QUERY/260
Programming Manual

HEWLETT
PACKARD

Herrenberger Straße 130, D-7030 Böblingen
Federal Republic of Germany

Manual Part number 45261-90021
E 1179

Printed in U.S.A.
November 1979

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revised date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

August 1978 ... **FIRST EDITION**
December 1978 ... **SECOND EDITION** revised pages: iii, 2-2, 4-2, 5-3, 5-4
June 1979...updated pages for O.S. rev 2.D: 1-1, 2-3, 2-4, 3-2
November 1979 ... **THIRD EDITION** revised pages: iii, 4-2, 5-3

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

Table of Contents

CHAPTER 1

Introduction

The QUERY/250 system software provides a means to access and edit an IMAGE/250 data base. This software is a basic program with subprograms which are located on an operating system disc labeled QUERY. The software is not configured into the system.

QUERY/250 is operator-oriented with many aids for composing commands. Some of the features of QUERY/250 require preparation by a programmer. Specifications for creating files, writing subprograms and program segments are discussed in this manual. Additions to the IMAGE/250 data base schema are also discussed.

It is assumed that the reader is knowledgeable about the BASIC, REPORT WRITER/250, FORMS/250 and IMAGE/250 system software.

CHAPTER 2

How to Create Files

The QUERY/250 commands DO, RUN and ADD...FROM require a file to be written prior to their execution. The process of creating these files is described here.

Creating a DO File

You create a DO file with the use of the EDITOR. (The EDITOR and all commands are fully described in the IMAGE/250 Programming Manual.

```
RUN "EDITOR [volume]"
```

The Editor prompts you with a slash (/) when it is ready for commands. The ADD command allows you to add new lines. If you have just begun, the first line is number 1. If there are lines already existing, the line is numbered one after the last line number. Two slashes entered causes the ADD command to terminate.

The MODIFY command allows you to change text in a particular line. If you need more than an 80 character line, use the SET LENGTH= command.

To save the file, enter:

```
KEEP "file name [volume]"
```

To terminate the Editor, enter:

```
EXIT
```

Each line in the DO file must be a QUERY command. There is no required first or last line. When control is transfered to the DO file, QUERY executes each command. When an End of File is reached, control transfers back to the operator.

Creating a Report Subprogram

The REPORT WRITER statements are described in the REPORT WRITER/250 Programming Manual.

The entry point of the subprogram must be:

```
SUB Report (#1,A$, INTEGER B(*),C)
```

#1 is the mass memory file number of the workfile.
A$ is the data base name.
B is a one-dimension integer array containing the set numbers for the workfile pointers.
C is the number of sets in B.
The subprogram cannot use COM statements.
The IMAGE statements necessary to access the data base are described in the IMAGE/250 Programming Manual.

The name of the file where the subprogram is stored is used in the RUN command.

