DMS-170

QUERY UPDATE
VERSION 3
APPLICATION
PROGRAMMING
USER'S GUIDE

For Use With:
CYBER Record Manager

CDC® OPERATING SYSTEMS:
NOS 1
NOS 2
NOS/BE 1
<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (04/01/77)</td>
<td>Original release.</td>
</tr>
<tr>
<td>B (08/15/78)</td>
<td>Updated to reflect Query Update Version 3.2, PSR level 472.</td>
</tr>
<tr>
<td>C (07/23/82)</td>
<td>Released at PSR level 564. This guide reflects Query Update Version 3.4 (or Version 3.3 for the NOS 1 user). The changes include the description of the MODIFY, REMOVE, STORE, and INVOKE directives, which replace the INSERT, UPDATE, DELETE, and USE directives respectively. The changes also include the description of the Query Update language under the NOS 2 operating system. This is a complete reprint.</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Page</th>
<th>Revision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Front Cover</td>
<td>-</td>
</tr>
<tr>
<td>Inside Front Cover</td>
<td>C</td>
</tr>
<tr>
<td>Title Page</td>
<td>-</td>
</tr>
<tr>
<td>ii</td>
<td>C</td>
</tr>
<tr>
<td>iii/iv</td>
<td>C</td>
</tr>
<tr>
<td>v</td>
<td>C</td>
</tr>
<tr>
<td>vi</td>
<td>C</td>
</tr>
<tr>
<td>vii</td>
<td>C</td>
</tr>
<tr>
<td>viii</td>
<td>C</td>
</tr>
<tr>
<td>1-1</td>
<td>C</td>
</tr>
<tr>
<td>2-1 thru 2-10</td>
<td>C</td>
</tr>
<tr>
<td>3-1 thru 3-8</td>
<td>C</td>
</tr>
<tr>
<td>4-1 thru 4-4</td>
<td>C</td>
</tr>
<tr>
<td>5-1 thru 5-5</td>
<td>C</td>
</tr>
<tr>
<td>6-1 thru 6-24</td>
<td>C</td>
</tr>
<tr>
<td>7-1 thru 7-9</td>
<td>C</td>
</tr>
<tr>
<td>8-1 thru 8-6</td>
<td>C</td>
</tr>
<tr>
<td>A-1 thru A-4</td>
<td>C</td>
</tr>
<tr>
<td>B-1</td>
<td>C</td>
</tr>
<tr>
<td>B-2</td>
<td>C</td>
</tr>
<tr>
<td>Index-i thru -3</td>
<td>C</td>
</tr>
<tr>
<td>Comment Sheet</td>
<td>C</td>
</tr>
<tr>
<td>Mailing</td>
<td>-</td>
</tr>
<tr>
<td>Back Cover</td>
<td>-</td>
</tr>
</tbody>
</table>
This user's guide describes the Query Update Version 3.4 language, which is designed for data storage and retrieval operations. Query Update operates under control of the following operating systems:

NOS 1 for the CONTROL DATA® CYBER 170 Computer Systems; CYBER 70 Computer System models 71, 72, 73, 74; 6000 Computer Systems

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

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

The Query Update language supports a wide variety of applications ranging from simple data file query to complex report production. This guide is organized by application; directives are introduced as they become appropriate. Some feature enhancements are not available to the NOS 1 user; these enhancements are noted in the guide.

Sections 1 through 5 discuss file concepts and directives that apply to data access and manipulation. Section 6 details report writing facilities and directives that apply to report catalog maintenance and production. Section 7 presents operations within a multiple-file data base environment. Section 8 illustrates special Query Update utility operations. All data base access is described for use with CYBER Record Manager.

This guide is designed for programmers who are familiar with Control Data standard software products. The Query Update reference manual should be consulted for additional information regarding the Query Update language.

Related material is contained in the following publications. The NOS 1, NOS 2, and NOS/BE 1 manual abstracts are pocket-sized manuals containing brief descriptions of the contents and intended audience of all NOS 1 and NOS 1 product set manuals, NOS 2 and NOS 2 product set manuals, and NOS/BE 1 and NOS/BE 1 product set manuals, respectively. The abstract manuals can be used to determine which manuals are of greatest interest. The Software Publications Release History can be used to determine 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:

<table>
<thead>
<tr>
<th>Publication</th>
<th>Publication Number</th>
<th>NOS 1</th>
<th>NOS 2</th>
<th>NOS/BE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Update Version 3 Reference Manual</td>
<td>60498300</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Query Update Version 3 User's Guide For Use With: CYBER Record Manager</td>
<td>60387700</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

The following manuals are of secondary interest:

<table>
<thead>
<tr>
<th>Publication</th>
<th>Publication Number</th>
<th>NOS 1</th>
<th>NOS 2</th>
<th>NOS/BE 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CYBER Record Manager Basic Access Methods Version 1.5 Reference Manual</td>
<td>60495700</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>DNS-170 Query Update/CYBER Record Manager Data Administration Reference Manual</td>
<td>60482100</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>INTERCOM Version 5 Reference Manual</td>
<td>60455010</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Publication</td>
<td>Publication Number</td>
<td>NOS 1</td>
<td>NOS 2</td>
<td>NOS/BE 1</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------------</td>
<td>-------</td>
<td>-------</td>
<td>----------</td>
</tr>
<tr>
<td>NOS Version 1 Manual Abstracts</td>
<td>84000420</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOS Version 1 Reference Manual, Volume 1 of 2</td>
<td>60435400</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>NOS Version 2 Manual Abstracts</td>
<td>60485500</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>NOS Version 2 Reference Set, Volume 3, System Commands</td>
<td>60459680</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>NOS/BE Version 1 Manual Abstracts</td>
<td>84000470</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>NOS/BE Version 1 Reference Manual</td>
<td>60498300</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Software Publications Release History</td>
<td>60481000</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

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

This guide describes a subset of the features and directives documented in the Query Update Version 3 Reference Manual. Control Data cannot be responsible for the proper functioning of features or directives not documented in the Query Update reference manual.
QUERY UPDATE AS A PROGRAMMING LANGUAGE

Query Update is a programming language that can be used to perform the following data processing operations:

- Store and remove data in existing files
- Modify data in records in existing files
- Sort and display data
- Perform arithmetic operations
- Generate special-purpose reports
- Prepare report formats and preserve them for later use
- Create data files

Instructions are input to the Query Update program in statement form. Each statement, called a Query Update directive, specifies one task to be performed. Query Update can be called to perform a single task or a group of tasks that collectively represent one Query Update session.

THE QUERY UPDATE USER

Query Update meets the needs of a variety of users. For example:

- A nonprogrammer can access data files to display selected fields of information, perform minor modifications, prepare simple reports, or request previously generated reports with only a few easy-to-learn instructions.

- A business programmer can search and manipulate data files through free-form directives rather than through complete program runs, and can construct and preserve various report layouts for subsequent reporting on data files.

- A scientific programmer can build individual data files either from scratch or from existing files, and perform any number of calculations.

COMPARING QUERY UPDATE TO COBOL

Query Update is a free-form data processing language. It contains no program divisions and imposes few restrictions on the order in which statements are submitted.

The Query Update language is similar to COBOL, which is also a data processing language, in the following ways:

- Both group English terms into sentence-type instructions.
- Both use reserved words such as DISPLAY, SORT, EQ, and IF in each instruction.
- Both require descriptions of the file organization and storage of data and the characteristics of data.

The terminology used in the descriptions is similar, but the methods for relaying the information to the program and operating system are different.

When Query Update is accessing a permanent data base file, the file organization and data description for the file must be predefined. Predefinition is performed by the data administrator; a program is written in RMS-170 Data Description Language (DDL), it is compiled into a DDL object directory, then it is stored into a permanent file via operating system procedures. The DDL object directory is called a Query Update subschema. One or more compiled Query Update subschemas are stored on a permanent file called a subschema library. The COBOL counterparts of a subschema are the Environment and Data Divisions.

When Query Update is accessing a sequential file that is not part of a permanent data base (this type of file is called a non-data-base file), the subschema concept does not apply. The physical storage characteristics of a non-data-base file are declared in the FILE control statement. The data description is generated within the series of directives that make up a Query Update session.
Before you consider the actual structure of the Query Update language, you should become familiar with how data is classified and handled within the Query Update data and processing environments.

THE DATA ENVIRONMENT

Query Update handles three distinct categories of data:

- Data base file data items
- Non-data-base file data items
- Temporary data items for operations related to data base or non-data-base files

A data base file is a file whose organization and content are described by a Query Update subschema. As mentioned in the previous section, the subschema must have been written in DDL source language by the data administrator, compiled into a DDL object directory, and stored onto a permanent file (called a subschema library) before the data base can be accessed. A data base file is accessed during a Query Update session through the CREATE or INVOKE directive.

A non-data-base file is a sequential file whose organization and content is not described by a Query Update subschema. A typical non-data-base file might be a transaction file that has been read in from tape for on-time reporting or data base updating purposes. The directory of a non-data-base file must be generated through the DESCRIBE directive before it can be accessed. A non-data-base file is accessed during a Query Update session through the DISPLAY or EXTRACT directives.

A temporary data item is an item that is defined with a DEFINE, DESCRIBE, or SPECIFY directive during a Query Update session. Once a temporary data item is defined, it can be used for comparing data items in a data base or non-data-base file or for modifying data items in a data base file.

QUERY UPDATE LANGUAGE COMPONENTS

Query Update language components include Query Update reserved words, recognized symbols, punctuation, and user-supplied elements (data-names, literals, expressions, conditions, and picture specifications). These components are grouped together into statements for input to the Query Update program. Each complete statement is called a directive. Each directive represents one task to be performed by the computer.

A reserved word is always the first word in a directive and identifies the task to be performed. The keyword can be followed by a number of user-supplied elements as well as additional Query Update keywords. For example, the following directive asks Query Update to make a subschema and its associated data base files available to the tasks that will follow it:

```
INVOKESUBSHEMNAME
FROMLIBRARYFRENANTFILENAME
(PERMRENTFILEPARAMETERS[FW])
```

Operating system parameters are required when permanent files are referenced in a directive.

The following directive describes the first data item in a non-data-base file directory:

```
DESCRIBENONDATAFILENAME
ANDDATANAME
ASCOLLECTBY$X(10)$
```

The $ delimiters are required when a non-numeric literal is specified.

The following directive defines a temporary item for use in subsequent tasks:

```
DEFINENATANMEASNUHRECICBY999
```

The $ delimiters are not required when a numeric literal is specified.

A summary of Query Update directives is shown in table 2-1. The directives are listed in alphabetic order by reserved word, which identifies the purpose of the directive. The comments column provides some rules and default options for each directive. See the Query Update Version 3 Reference Manual for details about the directives.
<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER</td>
<td>Locates retained report directives from the catalog for subsequent modification.</td>
<td>The designated report must have been created by a previous FORMAT directive and must reside on the current catalog.</td>
</tr>
<tr>
<td>BREAK</td>
<td>Indicates situations that cause 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.</td>
<td>BREAK is associated with a HEADING and FOOTING directive. Level number 0 cannot be specified.</td>
</tr>
<tr>
<td>COMPILE</td>
<td>Stores report specifications in encoded form on a table file.</td>
<td>After the table file has been generated by the COMPILE directive, it is available for input to the Report Utility. The REPORT control statement calls the Report Utility program to produce reports according to specifications in the table file.</td>
</tr>
<tr>
<td>CREATE</td>
<td>Establishes a data base file, known as an area, for subsequent insertion of data.</td>
<td>When alternate keys are defined for the area, the INDEX option is required if the index file name is not specified in the subschema. The FROM LIBRARY option is required if the subschema library permanent file name is different from the subschema name. The area and applicable index file must be made permanent if the files are to be saved for future use.</td>
</tr>
<tr>
<td>DATE</td>
<td>Specifies use and positioning of system-supplied date information within a report.</td>
<td>Default vertical positioning is line 1. Default horizontal positioning is column 2. Floating point is the default internal format. When the ITEMS option is included, an array is generated, and subscripting is subsequently required. When temporary data items are to be evaluated, the method of evaluation must be specified through the VALUE (or =) option.</td>
</tr>
<tr>
<td>DEFINE</td>
<td>Establishes temporary data names and storage requirements.</td>
<td>FILLER should be used for those portions of the record that are not to be referenced. When tag numbers are included, SELECT directives are required. The AT LINE default is positioned one line beyond the preceding line. The ONCE option can be used only when no more than one numbered DETAIL line directive is specified in the report format. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width or column 1 and the section width if multiple sections are specified. The default is no display of consecutive duplicate diagnostic messages.</td>
</tr>
<tr>
<td>DESCRIBE</td>
<td>Establishes a directory to the content of a non-data-base file.</td>
<td></td>
</tr>
<tr>
<td>DETAIL</td>
<td>Determines report line content and positioning of source data fields, literals, and computed values.</td>
<td></td>
</tr>
<tr>
<td>DIAGNOSTIC</td>
<td>Specifies whether or not consecutive duplicate diagnostic messages are to be displayed.</td>
<td></td>
</tr>
<tr>
<td>Directive</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>Displays information from a data base file, non-data-base file, or temporary storage.</td>
<td>The UPON default is the terminal or the output file. The FROM option requires a directory for the designated file; it must be produced by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive. Query Update rewinds the specified file before displaying FROM.</td>
</tr>
<tr>
<td>DUPLICATE</td>
<td>Copies recorded information between the user catalog and the default catalog.</td>
<td>Must be preceded by the VERSION directive, which attaches the user catalog.</td>
</tr>
<tr>
<td>END</td>
<td>Terminates Query Update operations and returns control to the operating system.</td>
<td>Default catalog, created area, and report dispositions are determined by the user and performed at this time through applicable operating system procedures. A single item in a matrix or a literal cannot be erased.</td>
</tr>
<tr>
<td>ERASE</td>
<td>Removes DEFINE items or SPECIFY items. Removes one or more report specifications or session-IDs from the current catalog. Eliminates a directory created by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive.</td>
<td></td>
</tr>
<tr>
<td>EVALUATE</td>
<td>Performs arithmetic operations to compute data name content or a cumulative function result. Selects the working storage data names for which values are to be calculated when a particular report production step occurs.</td>
<td>The IF directive can be used in conjunction with EVALUATE for manipulative operations, but not for report operations. For report operations, the EVALUATE directive must be preceded by an ALTER or FORMAT directive. If EVALUATE is initializing items to be included as part of a detail line, the directive should be related to the DETAIL directive and not to SELECT.</td>
</tr>
<tr>
<td>EXECUTE</td>
<td>Causes execution of a procedure that is external to Query Update.</td>
<td>The procedure name must be 1 through 7 characters in length. The procedure must be in relocatable format. Default report specifications, working storage data names, report names with associated report specifications, session IDs with associated directives, relations with associated record names, item names in record, and area names in use can be displayed. Rewind operations before and after an EXTRACT directive ensure correct programming. When data is renamed through the AS option, the data must be referenced by the new name while the EXTRACT is still in effect. Data items can be selected by using the EXTRACT directive with an IF directive. Items in an array must be individually extracted.</td>
</tr>
<tr>
<td>EXHIBIT</td>
<td>Lists information that is recorded in the current catalog.</td>
<td></td>
</tr>
<tr>
<td>EXTRACT</td>
<td>Creates a subset of information in a data base file, non-data-base file, or temporary storage, and creates a directory to the subset.</td>
<td></td>
</tr>
<tr>
<td>Directive</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>FOOTING</td>
<td>Provides content and determines line and column positioning for informative footings.</td>
<td>A FOOTING directive is associated with a BREAK directive. Level number 0 is associated with the end of data. The AT LINE default is positioned one line beyond the preceding line. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>Initiates grouping and retention of directives in the catalog under a report name for reference by other directives.</td>
<td>The report name must be unique in the current catalog.</td>
</tr>
<tr>
<td>HEADING</td>
<td>Provides content and determines line and column positioning for informative headings.</td>
<td>A HEADING directive is associated with a BREAK directive. Level number 0 occurs before any data is processed. The AT LINE default is positioned one line beyond the preceding line. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.</td>
</tr>
<tr>
<td>HELP</td>
<td>Presents descriptions of directives or explanations of diagnostic messages.</td>
<td>The HELP directive can be entered at any time during a Query Update session.</td>
</tr>
<tr>
<td>IF</td>
<td>Presents a test condition to determine whether subsequent directives are to be executed or bypassed.</td>
<td>An IF directive that references only temporary data items and literals can be used with any directive. An IF directive controls execution of the directives that follow it in the same transmission, up to but not including the next IF directive. IF directives cannot be nested.</td>
</tr>
<tr>
<td>INVOKE</td>
<td>Establishes areas, relations, and the subschema that is used for subsequent directives.</td>
<td>The FROM LIBRARY option identifies the subschema library that contains the subschema directory being used.</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Modifies a data item value in an existing record in a data base.</td>
<td>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 USING option or as a result of an IF directive with an associated MODIFY directive. The SETTING option specifies the names of the data items to be modified. Defined data items that require evaluation must be evaluated before the MOVE directive is executed. When one record is being updated, MOVE is used in conjunction with the MODIFY directive. For report operations, the MOVE directive must be preceded by a FORMAT or ALTER directive.</td>
</tr>
<tr>
<td>MOVE</td>
<td>Places values in data names.</td>
<td></td>
</tr>
<tr>
<td>Directive</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NOTE</td>
<td>Allows user comments to be included in transmissions.</td>
<td>User comments do not appear in output or as part of a report.</td>
</tr>
<tr>
<td>OS</td>
<td>Allows the user to enter an operating system command during a Query Update session.</td>
<td>An OS directive can be recorded as part of a session.</td>
</tr>
<tr>
<td>PAGE-NUMBER</td>
<td>Specifies use and positioning of system-supplied page number within a report.</td>
<td>The directive is not allowed in batch mode.</td>
</tr>
<tr>
<td>PAGE-SIZE</td>
<td>Specifies maximum number of vertical lines, horizontal columns, horizontal or vertical sectional page divisions, and multiple copy images.</td>
<td>Positioning default is the rightmost 10 characters of the page, line 1.</td>
</tr>
<tr>
<td>PERFORM</td>
<td>Retrieves and executes transmissions recorded in the current catalog.</td>
<td>Default page length is 60 lines; default page width is 136 columns, 1 section, and 1 image.</td>
</tr>
<tr>
<td>PREFACE</td>
<td>Causes lines of text or another report to precede the first page of a report.</td>
<td>A maximum of 10 sections can be specified. If sections are specified, the default number of columns divided by the number of sections rounded down is the width of a section.</td>
</tr>
<tr>
<td>PREPARE</td>
<td>Initiates execution of report directives.</td>
<td>When no options are included, the entire cataloged session is executed.</td>
</tr>
<tr>
<td>PREVIEW</td>
<td>Causes sample execution of report directives.</td>
<td>A specified report name must be in the current catalog.</td>
</tr>
<tr>
<td>RECAP</td>
<td>States content and positioning of recapitulative information generated at the end of each report page.</td>
<td>A preface for the specified report name is included in the output report.</td>
</tr>
<tr>
<td>RECORDING</td>
<td>Initiates the recording of subsequent transmissions in the current catalog.</td>
<td>The specified report name must be in the current catalog.</td>
</tr>
<tr>
<td>REMOVE</td>
<td>Removes specific records from a database file.</td>
<td>The source data file is automatically rewound before report preparation. The report output file is not rewound before or after report preparation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The specified report name must be in the current catalog.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The FROM default is dummy data values of X's and Y's for character information and 8's and 9's for numeric information.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The AT LINE default is positioned one line beyond the last detail or footing line.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Recording continues until RECORDING OFF is specified.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Report specification directives for a report format are not recorded; they are retained in the current catalog under the report name established by the FORMAT directive.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Only records from one area can be removed at a time; records from each area joined in a relation must be removed separately.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A complete record, not part of a record, is removed.</td>
</tr>
<tr>
<td>Directive</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td>RETURN</td>
<td>Releases a file, relation, or subschema that is no longer needed by Query Update.</td>
<td>Multiple files can be returned with one RETURN directive. A file established with a CREATE directive requires two RETURN directives: the first to return the area information, the second to return the file.</td>
</tr>
<tr>
<td>REWIND</td>
<td>Logically positions a file at the beginning-of-information.</td>
<td>Multiple files can be rewound with one REWIND directive.</td>
</tr>
<tr>
<td>SELECT</td>
<td>Indicates alternative DETAIL specifications are to be selected when stated conditions are met.</td>
<td>When DETAIL directives include tag numbers, SELECT directives are required. The tag number must correspond to a DETAIL tag number. A maximum of five tag numbers can be specified; the number can be modified at installation time. 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. When multiple selects exist, only the first qualifying select is executed.</td>
</tr>
<tr>
<td>SEPARATOR</td>
<td>Defines a character to be used for delimiting nonnumeric literals.</td>
<td>The dollar sign ($) is the default separator. A character defined as a universal character cannot be defined as a separator. The ITEM-SIZE option is not used in report production.</td>
</tr>
<tr>
<td>SORT</td>
<td>Specifies and initiates the resequencing of source data.</td>
<td>Before a file can be sorted, it must have a directory produced by a DESCRIBE, DISPLAY UPON (directory not generated on NOS 1), or EXTRACT directive. The default collating sequence is COBOL. The default ordering sequence is ascending. A maximum of 25 sorting fields can be defined; each can be defined as either ascending or descending. Specified conditions can be tested by IF and SELECT directives.</td>
</tr>
<tr>
<td>SPECIFY</td>
<td>Establishes a name for convenient reference to a condition.</td>
<td>Only one area can be modified at a time; each area joined in a relation must be modified separately. The SETTING option specifies the names of the data items to be stored.</td>
</tr>
<tr>
<td>STORE</td>
<td>Creates a new record and places it in a data base file.</td>
<td></td>
</tr>
<tr>
<td>SUMMARY</td>
<td>Causes lines of text or another report to follow the last page of a report.</td>
<td>A specified report name must be in the current catalog. A summary for the specified report name is included in the output report. Tab numbers need not be entered in sequence because the system sorts them in ascending order. Default vertical positioning is line 1; default horizontal positioning is column 90. If the page size is less than 99 columns, the default horizontal positioning is the page width minus 29 columns.</td>
</tr>
<tr>
<td>TABS</td>
<td>Relates tabular references to horizontal column numbers for report preparation.</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>Specifies use and positioning of system-supplied time information within a report.</td>
<td></td>
</tr>
</tbody>
</table>
Table 2-1. Query Update Directives (Cont'd)

