate migration to future hardware or software systems. Migration is restricted by continued use of COMPASS for stand-alone programs, by COMPASS subroutines embedded in programs using higher-level languages, and by COMPASS owncode routines used with CDC standard products. COMPASS should only be used to create part or all of an application program when the function cannot be performed in a higher-level language or when execution efficiency is more important than any other consideration.

FEATURE USE GUIDELINES

The recommendations in the remainder of this appendix ensure the easiest migration of an application program for use on future hardware or software systems. These recommendations are based on known or anticipated changes in the hardware or software system, or comply with proposed new industry standards or proposed changes to existing industry standards.

DMS-170

DMS-170 offers several features among which choices must be made. The following paragraphs indicate preferred usage of CDCS, DDL, and of Query Update in support of CDCS.

Multiple Record Descriptions

Do not include multiple record descriptions on a single file.

Repeating Groups

Avoid the use of the OCCURS clause, repeating groups, or arrays within records; as an alternative, the repeating data can be normalized into separate records on a different file. If repeating data must be used, limit usage to fixed length groups (no OCCURS DEPENDING ON clause) and to simple (unnested) OCCURS clauses.

Alternate Keys on Repeating Groups

Avoid the specification of alternate keys on repeating groups. The data can be normalized as indicated under Repeating Groups.

Collating Sequence

Use the default collating sequence or the ASCII collating sequence.

REDEFINES Clause

Use the REDEFINES clause only for alphanumeric-to-alphanumeric redefinitions, where the term alphanumeric has the meaning assigned by COBOL to data. In general, avoid the use of REDEFINES where use is based on a knowledge of the internal representation of data (floating-point layout, number of characters per word, and so forth).

Query Update Syntax

Use the new directives INVOKE, STORE, MODIFY, and REMOVE instead of the directives USE, INSERT, UPDATE, and DELETE.

INDEX

COMMENT SHEET

MANUAL TITLE: Query Update Version 3 Reference Manual

PUBLICATION NO.: 60498300

REVISION: J

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments on the back (please include page number references).

_____ Please reply _____ No reply necessary

CUT ALONG LINE

FOLD FOLD

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MN.

POSTAGE WILL BE PAID BY ADDRESSEE

CONTROL DATA

Technology and Publications Division
Mail Stop: SVL104
P.O. Box 3492
Sunnyvale, California 94088-3492

FOLD FOLD

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.
FOLD ON DOTTED LINES AND TAPE

NAME:

COMPANY:

STREET ADDRESS:

CITY/STATE/ZIP:

TAPE TAPE