Example Subprogram

```
1000 ! This is a report program to list customer names and addresses
1005 ! from the sales analysis data base. Any QUERY THREAD, FIND, or
1010 ! SORT (as long as the customer data set is found) may precede
1015 ! this program.
1020 !
1025 ! variables:
1030 !    Buf$ - a buffer for a customer data set record
1035 !    Db$  - the data base reference number and name
1040 !    I    - a loop index
1045 !    Rptr - an array of record pointers from the workfile
1050 !    Sets - an array of sets in the thread used with the FIND
1055 !           command that filled the workfile
1060 !    Slen - the length of the sets array
1065 !    Spos - the position of the customer data set in the thread
1070 !    Sts  - the status array for the DBGET call
1075 !    #1   - the mass memory file number for the workfile
1080 !
1085        SUB Report(#1,Db$,INTEGER Sets(*),Slen)
1090        INTEGER I,Spos,Sts(1:10)
1095        SHORT Rptr(1:10)
1100        DIM Buf$[250]
1105 !
1110 !      This is the report description section.
1115 Rpt1:     REPORT HEADER USING Fmt1
1120 Fmt1:     IMAGE 27X,"SALES ANALYSIS DATA BASE"
1125           PAGE LENGTH 66,2,3
1130           REPORT TRAILER WITH 2 LINES USING Fmt2;"**** TOTAL CUSTOMERS PRIN
TED",VAL$(WFLEN(1))
1135 Fmt2:     IMAGE 10X,K,2X,K
1140           PAGE HEADER WITH 5 LINES USING Fmt3
1145 Fmt3:     IMAGE 33X,"CUSTOMER LIST",4/
1150           PAGE TRAILER WITH 2 LINES USING Fmt4;VAL$(NUMPAGE)
1155 Fmt4:     IMAGE /,70X,"PAGE ",K
1160           END REPORT DESCRIPTION
1165 !
1170 !      If no entries in the workfile, error out.
1175        IF WFLEN(1)>0 THEN Cont1
1180           DISP "***  ERROR  ***  NO ENTRIES HAVE BEEN FOUND"
1185           GOTO End
```

```
1190 !
1195 !       Set the length of the record pointer array to the number
1200 !       of sets in the thread (and hence the number of pointers per
1205 !       record in the workfile).
1210 Cont1: REDIM Rptr(1:Slen)
1215 !
1220 !       Scan the thread for the customer data set (set number 6).
1225         Spos=0
1230         FOR I=1 TO Slen
1235            IF Sets(I)=6 THEN Spos=I
1240            NEXT I
1245 !
1250 !       If the customer set is not in the thread (and thus not in the
1255 !       workfile), error out.
1260         IF Spos<>0 THEN Cont2
1265            DISP "***  ERROR  ***  CUSTOMER SET NOT FOUND"
1270            GOTO End
1275 !
1280 !       Set an exit softkey and output to the printer.
1285 Cont2: ON KEY #8:"EXIT" GOTO End
1290 !
1291 !        Rewind the workfile
1292         READ #1,1
1293 !
1295 !
1300 !       This is the report execution section.
1305         BEGIN REPORT Rpt1
1310 !
1315 !          This loop prints customer data for every record in the workfile.
1320             FOR I=1 TO WFLEN(1)
1325 !
1330 !               Get a record from the workfile and customer set.
1335                 READ #1;Rptr(*)
1340                 DBGET (Db$,6,4,Sts(*),"@ ",Buf$,Rptr(Spos))
1345 !
1350 !               If a data base error, error out.
1355                 IF Sts(1)=0 THEN Cont3
1360                    DISP "***  ERROR  ***  DATA BASE ERROR",Sts(1)
1365                    GOTO End
1370 !
1375 !               Print the customer name.
1380 Cont3:          DETAIL LINE 1 WITH 6 LINES
1385                 DETAIL LINE 0 USING "5X,30A";Buf$[11,40]
1390 !
1395 !               Print the customer addresses.
1400                 IF TRIM$(Buf$[41,70])<>"" THEN DETAIL LINE 0 USING "5X,30A";Bu
f$[41,70]
1405                 IF TRIM$(Buf$[71,100])<>"" THEN DETAIL LINE 0 USING "5X,30A";B
uf$[71,100]
1410 !
1415 !               Print the city, state, and zip.
1420                 DETAIL LINE 0 USING "5X,33A";TRIM$(Buf$[101,116])&", "&TRIM$(B
uf$[117,122])&" "&Buf$[135,142]
1425 !
1430 !               Print the customer country and space.
1435                 DETAIL LINE 0 USING "5X,12A";Buf$[123,134]
1440                 DETAIL LINE 0 USING "X"
1445                 NEXT I
1450 !
1455 !            End the report.
1460               END REPORT
1465 !
1470 !       Done, stop the report and exit.
1475 End:    STOP REPORT
1480         SUBEND
```

Computer Museum

It is recommended that the printer be assigned with the OUTPUT command.

Creating a Form

The FORMS/250 Programming Manual describes how to create a form. When QUERY uses the form for adding items, it expects that the number of items in the item list is equal to the number of input fields (arrays require one field for each element) and that the form input order corresponds to the order of the data items in the input list.

The length of the input field must be equal to or less than the length of the data item if the data item is a string. If the data item is numeric, any length field is accepted.

Creating a Workfile

When QUERY creates its own workfile it requires a minimum of 122 sectors, and will use a large part of the largest hole available. QUERY purges this file before it terminates. The file name is QRwf1x, where x is the user number.

To create a workfile which QUERY will use instead of creating its own, use the CREATE or FCREATE command.

CREATE file spec, number of defined records[, record length]

You can create the file any size;, however, if QUERY runs out of room when using it an error will occur. A minimum of 122 sectors is needed for any sort operation to take place.

For further information of the size of the workfile refer to Appendix C of the SORT/250 Programming Manual.

CHAPTER 3

Syntax

The syntax of the QUERY commands are summarized here. A complete explanation of the commands and their parameters can be found in the QUERY/250 Operators Guide.