<table>
<thead>
<tr>
<th>Directive</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE</td>
<td>States content and positioning of title to start a report page.</td>
<td>Default vertical positioning is line 1; default horizontal positioning is column 1. When no points are specified by an IN clause, the CENTERED default horizontal points are column 1 and the page width.</td>
</tr>
<tr>
<td>UNIVERSAL</td>
<td>Establishes a character that marks a character position to be ignored during comparison testing.</td>
<td>The default universal character is # in the ASCII character set and = in the CDC graphic set. The current separator character cannot be defined as the universal character.</td>
</tr>
<tr>
<td>VERIFY</td>
<td>Specifies data names for terminal display in response to a previous VETO directive.</td>
<td>The specified data items are displayed in response to a directive that causes a record to be modified or deleted. The directive is not allowed in batch mode.</td>
</tr>
<tr>
<td>VERSION</td>
<td>Attaches a permanent file as the current catalog or reverts to the default catalog.</td>
<td>Permanent file parameters are required unless the subschema declares the area as TEMP.</td>
</tr>
<tr>
<td>VETO</td>
<td>Causes a terminal display of data subject to modification or deletion.</td>
<td>The directive is not allowed in batch mode.</td>
</tr>
<tr>
<td>VIA</td>
<td>Specifies which relation should be followed. VIA is needed when more than one relation is defined and the relation to be used cannot be determined by Query Update.</td>
<td>If more than one VIA directive is entered in a transmission, only the last one specified is used. A VIA directive remains in effect until another VIA directive is entered.</td>
</tr>
</tbody>
</table>

Using Query Update

Query Update functions in interactive mode through a user terminal. Query Update asks for input; the user enters directives, then Query Update responds. The results of the operation requested by the directives are displayed or printed on output devices. Output devices can include a terminal display screen, terminal hardcopy printer, line printer, or any combination of the three.

One or more directives can be used to specify a complete operation. A complete operation is called a transmission. A transmission begins with a directive keyword and ends when a SEND, RETURN, or equivalent transmit key is depressed. Query Update executes or records the transmission and then returns one or more of the following:

- The information requested by the transmission
- An error message
- A double hyphen (\----), which indicates that Query Update is ready to receive the next transmission
- A double greater-than symbol (\>>), which indicates that Query Update is ready to receive data

A physical transmission (a transmission as defined by the operating system) can contain a maximum of 150 characters on NOS and can contain an unlimited number of characters on NOS/BE. A logical transmission (a transmission as defined by Query Update) can contain a maximum of 1030 characters. The maximum logical transmission length can be changed by setting the TL parameter on the Query Update control statement.

When a logical transmission exceeds the line width of the terminal, the system performs an automatic carriage return and line feed, then continues to accept input on the next line.

Automatic continuation can be more convenient to use than manual continuation. Under automatic continuation, a user can backspace to the beginning of a physical transmission to correct a typing error.

Transmissions can be manually continued by entering one of two continuation characters (+ or =) as the last character on the line and depressing the transmit key. A manually continued transmission can include a maximum of 1030 characters or the number of characters specified by the TL parameter. At least one space is required between the last nonblank character and the continuation character.

Interactive mode is considered to be the most convenient method for using Query Update. Three exceptions might be the following:
When a large data base is being created.

When a long report format is being entered and terminal hardcopy output is not available.

When a long sequence of transmissions must be entered in the session.

Beginning a Query Update Session

Before beginning a Query Update session you must establish communication between the terminal and the host computer.

Hardware communication is established by depressing the switches necessary to turn on the remote terminal equipment, dialing into the host computer, and completing the connection between the terminal and the computer.

Software communication is established by a login procedure. The login procedure is a dialog that is required for user identification and accounting purposes. Login procedures for NOS and NOS/BE are shown in figure 2-1. User input is underscored.

A Query Update session is begun when you enter a QU control statement.

---

**Figure 2-1. Login Procedures**
Ending a Query Update Session

A Query Update session is ended when you enter an END directive.

Software communication is terminated by a logout procedure. The logout procedure is a single user command. Logout procedures for NOS and NOS/BE are shown in figure 2-2. User input is underscored.

```
NOS
---
? END
/BYE

NOS/BE
---
END
COMMAND LOGOUT
```

Figure 2-2. Logout Procedures

Hardware communications are terminated by depressing the switches necessary to turn off the remote terminal equipment and releasing the telephone connection between the terminal and the computer.

THE PROCESSING ENVIRONMENT

The Query Update processing environment using CYBER Record Manager (CRM) varies depending on the type of file (data base or non-data-base) being accessed. The basic requirement for either type of file access is the existence of a directory that describes the physical storage and characteristics of individual data items. Query Update processing is shown in figure 2-3.

The left portion shows user input to the Query Update program via a user terminal. Input consists of Query Update directives and data entered as a series of transmissions.

The center portion shows the Query Update program and how it interfaces with the Report Utility and CRM.

The Report Utility program is callable through the REPORT control statement of the operating system. This utility automates report writing. The report information table file contains encoded report specifications that are generated by the COMPILE directive of the Query Update language and used by the Report utility. The catalog of directives is a series of recorded report specifications that have been generated and stored in mass storage for subsequent use.

CRM performs input/output processing for data base files according to the information passed through the subschema. CRM performs input/output processing for non-data-base files according to information provided by the user.

The right portion shows output to the user from the Query Update program to a line printer or terminal display screen.
Note:

Shaded portions apply to a data base file described by a previously compiled subschema directory.

Nonshaded portions apply to operations on both data base and non-data-base files.

Figure 2-3. Query Update Processing
Every data base file accessed through the Query Update language must be described in a directory called a Query Update subschema. The subschema declares the organization of the file, indicates the physical storage of the data, defines data names, and describes the characteristics of each data item.

There are two types of Query Update subschemas: one in which a data base is defined by a subschema and controlled by CYBER Database Control System (CDCS), and one in which a data base is defined by a subschema and controlled by CYBER Record Manager (CRM). This guide only deals with Query Update in CRM data base access mode.

You can access an existing data base file after you generate and store a Query Update subschema that corresponds to the file. You can, alternatively, generate and store a Query Update subschema first, and then create a file to match the description set up in the subschema. This latter approach is taken in the following paragraphs.

**GENERATING A QUERY UPDATE SUBSCHEMA**

The generation of Query Update subschemas is normally performed under the direction of a data administrator. This conforms to the IMS-170 data base management concept that data base information should be centrally controlled. Detailed information regarding Query Update subschema generation is contained in the Query Update/CYBER Record Manager Data Administration reference manual.

The NOS job structure and DDL source statements required to generate a Query Update subschema are shown in figure 3-1.

The subschema contains the following divisions:

- **The Identification Division**, which names the subschema.
- **The Data Division**, which names the file, supplies file organization and key specification for CRM and describes the record.
- **The subschema is named QUSUB.**
- **The data base file is named INVENTORY.** Notice that the subschema refers to the file as an area. Several areas could have been included in this subschema, but only one was selected.

File organization is new indexed sequential (which means extended indexed sequential). The primary key is INV-NO, the alternate key is BACK-ORDER.

The index file is named INVIDX. The INDEX statement is optional; when it is not included in the subschema, the index for files with alternate keys must be declared at Query Update execution time.

```plaintext
Job statement
USER control statement
CHARGE control statement
DEFINE(QSUB/CT=FU,M=R)
DRL3(QS,SB=QUSUB)
End-of-record

IDENTIFICATION DIVISION.
SUBSCHEMA NAME IS QUSUB
DATA DIVISION.
AREA-NAME IS INVENTORY UN IS username M IS W
INDEX IS INVIDX UN IS username M IS W
ORGANIZATION IS INDEXED NEW
KEY IS INV-NO
KEY IS ALTERNATE BACK-ORDER Duplicates INDEXED

RECORD-NAME IS INV-REC
02 INV-NO PICTURE X(6)
02 IN-Stock PICTURE Z(3)9
02 BACK-ORDER PICTURE Z(3)9
02 ON-Order PICTURE Z(3)9
02 REORDER-PT PICTURE Z(3)9
02 UNIT-COST PICTURE Z(4).99
02 UNIT-PRICE PICTURE Z(4).99
02 DESCRIPTION PICTURE X(17)

End-of-information
```

![Figure 3-1. Generating a Subschema Under NOS](Image)
The record is named INV-REC and has eight fields, which are alphanumeric, as indicated by the X descriptor. The remaining fields are numeric. Because reports are to be generated from the INVENTORY file, editing characters are included in appropriate picture specifications.

The DEFINE control statement effects permanent file storage. The file is declared public by the CT parameter and the file access mode is declared read-only by the M parameter. (The file access mode only affects users who have a different user number.)

The DDL control statement specifies two required parameters: QD specifies that a Query Update subschema is to be compiled and SB=QUSUB denotes the file name upon which the subschema is to be written.

The NOS/BE job structure and DDL statements required to generate a subschema are shown in figure 3-2.

The REQUEST and CATALOG control statements are required for permanent file storage of the subschema. The local file name is X and the permanent file name is QUSUB.

The DDL control statement specifies two required parameters: QD denotes that a Query Update subschema is to be compiled and SB=X denotes the local file name upon which the subschema is to be written.

**CREATING A DATA BASE FILE**

Now that a subschema exists, a data base file named INVENTORY can be created through the CREATE directive. The use of this directive is shown on NOS in figure 3-3.

```plaintext
job statement
ACCOUNT control statement
REQUEST(X,P,F)
DDL3(QD,SB=X)
CATALOG(X,QUSUB,ID=QUSER)
End-of-record

IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS QUSUB
DATA DIVISION.
AREA-NAME IS INVENTORY ID IS QUSER
INDEX IS INVIDX ID IS QUSER
ORGANIZATION IS INDEXED NEW
KEY IS INV-NO
KEY IS ALTERNATE BACK-ORDER Duplicates INDEXED
RECORD-NAME IS INV-REC
02 INV-NO PICTURE X(6)
02 IN-STOCK PICTURE Z(39)
02 BACK-ORDER PICTURE Z(39)
02 ON-ORDER PICTURE Z(39)
02 REORDER-PT PICTURE Z(39)
02 UNIT-COST PICTURE Z(4),99
02 UNIT-PRICE PICTURE Z(4),99
02 DESCRIPTION PICTURE X(17)

End-of-information
```

Figure 3-2. Generating a Subschema Under NOS/BE
Each data entry includes an item for each field specified in the STORE directive. Data is entered in the order indicated by the data names in the SETTING option; INV-NO data is first, IN-STOCK data is second, and so forth. Data fields are separated from each other by at least one space (a comma could be used instead of a space). Several additional points concerning the data entries should be noted:

The INV-NO and DESCRIPTION fields are described as alphanumeric in the subschema; therefore, they must be enclosed in $ delimiters. These fields are stored with the appropriate number of trailing blanks.

The numeric fields do not require $ delimiters. These fields are stored with the appropriate number of leading zeros. The Z replacement character in the subschema causes suppression of leading zeros when the fields are output in display mode.

The data entries are followed by an *END directive, which terminates the STORE directive.

The last entry is the END directive, which terminates the Query Update session.

The INVENTORY data base file can be created under NOS/BE with the job stream shown in figure 3-4.

Two CATALOG control statements effect permanent file storage of the INVENTORY file and its associated index file.
QUERYING A DATA BASE FILE

Interactive query always begins with a call to the Query Update program. This is done by typing the letters QU in response to the operating system request for input. NOS requests input by printing a slash (/). NOS/BE requests input by printing the word COMMAND-

When the Query Update program is loaded into memory, a Query Update heading is printed followed by a request for input. Query Update requests input by printing double hyphens (--) followed by a linefeed. On NOS, Query Update also prints a question mark (?)

A data base file must be attached before it can be accessed for interactive query. This is done through the INVOKE directive, which names the subschema and the subschema library (if the subschema library name is different from the subschema name) and provides user identification. Query Update automatically attaches the data base file, together with its associated index file, when it is needed and returns the files when they are not needed.

Sample directives for interactive query are shown in figure 3-5. The following points should be noted:

- The INVOKE directive references permanent files.
- Certain parameters are required as shown in the figure.

Query Update displays only 14 lines at a time. After each group of lines, the program pauses and offers the option to continue the display or terminate the directive.

The IF directive can be used in conjunction with the DISPLAY directive to provide conditional query.

MODIFYING A DATA BASE FILE

Modifications to a data base file are made by three Query Update directives: MODIFY, STORE, and REMOVE. Specific operations and general directive formats are listed as follows:

- Modify like fields in several records.
  
  MODIFY USING key-name
  MOVE expression TO data-name-1
  >> key-name-value

- Store values in one or more fields for one new record.

  STORE MOVE expression-1 TO data-name-1
  AND expression-2 TO data-name-2

- Store values for like fields in several new records.

  STORE SETTING
  data-name-1 data-name-2 data-name-3
  >> data-name-1-value data-name-2-value data-name-3-value

- Remove one or more records.

  REMOVE USING key-name
  >> key-name-value
Call Query Update.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Display six fields.

A Y response continues the display.

Request a conditional display by primary key. The primary key is alphanumeric and requires the $'s. Full record display is never in display format.

Request a conditional display using the relational operator less than or equal to.

Request a conditional display using the relational operator greater than or equal to.

An N response terminates the display.

Figure 3-5. Querying a Data Base File (Sheet 1 of 2)
DISPLAY DESCRIPTION $$$$ UNIT-PRICE
METAL DESKS $ 389.95
OAK DESKS $ 1282.50
WALNUT DESK $ 1300.00
BULLETIN BOARD $ 15.00
CHALK BOARD $ 19.52
1-DR FILE CABINET $ 45.00
3-DR FILE CABINET $ 60.00
5-DR FILE CABINET $ 90.00
ARM CHAIR $ 295.00
DESK CHAIR $ 149.95
SWIVEL CHAIR $ 96.00
LETTER RACK $ 3.98
3-SHELF BOOK CASE $ 39.95
STOOL $ 16.20
(MORE... ANSWER Y OR N)
? Y
ELECT TYPewriter $ 369.00
COFFEE TABLE $ 95.00
DESK LAMP $ 19.95
FLOOR LAMP $ 69.95
TABLE LAMP $ 39.95
19 ACCESSIES, 19 HITS, 19 TO-S

? IF REORDER-PT = 3 AND UNIT-COST NE 130 +
? DISPLAY DESCRIPTION REORDER-PT UNIT-COST
DESK CHAIR 3 89.00
SWIVEL CHAIR 3 35.00
STOOL 3 9.50
19 ACCESSIES, 3 HITS, 19 TO-S

? IF REORDER-PT = 3 OR UNIT-COST = 89 +
? DISPLAY DESCRIPTION REORDER-PT UNIT-COST
ARM CHAIR 3 130.00
DESK CHAIR 3 89.00
SWIVEL CHAIR 3 35.00
STOOL 3 9.50
19 ACCESSIES, 4 HITS, 19 TO-S

? DISPLAY (UNIT-PRICE - UNIT-COST) +
? / UNIT-COST = 100
000000000121.83
000000000156.50
000000000044.44
000000000200.00
000000000144.00
000000000200.00
000000000200.00
000000000181.25
000000000126.92
000000000068.48
000000000174.29
000000000306.12
000000000099.75
000000000070.53
(MORE... ANSWER Y OR N)
? N
15 ACCESSIES, 14 HITS, 15 TO-S

? END

Terminate Query Update.
Sample directives for data base file modification are shown in figure 3-6. The following points should be noted:

The REMOVE directive deletes an entire record, not part of a record.

The IF directive can be used in conjunction with the MODIFY and REMOVE directives to provide conditional modification.

The ACCESS message indicates the total number of records accessed for modification; the HITS message indicates the total number of successful operations; the IO-S message indicates the total number of input/output operations required.

Call Query Update.

Query Update 3.4 SYSEDIT-82110 82/04/29 13:05:46

? INVOKE QSUB (UN=username)

? MODIFY USING INV-NO SETTING ON-ORDER

>> $AB5972$ 12

>> $AB5973$ 7

>> $AB5975$ 12

>> *END

3 ACCESSES, 3 HITS, 6 IO-S

? IF INV-NO EQ $SHD011$ MODIFY MOVE 15 TO IN-STOCK

1 ACCESSES, 1 HITS, 3 IO-S

Request a conditional modification.

Store three records.

Terminal the MODIFY USING directive.

Call Query Update.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Modify one field in three records.

Termiate the MODIFY USING directive.

Call Query Update.

Request a conditional modification.

Store three records.

Remove three records.

Terminale the REMOVE USING directive.

The VETO directive displays the first 40 characters. A NO response cancels the remove request.

Figure 3-6. Modifying a Data Base File (Sheet 1 of 2)
-- VERIFY DESCRIPTION

--

? REMOVE USING INV-NO VETO

>> $ST0592$

STOOL
VETO- YES

>> *END

  1 ACCESSES, 1 HITS, 2 IO-S

--

? IF INV-NO EQ $XN6158$ REMOVE VETO

COFFEE TABLE
VETO- NO

  1 ACCESSES, 1 HITS, 2 IO-S

--

? DISPLAY INV-NO DESCRIPTION
AB5972 METAL DESK
AB5973 OAK DESK
AB5975 WALNUT DESK
BB0013 BULLETIN BOARD
CB0168 CHALK BOARD
CB1002 2-DR FILE CABINET
CB1003 3-DR FILE CABINET
CB1004 4-DR FILE CABINET
CHO059 ARM CHAIR
CHO060 DESK CHAIR
CHO080 SWIVEL CHAIR
CHO100 TYPING CHAIR
CHO075 LETTER RACK
SH0011 3-SHELF BOOK CASE

(MORE... ANSWER Y OR N)

? Y

TY5015 ELECT TYPEWRITER
XN6158 COFFEE TABLE
YB0020 DESK LAMP
YB0060 TABLE LAMP

  18 ACCESSES, 18 HITS, 18 IO-S

--

? END

The VERIFY directive requests display of the description field, rather than the first 40 characters.

A YES response effects the remove operation.

Request a conditional remove with the VERIFY option in effect.

A DISPLAY operation shows that three records have been stored (CB1002, CB1004, and CHO100) and four records have been removed (CB1001, CB1003, ST0592, and YB0059).

Terminate Query Update.

Figure 3-6. Modifying a Data Base File (Sheet 2 of 2)
A non-data-base file, as defined within the context of this guide, is a sequential file that has no corresponding Query Update subchema. When a non-data-base file is to be accessed by Query Update, the following requirements must be met:

The file must be a local file. If the file is a permanent file, it must be attached via the operating system permanent file attach procedure.

The storage characteristics of the file must be conveyed to CYBER Record Manager (CRM) with a FILE control statement.

The file must be described for Query Update with the DESCRIBE directive.

For purposes of illustration, a non-data-base file has been stored on disk. The layout of the file is shown in figure 4-1. The file has the following characteristics:

The file name is INDFILE.

The file contains six records.

Each record contains two fields.

Field 1 contains six alphanumeric characters and field 2 contains four numeric characters.

This file is a transaction file related to the data base file INVENTORY. INDFILE includes values that correspond to values within the primary key field of INVENTORY.

DESCRIBING A NON-DATA-BASE FILE

Query Update accesses a non-data-base file according to the rules specified in a directory. Unlike

<table>
<thead>
<tr>
<th>Field 1</th>
<th>Field 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 0</td>
<td>A 8 5 9 7 2 0 0 2 5</td>
</tr>
<tr>
<td>Word 1</td>
<td>C 8 0 1 6 8 0 0 5 0</td>
</tr>
<tr>
<td>Word 2</td>
<td>Y 8 0 0 2 0 0 0 3 5</td>
</tr>
<tr>
<td>Word 3</td>
<td>B 8 0 0 1 3 0 0 4 0</td>
</tr>
<tr>
<td>Word 4</td>
<td>A 8 5 9 7 5 0 0 2 2</td>
</tr>
<tr>
<td>Word 5</td>
<td>C 8 0 0 6 0 0 0 3 3</td>
</tr>
</tbody>
</table>

Figure 4-1. INDFILE Format

a data base file subchema directory that is generated by DDL, a non-data-base file directory is generated by the Query Update DESCRIBE directive.

The operating system control statements and Query Update directive required to establish access to the non-data-base file INDFILE are shown in figure 4-2. The ATTACH and FILE control statements are required; they can precede the QU control statement and be entered in the usual manner or they can be entered through the Query Update OS directive. Only the BT and RT parameters are included on the FILE control statement; default values for other required parameters apply.

The DESCRIBE directive includes the file name and assigns names for the two fields (IND-INV-NO and IND-ON-ORDER). The fields correspond to the INVENTORY file fields INV-NO and ON-ORDER. Data item IND-INV-NO is described as a six-character alphanumeric field and IND-ON-ORDER is described as a four-character zero-suppressed numeric field. The $ delimiters are required because of the non-numeric characters X Z (.) .

/ QU
QUERY UPDATE 3.4 SYSEDIT-B2110 82/05/03 09.26.41
--
? OS,ATTACH,INDFILE
--
? OS,FILE,INDFILE BT=C,RT=F
--
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +
? AND IND-ON-ORDER AS NUM BY $Z(3)$+% 
--

Figure 4-2. Describing a Non-Data-Base File
QUERYING A NON-DATA-BASE FILE

You can query a non-data-base file after it has been attached, defined for CRM, and described for Query Update. Sample display operations are shown in figure 4-3.

The following points should be noted for non-data-base file query operations:

When the OS directives reference permanent files, certain parameters are required, as shown in the figure.

The word FROM and the file name must be included in the DISPLAY directive.

The file is always automatically rewound before a DISPLAY FROM directive is executed.

Conditional testing with the IF directive is not allowed.

ENTERING DATA IN A NON-DATA-BASE FILE

Data can be placed in a non-data-base file by using the DISPLAY UPON format of the DISPLAY directive.

Data is entered and displayed upon a non-data-base file rather than upon the terminal display screen. Data entered with the DISPLAY UPON directive under NOS is retained in a direct access file automatically, whereas data entered under NOS/BE with the directive must be retained with permanent file operations.

Sample DISPLAY UPON operations are shown in figure 4-4 for NOS and in figure 4-5 for NOS/BE. The following points should be noted for the DISPLAY UPON directive:

The non-data-base file must be positioned at the end of information before the DISPLAY UPON directive is issued. Figure 4-4 shows the DISPLAY FROM directive positioning the file; figure 4-5 shows the OS directive positioning the file. The DISPLAY FROM directive for positioning purposes is practical only when the file is small.

The DISPLAY UPON directive eliminates any previously established directory and creates a directory for the file being displayed (directory not created on NOS 1).

The SEPARATOR (SEP ITEM-SIZE) directive tells Query Update to concatenate the fields in INDFILE. If SEP ITEM-SIZE had not been specified, Query Update would have written one blank character before each item.

```
? DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER
AB5972 25
CB0168 50
YB0020 35
BB0013 40
AB5975 22
CH0060 33
6 ACCESSES, 6 HITS, 6 IO-S

--

? DISPLAY FROM INDFILE $INVENTORY NUMBER IS$ +
? IND-INV-No AND ORDER AMOUNT IS$ IND-ON-ORDER
INVENTORY NUMBER IS AB5972 AND ORDER AMOUNT IS 25
INVENTORY NUMBER IS CB0168 AND ORDER AMOUNT IS 50
INVENTORY NUMBER IS YB0020 AND ORDER AMOUNT IS 35
INVENTORY NUMBER IS BB0013 AND ORDER AMOUNT IS 40
INVENTORY NUMBER IS AB5975 AND ORDER AMOUNT IS 22
INVENTORY NUMBER IS CH0060 AND ORDER AMOUNT IS 33
6 ACCESSES, 6 HITS, 6 IO-S

--

? END
```

Display the INDFILE fields.

Include nonnumeric literals in a display.

Figure 4-3. Querying a Non-Data-Base File
Call Query Update.

Issue operating system ATTACH and FILE control statements.

Establish the directory for INDFILE.

Position INDFILE at the end-of-information.

Indicate to Query Update to concatenate the fields.

Enter four records in INDFILE. Under NOS 1, the DISPLAY UPON directive eliminates the old directory. Under NOS 2 and NOS/BE, the directive creates a new directory to INDFILE.

Establish the directory for INDFILE under NOS 1.

The DISPLAY operation verifies that four records have been added to the file.

The added data is made permanent.

Figure 4-4. Entering Data Into a Non-Data-Base File Under NOS
COMMAND: QU

QUERY UPDATE 3.3 538-81089  82/05/03  10.09.16

---
OS,ATTACH,INDFILE,ID=QUSER   Call Query Update.
---
OS,FILE,INDFILE,DT=C,RT=F
---
OS,SKIPF,INDFILE,300000       Issue operating system ATTACH and FILE control statements.
---
SEP ITEM-SIZE
---
DISPLAY UPON INDFILE $CM0575$ $0006$
---
DISPLAY UPON INDFILE $SH0011$ $0004$
---
DISPLAY UPON INDFILE $TY5015$ $0012$
---
DISPLAY UPON INDFILE $XN6158$ $0003$
---
DISPLAY FROM INDFILE IND-INV-NO IND-ON-ORDER
AB5972  25
CB0168  30
VB0020  35
VB0013  40
AB5975  22
CH0060  33
CM0575  6
SH0011  4
TY5015  12
XN6158  3
10 ACCESSES, 10 HITS, 10 IO-S
---
OS,EXTEND,INDFILE
---
END

Figure 4-5. Entering Data Into a Non-Data-Base File Under NOS/BE
COMBINING DATA BASE AND NON-DATA-BASE FILE OPERATIONS

Section 3 introduces the MODIFY, STORE, and REMOVE directives and shows their use with data base files. The input source in each case was the user terminal. This section presents a variation of these directives and shows how they can be used to modify a data base file using a non-data-base file as the input source. Three operations can be performed:

MODIFY ... FROM


REMOVE ... FROM

Removes records from a data base file based on values contained in a non-data-base file.

STORE ... FROM


Whenever combined data base and non-data-base file operations are anticipated, the following requirements must be met:


Information in a non-data-base file must be arranged in the order in which it is to be referenced in a Query Update directive.

Data base file INVENTORY and non-data-base file INDFILE contain common fields: inventory number and amount on order. The following paragraphs describe modification of INVENTORY based on values contained in INDFILE.

MODIFYING ACROSS FILES

The INVENTORY ON-ORDER fields, which are assumed to be noncumulative, can be modified with values contained in the corresponding INDFILE IN-ON-ORDER fields. The MODIFY ... FROM directive performs this type of operation as shown in figure 5-1.

INDFILE is appropriately attached and described and the subschema is named with the INVOKE directive. Displays against INVENTORY and INDFILE indicate the present status of the ON-ORDER fields.

The SEF ITEM-SIZE directive tells Query Update to use the directory description (established by the DESCRIBE directive) to determine the size of each item in INDFILE. The MODIFY ... FROM directive specifies the INVENTORY file primary key field and the field to be modified. CTBER Record Manager (CRM) locates the appropriate data base record and Query Update moves the new value into the corresponding ON-ORDER field. A second display against INVENTORY shows that data records with primary keys AB5972, AB5975, BB0013, CB0168, CH0060, and YB0020 reflect new totals for the ON-ORDER fields.

The following should be noted about the MODIFY ... FROM directive:

The fields specified in the USING option must be a primary or an alternate key field in the data base file. If a primary key is specified, only one record will be modified because the key value is unique. If an alternate key is specified, all records with the same value as the key are modified.

The non-data-base file is automatically rewound before a MODIFY ... FROM directive is executed.

Conditional testing on a data base field with the IF directive is not allowed when the USING option has been specified.
Call Query Update.

Issue operating system ATTACH and FILE control statements. NOS/BE requires an ID parameter instead of a UN parameter.

Establish the directory for INDFILE.

Name the subschema.

This is the present status of INVENTORY ON-ORDER fields.

This is the present status of INDFILE IND-ON-ORDER fields.

Indicate that the $ is not present in INDFILE; Query Update must use the directory descriptions.

Modify INVENTORY from INDFILE.
REMOVING ACROSS FILES

Records can be removed from INVENTORY based on information contained in INDFILE. Removal is performed by the REMOVE ... FROM directive. Key field values in the non-data-base file are compared to key field values in the data base file; when the values match, the appropriate data base file record is removed. This type of operation is shown in figure 5-2.

INDFILE is appropriately attached and described, and the subschema QUSUB is named with the INVOKE directive. A display of INVENTORY indicates the total number of records currently stored.

The REMOVE ... FROM directive specifies the INVENTORY primary key field. CRN locates the data base record whose key field matches the non-data-base file field, and Query Update removes the data base record. A second display of INVENTORY reflects a new total, indicating that the removal took place.

The following should be noted about the REMOVE ... FROM directive:

- A field specified in the USING option must be a primary or an alternate key field in the data base file. Additional fields, if included, are used for spacing in the file specified in the FROM clause.

- A non-data-base file is automatically rewound before a REMOVE ... FROM directive is executed.

- Conditional testing on a data base field with an IF directive is not allowed when the USING option has been specified.

STORING ACROSS FILES

Records in INDFILE can be stored into INVENTORY. The STORE ... FROM directive performs this type of operation as shown in figure 5-3.

INDFILE is appropriately attached and described, and the subschema QUSUB is named with the INVOKE directive. A display of INVENTORY indicates the total number of records currently stored.

The STORE ... FROM directive specifies the INVENTORY file primary key INV-NO field and the ON-ORDER field. Query Update stores the new record entries in the data base file. A second display of INVENTORY reflects a new total, indicating that the store took place. A display of inventory record AB5975 shows that Query Update numeric fields are zero filled and alphanumeric fields are blank filled.

The following should be noted about the STORE ... FROM directive:

- The field specified in the SETTING option must be a primary or an alternate key field in the data base file.

- The non-data-base file is automatically rewound before a STORE ... FROM directive is executed.

- Conditional testing on a data base field with the IF directive is not allowed when the SETTING option has been specified.
QUERY UPDATE 3.4 SYSEDIT-82110  82/05/03 10.09.16

--
? OS,ATTACH,INDFILE (UN=username)

--
? OS,FILE,INDFILE,BT=C,RT=F

--
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +
? AND IND-ON-ORDER AS NUM BY $Z(3)$9$

--
? INVOKE QUSUB (UN=CAH0220)

--
? DISPLAY INV-NO
AB5972
AB5973
AB5975
BB0013
CB0168
CB1002
CB1003
CB1004
CHO059
CHO060
CHO080
CHO100
CHO575
SH0011
(MORE... ANSWER Y OR N)
? Y
TY5015
XN6158
YB0020
YB0060
18 ACCESSES, 18 HITS, 18 IO-S

--
? SEP ITEM-SIZE

--
? REMOVE USING INV-NO FROM INDFILE
10 ACCESSES, 10 HITS, 20 IO-S

--
? DISPLAY INV-NO
AB5973
CB1002
CB1003
CB1004
CHO059
CHO080
CHO100
YB0060
8 ACCESSES, 8 HITS, 8 IO-S

--
? END

Issue operating system ATTACH and FILE
control statements. NOS/BE requires
an ID parameter instead of a UN
parameter.

Establish the directory for INDFILE.

Name the subschema. NOS/BE requires an
ID parameter instead of a UN parameter.

A display of INVENTORY primary key
fields shows that 18 records are stored
in the file.

Indicate that the $ is not present in
INDFILE; Query Update must use the
directory descriptions.

Remove INVENTORY records whose primary
key values equal those in INDFILE.
The file is positioned at the beginning-
of-information because no operation has
been performed against INDFILE.

A display of INVENTORY primary key
fields shows that 10 records have been
removed.

Terminate Query Update.

Figure 5-2. Removing INVENTORY Records Via INDFILE
QUERY UPDATE 3.4 SYSEDIT-82110 82/05/03 10.09.16

--
? OS,ATTTACH,INFILE (UN=username)

--
? OS,FILE,INDFILE,BT=C,RT=F

--
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +
? AND IND-ON-ORDER AS NUM BY $Z(3)$

--
? INVOKE QUSUB (UN=XXXXXX)

--
? DISPLAY INV-NO ON-ORDER
AB5973 7
CB1002 0
CB1003 0
CB1004 0
CH059 6
CH0080 0
CH0100 2
Y60060 4

8 ACCESSES, 8 HITS, 8 IO-S

--
? SEP ITEM-SIZE

--
? STORE SETTING INV-NO ON-ORDER FROM INDFILE
10 ACCESSES, 10 HITS, 10 IO-S

--
? DISPLAY INV-NO ON-ORDER
AB5972 25
AB5973 7
AB5975 22
BB0013 40
CB0168 50
CB1002 0
CB1003 0
CB1004 0
CH059 6
CH0080 0
CH0100 2
CH0575 6
SH011 4

(MORE... ANSWER Y OR N)
? Y
TY5015 12
X6158 3
Y60020 0
Y60060 4

18 ACCESSES, 18 HITS, 18 IO-S

--
? IF INV-NO EQ AB5975$ DISPLAY INV-REC
(209) REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT
AB5975000000002200000000000000
1 ACCESSES, 1 HITS, 2 IO-S

--
? END

Figure 5-3. Storing INVENTORY Records Via INDFILE
A variety of Query Update directives are available for preparing and generating reports. Reports can be displayed at the terminal, printed on either a terminal hardcopy printer or a line printer, stored as permanent files, or any combination of the four.

Reports can be generated from a data base or non-data-base file. Data for the report is retrieved according to a directory established at report time.

Data base file report operations use a temporary directory that points to a subset of the data base file. The subset is a sequential file composed of the fields selected for the report. Both the file subset and the temporary directory are created by an EXTRACT directive.

Non-data-base file report operations use the directory established by a DESCRIBE directive.

Reports are named by the FORMAT directive. The layout directives that follow the FORMAT directive provide the report titles, column headings, column positioning, page numbering, and so forth.

Production of the report is initiated through a PREPARE directive. Operating system commands are used to direct the finished report to the desired output device.

Reports shown in this section combine operations under NOS and NOS/BE. Some differences are the following:

The operating system request for input. NOS prints a slash (/), NOS/BE prints the word COMMAND.

The permanent file parameters. Directives that reference permanent files require certain parameters, as shown in the figures.

The permanent file storage commands. NOS requires DEFINE and COPY control statements, NOS/BE requires a CATALOG control statement.

The report output commands. NOS requires a ROUTE control statement, NOS/BE requires a BATCH control statement.

The commands for viewing reports at the terminal (not shown in figures).

NOS requires you to enter the word EDIT, a comma, and the name of the report to be displayed. Then the text editor prints a heading and waits for input. You enter the word PRINT followed by an asterisk to display the entire report. Printing is terminated when you enter the word END.

NOS/BE requires you to enter the word PAGE, a comma, and the name of the report to be displayed. Then the PAGE system prints a Ready message and waits for input. You enter the number of the first line to be displayed. Paging continues if you enter a plus sign (+), paging terminates if you enter the letter E for end.

The general sequence of commands and directives for report generation is summarized in table 6-1.

Reports prepared in this manner are referred to as one-time reports because they are available only during the current session.

<table>
<thead>
<tr>
<th>TABLE 6-1. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR ONE-TIME REPORTING</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-Data-Base File Operations</strong></td>
</tr>
<tr>
<td>QU</td>
</tr>
<tr>
<td>OS†, ATTACH, non-data-base file</td>
</tr>
<tr>
<td>OS†, FILE ...</td>
</tr>
<tr>
<td>DESCRIBE</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>FORMAT</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>PREPARE</td>
</tr>
<tr>
<td>END</td>
</tr>
<tr>
<td>Output Commands</td>
</tr>
</tbody>
</table>

†You can enter the operating system ATTACH and FILE control statements, rather than the Query Update OS directives, before the QU control statement.

The basic commands and directives for a one-time report from INVENTORY are shown in figure 6-1.

The basic commands and directives for a one-time report from INDFILE are shown in figure 6-2.

In most cases it is better to prepare report formats and store them as permanent files so they can be used over and over again. This convention is called cataloging and is described in the following paragraphs.
Call Query Update.

NAME the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Build a two-field subset of INVENTORY and an associated directory. The subset sequential file is named FILE1.

Supply the name of the report and indicate that layout directives follow.

Supply the report line content and vertical positioning for the fields.

Supply the report title and specify horizontal positioning on line 2. A blank line is to appear at line 3.

Generate REPORT1 using data from FILE1.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND- BATCH,REPORT1,PRINT

Figure 6-1. Generating a One-Time Report From INVENTORY
Call Query Update.

Attach INDFILE and declare file organization for CRM. NOS/BE requires an ID parameter instead of a UN parameter.

Establish the directory for INDFILE.

Supply the name of the report and indicate that layout directives follow.

Supply report line content and vertical positioning for the fields.

Supply report title and specify horizontal positioning on line 2. A blank line is to appear at line 3.

Rewind INDFILE and execute REPORT2 directives.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND- BATCH,REPORT2,PRINT

**END OF REPORT REPORT2**

Figure 6-2. Generating a One-Time Report From INDFILE

QUERY UPDATE CATALOGS

Report directives are stored on a local file named ZZZZZQ2, which is known as the default catalog. At the beginning of each new Query Update session (that is, whenever the QU control statement is entered), the previous default catalog is returned to the operating system and a new default catalog is made available. This means that the report directives entered on the default catalog for one session cannot be made available for another session. All report directives for that specific report must be entered again.

This problem is solved by storing the default catalog as a permanent file. When the report is needed at a later time, the permanent catalog can be attached, the Query Update session directives initiated, and the report generated with minimum effort.
CATALOGING REPORTS

Recording Query Update sessions on a permanent catalog represents an efficient method of generating reports. Report directives need to be entered once, then the report itself can be generated over and over again. Reports can always be changed, even after they are cataloged. The ALTER directive, which is discussed later in this section, accepts subsequent directives that remove, replace, or add report specifications to the cataloged report.

The general sequence of commands and directives for cataloging reports is summarized in Table 6-2.

<table>
<thead>
<tr>
<th>TABLE 6-2. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR CATALOGING REPORTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Data-Base File Operations</td>
</tr>
<tr>
<td>QU</td>
</tr>
<tr>
<td>OS†, ATTACH, non-data-base file, ...†</td>
</tr>
<tr>
<td>OS, FILE, ...†</td>
</tr>
<tr>
<td>RECORDING</td>
</tr>
<tr>
<td>DESCRIBE</td>
</tr>
<tr>
<td>PREPARE</td>
</tr>
<tr>
<td>END†††</td>
</tr>
<tr>
<td>RECORDING OFF</td>
</tr>
<tr>
<td>FORMAT</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>END</td>
</tr>
<tr>
<td>Permanent File Storage Commands</td>
</tr>
</tbody>
</table>

†You can use the operating system ATTACH and FILE statements, rather than the Query Update OS directives, before the QU control statement. The OS directives can be positioned immediately after the RECORDING directive and be recorded as part of a session.

††The INVOKE directive can follow the QU control statement and remain outside of the recording limits.

†††The END directive within the recording limits is optional. When this directive is included, control reverts to the operating system immediately after the cataloged session is subsequently performed. When this directive is omitted, control remains with Query Update.

Notice that the FORMAT directive initiates retention of the associated layout directives. These directives are retained in the catalog under the report name specified in the FORMAT directive.

Figure 6-3 shows the basic commands and directives for cataloging a report from INVENTORY.

Figure 6-4 shows the basic commands and directives for cataloging a report from INDFILE.

USING CATALOGED REPORTS

Cataloged reports can be produced with a minimum of two directives: VERSION and PERFORM. The VERSION directive attaches the named permanent catalog. The PERFORM directive retrieves and executes the directives recorded under the named session. The general sequence of commands and directives for using cataloged reports is summarized in Table 6-3.