RUN "QUERY [volume]"
Begins QUERY operation.

ADD item list [FROM "form name"]
Adds entries to the data items or data set listed. A form can be used to input the values.

[BREAK ON item] [TOTAL item list]
Sets report breaks and their associated totals for the LIST or LINEAR LIST commands. TOTAL alone causes a grand total to be printed. A maximum of ten items can be totaled.

DATA BASE data base name [volume]
Causes future commands to operate on the specified data base.

DELETE item list
Deletes data item values or entries from the data set. The data items must have been found by the previous FIND command.

DO "file name [volume]"
Transfers control to a file containing QUERY commands. At the end of the file, control transfers back to the operator.

EXIT
Termintates QUERY.

FIND item list FOR search expression
Finds entries which satisfy the search expression and places the data items listed in the item list or the entire data entry (if the set name is in the item list) into the workfile. The search expression is a mathematical expression using data items as variables. The Data Variables chapter of the Basic Programming Manual describes the process of rounding which is done with numerical expressions.

INFO
Prints a modified schema listing of the data base on the current output device.

LINEAR LIST ["string"] [item list]
Lists data items from the workfile in a linear format (one item per line) on the current output device using all BREAKS and TOTALS specified.

LIST ["string"] [item list]
Lists data items from the workfile in columnar format (one data entry per line) on the current output device using all BREAKS and TOTALS specified.

OUTPUT TO device [width [; length]]
Changes the output device for future LIST, LINEAR LIST and INFO commands. The device may be a spool file.

PASSWORD password
Defines the data base password to be used for subsequent commands.

REPLACE item list
Replaces values for the data items specified which are in the workfile.

RUN "report name [volume]"
Causes a report subprogram to be run.

SORT BY item list
Sorts the entries in the workfile by the data items listed. Data items are sorted in ascending order unless a D follows the data item name.

THREAD set list
Defines the order in which data sets are accessed during a FIND command. The first data set is searched sequentially and all others are searched via the data chains and data paths.

WORKFILE "file name [volume]"
Specifies the workfile to be used for subsequent commands.

CHAPTER 4

Control Number

The QUERY control number is specified in the data base schema and is used by QUERY to determine whether an item may be modified and to determine the list format for the item.

The control numbers are:

Option	Value
1) Write Inhibit	
No Inhibit	0
Write Inhibit	1
2) Item Type	
Default	0
Date Type	2
Currency Type	4
Undefined Type	6
3) Numeric Spacing	
Default	0
Comma Every 3 Digits	8
4) Digits Right of Decimal Point	
Default	0
0 Digits	16
1 Digit	32
2 Digits	48
3 Digits	64
4 Digits	80
5 Digits	96
6 Digits	112

Computer Museum

To find the control number, one value from each option is selected and the values are summed.

For example, the data base item TEST_HOURS may be specified as:

2 Digits Right of Decimal Point		48
Comma Every 3 Digits		8
Default Type		0
QUERY Write Inhibit	+	1
Control Number =		57

Then in the schema this item is:

```
TEST_HOURS        L(57);
```

Write Inhibit specifies whether or not an operator can make changes to this data item using QUERY.

Item Type specifies a particular format.

- Date Type – The data item values are input and output as MM/DD/YY, (or DD/MM/YY in the UK and Australia), but are stored differently (refer to Chapter 5).
- Currency Type – The data item values are output with a preceding dollar sign, $XXXX. The UK currency is preceded with a pound sign, £.
- Undefined Type - QUERY ignores the spacing and digits specifications for this data item.

Numeric Spacing specifies whether or not commas are inserted when the data item values are output. If commas are specified, they are output after every third digit, counting from the decimal point to the left, X,XXX,XXX.

Digits Right of Decimal Point specifies the number of digits to the right of the decimal point to be output. If 5 digits are specified, the number 3.45 is output as 3.45000, and the number 3.451236 is output as 3.45124.