<table>
<thead>
<tr>
<th>TABLE 6-3. SUMMARY OF BASIC COMMANDS AND DIRECTIVES FOR USING CATALOGED REPORTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-Data-Base File Operations</td>
</tr>
<tr>
<td>QU</td>
</tr>
<tr>
<td>OS, ATTACH, non-data-base file ...†</td>
</tr>
<tr>
<td>OS, FILE, ...†</td>
</tr>
<tr>
<td>RECORDING</td>
</tr>
<tr>
<td>INVOKE subschema ...††</td>
</tr>
<tr>
<td>EXTRACT</td>
</tr>
<tr>
<td>PREPARE</td>
</tr>
<tr>
<td>END†††</td>
</tr>
<tr>
<td>RECORDING OFF</td>
</tr>
<tr>
<td>FORMAT</td>
</tr>
<tr>
<td>. (layout directives)</td>
</tr>
<tr>
<td>END</td>
</tr>
<tr>
<td>Permanent File Storage Commands</td>
</tr>
</tbody>
</table>

†The ATTACH and FILE control statements can assume three positions: before the QU control statement as normal operating system control statements, after the QU control statement as Query Update OS directives, or within the recorded session called by the Query Update VERSION directive.

Figure 6-5 shows the generation of data base file REPORT3 from the QUCAT catalog.

Figure 6-6 shows the generation of non-data-base file REPORT4 from the QUCAT catalog.

When the VERSION directive references the permanent catalog, some permanent file parameters are required as shown in the figures.
Call Query Update.

QUERY UPDATE 3.4 SYSEDIT-82110  82/05/12  16.42.18

---
? RECORDING SESS3

001---
? INVOKE GSUB (UN=username)

002---
? EXTRACT UPON FILE3 INV-NO ON-ORDER

003---
? PREPARE REPORT3 FROM FILE3

004---
? END

005---
? RECORDING OFF

END OF SESSION SESS3

---

? FORMAT REPORT3

---

? DETAIL IS INV-NO IN COLUMN 7 +
? ON-ORDER IN COLUMN 16

---

? TITLE AT LINE 2 IS +
? SRECORDED DATA BASE FILE REPORTS +
? AT LINE 3 IS $    $

---

? END

**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2

/ DEFINE,QUICAT
/ COPY,ZZZZZZQ2,QUICAT

NOS/BE control statements would be the following:
COMMAND= CATALOG,ZZZZZZQ2,QUICAT,ID=QUSER

Figure 6-3. Cataloging a Report From INVENTORY

---

Call Query Update.

QUERY UPDATE 3.4 SYSEDIT-82110  82/05/12  16.55.44

---
? OS,ATTACH,INDFILE (UN=username)

---
? OS,FILE,INDFILE,DT=C,RT=F

---

? RECORDING SESS4

001---
? DESCRIBE INDFILE AND IND-INV-NO AS CHAR BY $X(6)$ +
? AND IND-ON-ORDER AS NUM BY $Z(3)$

002---

? PREPARE REPORT4 FROM INDFILE

Attach INDFILE. NOS/BE requires an ID parameter instead of a UN parameter.

DECLARE file organization for CRM.

Initiate recording and name the session SESS4.

Query Update prints transmission IDs numbered sequentially from 001. Each executable directive is entered for a report named REPORT4.

Figure 6-4. Cataloging a Report From INDFILE (Sheet 1 of 2)
DD3--
  ? END

DD4--
  ? RECORDING OFF
  END OF SESSION SESS4

- -
  ? FORMAT REPORT4

- -
  ? DETAIL IS IND-INV-NO IN COLUMN 7 +
  ? IND-ON-ORDER IN COLUMN 16

- -
  ? TITLE AT LINE 2 IS +
  ? RECORDED NON-DATA-BASE FILE REPORTS +
  ? AT LINE 3 IS $ $

- -
  ? END

**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZQ2

/ DEFINE, QUCAT
/ COPY, ZZZZQ2, QUCAT

NOS/BE control statement would be the following:
COMMAND- CATALOG, ZZZZQ2, QUCAT, ID=QUSER

Figure 6-4. Cataloging a Report From INDFILE (Sheet 2 of 2)

/ QU
QUERY UPDATE 3.4 SYSEDIT-B2110 82/05/17 10.35.49

- -
  ? VERSION QUCAT (UN=username)

- -
  ? PERFORM SESS3

- -
  ? END

/ ROUTE, REPORT3, DC=PR

RECORDED DATA BASE FILE REPORT

AB5972  9
AB5973  0
AB5975  3
B00013  8
CB0168  0
CB1001  0
CB1003  0
CB1005  0
CH0059  6
CH0060  0
CH0060  0
CM0575  6
SH0011  3
ST0592  0
TY5015  0
XM6515  0
YB0020  0
YB0059  0
YB0060  4

** END OF REPORT REPORT3 **

Figure 6-5. Generating Data Base File REPORT3 From a Permanent Catalog
DUPLICATING CATALOGED REPORTS

Reports that are recorded on a permanent catalog can be duplicated on the default catalog. Conversely, reports that are recorded on the default catalog can be duplicated on a permanent catalog. Session names, report names, and transmission IDs (the three-digit number appended to a directive at recording time) can be selected for duplication.

A duplicate report operation from the default catalog to the permanent catalog is shown in figure 6-7.

This option is a convenient way to add reports to an existing permanent catalog.

A duplicate report operation from the permanent catalog to the default catalog is shown in figure 6-8. This option is a convenient way to access cataloged report formats that require modification for one-time use. Changes can be made to the duplicated report format on the default catalog without affecting the report format stored on the permanent file catalog.
REPORTS is formatted on the default catalog.

Attach the QUCAT catalog. NOS/BE requires an ID parameter instead of a UN parameter.

Copy SESS5.

Copy REPORTS.

Retrieve and execute the directives recorded on permanent file catalog QUCAT.

Terminate Query Update.

**CAUTION**

DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQZ

/ ROUTE,REPORTS,DC=PR

NOS/BE control statement would be the following: COMMAND- BATCH,REPORTS,PRINT

<table>
<thead>
<tr>
<th>REPORT</th>
<th>COUNT</th>
</tr>
</thead>
<tbody>
<tr>
<td>AB5972</td>
<td>9</td>
</tr>
<tr>
<td>AB5973</td>
<td>0</td>
</tr>
<tr>
<td>AB5976</td>
<td>3</td>
</tr>
<tr>
<td>B80013</td>
<td>8</td>
</tr>
<tr>
<td>CB0168</td>
<td>0</td>
</tr>
<tr>
<td>CB1001</td>
<td>0</td>
</tr>
<tr>
<td>CB1003</td>
<td>0</td>
</tr>
<tr>
<td>CB1005</td>
<td>0</td>
</tr>
<tr>
<td>CH0059</td>
<td>6</td>
</tr>
<tr>
<td>CH0060</td>
<td>0</td>
</tr>
<tr>
<td>CH0080</td>
<td>0</td>
</tr>
<tr>
<td>CM0575</td>
<td>6</td>
</tr>
<tr>
<td>SH0011</td>
<td>3</td>
</tr>
<tr>
<td>ST0592</td>
<td>0</td>
</tr>
<tr>
<td>TY0015</td>
<td>0</td>
</tr>
<tr>
<td>XN6158</td>
<td>0</td>
</tr>
<tr>
<td>YB0020</td>
<td>0</td>
</tr>
<tr>
<td>YB0059</td>
<td>0</td>
</tr>
<tr>
<td>YB0060</td>
<td>4</td>
</tr>
</tbody>
</table>

** END OF REPORT REPORTS **
Call Query Update.

Attach the permanent file catalog QCAT. NOS/BE requires an ID parameter instead of a UN parameter.

Copy SESS3.

Copy REPORT4.

Future PERFORM directives will use the default catalog.

Retrieve and execute the directives duplicated on the default catalog. Notice that the new session name DUP3 is used; SESS3 would not be found on the default catalog.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND BATCH REPORT3 PRINT

Figure 6-8. Duplicating a Report From a Permanent to a Default Catalog
ALTERING REPORTS

Report formats can be altered by adding or removing one or more layout directives. The report whose format is to be altered can reside on the permanent or default catalog. The operation is initiated by the ALTER directive, which names the report. The directives that immediately follow effect the changes.

Adding a Directive

A new directive is to be added to REPORT3. Each time the report is produced, the system is to supply the data and position it on the title line at column 45. The addition of the new directive is shown in figure 6-9.

The QUCAT catalog on which REPORT3 resides is attached with the VERSION directive. The report to be altered is named with the ALTER directive. The horizontal and vertical positioning of the date is supplied with the DATE directive.

Report modification ends when a directive not associated with report modification is entered. Since the PERFORM directive is not related to report modification, no further modifications are made. The subsequent report reflects the current date.

```
/ QU
QUERY UPDATE 3.4 SYSEDIT-82110 82/05/17 14.45.09

...?

? VERSION QUCAT (UN=username/M=M)

...

? ALTER REPORT3

...

? DATE AT TITLE-LINE COLUMN 45

...

? PERFORM SESS3

...

? END

/ ROUTE,REPORT3,DC=PR

RECORDED DATA BASE FILE REPORT 81/06/17

AB5972 9
AB5973 0
AB5975 3
BD0013 8
CB0168 0
CB1001 0
CB1003 0
CB1005 0
CH0059 6
CH0060 0
CH0080 0
CM0575 6
SH0011 3
ST0592 0
TY5015 0
YN6158 0
YB0020 0
YB0059 0
YB0060 4

** END OF REPORT REPORT3 **
```

Figure 6-9. Adding a Directive to a Report
Erasing a Directive

A new title is to be given to REPORT3. The present TITLE directive is to be removed and a new TITLE directive is to be stored. The use of the ERASE directive is shown in figure 6-10.

The QUCAT catalog on which REPORT3 resides is attached with the VERSION directive and the report to be altered is named with the ALTER directive. The ERASE directive indicates the TITLE directive is to be removed. Query Update erases the directive and prints its contents. The TITLE directive adds a new title. The PERFORM directive modifies the session.

The subsequent report reflects the new title.

```
/ QU
QUERY UPDATE 3.4 SYSEDIT-82110  82/05/17  14.52.45
--
? VERSION QUCAT (UN=username/M=W)
--
? ALTER REPORT3
--
? ERASE REPORT REPORT3 TITLE
  TITLE AT LINE 2 IS $RECORDED DATA BASE
  FILE REPORTS AT LINE 3 IS $   $
--
? TITLE AT LINE 2 IS +
? ALTERED DATA BASE FILE REPORTS +
? AT LINE 3 IS $   $
--
? PERFORM SESS3
--
? END
/ ROUTE,REPORT3,DC=PR

ALTERED DATA BASE FILE REPORT 81/06/17
AB5972  9
AB5973  0
AB5975  3
BB0013  8
CB0168  0
CB1001  0
CB1003  0
CB1005  0
CH0059  6
CH0060  0
CH0083  0
CM0575  6
SH0011  3
ST0592  0
TY0515  0
XN6158  0
YB0020  0
YB0059  0
YB0060  4
** END OF REPORT REPORT3 **
```

Call Query Update.

Attach the permanent file catalog QUCAT. NOS/BE requires an ID parameter instead of a UN parameter.

Locate the report directives for REPORT3.

Request the title line to be erased.

Query Update erases the title line after printing its contents.

Supply a new title line.

Terminate report modification. Retrieve and execute the directives in SESS3.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND- BATCH,REPORT3,PRINT

Figure 6-10. Erasing a Directive From a Report
SAMPLE REPORTS

Sample reports are shown on the following pages. All reports are recorded on the permanent catalog QUCAT and the reference data base file INVENTORY. For NOS, the write permission mode parameter (M=W) must be included on the VERSION directive for both the owner and the alternate user. Many of the Query Update directives that appear in the reports will be familiar because they have already been discussed in relation to cataloging operations. For this reason, only new directives are detailed in the figures.

Figure 6-11 is a sample report that extracts information from four fields, sorts the information on the IN-STOCK field, positions the information in four columns, and supplies the title, date, and time. This report shows the most commonly used report directives and introduces two other directives: SORT and TIME.

Figure 6-12 is a sample report that shows the definition of a temporary item. As mentioned in an earlier section, a temporary item is one of the three categories of data handled by Query Update.

This report defines temporary item PROFIT-MARGIN and uses it to calculate entries for one of the report columns. Three directives are introduced: DEFINE, TABS, and EVALUATE.

Figure 6-13 is a sample report that presents one of many possible applications for the universal character. This report declares the asterisk as the universal character and uses it for special testing against the INV-NO field. Three directives are introduced: UNIVERSAL, SPECIFY, and SELECT.

Figure 6-14 is a sample report that extracts information from four fields, sorts the information on the IN-STOCK field, and specifies a break condition based on the contents of that field. Three directives are introduced: HEADING, PAGE-NUMBER, and BREAK.

Figure 6-15 is a sample report that specifies vertical positioning for the output report. Two directives are introduced: PAGE-SIZE and RECAP.

Figure 6-16 is a sample report that shows how lines of text can be designated to precede and follow a report. Two directives are introduced: PREFACE and SUMMARY.

```
/ QU
QUERY UPDATE 3.4 SYSEDIT-82110  82/05/17  15.51.19

? VERSION QUCAT (UN=username)

? RECORDING SESS6

001--
? INVOKE QUSUB (UN=username)

002--
? EXTRACT UPON FILE6 INV-NO DESCRIPTION +
? IN-STOCK UNIT-PRICE

003--
? REWIND FILE6 SORT6

004--
? SORT FILE6 UPON SORT6 ON IN-STOCK

005--
? PREPARE REPORT6 FROM SORT6

006--
? RECORDING OFF
    END OF SESSION SESS6

? FORMAT REPORT6

? DETAIL IS IN-STOCK IN COLUMN 1 +
? INV-NO IN COLUMN 11 +
? DESCRIPTION IN COLUMN 24 +
? UNIT-PRICE IN COLUMN 48
```

Call Query Update.

Attach the permanent catalog. NOS/BE requires an ID parameter instead of a UN parameter.

Begin recording session SESS6.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Extract four fields from FILE6.

Rewind FILE6 and the file to be used for sorting operations.

Resequence records in FILE6 and place them on file SORT6.

Prepare REPORT6 from the sorted file.

Terminate recording of SESS6.

Format a report named REPORT6.

Supply the report line content.

Figure 6-11. Sample Report Illustrating the Basic Directives (Sheet 1 of 2)
Supply the report title.

Request output of the current date.

Request output of the current time.

Retrieve and execute SESS6, which contains the directives in REPORT6.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND BATCH, REPORT6, PRINT

BASIC REPORT

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Quantity</th>
<th>Unit Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>AO5973</td>
<td>128.50</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>AR5975</td>
<td>1300.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TY5015</td>
<td>369.00</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>AO5972</td>
<td>389.95</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CH0060</td>
<td>149.95</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CH0059</td>
<td>295.00</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ST0592</td>
<td>16.20</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>CB0168</td>
<td>19.52</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>CH0080</td>
<td>96.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BB0013</td>
<td>15.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CB1001</td>
<td>45.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CB1003</td>
<td>60.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CB1005</td>
<td>90.00</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>SH0011</td>
<td>39.95</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>XN6158</td>
<td>95.00</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>YB0060</td>
<td>39.95</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>YB0059</td>
<td>69.95</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>CM0575</td>
<td>3.98</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>YB0020</td>
<td>19.95</td>
<td></td>
</tr>
</tbody>
</table>

** END OF REPORT REPORT6 **

Figure 6-11. Sample Report Illustrating the Basic Directives (Sheet 2 of 2)

/ QU

QUERY UPDATE 3.4 SYSEDIT-82110 82/05/18 09.15.12

Call Query Update.

Attach the permanent file catalog. NOS/BE requires an ID parameter instead of a UN parameter.

Begin recording session SESS7.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Extract three fields upon FILE7.

Figure 6-12. Sample Report Illustrating the Definition of a Temporary Date Item (Sheet 1 of 2)
Establish a temporary data item named PROFIT-MARGIN. The value of the item is a mathematical expression that is to be evaluated by Query Update.

Prepare REPORT7 from the extracted file.

Terminate execution of SESS7.

Terminate recording of SESS7.

Format a report named REPORT7.

Establish tab positions at columns 1 and 20 for the report layout.

Perform the mathematical operation indicated in the above DEFINE directive.

Supply the report line content.

Supply the report title and set up two column headings.

Retrieve and execute SESS7.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND- BATCH,REPORT7,PRINT

** END OF REPORT REPORT7 **

Figure 6-12. Sample Report Illustrating the Definition of a Temporary Data Item (Sheet 2 of 2)
Call Query Update.

Attach the permanent file catalog. NOS/BE requires an ID parameter instead of a UN parameter.

Begin recording SESSB.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Extract three fields upon FILE8.

Establish a character (*) that marks a character position to be ignored during comparison testing.

Establish a name for convenient reference to a condition. The condition name in this directive is CHAIRS. The condition is an inventory number (INV-NO) whose first two characters are equal to CH. The universal character prohibits testing on the last four character positions of INV-NO.

Prepare REPORT8 from the extracted file.

Terminate recording of SESSB.

Format a report named REPORT8.

Test each record to see if it satisfies the condition in the SPECIFY directive.

Supply the report line content.

Supply the report title and set up two column headings.

Retrieve and execute the directives in SESSB.

Terminate Query Update.

NOS/BE control statement would be the following: COMMAND- BATCH, REPORT8, PRINT

** END OF REPORT REPORT8 **

Figure 6-13. Sample Report Illustrating the Universal Character

<table>
<thead>
<tr>
<th>ITEM</th>
<th>IN STOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM CHAIR</td>
<td>7</td>
</tr>
<tr>
<td>DESK CHAIR</td>
<td>5</td>
</tr>
<tr>
<td>SWIVEL CHAIR</td>
<td>9</td>
</tr>
</tbody>
</table>

60499000 C 6-15 


Call Query Update.

Attach the permanent file catalog. NOS/BE requires an ID parameter instead of the UN parameter.

Begin recording SESS9.

Name the subschema. NOS/BE requires an ID parameter instead of the UN parameter.

Extract four fields upon FILE9.

Rewind FILE9 and the file to be used for sorting operations.

Sort FILE9 on the IN-STOCK field.

Prepare REPORT9 from the sorted file.

Terminate recording of SESS9.

Format a report named REPORT9.

Supply the report line content.

Supply the report title.

Specify the content and positioning of a heading. The level number 0 indicates that the heading is to appear before any input data is processed. The heading is to appear on each page of the report.

Indicate vertical and horizontal positioning of page numbers.

Establish when the body of the report page is to be interrupted for a heading or footing. The break occurs when the content of IN-STOCK changes. A printer page-eject is included.

Retrieve and execute SESS9.

Terminate Query Update.

Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 1 of 3)
Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 2 of 3)
<table>
<thead>
<tr>
<th>IN STOCK</th>
<th>PART NO</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>DB0013</td>
<td>BULLETIN BOARD</td>
<td>15.00</td>
</tr>
<tr>
<td>10</td>
<td>CB1001</td>
<td>1-DR FILE CABINET</td>
<td>45.00</td>
</tr>
<tr>
<td>10</td>
<td>CB1003</td>
<td>3-DR FILE CABINET</td>
<td>60.00</td>
</tr>
<tr>
<td>10</td>
<td>CB1005</td>
<td>5-DR FILE CABINET</td>
<td>90.00</td>
</tr>
<tr>
<td>10</td>
<td>SH0011</td>
<td>3-SHELF BOOK CASE</td>
<td>39.95</td>
</tr>
<tr>
<td>10</td>
<td>XN6158</td>
<td>COFFEE TABLE</td>
<td>95.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN STOCK</th>
<th>PART NO</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>YB0060</td>
<td>TABLE LAMP</td>
<td>39.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN STOCK</th>
<th>PART NO</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>YB0059</td>
<td>FLOOR LAMP</td>
<td>69.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN STOCK</th>
<th>PART NO</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>CM0575</td>
<td>LETTER RACK</td>
<td>3.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>IN STOCK</th>
<th>PART NO</th>
<th>DESCRIPTION</th>
<th>PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>YB0020</td>
<td>DESK LAMP</td>
<td>19.95</td>
</tr>
</tbody>
</table>

** END OF REPORT REPORT9 **

Figure 6-14. Sample Report Illustrating a Break on an Item (Sheet 3 of 3)
Call Query Update.

Attach the permanent file catalog. NOS/BE requires an ID parameter instead of a UN parameter.

Begin recording SESS10.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Extract two fields upon FILE10.

Rewind FILE10 and the file to be used for sorting operations.

Sort the file on the description field.

Prepare REPT10 from the sorted file.

Terminate recording of SESS10.

Format a report named REPT10.

Supply the report line content.

Supply the report title.

Establish special placement of detail lines on a report page. Two sectional divisions are to appear across the page. The VERTICAL option places successive entries in one section and then continues to the second section.

Generate a line of information at the completion of a report page. A literal expression is selected to appear two lines after the last detail line.

Retrieve and execute SESS10.

Terminate Query Update.

Figure 6-15. Sample Report Illustrating Report Page Positioning (Sheet 1 of 2)
VERTICAL SECTIONS REPORT

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM CHAIR</td>
<td>295.00</td>
</tr>
<tr>
<td>BULLETIN BOARD</td>
<td>15.00</td>
</tr>
<tr>
<td>CHALK BOARD</td>
<td>19.52</td>
</tr>
<tr>
<td>COFFEE TABLE</td>
<td>95.00</td>
</tr>
<tr>
<td>DESK CHAIR</td>
<td>149.95</td>
</tr>
<tr>
<td>DESK LAMP</td>
<td>19.95</td>
</tr>
<tr>
<td>ELECT TYPEWRITER</td>
<td>369.00</td>
</tr>
<tr>
<td>FLOOR LAMP</td>
<td>69.95</td>
</tr>
<tr>
<td>LETTER RACK</td>
<td>3.98</td>
</tr>
<tr>
<td>METAL DESK</td>
<td>389.95</td>
</tr>
<tr>
<td>OAK DESK</td>
<td>1282.50</td>
</tr>
<tr>
<td>STOOL</td>
<td>16.20</td>
</tr>
<tr>
<td>SWIVEL CHAIR</td>
<td>96.00</td>
</tr>
<tr>
<td>TABLE LAMP</td>
<td>39.95</td>
</tr>
<tr>
<td>WALNUT DESK</td>
<td>1300.00</td>
</tr>
<tr>
<td>1-DR FILE CABINET</td>
<td>45.00</td>
</tr>
<tr>
<td>3-DR FILE CABINET</td>
<td>60.00</td>
</tr>
<tr>
<td>3-SHELF BOOK CASE</td>
<td>39.95</td>
</tr>
<tr>
<td>5-DR FILE CABINET</td>
<td>90.00</td>
</tr>
</tbody>
</table>

END OF VERTICAL SECTIONS REPORT

** END OF REPORT REPT10 **

Figure 6-15. Sample Report Illustrating Report Page Positioning (Sheet 2 of 2)

NOS/BE control statements. The files contain preface and summary information for the report.

COMMAND- ATTACH,PROFILE,ID=USER
COMMAND- FILE,PROFILE,BT=C,RT=Z
COMMAND- ATTACH,SUMFILE,ID=USER
COMMAND- FILE,SUMFILE,BT=C,RT=Z

NOS control statements.

/ ATTACH,PROFILE,SUMFILE
/ FILE,PROFILE,BT=C,RT=Z
/ FILE,SUMFILE,BT=C,RT=Z

Call Query Update.

/ QU
QUERY UPDATE 3.4 SYESEDT-82110 82/05/18 17.16.34

? VERSION QUCAT (UN=username)

Begin recording SESS11.

? RECORDING SESS11

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

001--
? INVOKE QUSUB (UN=username)

Extract two fields upon FILE11.

002--
? EXTRACT UPON FILE11 DESCRIPTION UNIT-PRICE

Specify a condition named SALE-ITEMS. The condition is a price (UNIT-PRICE) less than $50 dollars.

003--
? SPECIFY SALE-ITEMS AS UNIT-PRICE LT 50

Prepare REPT11 from the extracted file.

004--
? PREPARE REPT11 FROM FILE11

Figure 6-16. Sample Report Illustrating a Report Preface and Summary (Sheet 1 of 2)
DOS--
? RECORDING OFF
END OF SESSION SESS11
--

? FORMAT REPT11

--

? PREFACE IS TEXT FROM PREFILE

--

? SUMMARY IS TEXT FROM SUMFILE

--

? SELECT 1 ON SALE-ITEMS

--

? DETAIL 1 IS DESCRIPTION IN COLUMN 1 +
? UNIT-PRICE IN COLUMN 20

--

? TITLE AT LINE 1 IS $PREFACE/SUMMARY REPORTS +
? IN COLUMN 3 AT LINE 2 IS $ $ AT LINE 3 +
? IS $SALE ITEMS$ IN COLUMN 9 +
? AT LINE 4 IS $ $ +

--

? PERFORM SESS11

--

? END

/ ROUTE,REPT11,CD=PR QU.

DISTRIBUTE TO ALL OUTLETS BY THE 16TH OF THE MONTH

PREFACE/SUMMARY REPORT

SALE ITEMS

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>BULLETIN BOARD</td>
<td>15.00</td>
</tr>
<tr>
<td>CHALK BOARD</td>
<td>19.52</td>
</tr>
<tr>
<td>1-DR FILE CABINET</td>
<td>45.00</td>
</tr>
<tr>
<td>LETTER RACK</td>
<td>3.98</td>
</tr>
<tr>
<td>3-SHELF BOOK CASE</td>
<td>39.95</td>
</tr>
<tr>
<td>STOOL</td>
<td>16.20</td>
</tr>
<tr>
<td>DESK LAMP</td>
<td>19.95</td>
</tr>
<tr>
<td>TABLE LAMP</td>
<td>39.95</td>
</tr>
</tbody>
</table>

SALE ENDS ON THE LAST DAY OF THE MONTH

** END OF REPORT REPT11 **

Figure 6-16. Sample Report Illustrating a Report Preface and Summary (Sheet 2 of 2)
PREVIEWING REPORTS

A two-page sample printout can be produced for any report that has an existing directory. A data base file requires a directory created with the EXTRACT directive; a non-data-base file requires a directory created with the DESCRIBE directive. The sample report is produced with the PREVIEW directive.

When the PREVIEW directive is issued with no options, dummy values of X's and Y's are substituted for alphanumeric data and dummy values of 8's and 9's are substituted for numeric data. When the PREVIEW directive is issued with reference to a file name, actual data values are used.

Use of the PREVIEW directive is shown in figure 6-17. A report entitled PREVIEWED REPORT is previewed before it is prepared from FILE12. This application shows how a report format can be viewed and adjusted, if necessary, before preparation.

```
/ QU
QUERY UPDATE 3.4 SYSEDIT-82110 82/05/19 10.22.37
--
? INVOKE QSUB (UN=username)
--
? EXTRACT UPON FILE12 INV-NO +
? DESCRIPTION UNIT-COST
  19 ACCESSES, 19 HITS, 19 10-S
--
? FORMAT REPT12
--
? DETAIL IS INV-NO IN COLUMN 1 +
? DESCRIPTION IN COLUMN 8 +
? UNIT-COST IN COLUMN 26
--
? TITLE AT LINE 1 IS $PREVIEWED REPORTS +
? AT LINE 2 IS $   $
--
? PREVIEW REPT12
--
? OS,EDIT,REPT12
PRINT,1
PREVIEWED REPORT
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
XXXXXX YYYYYYYYYYYYYYYY 9999999
--
? END
** END OF REPORT REPT12 **
**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQ2
```

Figure 6-17. Previewing a Report
THE REPORT UTILITY

All reports illustrated up to this point are shown being prepared by Query Update through the PREPARE directive. When a PREPARE directive is encountered, Query Update compiles internal tables and uses them to produce the report. The tables, which are collectively referred to as a report information table, serve the following purposes:

- Describe the sequential file that is created as a result of an EXTRACT or DESCRIBE directive.
- Associate the sequential file with the report layout that is established by the FORMAT directive.

Query Update compiles a report information table each time a report is prepared and discards the table after the report is generated.

Reports can be prepared for use by the Report Utility through the COMPILE directive. When a COMPILE directive is used in lieu of the PREPARE directive, Query Update compiles the report information table and stores it on a local file. When the local file is made permanent, reports can be subsequently generated by the following steps:

1. Attach the permanent file containing the compiled report information table.
2. Attach the sequential file.
3. Define the sequential file characters for CRM.

When a report format is prepared in this manner, compilation of the report information table occurs only once; a new report information table has to be generated only when the report layout is changed. This approach results in savings in central processor time and central memory usage.

Use of the COMPILE directive is shown in figure 6-18. A sequential file named COMFILE is generated from database file INVENTORY and the report information table is compiled on local file TBLFILE. Both files are made permanent for subsequent use by the Report Utility.

Use of the Report Utility is shown in figure 6-19. Two sets of control statements are shown. The first set uses COMPILE as the input data file; the second set uses non-data-base file NDINDEX to obtain the same results.

```
COMMAND= REQUEST, COMFILE, *PF
COMMAND= REQUEST, TBLFILE, *PF
/ DEFINE, COMFILE/CT=PU, M=W
/ DEFINE, TBLFILE/CT=PU, M=W
COMMAND= QU
QUERY UPDATE 3,4 SYSEDIT-82110 82/05/19 10,41,32
--
INVOKE QSUB (UN=user name)
--
EXTRACT UPON COMFILE INVNO ON-ORDER
   19 ACCESSES, 19 HITS, 19 10-5
--
FORM COMRPT
--
DETAIL IS INVNO IN COLUMN 1 +
   ON-ORDER IN COLUMN 12
--
TITLE AT LINE 1 IS $COMPILED REPORTS
   AT LINE 2 IS $  $
--
COMPILE COMRPT UPON TBLFILE
PREFACES/SUMMARIES
   (NONE)
--
END
```

Figure 6-18. Compiling Report Specifications (Sheet 1 of 2)
**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE ZZZZZQZ

```
COMMAND= CATALOG,COMFILE,ID=GUSER,RP=999
COMMAND= CATALOG,TBLFILE,ID=GUSER,RP=999
NOS/BE control statements.
```

Figure 6-18. Compiling Report Specifications (Sheet 2 of 2)

```
COMMAND= ATTACH,COMFILE,ID=GUSER
COMMAND= FILE,COMFILE,RT=F
COMMAND= ATTACH,TBLFILE,ID=GUSER
COMMAND= REPORT,R=COMRPT,T=TBLFILE,I=COMFILE
COMMAND= BATCH,COMRPT,PRINT

ATTACH control statements are issued for sequential file COMFILE and report information table TBLFILE. A file control statement is issued for COMFILE to describe the file for CRM. A REPORT control statement is issued to call the Report Utility program to produce the report according to specifications in TBLFILE. The R parameter names the report; the T parameter names the report information table; and the I parameter names the input data file. Appropriate commands direct the finished report to the line printer.

/ ATTACH,COMFILE,TBLFILE
/ FILE,COMFILE,RT=F
/ REPORT,R=COMRPT,T=TBLFILE,I=COMFILE
/ ROUTE,COMRPT,DC=PR

COMPILED REPORT

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AB5972</td>
<td>9</td>
</tr>
<tr>
<td>AB5973</td>
<td>0</td>
</tr>
<tr>
<td>AB5975</td>
<td>3</td>
</tr>
<tr>
<td>BB0013</td>
<td>8</td>
</tr>
<tr>
<td>CB0168</td>
<td>0</td>
</tr>
<tr>
<td>CB1001</td>
<td>0</td>
</tr>
<tr>
<td>CB1003</td>
<td>0</td>
</tr>
<tr>
<td>CB1005</td>
<td>0</td>
</tr>
<tr>
<td>CM0059</td>
<td>6</td>
</tr>
<tr>
<td>CH0060</td>
<td>0</td>
</tr>
<tr>
<td>CH0080</td>
<td>0</td>
</tr>
<tr>
<td>CM0575</td>
<td>6</td>
</tr>
<tr>
<td>SH0011</td>
<td>3</td>
</tr>
<tr>
<td>ST0592</td>
<td>0</td>
</tr>
<tr>
<td>TV0016</td>
<td>0</td>
</tr>
<tr>
<td>XN6189</td>
<td>0</td>
</tr>
<tr>
<td>YB0020</td>
<td>0</td>
</tr>
<tr>
<td>YB0059</td>
<td>0</td>
</tr>
<tr>
<td>YB0060</td>
<td>4</td>
</tr>
</tbody>
</table>

** END OF REPORT COMRPT **

```
COMMAND= ATTACH,INDFILE,ID=GUSER
COMMAND= FILE,INDFILE,RT=F
COMMAND= REPORT,R=COMRPT,T=TBLFILE,I=INDFILE
COMMAND= BATCH,COMRPT,PRINT

/ ATTACH,INDFILE
/ FILE,INDFILE,RT=F
/ REPORT,R=COMRPT,T=TBLFILE,I=INDFILE
/ ROUTE,COMRPT,DC=PR

COMPILED REPORT

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AB5972</td>
<td>25</td>
</tr>
<tr>
<td>CB0168</td>
<td>50</td>
</tr>
<tr>
<td>YB0020</td>
<td>35</td>
</tr>
<tr>
<td>BB0013</td>
<td>40</td>
</tr>
<tr>
<td>AB5975</td>
<td>22</td>
</tr>
<tr>
<td>CH0060</td>
<td>33</td>
</tr>
</tbody>
</table>

** END OF REPORT COMRPT **

The data file can vary from run to run; only its characteristics must remain the same. These control statements attach INDFILE and use it as input data to the Report Utility.

Figure 6-19. Using the Report Utility
One Query Update subschema can reference up to 64 separate data base files. This section of the guide shows the declaration of multiple files and how they can be individually accessed, and presents the relational data base capability in which files can be accessed simultaneously.

DECLARING MULTIPLE FILES

A subschema MULTSUB is shown in figure 7-1. Unlike subschema QSUB, which was presented in earlier sections, this subschema references multiple files. MULTSUB describes the original INVENTORY file along with three additional files named ORDERS, LINEITEMS, and OUTLET.

For ease of illustration, assume that information is stored in the new files. Figure 7-2 shows sample display operations for the new files and introduces the RETURN directive. RETURN allows files to be released to the operating system when they are no longer needed by Query Update. Returning a file frees a portion of the central memory.

If the returned file is a data base area, its internal tables are released; the file itself is not released to the operating system. If the file is a non-data-base file, a RETURN releases the file to the operating system. There is no need to release a data base area specified through an INVOKE directive, which is not a temporary area, because INVOKE does not attach the file; directives that reference the area (such as DISPLAY) attach the file, reference it, and release it.

Two RETURN directives will release to the operating system a temporary area or an area specified through a CREATE directive. The first RETURN releases internal tables; the second performs a RETURN on a non-data-base file and releases it to the operating system.

The LINEITEMS file lends itself to arithmetic operations. Figure 7-3 is a sample report that illustrates subtotaling facilities. Two directives are introduced: MOVE and FOOTING.

MODIFYING MULTIPLE FILES

When modifying files that are data base areas in a multiple file subschema, the UPDATE directive tells Query Update which file is to be modified. The UPDATE directive is shown in figure 7-4.
Call Query Update.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Attach ORDERS and ORDXID, display all fields of ORDERS, and release ORDERS and ORDXID to the operating system.

Release the internal tables of file ORDERS.

Attach LINEITEMS and ITMIDX, display all LINEITEMS, and release LINEITEMS and ITMIDX to the operating system.

Release the internal tables of file LINEITEMS.

Attach OUTLET, display all fields of OUTLET, and release OUTLET to the operating system.

Build a one-field subset of OUTLET and an associated directory.

Release the internal tables associated with file FILE1 and release FILE1 to the operating system.

---

<table>
<thead>
<tr>
<th>ORDER-DATE</th>
<th>ORDER-REC</th>
<th>OUTLET-CODE</th>
<th>INC-NO</th>
<th>ITEM-PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>07/01/81</td>
<td>2374.45</td>
<td>AZ43</td>
<td>5</td>
<td>19.52</td>
</tr>
<tr>
<td>07/02/81</td>
<td>8647.20</td>
<td>NV44</td>
<td>3</td>
<td>1300.00</td>
</tr>
<tr>
<td>07/05/81</td>
<td>754.00</td>
<td>CA46</td>
<td>6</td>
<td>1282.50</td>
</tr>
<tr>
<td>12/03/81</td>
<td>70.00</td>
<td>CA46</td>
<td>9</td>
<td>15.00</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>ITEM-PRICE</th>
<th>OUTLET-CODE</th>
<th>ITEM-NO</th>
<th>ITEM-PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>309.95</td>
<td>AZ43</td>
<td>3</td>
<td>1282.50</td>
</tr>
<tr>
<td>369.00</td>
<td>AZ43</td>
<td>5</td>
<td>3.98</td>
</tr>
<tr>
<td>19.52</td>
<td>AZ43</td>
<td>5</td>
<td>15.00</td>
</tr>
<tr>
<td>1300.00</td>
<td>NV44</td>
<td>3</td>
<td>60.00</td>
</tr>
<tr>
<td>1282.50</td>
<td>NV44</td>
<td>6</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>ITEM-PRICE</th>
<th>ITEM-NO</th>
<th>ITEM-PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>309.95</td>
<td>3</td>
<td>1282.50</td>
</tr>
<tr>
<td>369.00</td>
<td>5</td>
<td>3.98</td>
</tr>
<tr>
<td>19.52</td>
<td>5</td>
<td>15.00</td>
</tr>
<tr>
<td>1300.00</td>
<td>3</td>
<td>60.00</td>
</tr>
<tr>
<td>1282.50</td>
<td>6</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>ITEM-PRICE</th>
<th>ITEM-NO</th>
<th>ITEM-PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>309.95</td>
<td>3</td>
<td>1282.50</td>
</tr>
<tr>
<td>369.00</td>
<td>5</td>
<td>3.98</td>
</tr>
<tr>
<td>19.52</td>
<td>5</td>
<td>15.00</td>
</tr>
<tr>
<td>1300.00</td>
<td>3</td>
<td>60.00</td>
</tr>
<tr>
<td>1282.50</td>
<td>6</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>ITEM-PRICE</th>
<th>ITEM-NO</th>
<th>ITEM-PRICE</th>
</tr>
</thead>
<tbody>
<tr>
<td>309.95</td>
<td>3</td>
<td>1282.50</td>
</tr>
<tr>
<td>369.00</td>
<td>5</td>
<td>3.98</td>
</tr>
<tr>
<td>19.52</td>
<td>5</td>
<td>15.00</td>
</tr>
<tr>
<td>1300.00</td>
<td>3</td>
<td>60.00</td>
</tr>
<tr>
<td>1282.50</td>
<td>6</td>
<td>100.00</td>
</tr>
</tbody>
</table>

---

Figure 7-2. Querying Multiple Files
Call Query Update.

Attach the permanent catalog.

Begin recording session MSESS.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Extract 5 fields upon MFILE. The AS option renames each field.

Rewind MFILE and the file to be used for sorting operations.

Sort the file on the ON-ORDER field, renamed Q.

Define a temporary item named TOTAL whose value is evaluated by Query Update.

Define a temporary item named SUBTOTAL.

Define a temporary item named GRANDTOTAL.

Prepare MRPT from the sorted file.

Terminate recording of MSESS. Query Update acknowledges the end of the session.

Format a report named MRPT.

Evaluate temporary item TOTAL before the DETAIL production step.

Supply the report title.

Supply the report line content. The new field names must be used.

Break when the content of field 0 (the renamed ORDER-NO field) changes.

Specify the heading that is to appear before any input data is processed.

Figure 7-3. Sample Report Illustrating Subtotaling Operations (Sheet 1 of 2)
The FOOTING directive shows the content and positioning of a footing. Level 0 indicates that this footing is associated with the end of data.

FOOTING 1 is associated with BREAK 1. This footing will appear when the content of field 0 (the renamed ORDER-NO field) changes.

The MOVE directive moves values to data names. Values are moved before the DETAIL production step.

Set the content of SUBTOTAL to zero after a level 1 footing is executed.

Terminate Query Update.