CHAPTER 5

Date Conversion

If a data item has a control number which specifies it to be a date, QUERY uses an algorithm to convert a date format (MM/DD/YY) into an integer for storage. The inverse algorithm is used when the value is output. These algorithms are used to save storage space in the data base and to allow sorting by date. The algorithms are listed below.

```
1000 ! This function converts a date string of the form mm/dd/yy, m/dd/yy,
1005 ! mm/d/yy, or m/d/yy to an integer value. The date must be from
1010 ! 1/1/72 to 12/31/99 inclusive. The integer value returned will be
1015 ! in the range 1 to 10227. If an invalid date string is input, the
1020 ! return integer value will be 0.
1025 !
1030 ! variables:
1035 !     D    - the day value
1040 !     Dt$  - the date string
1045 !     Dval - the numeric value of the date
1050 !     I    - a loop index
1055 !     M    - the month value
1060 !     Y    - the year value
1065 !
1070         DEF FNDate(Dt$)
1075         INTEGER D,Dval,I,M,Y
1080 !
1085 !       Check for proper date string length.
1090         IF (LEN(Dt$)<=8) AND (LEN(Dt$))=6) THEN Cont1
1095            Dval=0
1100            GOTO End
1105 !
1110 !       Insert leading zeros in month and day, if necessary,
1115 !       to get mm/dd/yy.
1120 Cont1:  IF LEN(Dt$)=6 THEN Dt$="0"&Dt$
1125         IF (LEN(Dt$)=7) AND (Dt$[2,2]="/") THEN Dt$="0"&Dt$
1130         IF LEN(Dt$)=7 THEN Dt$=Dt$[1,3]&"0"&Dt$[4]
1135 !
1140 !       Check for slashes between month, day, and year.
1145         IF (Dt$[3,3]="/") AND (Dt$[6,6]="/") THEN Cont2
1150            Dval=0
1155            GOTO End
1160 !
1165 !       Check for only digits in month, day, and year.
1170 Cont2:  FOR I=1 TO 8
1175            IF (I=3) OR (I=6) THEN Cont3
1180               IF POS("1234567890",Dt$[I,I])>0 THEN Cont3
1185                  Dval=0
1190                  GOTO End
1195 Cont3:     NEXT I
1200 !
1205 !       Convert month, day, and year to numeric.
1210         M=VAL(Dt$[1,2])
1215         D=VAL(Dt$[4,5])
1220         Y=VAL(Dt$[7,8])+1900
1225 !
1230 !       Compute the numeric value of the date.
1235         Dval=INT(365.25*(Y-1972)+.75)+INT(30.55*M-29.95)-2*(M)2)+D+((M)2) AN
D (Y/4=INT(Y/4)))
1240 !
1245 !       Convert the numeric value of the date back to a string to check
1250 !       for valid date.
1255 Cont4:  IF Dt$<>FNDate$(Dval) THEN Dval=0
1260 End:    RETURN Dval
1265         FNEND
1270 !
```

```
1275 !
1280 ! This function converts a date integer value in the range 1 to 10227
1285 ! to a date string of the form mm/dd/yy. If an invalid date integer
1290 ! value is input, the date string will be set to null.
1295 !
1300 ! variables:
1305 !    D    - the day value
1310 !    Dt$  - the date string
1315 !    Dval - the numeric value of the date
1320 !    M    - the month value
1325 !    N    - an intermediate value
1330 !    Y    - the year value
1335 !
1340       DEF FNDate$(INTEGER Dval)
1345       INTEGER D,M,N,Y
1350       DIM Dt$[8]
1355 !
1360 !     Check the numeric value for valid range.
1365       IF (Dval)0) AND (Dval(10228) THEN Cont1
1370          Dt$=""
1375          GOTO End
1380 !
1385 !     Convert numeric value of the date to month, day and year.
1390 Cont1: Y=INT((Dval-1)/365.25)+1972
1395       N=Dval-INT(365.25*(Y-1972)+.75)
1400       M=INT((N+31)/30)
1405       IF INT(30.55*M-29.95)-2*(M)2)+((M)2) AND (Y/4=INT(Y/4)))=N THEN M=M
-1
1410       D=N-INT(30.55*M-29.95)+2*(M)2)-((M)2) AND (Y/4=INT(Y/4)))
1415 !
1420 !     Assemble the date string.
1425       Dt$=VAL$(M)&"/"&VAL$(D)&"/"&VAL$(Y-1900)
1430 !
1435 !     Insert leading zeros in month and day, if necessary,
1440 !     to get mm/dd/yy.
1445       IF LEN(Dt$)=6 THEN Dt$="0"&Dt$
1450       IF (LEN(Dt$)=7) AND (Dt$[2,2]="/") THEN Dt$="0"&Dt$
1455       IF LEN(Dt$)=7 THEN Dt$=Dt$[1,3]&"0"&Dt$[4]
1460 End:  RETURN Dt$
1465       FNEND
```