Print the report. Under NOS/BE the control statement would be COMMAND- BATCH,MRPT,PRINT

<table>
<thead>
<tr>
<th>ORDER</th>
<th>ITEM</th>
<th>QTY</th>
<th>PRICE</th>
<th>TOTAL</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>430001</td>
<td>AB5972</td>
<td>3</td>
<td>389.95</td>
<td>1169.85</td>
<td>AZ43</td>
</tr>
<tr>
<td>430001</td>
<td>TY5015</td>
<td>3</td>
<td>369.00</td>
<td>1107.00</td>
<td>AZ43</td>
</tr>
<tr>
<td>430001</td>
<td>CB0168</td>
<td>5</td>
<td>19.52</td>
<td>97.60</td>
<td>AZ43</td>
</tr>
</tbody>
</table>

SUBTOTAL 2374.45

<table>
<thead>
<tr>
<th>ORDER</th>
<th>ITEM</th>
<th>QTY</th>
<th>PRICE</th>
<th>TOTAL</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>440001</td>
<td>AB5975</td>
<td>3</td>
<td>1300.00</td>
<td>3900.00</td>
<td>NV44</td>
</tr>
<tr>
<td>440001</td>
<td>CH0060</td>
<td>6</td>
<td>149.95</td>
<td>899.70</td>
<td>NV44</td>
</tr>
<tr>
<td>440001</td>
<td>AB5975</td>
<td>3</td>
<td>1282.50</td>
<td>3847.50</td>
<td>NV44</td>
</tr>
</tbody>
</table>

SUBTOTAL 8647.20

<table>
<thead>
<tr>
<th>ORDER</th>
<th>ITEM</th>
<th>QTY</th>
<th>PRICE</th>
<th>TOTAL</th>
<th>CODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>460001</td>
<td>CB1003</td>
<td>7</td>
<td>60.00</td>
<td>420.00</td>
<td>CA46</td>
</tr>
<tr>
<td>460001</td>
<td>BB0013</td>
<td>9</td>
<td>15.00</td>
<td>135.00</td>
<td>CA46</td>
</tr>
<tr>
<td>460001</td>
<td>CM0575</td>
<td>50</td>
<td>3.98</td>
<td>199.00</td>
<td>CA46</td>
</tr>
</tbody>
</table>

SUBTOTAL 754.00

GRAND TOTAL 11775.65

** END OF REPORT MRPT **

Figure 7-3. Sample Report Illustrating Subtotaling Operations (Sheet 2 of 2)
Call Query Update.

Name the subschema. NOS/BE requires an ID parameter instead of a UN parameter.

Name the file.

Modify two fields.

Terminate the MODIFY directive.

Verify that outlet CA46 has a new address.

Name another file.

Store a new record.

Terminate the STORE directive.

Terminate Query Update.

Figure 7-4. Modifying Multiple Files
ESTABLISHING RELATIONSHIPS

The multiple file subschema MULTSUB presented in
the preceding paragraphs describes several files. Each file could be accessed for query, update, or
report operations — but data could be retrieved
from only one file at a time.

Inherent relationships exist among INVENTORY,
ORDERS, LINEITEMS, and OUTLET. Some data items
appear in more than one file. For example, data
item INV-NO appears in INVENTORY and LINEITEMS, and
data item OUTLET-CODE appears in ORDERS, LINEITEMS,
and OUTLET. A logical connection can be estab-
lished within the subschema to allow two or more
files to be queried as one file.

A relational subschema is shown in Figure 7-5.
MULTSUB, renamed RELSUB, is expanded to include a
Relation Division that specifies two relations:
CONTEST and ACTIVITY-CHECK.

<table>
<thead>
<tr>
<th>IDENTIFICATION DIVISION.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUBSCHEMA NAME IS RELSUB</td>
</tr>
<tr>
<td>DATA DIVISION.</td>
</tr>
<tr>
<td>AREA-NAME IS INVENTORY M IS W</td>
</tr>
<tr>
<td>INDEX IS INVIDX M IS W</td>
</tr>
<tr>
<td>ORGANIZATION IS INDEXED NEW</td>
</tr>
<tr>
<td>KEY IS INV-NO</td>
</tr>
<tr>
<td>KEY IS ALTERNATE BACK-ORDER DUPLICATES INDEXED</td>
</tr>
<tr>
<td>RECORD-NAME IS INV-REC</td>
</tr>
<tr>
<td>02 INV-NO PIC X(6)</td>
</tr>
<tr>
<td>02 IN-STOCK PIC Z(3)9</td>
</tr>
<tr>
<td>02 BACK-ORDER PIC Z(3)9</td>
</tr>
<tr>
<td>02 ON-ORDER PIC Z(3)9</td>
</tr>
<tr>
<td>02 REORDER-PT PIC Z(3)9</td>
</tr>
<tr>
<td>02 UNIT-COST PIC Z(4).99</td>
</tr>
<tr>
<td>02 UNIT-PRICE PIC Z(4).99</td>
</tr>
<tr>
<td>02 DESCRIPTION PIC X(17)</td>
</tr>
<tr>
<td>AREA-NAME IS ORDERS M IS W</td>
</tr>
<tr>
<td>INDEX IS ORIDX M IS W</td>
</tr>
<tr>
<td>ORGANIZATION IS INDEXED NEW</td>
</tr>
<tr>
<td>KEY IS ORDER-NO</td>
</tr>
<tr>
<td>KEY IS ALTERNATE OUTLET-CODE DUPLICATES INDEXED</td>
</tr>
<tr>
<td>RECORD-NAME IS ORDER-REC</td>
</tr>
<tr>
<td>02 ORDER-NO PIC 9(6)</td>
</tr>
<tr>
<td>02 ORDER-DATE PIC X(10)</td>
</tr>
<tr>
<td>02 TOT-ORDER PIC 2(6).99</td>
</tr>
<tr>
<td>02 OUTLET-CODE PIC X(4)</td>
</tr>
<tr>
<td>AREA-NAME IS LINEITEMS M IS W</td>
</tr>
<tr>
<td>INDEX IS LINDEX M IS W</td>
</tr>
<tr>
<td>ORGANIZATION IS ACTUAL NEW</td>
</tr>
<tr>
<td>KEY IS ACT-KEY</td>
</tr>
<tr>
<td>KEY IS ALTERNATE ORDER-NO DUPLICATES INDEXED</td>
</tr>
<tr>
<td>KEY IS ALTERNATE INV-NO DUPLICATES INDEXED</td>
</tr>
<tr>
<td>RECORD-NAME IS ITEM-REC</td>
</tr>
<tr>
<td>02 ACT-KEY PIC 9(5) USAGE IS INTEGER</td>
</tr>
<tr>
<td>02 ORDER-NO PIC 9(6)</td>
</tr>
<tr>
<td>02 INV-NO PIC X(6)</td>
</tr>
<tr>
<td>02 QUANTITY PIC Z(4).99</td>
</tr>
<tr>
<td>02 ITEM-PRICE PIC Z(4).99</td>
</tr>
<tr>
<td>02 OUTLET-CODE PIC X(4)</td>
</tr>
<tr>
<td>AREA-NAME IS OUTLET M IS W</td>
</tr>
<tr>
<td>ORGANIZATION IS DIRECT NEW</td>
</tr>
<tr>
<td>KEY IS OUTLET-CODE</td>
</tr>
<tr>
<td>RECORD-NAME IS OUTLET-REC</td>
</tr>
<tr>
<td>02 OUTLET-CODE PIC X(4)</td>
</tr>
<tr>
<td>02 OUTLET-ADDR</td>
</tr>
<tr>
<td>05 OUTLET-STREET PIC X(16)</td>
</tr>
<tr>
<td>05 OUTLET-CITY PIC X(16)</td>
</tr>
<tr>
<td>05 OUTLET-STATE PIC A(2)</td>
</tr>
<tr>
<td>05 OUTLET-ZIP PIC 9(5)</td>
</tr>
</tbody>
</table>

RELATION DIVISION,
RELATION-NAME IS CONTEST
JOIN WHERE OUTLET-CODE OF ORDER-REC EQ OUTLET-CODE OF OUTLET-REC
RESTRICT ORDER-REC WHERE TOT-ORDER GE 5000
RELATION-NAME IS ACTIVITY-CHECK
JOIN WHERE INV-NO OF INV-REC EQ INV-NO OF ITEM-REC
OUTLET-CODE OF ITEM-REC EQ OUTLET-CODE OF OUTLET-REC
RESTRICT OUTLET-REC WHERE OUTLET-CITY EQ "PHOENIX"

Figure 7-5. A Relational Subschema
The CONTEST relation joins the ORDERS and OUTLET areas. Data from both files can be returned whenever these two files are traversed by Query Update. The files are traversed when at least one data item in each file is referenced in a Query Update transmission. The CONTEST relation imposes one restriction on the ORDERS file. Data is returned only when the TOT-ORDER field contains a value greater than or equal to 5000.

The ACTIVITY-CHECK relation joins the INVENTORY, LINEITEMS, and OUTLET areas. Data can be returned from these files whenever they are traversed by Query Update. The ACTIVITY-CHECK relation imposes one restriction on the OUTLET file. Data is returned only when the OUTLET-CITY field contains the value PHOENIX.

The relationships are shown in the diagram in figure 7-6. The solid arrow traces CONTEST. The shaded arrows trace ACTIVITY-CHECK.

QUERYING RELATED FILES

Related files are queried in the same manner as individual files. When queries are issued against files joined in a relation, Query Update searches the joined files and returns the qualifying data as one projected record. Query Update determines which relation to use by the data names specified in a directive.

Interactive query begins with the INVOKE directive, which makes all areas and relations described in the subschema available. Sample queries involving relations are shown in figure 7-7.

RECONCILING AMBIGUITIES

Ambiguities can exist if the same files are joined in different relations and the data items named in the query are from these files only. The VIA directive specifies the relation that should be followed when such ambiguities exist. The directive can be entered alone as a single transmission or included in a query transmission. The VIA directive is illustrated in figure 7-8.

MODIFYING RELATED FILES

Relations are limited to read-only operations. When files joined in a relationship are to be modified, the modification is made on each individual file, one at a time.

Care should be taken when joined files are modified. If, for example, an inventory number (INV-NO) in the INVENTORY file is changed, appropriate changes must be reflected in the LINEITEMS file to maintain the integrity of projected records.

![Diagram of relationships between areas](image-url)

Figure 7-6. The Relationships Between Areas
CALL QUERY UPDATE 3.4 SYSEDIT-82110 81/06/26 13.30.26

--

? INVOKE RELSUB (UN=username)

--

? DISPLAY TOT-ORDER OUTLET-CITY

8647.20 LAS VEGAS
   1 ACCESSES, 1 HITS, 4 IO-S

--

? DISPLAY $THE ORDER THAT WAS EQUAL TO +
? OR GREATER THAN $5000 Was TOT-ORDER +
? $ ****THE WINNER IS$ OUTLET-CITY

THE ORDER THAT WAS EQUAL TO OR GREATER THAN $5000 WAS 8647.20
 **** THE WINNER IS LAS VEGAS
   1 ACCESSES, 1 HITS, 4 IO-S

--

? DISPLAY DESCRIPTION QUANTITY OUTLET-CITY
METAL DESK 3 PHOENIX
CHALK BOARD 5 PHOENIX
ELECT TYPewriter 3 PHOENIX
   19 ACCESSES, 3 HITS, 35 IO-S

--

? DISPLAY DESCRIPTION INV-NO OF ITEM-REC +
? OUTLET-CITY
METAL DESK 905972 PHOENIX
CHALK BOARD C00168 PHOENIX
ELECT TYPewriter 950075 PHOENIX
   19 ACCESSES, 3 HITS, 35 IO-S

--

? EXTRACT UPON RELFILE DESCRIPTION QUANTITY +
? ITEM-PRICE OUTLET-CITY
   19 ACCESSES, 3 HITS, 35 IO-S

--

? FORMAT RELRPT

--

? TITLE AT LINE ↑ IS +
? PHOENIX ACTIVITY REPORTS IN COLUMN 11 +
? AT LINE 2 IS $ $

--

? DETAIL IS DESCRIPTION IN COLUMN 1 +
? QUANTITY IN COLUMN 20 +
? ITEM-PRICE IN COLUMN 26 +
? OUTLET-CITY IN COLUMN 35

--

? PREPARE RELRPT FROM RELFILE

--

? END

Figure 7-7. Querying a Relation (Sheet 1 of 2)
**CAUTION**
DEFAULT CATALOG REMAINS AS LOCAL FILE IZZIZAR

PRINT the report. Under NOS/8E the control statement would be COMMAND-BATCH, RELRPT, PRINT.

PHOENIX ACTIVITY REPORT

| METAL DESK | 3 | 389.95 | PHOENIX |
| CHALK BOARD | 5 | 19.52 | PHOENIX |
| ELECT TYPEWRITER | 3 | 369.00 | PHOENIX |

** END OF REPORT RELRPT **

Figure 7-7. Querying a Relation (Sheet 2 of 2)

```plaintext
INVOLVE RLSUB (UN=username)

VIA ACTIVITY-CHECK

VIA can be a separate transmission.

DISPLAY ...

Where the ellipses indicate data names from areas INVENTORY and LINE ITEMS, which are joined in relation ACTIVITY-CHECK.

VIA ACTIVITY-CHECK DISPLAY ...

VIA can precede a query.

DISPLAY ... VIA ACTIVITY-CHECK

VIA can follow a query.
```

Figure 7-8. The VIA Directive
A number of special Query Update utilities are available to assist programmers by providing access to and clarification of stored information and offering subscribing capabilities.

Query Update uses a large amount of stored reference information within its processing environment. Typical information includes subschema descriptions and permanent catalog contents. This type of information is made available to the programmer through several informative directives.

Subscribing operations can be performed within Query Update. Elements within an array or repeating group can be accessed by supplying an integer, data name, or figurative subscript.

INFORMATIVE DIRECTIVES

Query Update includes the following four directives that supply information that can be helpful to the new user:

DIAGNOSTIC

- Expands diagnostic messages to aid in debugging

EXHIBIT

- Displays current information related to permanent and temporary data names, relations, reports, and sessions

Help

- Provides descriptions of directives and explains diagnostic messages

Note

- Allows comments to be included in transmissions

These directives can be issued at any time without affecting the operation in progress. Each is described in the following paragraphs.

DIAGNOSTIC

Query Update analyzes the syntax of a directive after it is transmitted. When errors appear in the input string, Query Update prints a three-digit number followed by a diagnostic message. Full or partial diagnostics can be displayed. When diagnostics are full, Query Update displays all diagnostic messages. When diagnostics are partial (default), Query Update does not display consecutive duplicate messages but indicates instead the number of times the diagnostic occurred.

Use of the DIAGNOSTIC directive with both the FULL and PART options is shown in figure 8-1. In both cases, the INVOKE directive includes an invalid character. When the FULL option is in effect, message number 007 is repeated. When the PART option is in effect, consecutive appearances of message number 007 are not repeated.

```
/ QU
QUERY UPDATE 3.4 SYSEDIT-82110 81/06/24 13.40.25
--
? DIAGNOSTIC FULL
--
? INVOKE> QSU>
(007) INVALID CHARACTER IN A NAME OR KEYWORD INVOKE>
(007) INVALID CHARACTER IN A NAME OR KEYWORD QSU>
(077) INVOKE> INVALID QU DIRECTIVE
--
? DIAGNOSTIC PART
--
? INVOKE> QSU>
(007) INVALID CHARACTER IN A NAME OR KEYWORD INVOKE>
THERE WERE 001 MORE 007 DIAGNOSTIC(S).
(077) INVOKE> INVALID QU DIRECTIVE
--
```

Figure 8-1. The DIAGNOSTIC Directive
EXHIBIT

Query Update deals with large amounts of information. Any number of subschemas can be stored in the system, and each subschema can reference up to 64 files. Any number of permanent file catalogs can also be stored in the system, and no limitation is imposed on the number of recorded sessions and report formats that can be retained. All information referenced by and contained within these files can be made available through the EXHIBIT directive.

Use of the EXHIBIT directive is shown in figure 8-2. Selected information concerning the subschema and the QUCAT catalog is displayed.

```
? INVOKE RELSUB (UN=USERNAME)

? 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):
    INVENTORY
    ORDERS
    LINEITEMS
    OUTLET
  SUBSCHEMA NAME = RELSUB
  SUBSCHEMA LIBRARY NAME = RELSUB
  UN = USERNAME

? EXHIBIT RELATION
  CONTEST RELATES THE RECORDS:
    ORDER-REC IN ORDERS
    OUTLET-REC IN OUTLET
  ACTIVITY-CHECK RELATES THE RECORDS:
    INV-REC IN INVENTORY
    ITEM-REC IN LINEITEMS
    OUTLET-REC IN OUTLET

? EXHIBIT INVENTORY
  RECORD NAME IS INV-REC
  KEY IS INV-NO
  ALT KEY BACK-ORDER
  AREA PF NAME = INVENTORY
  UN = USERNAME
  INDEX PF NAME = INVIDX
  M = W

? EXHIBIT DESCRIPTION
  DESCRIPTION OF INV-REC
  TYPE CHAR ITEM PIC SIZE 0017

? VERSION QUCAT (UN=USERNAME)

? EXHIBIT REPORTS REPORT7 TITLE
  TITLE AT LINE 1 IS $TEMPORARY ITEM
  REPORTS IN COLUMN 6 AT LINE 2 IS $ AT
  LINE 3 IS $ITEMS IN TAB 1 IS $PROFIT MARGINS
  IN TAB 2 AT LINE 4 IS $ $ $ 

? EXHIBIT SESSIONS SESS8 001 TO 003
  1  INVOKE QSUB (UN=USERNAME)
  2  EXTRACT UPON FILE8 DESCRIPTION INV-NO IN-STOCK
  3  UNIVERSAL IS *
```

Figure 8-2. The EXHIBIT Directive
HELP

Query Update provides over 50 directives and issues over 500 diagnostic messages. Definitions of the directives and detailed explanations of the diagnostic messages can be made available through the HELP directive.

? HELP PREFACE

PREFACE CAUSES LINES OF TEXT OR ANOTHER REPORT TO PRECEDE THE CURRENT REPORT.

? HELP SELECT

(212) -SELECT- IS NOT VALID FOLLOWING HELP.
(Q29) ERRONEOUS DIRECTIVE AND REST OF TRANSMISSION IGNORED

? HELP 212

THE ONLY VALID HELP PARAMETERS ARE EITHER A DIRECTIVE NAME OR A DIAGNOSTIC NUMBER.

? HELP SELECT

SELECT STATES CRITERIA FOR SELECTION OF DETAIL SPECIFICATIONS.

? HELP

THE DIRECTIVES IMPLEMENTED IN THIS RELEASE ARE

ACCESS ALTER BREAK
COMPILE CREATE
DATE DEFINE DELETE
DESCRIBE DETAIL DIAG
DISPLAY DUPLICATE END
ERASE EVALUATE
EXECUTE EXHIBIT EXTRACT
FOOTING FORMAT HEADING
HELP IF INSERT
INVOKE MOVE NOTE
OS PAGE-NUMBER
PAGE-SIZE PERFORM PREFACE
PREPARE PREVIEW RECAP
PRINT RECORDING REMOVAL
RETURNREWIND SELECT
SEPARATOR SORT SPECIFY
STOP STORE SUMMARY
TABSTIME TITLE UNIVERSAL
UPDATE USE VERIFY
VERSION VETO VIA
WRITE

Ask for the definition of the PREFACE directive.

Ask for the definition of the SELECT directive and misspell its name.

Ask for a more detailed explanation of diagnostic message 212.

Ask for the definition of the SELECT directive.

Ask for a list of available directives.

Figure 8-3. The HELP Directive
NOTE

Notes and comments can be useful, particularly when recorded sessions and report formats must be maintained by someone other than the originator. Notes can be incorporated through the NOTE directive as complete transmissions or can be appended as the last portion of a transmission containing other directives. All characters between keyword NOTE and the end of transmission are accepted as comments.

Use of the NOTE directive is shown in figure 8-4. Two notes are recorded and exhibited.

```
--
? VERSION QUCAT (UN=username)
--
? RECORDING SESS13

001--
? INVOKE QUSUB (UN=username)

002--
? NOTE THIS REPORT EXTRACTS DATA FROM ONE FIELD

003--
? EXTRACT UPON FILE13 DESCRIPTION NOTE THIS IS +
? A 17-CHARACTER ALPHANUMERIC FIELD

004--
? PREPARE REPT13 FROM FILE13

005--
? RECORDING OFF
     END OF SESSION SESS13
--

? FORMAT RPT13
--

? DETAIL IS DESCRIPTION IN COLUMN 1
--

? EXHIBIT SESSION SESS13
  1  INVOKE QUSUB (UN=username)
  2  NOTE THIS REPORT EXTRACTS DATA FROM ONE FIELD
  3  EXTRACT UPON FILE13 DESCRIPTION NOTE THIS IS A 17-CHARACTER A
  LPHANUMERIC FIELD
  4  PREPARE REPT13 FROM FILE13
--
```

NOS/BE requires an ID parameter instead of a UN parameter.

Insert a note as a complete transmission.

Append a note to a transmission.

Notes are printed through the EXHIBIT directive.

Figure 8-4. The NOTE Directive
SUBSCRIPTING

Subscripting operations can be performed when a series of data items is referenced by one data name. The data items can be represented as repeating group or elementary data items in the subschema, or as an array entered through the DEFINE directive. The subscript can be an integer, the data name of an item containing an integer, or one of four figurative subscripts: ALL, LAST, NEXT, or ANY. Figurative subscript ANY is restricted to the IF, BREAK, SELECT, SPECIFY, and PERFORM UNTIL directives. Figurative subscript NEXT is restricted to the MOVE and STORE SETTING directives.

An application illustrating the subscripting capability is shown in figure 8-5. The subschema contains a repeating elementary item named CHILD. The CHILD matrix is constructed by including figurative subscript NEXT in the STORE SETTING directive that creates the sample data base. The display operations that follow illustrate subscripts specified as integers and as figurative constants ALL and LAST.

```
IDENTIFICATION DIVISION.
SUB-SCHEMA NAME IS PERS
DATA DIVISION.
AREA-NAME IS PERSONNEL M IS W
ORGANIZATION IS INDEXED NEW
KEY IS SSN
RECORD-NAME IS PERSRECORD
  02 SSN PIC X(9)
  02 EMPNAME PIC X(20)
  02 KIDSNO PIC 9
  02 CHILD PIC X(10) OCCURS 0 TO 9 TIMES DEPENDING ON KIDSNO

/ QU

QUERY UPDATE 3,4 SYSEDIT-82110 04/25/82 14.43.02

? CREATE PERSONNEL OF PERS (UN=username)

? STORE SETTING SSN EMPNAME CHILD(NEXT) CHILD(NEXT) CHILD(NEXT)
  $1112233335 $JOHN R. DOES $JEFFS $LARRY$ $CEDRICK$  
  $2223344445 $JACK C. JONES$  
  $3334455555 $GIL F. GULLES $ALICE$ SERICAS$  
  $4445566666 $SABE A. ABBOTT$ $JONASS$  
  *END  
  4 ACCESSES, 4 HITS, 4 IO-S

? DISPLAY KIDSNO CHILD(ALL)
  3 JEFF LARRY CEDRIC
  0  
  2 ALICE ERICA
  1 JONAS  
  4 ACCESSES, 4 HITS, 4 IO-S

? IF KIDSNO GE 1 DISPLAY SSN KIDSNO CHILD(LAST)
  1112233335 3 CEDRIC
  333445555 2 ERICA
  444556666 1 JONAS  
  4 ACCESSES, 3 HITS, 4 IO-S
```

Figure 8-5. Subscripting (Sheet 1 of 2)
? DISPLAY CHILD(1) CHILD(2) CHILD(3)
JEFF       LARRY    CEDRIC
(941) SUBSCRIPT OUT OF BOUNDS
THERE WERE 002 MORE 941 DIAGNOSTIC(S).
4 ACCESSES, 4 HITS, 4 IO-S

--

? DISPLAY PERSRECORD NOTE - THIS IS TO SEE THE COMPLETE RECORD
(209) REQUESTED DATA MAY NOT BE IN DISPLAY FORMAT
111223333JOHN R. DOE    3JEFF    LARRY    CEDRIC
222334444JACK C. JONES   0
333445555GIL F. GULLES   2ALICE   ERICA
444556666ABE A. ABBOTT   1JONAS
4 ACCESSES, 4 HITS, 4 IO-S

--

? END

Figure 8-5. Subscripting (Sheet 2 of 2)
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.

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>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>: (colon)††</td>
<td>8.2</td>
<td>00</td>
<td>: (colon)††</td>
<td>8.2</td>
<td>072</td>
</tr>
<tr>
<td>01</td>
<td>A</td>
<td>12.1</td>
<td>61</td>
<td>A</td>
<td>12.1</td>
<td>101</td>
</tr>
<tr>
<td>02</td>
<td>B</td>
<td>12.2</td>
<td>62</td>
<td>B</td>
<td>12.2</td>
<td>102</td>
</tr>
<tr>
<td>03</td>
<td>C</td>
<td>12.3</td>
<td>63</td>
<td>C</td>
<td>12.3</td>
<td>103</td>
</tr>
<tr>
<td>04</td>
<td>D</td>
<td>12.4</td>
<td>64</td>
<td>D</td>
<td>12.4</td>
<td>104</td>
</tr>
<tr>
<td>05</td>
<td>E</td>
<td>12.5</td>
<td>65</td>
<td>E</td>
<td>12.5</td>
<td>105</td>
</tr>
<tr>
<td>06</td>
<td>F</td>
<td>12.6</td>
<td>66</td>
<td>F</td>
<td>12.6</td>
<td>106</td>
</tr>
<tr>
<td>07</td>
<td>G</td>
<td>12.7</td>
<td>67</td>
<td>G</td>
<td>12.7</td>
<td>107</td>
</tr>
<tr>
<td>10</td>
<td>H</td>
<td>12.8</td>
<td>70</td>
<td>H</td>
<td>12.8</td>
<td>110</td>
</tr>
<tr>
<td>11</td>
<td>I</td>
<td>12.9</td>
<td>71</td>
<td>I</td>
<td>12.9</td>
<td>111</td>
</tr>
<tr>
<td>12</td>
<td>J</td>
<td>11.1</td>
<td>41</td>
<td>J</td>
<td>11.1</td>
<td>112</td>
</tr>
<tr>
<td>13</td>
<td>K</td>
<td>11.2</td>
<td>42</td>
<td>K</td>
<td>11.2</td>
<td>113</td>
</tr>
<tr>
<td>14</td>
<td>L</td>
<td>11.3</td>
<td>43</td>
<td>L</td>
<td>11.3</td>
<td>114</td>
</tr>
<tr>
<td>15</td>
<td>M</td>
<td>11.4</td>
<td>44</td>
<td>M</td>
<td>11.4</td>
<td>115</td>
</tr>
<tr>
<td>16</td>
<td>N</td>
<td>11.5</td>
<td>45</td>
<td>N</td>
<td>11.5</td>
<td>116</td>
</tr>
<tr>
<td>17</td>
<td>O</td>
<td>11.6</td>
<td>46</td>
<td>O</td>
<td>11.6</td>
<td>117</td>
</tr>
<tr>
<td>18</td>
<td>P</td>
<td>11.7</td>
<td>47</td>
<td>P</td>
<td>11.7</td>
<td>118</td>
</tr>
<tr>
<td>19</td>
<td>Q</td>
<td>11.8</td>
<td>50</td>
<td>Q</td>
<td>11.8</td>
<td>119</td>
</tr>
<tr>
<td>20</td>
<td>R</td>
<td>11.9</td>
<td>51</td>
<td>R</td>
<td>11.9</td>
<td>120</td>
</tr>
<tr>
<td>21</td>
<td>S</td>
<td>11.0</td>
<td>52</td>
<td>S</td>
<td>11.0</td>
<td>121</td>
</tr>
<tr>
<td>22</td>
<td>T</td>
<td>11.1</td>
<td>53</td>
<td>T</td>
<td>11.1</td>
<td>122</td>
</tr>
<tr>
<td>23</td>
<td>U</td>
<td>11.2</td>
<td>54</td>
<td>U</td>
<td>11.2</td>
<td>123</td>
</tr>
<tr>
<td>24</td>
<td>V</td>
<td>11.3</td>
<td>55</td>
<td>V</td>
<td>11.3</td>
<td>124</td>
</tr>
<tr>
<td>25</td>
<td>W</td>
<td>11.4</td>
<td>56</td>
<td>W</td>
<td>11.4</td>
<td>125</td>
</tr>
<tr>
<td>26</td>
<td>X</td>
<td>11.5</td>
<td>57</td>
<td>X</td>
<td>11.5</td>
<td>126</td>
</tr>
<tr>
<td>27</td>
<td>Y</td>
<td>11.6</td>
<td>58</td>
<td>Y</td>
<td>11.6</td>
<td>127</td>
</tr>
<tr>
<td>28</td>
<td>Z</td>
<td>11.7</td>
<td>59</td>
<td>Z</td>
<td>11.7</td>
<td>128</td>
</tr>
<tr>
<td>29</td>
<td>0</td>
<td>12.0</td>
<td>60</td>
<td>0</td>
<td>12.0</td>
<td>129</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>12.1</td>
<td>61</td>
<td>1</td>
<td>12.1</td>
<td>130</td>
</tr>
<tr>
<td>31</td>
<td>2</td>
<td>12.2</td>
<td>62</td>
<td>2</td>
<td>12.2</td>
<td>131</td>
</tr>
<tr>
<td>32</td>
<td>3</td>
<td>12.3</td>
<td>63</td>
<td>3</td>
<td>12.3</td>
<td>132</td>
</tr>
<tr>
<td>33</td>
<td>4</td>
<td>12.4</td>
<td>64</td>
<td>4</td>
<td>12.4</td>
<td>133</td>
</tr>
<tr>
<td>34</td>
<td>5</td>
<td>12.5</td>
<td>65</td>
<td>5</td>
<td>12.5</td>
<td>134</td>
</tr>
<tr>
<td>35</td>
<td>6</td>
<td>12.6</td>
<td>66</td>
<td>6</td>
<td>12.6</td>
<td>135</td>
</tr>
<tr>
<td>36</td>
<td>7</td>
<td>12.7</td>
<td>67</td>
<td>7</td>
<td>12.7</td>
<td>136</td>
</tr>
<tr>
<td>37</td>
<td>8</td>
<td>12.8</td>
<td>68</td>
<td>8</td>
<td>12.8</td>
<td>137</td>
</tr>
<tr>
<td>38</td>
<td>9</td>
<td>12.9</td>
<td>69</td>
<td>9</td>
<td>12.9</td>
<td>138</td>
</tr>
<tr>
<td>39</td>
<td>+</td>
<td>12.10</td>
<td>60</td>
<td>+</td>
<td>12.10</td>
<td>139</td>
</tr>
<tr>
<td>40</td>
<td>+</td>
<td>12.11</td>
<td>61</td>
<td>+</td>
<td>12.11</td>
<td>140</td>
</tr>
<tr>
<td>41</td>
<td>=</td>
<td>11.1</td>
<td>73</td>
<td>=</td>
<td>11.1</td>
<td>141</td>
</tr>
<tr>
<td>42</td>
<td>/</td>
<td>11.2</td>
<td>74</td>
<td>/</td>
<td>11.2</td>
<td>142</td>
</tr>
<tr>
<td>43</td>
<td>(</td>
<td>11.3</td>
<td>75</td>
<td>(</td>
<td>11.3</td>
<td>143</td>
</tr>
<tr>
<td>44</td>
<td>)</td>
<td>11.4</td>
<td>76</td>
<td>)</td>
<td>11.4</td>
<td>144</td>
</tr>
<tr>
<td>45</td>
<td>,</td>
<td>11.5</td>
<td>77</td>
<td>,</td>
<td>11.5</td>
<td>145</td>
</tr>
<tr>
<td>46</td>
<td>;</td>
<td>11.6</td>
<td>78</td>
<td>;</td>
<td>11.6</td>
<td>146</td>
</tr>
<tr>
<td>47</td>
<td>'</td>
<td>11.7</td>
<td>79</td>
<td>'</td>
<td>11.7</td>
<td>147</td>
</tr>
<tr>
<td>48</td>
<td>&quot;</td>
<td>11.8</td>
<td>80</td>
<td>&quot;</td>
<td>11.8</td>
<td>148</td>
</tr>
<tr>
<td>49</td>
<td>#</td>
<td>11.9</td>
<td>81</td>
<td>#</td>
<td>11.9</td>
<td>149</td>
</tr>
<tr>
<td>50</td>
<td>.</td>
<td>12.0</td>
<td>82</td>
<td>.</td>
<td>12.0</td>
<td>150</td>
</tr>
<tr>
<td>51</td>
<td>(comma)</td>
<td>12.1</td>
<td>83</td>
<td>(comma)</td>
<td>12.1</td>
<td>151</td>
</tr>
<tr>
<td>52</td>
<td>(period)</td>
<td>12.2</td>
<td>84</td>
<td>(period)</td>
<td>12.2</td>
<td>152</td>
</tr>
<tr>
<td>53</td>
<td>(quote)</td>
<td>12.3</td>
<td>85</td>
<td>(quote)</td>
<td>12.3</td>
<td>153</td>
</tr>
<tr>
<td>54</td>
<td>(underline)</td>
<td>12.4</td>
<td>86</td>
<td>(underline)</td>
<td>12.4</td>
<td>154</td>
</tr>
<tr>
<td>55</td>
<td>blank</td>
<td>no punch</td>
<td>20</td>
<td>blank</td>
<td>no punch</td>
<td>040</td>
</tr>
<tr>
<td>56</td>
<td>,</td>
<td>12.5</td>
<td>87</td>
<td>,</td>
<td>12.5</td>
<td>155</td>
</tr>
<tr>
<td>57</td>
<td>-</td>
<td>12.6</td>
<td>88</td>
<td>-</td>
<td>12.6</td>
<td>156</td>
</tr>
<tr>
<td>58</td>
<td>+</td>
<td>12.7</td>
<td>89</td>
<td>+</td>
<td>12.7</td>
<td>157</td>
</tr>
<tr>
<td>59</td>
<td>*</td>
<td>12.8</td>
<td>90</td>
<td>*</td>
<td>12.8</td>
<td>158</td>
</tr>
<tr>
<td>60</td>
<td>/</td>
<td>12.9</td>
<td>91</td>
<td>/</td>
<td>12.9</td>
<td>159</td>
</tr>
<tr>
<td>61</td>
<td>)</td>
<td>13.0</td>
<td>92</td>
<td>)</td>
<td>13.0</td>
<td>160</td>
</tr>
<tr>
<td>62</td>
<td>(</td>
<td>13.1</td>
<td>93</td>
<td>(</td>
<td>13.1</td>
<td>161</td>
</tr>
<tr>
<td>63</td>
<td>)</td>
<td>13.2</td>
<td>94</td>
<td>)</td>
<td>13.2</td>
<td>162</td>
</tr>
<tr>
<td>64</td>
<td>'</td>
<td>13.3</td>
<td>95</td>
<td>'</td>
<td>13.3</td>
<td>163</td>
</tr>
<tr>
<td>65</td>
<td>&quot;</td>
<td>13.4</td>
<td>96</td>
<td>&quot;</td>
<td>13.4</td>
<td>164</td>
</tr>
<tr>
<td>66</td>
<td>,</td>
<td>13.5</td>
<td>97</td>
<td>,</td>
<td>13.5</td>
<td>165</td>
</tr>
<tr>
<td>67</td>
<td>;</td>
<td>13.6</td>
<td>98</td>
<td>;</td>
<td>13.6</td>
<td>166</td>
</tr>
<tr>
<td>68</td>
<td>'</td>
<td>13.7</td>
<td>99</td>
<td>'</td>
<td>13.7</td>
<td>167</td>
</tr>
<tr>
<td>69</td>
<td>&quot;</td>
<td>13.8</td>
<td>100</td>
<td>&quot;</td>
<td>13.8</td>
<td>168</td>
</tr>
</tbody>
</table>

† 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 zeros.
†† 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 related card codes do not exist and translations yield a blank (20H).
<table>
<thead>
<tr>
<th>Collating Sequence</th>
<th>CDC Graphic</th>
<th>Display Code</th>
<th>External BCD</th>
<th>Collating Sequence</th>
<th>CDC Graphic</th>
<th>Display Code</th>
<th>External BCD</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00</td>
<td>blank</td>
<td>55</td>
<td>20</td>
<td>32 40</td>
<td>H</td>
<td>10</td>
<td>70</td>
</tr>
<tr>
<td>01 01</td>
<td>&lt;=</td>
<td>74</td>
<td>15</td>
<td>33 41</td>
<td>I</td>
<td>11</td>
<td>71</td>
</tr>
<tr>
<td>02 02</td>
<td>%</td>
<td>63 †</td>
<td>16 †</td>
<td>34 42</td>
<td>V</td>
<td>66</td>
<td>52</td>
</tr>
<tr>
<td>03 03</td>
<td>(</td>
<td>61</td>
<td>17</td>
<td>35 43</td>
<td>J</td>
<td>12</td>
<td>41</td>
</tr>
<tr>
<td>04 04</td>
<td>–</td>
<td>65</td>
<td>35</td>
<td>36 44</td>
<td>K</td>
<td>13</td>
<td>42</td>
</tr>
<tr>
<td>05 05</td>
<td>≡</td>
<td>60</td>
<td>36</td>
<td>37 45</td>
<td>L</td>
<td>14</td>
<td>43</td>
</tr>
<tr>
<td>06 06</td>
<td>^</td>
<td>67</td>
<td>37</td>
<td>38 46</td>
<td>M</td>
<td>15</td>
<td>44</td>
</tr>
<tr>
<td>07 07</td>
<td>†</td>
<td>70</td>
<td>55</td>
<td>39 47</td>
<td>N</td>
<td>16</td>
<td>45</td>
</tr>
<tr>
<td>08 10</td>
<td>↓</td>
<td>71</td>
<td>56</td>
<td>40 50</td>
<td>O</td>
<td>17</td>
<td>46</td>
</tr>
<tr>
<td>09 11</td>
<td>&gt;</td>
<td>73</td>
<td>57</td>
<td>41 51</td>
<td>P</td>
<td>20</td>
<td>47</td>
</tr>
<tr>
<td>10 12</td>
<td>≥</td>
<td>75</td>
<td>75</td>
<td>42 52</td>
<td>Q</td>
<td>21</td>
<td>50</td>
</tr>
<tr>
<td>11 13</td>
<td>–</td>
<td>76</td>
<td>76</td>
<td>43 53</td>
<td>R</td>
<td>22</td>
<td>51</td>
</tr>
<tr>
<td>12 14</td>
<td>.</td>
<td>57</td>
<td>73</td>
<td>44 54</td>
<td>J</td>
<td>62</td>
<td>32</td>
</tr>
<tr>
<td>13 15</td>
<td>)</td>
<td>52</td>
<td>74</td>
<td>45 55</td>
<td>S</td>
<td>23</td>
<td>22</td>
</tr>
<tr>
<td>14 16</td>
<td>;</td>
<td>77</td>
<td>77</td>
<td>46 56</td>
<td>T</td>
<td>24</td>
<td>23</td>
</tr>
<tr>
<td>15 17</td>
<td>+</td>
<td>45</td>
<td>60</td>
<td>47 57</td>
<td>U</td>
<td>25</td>
<td>24</td>
</tr>
<tr>
<td>16 20</td>
<td>$</td>
<td>53</td>
<td>53</td>
<td>48 60</td>
<td>V</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>17 21</td>
<td>*</td>
<td>47</td>
<td>54</td>
<td>49 61</td>
<td>W</td>
<td>27</td>
<td>26</td>
</tr>
<tr>
<td>18 22</td>
<td>–</td>
<td>46</td>
<td>40</td>
<td>50 62</td>
<td>X</td>
<td>30</td>
<td>27</td>
</tr>
<tr>
<td>19 23</td>
<td>/</td>
<td>50</td>
<td>21</td>
<td>51 63</td>
<td>Y</td>
<td>31</td>
<td>30</td>
</tr>
<tr>
<td>20 24</td>
<td>,</td>
<td>56</td>
<td>33</td>
<td>52 64</td>
<td>Z</td>
<td>32</td>
<td>31</td>
</tr>
<tr>
<td>21 25</td>
<td>(</td>
<td>51</td>
<td>34</td>
<td>53 65</td>
<td>:</td>
<td>00 †</td>
<td>none †</td>
</tr>
<tr>
<td>22 26</td>
<td>=</td>
<td>54</td>
<td>13</td>
<td>54 66</td>
<td>0</td>
<td>33</td>
<td>12</td>
</tr>
<tr>
<td>23 27</td>
<td>≠</td>
<td>64</td>
<td>14</td>
<td>55 67</td>
<td>1</td>
<td>34</td>
<td>01</td>
</tr>
<tr>
<td>24 30</td>
<td>&lt;</td>
<td>72</td>
<td>72</td>
<td>56 70</td>
<td>2</td>
<td>35</td>
<td>02</td>
</tr>
<tr>
<td>25 31</td>
<td>A</td>
<td>01</td>
<td>61</td>
<td>57 71</td>
<td>3</td>
<td>36</td>
<td>03</td>
</tr>
<tr>
<td>26 32</td>
<td>B</td>
<td>02</td>
<td>62</td>
<td>58 72</td>
<td>4</td>
<td>37</td>
<td>04</td>
</tr>
<tr>
<td>27 33</td>
<td>C</td>
<td>03</td>
<td>63</td>
<td>59 73</td>
<td>5</td>
<td>40</td>
<td>05</td>
</tr>
<tr>
<td>28 34</td>
<td>D</td>
<td>04</td>
<td>64</td>
<td>60 74</td>
<td>6</td>
<td>41</td>
<td>06</td>
</tr>
<tr>
<td>29 35</td>
<td>E</td>
<td>05</td>
<td>65</td>
<td>61 75</td>
<td>7</td>
<td>42</td>
<td>07</td>
</tr>
<tr>
<td>30 36</td>
<td>F</td>
<td>06</td>
<td>66</td>
<td>62 76</td>
<td>8</td>
<td>43</td>
<td>10</td>
</tr>
<tr>
<td>31 37</td>
<td>G</td>
<td>07</td>
<td>67</td>
<td>63 77</td>
<td>9</td>
<td>44</td>
<td>11</td>
</tr>
</tbody>
</table>

†In installations using the 63-graphic set, the % graphic does not exist. The : graphic is display code 63, External BCD code 16.
<table>
<thead>
<tr>
<th>Collating Sequence Decimal/Octal</th>
<th>ASCII Graphic Subset</th>
<th>Display Code</th>
<th>ASCII Code</th>
<th>Collating Sequence Decimal/Octal</th>
<th>ASCII Graphic Subset</th>
<th>Display Code</th>
<th>ASCII Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>00 00</td>
<td>blank</td>
<td>55</td>
<td>20</td>
<td>32 40</td>
<td>@</td>
<td>74</td>
<td>40</td>
</tr>
<tr>
<td>01 01</td>
<td>!</td>
<td>66</td>
<td>21</td>
<td>33 41</td>
<td>A</td>
<td>01</td>
<td>41</td>
</tr>
<tr>
<td>02 02</td>
<td>&quot;</td>
<td>64</td>
<td>22</td>
<td>34 42</td>
<td>B</td>
<td>02</td>
<td>42</td>
</tr>
<tr>
<td>03 03</td>
<td>#</td>
<td>60</td>
<td>23</td>
<td>35 43</td>
<td>C</td>
<td>03</td>
<td>43</td>
</tr>
<tr>
<td>04 04</td>
<td>$</td>
<td>53</td>
<td>24</td>
<td>36 44</td>
<td>D</td>
<td>04</td>
<td>44</td>
</tr>
<tr>
<td>05 05</td>
<td>%</td>
<td>63†</td>
<td>25</td>
<td>37 45</td>
<td>E</td>
<td>05</td>
<td>45</td>
</tr>
<tr>
<td>06 06</td>
<td>&amp;</td>
<td>67</td>
<td>26</td>
<td>38 46</td>
<td>F</td>
<td>06</td>
<td>46</td>
</tr>
<tr>
<td>07 07</td>
<td>'</td>
<td>70</td>
<td>27</td>
<td>39 47</td>
<td>G</td>
<td>07</td>
<td>47</td>
</tr>
<tr>
<td>08 10</td>
<td>(</td>
<td>51</td>
<td>28</td>
<td>40 50</td>
<td>H</td>
<td>10</td>
<td>48</td>
</tr>
<tr>
<td>09 11</td>
<td>)</td>
<td>52</td>
<td>29</td>
<td>41 51</td>
<td>I</td>
<td>11</td>
<td>49</td>
</tr>
<tr>
<td>10 12</td>
<td>*</td>
<td>47</td>
<td>2A</td>
<td>42 52</td>
<td>J</td>
<td>12</td>
<td>4A</td>
</tr>
<tr>
<td>11 13</td>
<td>+</td>
<td>45</td>
<td>2B</td>
<td>43 53</td>
<td>K</td>
<td>13</td>
<td>4B</td>
</tr>
<tr>
<td>12 14</td>
<td>.</td>
<td>56</td>
<td>2C</td>
<td>44 54</td>
<td>L</td>
<td>14</td>
<td>4C</td>
</tr>
<tr>
<td>13 15</td>
<td>-</td>
<td>46</td>
<td>2D</td>
<td>45 55</td>
<td>M</td>
<td>15</td>
<td>4D</td>
</tr>
<tr>
<td>14 16</td>
<td>.</td>
<td>57</td>
<td>2E</td>
<td>46 56</td>
<td>N</td>
<td>16</td>
<td>4E</td>
</tr>
<tr>
<td>15 17</td>
<td>/</td>
<td>50</td>
<td>2F</td>
<td>47 57</td>
<td>O</td>
<td>17</td>
<td>4F</td>
</tr>
<tr>
<td>16 20</td>
<td>0</td>
<td>33</td>
<td>30</td>
<td>48 60</td>
<td>P</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td>17 21</td>
<td>1</td>
<td>34</td>
<td>31</td>
<td>49 61</td>
<td>Q</td>
<td>21</td>
<td>51</td>
</tr>
<tr>
<td>18 22</td>
<td>2</td>
<td>35</td>
<td>32</td>
<td>50 62</td>
<td>R</td>
<td>22</td>
<td>52</td>
</tr>
<tr>
<td>19 23</td>
<td>3</td>
<td>36</td>
<td>33</td>
<td>51 63</td>
<td>S</td>
<td>23</td>
<td>53</td>
</tr>
<tr>
<td>20 24</td>
<td>4</td>
<td>37</td>
<td>34</td>
<td>52 64</td>
<td>T</td>
<td>24</td>
<td>54</td>
</tr>
<tr>
<td>21 25</td>
<td>5</td>
<td>40</td>
<td>35</td>
<td>53 65</td>
<td>U</td>
<td>25</td>
<td>55</td>
</tr>
<tr>
<td>22 26</td>
<td>6</td>
<td>41</td>
<td>36</td>
<td>54 66</td>
<td>V</td>
<td>26</td>
<td>56</td>
</tr>
<tr>
<td>23 27</td>
<td>7</td>
<td>42</td>
<td>37</td>
<td>55 67</td>
<td>W</td>
<td>27</td>
<td>57</td>
</tr>
<tr>
<td>24 30</td>
<td>8</td>
<td>43</td>
<td>38</td>
<td>56 70</td>
<td>X</td>
<td>30</td>
<td>58</td>
</tr>
<tr>
<td>25 31</td>
<td>9</td>
<td>44</td>
<td>39</td>
<td>57 71</td>
<td>Y</td>
<td>31</td>
<td>59</td>
</tr>
<tr>
<td>26 32</td>
<td>:</td>
<td>00†</td>
<td>3A</td>
<td>58 72</td>
<td>Z</td>
<td>32</td>
<td>5A</td>
</tr>
<tr>
<td>27 33</td>
<td>;</td>
<td>77</td>
<td>3B</td>
<td>59 73</td>
<td>[</td>
<td>61</td>
<td>5B</td>
</tr>
<tr>
<td>28 34</td>
<td>&lt;</td>
<td>72</td>
<td>3C</td>
<td>60 74</td>
<td>\</td>
<td>75</td>
<td>5C</td>
</tr>
<tr>
<td>29 35</td>
<td>=</td>
<td>54</td>
<td>3D</td>
<td>61 75</td>
<td>)</td>
<td>62</td>
<td>5D</td>
</tr>
<tr>
<td>30 36</td>
<td>&gt;</td>
<td>73</td>
<td>3E</td>
<td>62 76</td>
<td>~</td>
<td>76</td>
<td>5E</td>
</tr>
<tr>
<td>31 37</td>
<td>?</td>
<td>71</td>
<td>3F</td>
<td>63 77</td>
<td>_</td>
<td>65</td>
<td>5F</td>
</tr>
</tbody>
</table>

†In installations using a 63-graphic set, the % graphic does not exist. The : graphic is display code 63.
GLOSSARY

Alphanumeric -
The description of a data item that can be any character A through Z, digit 0 through 9, or special character recognized by Query Update.

Area -
A uniquely named data base subdivision that contains data records; a file.

Beginning-of-Information -
The start of the first user record in a file.

Break -
The point during preparation of a report page when headings and/or footings are to be inserted.

Character Set -
Set of graphic and/or control characters.

Code Set -
System of symbols used to represent each character within a character set.

Condition -
One of a set of specified values for which a data item can be tested.

Current Catalog -
The catalog (default or permanent) that is available for recording Query Update transmissions.

CYBER Record Manager (CRM) -
A software package running under the NOS and NOS/RE operating systems that allows a variety of record types, blocking types, and file organizations to be created and accessed. All CYBER Record Manager file processing requests ultimately pass through the operating system input/output routines.

Data Administrator -
The person, or group of people, that lead the design, programming, implementation, and maintenance efforts associated with a DMS-170 controlled data base.

Data Base File -
A file whose organization and content is described by one or more subschemas.

Data Description Language (DDL) -
The language used to generate a subschema.

Default Catalog -
A local file (ZZZZZQ2) that is available for recording Query Update transmissions.

Delimiter -
One of a set of characters used to separate and organize data items; synonymous with separator.

Directive -
A Query Update statement that describes an operation to be performed. A directive consists of a reserved word in the Query Update language and a combination of recognized symbols, punctuation, and user-supplied elements.

Directory -
A file that contains area and record attributes. The permanent file directory (subschema) for a data base file is created by DDL; the temporary file directory for a data base file is created by an EXTRACT or DISPLAY UPON (directory not generated on NOS 1) directive. The temporary file directory for a non-data-base file is created by a DESCRIBE or DISPLAY UPON (directory not generated on NOS 1) directive.

File Organization -
The predetermined arrangement of stored data. CYBER Record Manager recognizes the following organizations: sequential, extended indexed sequential, extended actual key, extended direct access, initial indexed sequential, initial actual key, and initial direct access.

Footing -
Lines of print that comprise a caption and occur after a break.

Heading -
Lines of print that comprise a caption and occur after a break.

Index File -
A file that contains an entry for each unique value of alternate key and associates it with a list of primary keys for all records containing that value.

Key -
One or more data items, the contents of which identify a record or set of records.

Layout Directives -
Directives that supply arrangement and structure of a report on a printed page.

Logical Operator -
A word defining the logical connections between two terms. Query Update recognizes AND, OR, XOR, and NOT.

Non-Data-Base File -
A sequential file whose organization and content is not described by a subschema.

Numeric -
The description of a data item that can be any digit 0 through 9.

Permanent Catalog -
A permanent file that contains recorded Query Update transmissions.
Permanent File -
A disk file known to the system because the file name has been cataloged in a permanent file table.

Picture -
The description of the general characteristics and editing requirements of a data item.

Record -
CYBER Record Manager defines a record as a group of related characters. A record or a portion of a record is the smallest collection of information passed between CYBER Record Manager and a user program.

Register -
Query Update locations that retain current data for display purposes.

Relation -
The logical structure formed by the joining of records.

Relational Operator -
An abbreviation or correspondence symbol that is used to describe a relationship between two terms. Query Update recognizes EQ, NE, GT, LT, GE, and LE.

Report Information Table -
An internal table that is generated by Query Update and used to produce a report.

Reserved Word -
The first word of a Query Update directive.

Separator -
A character used by Query Update as a delimiter.

Session -
Series of transmissions sent by a user between the QU control statement and the END directive.

Session Id -
The six-character session name assigned by the user.

Subschema -
Plan or outline described with DDL statements regarding names and characteristics of data items, records, areas, and relationships that must be maintained among data base elements.

Subscripting -
Use of an integer or variable to identify a particular element in an array.

Temporary Data Item -
An item established through a DEFINE, DESCRIBE, or SPECIFY directive for temporary use with a data base or non-data-base file.

Transmission -
One or more directives submitted as a unit.

Transmission Id -
The three-digit system-supplied identifier assigned to one or more directives in a session catalog.
INDEX

ACCESS/HITS message 3-7
Adding a directive 6-10
Alphanumeric
  Definition 8-1
  Fields 3-2
ALTER directive 2-2, 6-4, 6-10
Altering reports 6-10
Ambiguities 7-7
Area 3-1, B-1
Arithmetic operations 7-3

Beginning-of-Information (BOI) B-1
Break B-1
BREAK directive 2-2, 6-16

CATALOG control statement 3-2, 6-1
Catalog of directives 6-3
Cataloging reports 6-4
Character sets A-1, B-1
COBOL 1-1
Code set B-1
Column headings 6-14
COMPILE directive 2-2, 6-23
Condition B-1
Conditional
  Delete 3-7
  Display 3-5
  Modification 3-7
  Query 3-4
Continuation characters 2-7
Continuing display 3-5
Control statements
  ATTACH 4-1
  CATALOG 3-2, 6-1
  COPY 6-1
  DDL 3-1
  DEFINE 3-1, 6-1
  FILE 4-1
  QU 2-8
  REPORT 2-4
  REQUEST 3-2
  ROUTE 6-1
COPY control statement 6-1
CREATE directive 2-2, 3-2
Current catalog B-1
Current date 6-10
CYBER Record Manager (CRM) 2-9, B-1

Data administrator 1-1, B-1
Data base file
  Definition 2-1, B-1
  File operations 3-1
  Modification 3-4
  Query 3-4
Data Description Language (DDL)
  Control statement 3-1
  Language 1-1, 2-1, B-1
DATE directive 2-2, 6-10
Default catalog
  Definition 6-3, B-1
  Storing 6-3
Default display 3-6

DEFINE control statement 3-1, 6-1
DEFINE directive 2-2, 3-1
delimiter B-1
DESCRIBE directive 2-2, 4-1
DETAIL directive 2-2, 6-2
DIAGNOSTIC directive 2-2, 8-1
Directives
  Alphabetical listing 2-2
  ALTER 2-2, 6-10
  BREAK 2-2, 6-16
  COMPILE 2-2, 6-23
  CREATE 2-2, 3-2, 3-3
  DATE 2-2, 6-10
  DEFINE 2-2, 6-14
  Definition 2-1, B-1
  DESCRIBE 2-2, 4-1
  DETAIL 2-2, 6-2
  DIAGNOSTIC 2-2, 8-1
  DISPLAY 2-3, 3-3
  DISPLAY FROM 4-2
  DISPLAY UPON 4-2
  DUPLICATE 2-3, 6-8
  END 2-3, 3-3
  ERASE 2-3, 6-11
  EVALUATE 2-3, 6-14
  EXECUTE 2-3
  EXHIBIT 2-3, 8-1
  EXTRACT 2-3, 6-2
  FOOTING 2-4, 7-4
  FORMAT 2-4, 6-2
  HEADING 2-4, 6-16
  HELP 2-4, 6-1, B-3
  IF 2-4, 3-5
  INVOKE 2-4, 3-4
  MODIFY 2-4, 3-4
  MODIFY FROM 5-1
  MOVE 2-4, 7-4
  NOTE 2-5, 8-1, 8-4
  OS 2-5, 4-1
  PAGE-NUMBER 2-5, 6-16
  PAGE-SIZE 2-5, 6-19
  PERFORM 2-5, 6-4, 6-6
  PREFACE 2-5, 6-21
  PREPARE 2-5, 6-2
  PREVIEW 2-5, 6-22
  RECAPP 2-5, 6-19
  RECORDING 2-5, 6-5
  REMOVE 2-5, 3-7
  REMOVE FROM 5-3
  RETURN 2-6, 7-1, 7-2
  REWIND 2-6, 5-1
  SELECT 2-6, 6-13
  SEPARATOR 2-6, 4-2
  SORT 2-6, 6-12
  SPECIFY 2-6, 6-15
  STORE 2-6, 3-3
  STORE FROM 5-3
  SUMMARY 2-6, 6-21
  TABS 2-6, 6-14
  TIME 2-6, 6-13
  TITILE 2-7, 6-2
  UNIVERSAL 2-7, 6-15
  VERIFY 2-7, 3-8
  VERSION 2-7, 6-6
  VETO 2-7, 3-8
  VIA 2-7, 7-8

60499000 C

Index-1 ●
Directory
Data base file 3-1
Definition 3-1, B-1
Non-data-base file 4-1
Display
Conditional 3-5
Continuing 3-5
Default 3-6
Full record 3-5
Terminating 3-5
$ literal 3-6
DISPLAY directive 2-3, 3-5
DISPLAY FROM directive 4-2
DISPLAY UNTIL directive 4-2
DUPLICATE directive 2-3, 6-8
Duplicating reports 6-7
EDIT command 6-1
END directive 2-3, 3-3
ERASE directive 2-3, 6-11
Erasing a directive 6-11
EVALUATE directive 2-3, 6-14
EXECUTE directive 2-3
EXHIBIT directive 2-3, 8-1
EXTRACT directive 2-3, 6-1

Fields
Alphanumeric 3-2
Numeric 3-2
Zero-suppressed 3-3
FILE control statement 4-1
File organization 3-1, B-1
Files
Data base 2-1
Non-data-base 2-1, 4-1
Preface 6-20
Related 7-6
Summary 6-20
Fothing B-1
FOOTING directive 2-4, 7-4
FORMAT directive 2-4, 6-1
Full record display 3-5

Grand totaling 7-3

Heading B-1
HEADING directive 2-4, 6-16
HELP directive 2-4, 8-1, 8-3

IF directive 2-4, 3-5
Index file B-1
Input/output processing 2-9
Interactive mode
Definition 2-7
Entering 2-8
Hardware communications 2-8
Login procedure 2-8
Logout procedure 2-9
Output equipment 2-9
Request for input 2-7, 3-4, 6-1
Software communications 2-8
Interactive query
Data base files 3-4
Multiple files 7-1
Non-data-base files 4-2
Related files 7-7
Invoke directive 2-4, 3-5

Key B-1
Language components 2-1
Layout directive definition B-1
Logical operators 3-6, B-1
Login procedure 2-8
Logout procedure 2-9

Modification
Data base files 3-4
Related files 7-7
MODIFY directive 2-4, 3-4
MODIFY FROM directive 5-1
MOVE directive 2-4, 7-4
Multiple file query 7-1
Multiple file subschema 7-1

Non-data-base file
Definition 2-1, 4-1, B-1
Directory 4-1
Interactive query 4-2
Positioning 4-2, 5-4
NOTE directive 2-5, 8-1, 8-4
Numeric
Definition B-1
Files 3-2

Operating system commands
EDIT 6-1
PAGE 6-1
ROUTE 6-1
Operators
Logical 3-6
Relational 3-5
OS directive 2-5, 4-1

PAGE command 6-1
Page eject 6-16
Page positioning 6-19
PAGE-WIDTH directive 2-5, 6-16
PAGE-SIZE directive 2-5, 6-18
PERFORM directive 2-5, 6-4, 6-6
Permanent catalog B-2
Permanent file
Definition B-2
Storage 3-2, 6-1
Picture B-2
Positioning non-data-base files 4-2, 5-1, 5-3
PRETTY directive 2-5, 6-21
PREPARE directive 2-5, 6-2
PREVIEW directive 2-5, 6-22
Previewing reports 6-22
Projected record 7-7

QU control statement 2-8
Qualification 7-7
Query update
Directives 2-2
Language components 2-1
Operations 1-1
Processing 2-9
Reports 6-1
Subschema 1-1, 3-1
Users 1-1
Utilities 8-1
RECAP directive 2-5, 6-19
Record B-2
RECORDING directive 2-5, 6-5
Register B-2
Related files
    Establishing 7-6
    Modifying 7-7
    Querying 7-7
Relation B-2
Relational operators 3-5, B-2
Relational subschema 7-6
REMOVE directive 2-5, 3-4
REMOVE FROM directive 5-3
REPORT control statement 2-9, 6-24
Report information table 6-23, B-2
Report utility 2-9, 6-23
Reports
    Adding a directive 6-10
    Altering 6-10
    Cataloging 6-4
    Column headings 6-15
    Duplicating 6-7
    Erasing a directive 6-11
    Generating 6-4
    Page positioning 6-19
    Previewing 6-22
Terminating report modification 6-10
REQUEST control statement 3-2
Request for input 2-7, 3-4, 6-1
Reserved word B-2
RETURN directive 2-6, 7-2
REWIND directive 2-6
ROUTE control statement 6-1

SELECT directive 2-6, 6-15
Separator B-2
SEPARATOR directive 2-6, 4-2
Session B-2
Session ID B-2
SORT directive 2-6, 6-12
SPECIFY directive 2-6, 6-15
STORE directive 2-6, 3-3

STORE FROM directive 5-3
Storing default catalogs 6-4
Subschema
    Data Division 3-1
    Definition 1-1, 3-1, B-2
    Identification Division 3-1
    Maximum number 8-2
    Multiple-file 7-1
    Relation Division 7-6
    Relational 7-6
    Subscripting B-5, B-2
    Subtotaling 7-4
SUMMARY directive 2-6, 6-21

TABS directive 2-6, 6-14
Temporary data item 2-1, 6-14, B-2
Terminating display 3-5
Terminating report modification 6-10
TIME directive 2-6, 6-13
TITLE directive 2-7, 6-2
Transmission
    Continuation characters 2-7
    Definition 2-7, B-2
    Maximum characters 2-7
    Transmission ID B-2

UNIVERSAL directive 2-7, 6-15
Using cataloged reports 6-4

VERIFY directive 2-7, 3-8
VERSION directive 2-7, 6-6
VETO directive 2-7, 3-8
VIA directive 2-7, 7-9

Zero suppression 3-3
ZZZZZZQ2 file 6-3

$ literal display 3-3
*END 3-3
COMMENT SHEET


PUBLICATION NO.: 60499000

REVISION: C

NAME:

COMPANY:

STREET ADDRESS:

CITY: STATE: ZIP CODE:

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 below (please include page number references).

___ Please reply  ___ No reply necessary

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND TAPE