Alternate Date Conversion

The United Kingdom and Australian date format (DD/MM/YY) is converted using the following algorithms.

```
1000 ! This function converts a date string of the form mm/dd/yy,
1010 ! m/dd/yy,d/m/y,dd/m/y,d/mm/y,dd/mm/y, mm/d/yy or m/d/yy to an
1020 ! integer value. The date must be from 1/1/72 to 31/12/60 (60=
1030 ! the year 2060) inclusive. The integer value returned will be
1040 ! in the range 1 to 32508. If an invalid date string is input,
1050 ! the return integer value will be 0.
1060 !
1070 !
1080 ! variables:
1090 !    D     - the day value
1100 !    Dt$   - the date string
1110 !    Dval  - the numeric value of the date
1120 !    I     - a loop index
1130 !    M     - the month value
1140 !    Y     - the year value
1150 !
1160          DEF FNDate(Dt$)
1170          INTEGER D,Dval,I,M,Y
1180 !
1190 !
1200 !        Check for proper date string length.
1210          IF (LEN(Dt$)<=8) AND (LEN(Dt$)>=5) THEN Cont1
1220             Dval=0
1230             GOTO End
1240 !
1250 !        Insert leading zeros in month and day, if necessary,
1260 !        to get dd/mm/yy.
1270 Cont1:   IF Dt$[2,2]="/" THEN Dt$="0"&Dt$
1280          IF Dt$[5,5]="/" THEN Dt$=Dt$[1,3]&"0"&Dt$[4]
1290          IF LEN(Dt$)<>8 THEN End
1300          IF LEN(Dt$)=7 THEN Dt$=Dt$[1,6]&"0"&Dt$[7]
1310 !
1320 !        Check for slashes between month, day and year.
1330          IF (Dt$[3,3]="/") AND (Dt$[6,6]="/") THEN Cont2
1340             Dval=0
1350             GOTO End
1360 !
1370 !        Check for only digits in month, day and year.
1380 Cont2:   FOR I=1 TO 8
1390             IF (I=3) OR (I=6) THEN Cont3
1400                IF POS("1234567890",Dt$[I,1])>0 THEN Cont3
1410                   Dval=0
1420                   GOTO End
1430 Cont3:   NEXT I
1440 !
1450 !        Convert month, day and year to numeric.
1460          M=VAL(Dt$[1,2])
1470          D=VAL(Dt$[4,5])
1480          Y=VAL(Dt$[7,8])+1900
1490          IF Y<1972 THEN Y=Y+100
1500 !
1510 !        Compute the numeric value of the date.
1520          Dval=INT(365.25*(Y-1972)+.75)+INT(30.55*M-29.95)-2*(M>2)+D+((M>2) A
ND (Y/4=INT(Y/4)))
1530 !
1540 !        Convert the numeric value of the date back to a string to check
1550 !        for valid date.
1560 Cont4:   IF Dt$<>FNDate$(Dval) THEN Dval=0
1570 End:     RETURN Dval
1580          FNEND
```

```
1590 !
1600 !
1610 ! This function converts a date integer value in the range 1 to 32508
1620 ! to a date string of the form dd/mm/yy.  If an invalid date integer
1630 ! value is input, the date string will be set to null.
1640 !
1650 ! variables:
1660 !    D    - the day value
1670 !    Dt$  - the date string
1680 !    Dval - the numeric value of the date
1690 !    M    - the month value
1700 !    N    - an intermediate value
1710 !    Y    - the year value
1720 !
1730         DEF FNDate$(INTEGER Dval)
1740         INTEGER D,M,N,Y
1750         DIM Dt$[8]
1760 !
1770 !       Check the numeric value for valid range.
1780         IF (Dval>0) AND (Dval<32509) THEN Cont1
1790            Dt$=""
1800            GOTO End
1810 !
1820 !       Convert numeric value of the date to day, month and year.
1830 Cont1:  Y=INT((Dval-1)/365.25)+1972
1840         N=Dval-INT(365.25*(Y-1972)+.75)
1850         M=INT((N+31)/30)
1860         IF INT(30.55*M-29.95)-2*(M>2)+((M>2) AND (Y/4=INT(Y/4)))>=N THEN M=
M-1
1870         D=N-INT(30.55*M-29.95)+2*(M>2)-((M>2) AND (Y/4=INT(Y/4)))
1880 !
1890 !       Assemble the date string.
1900         IF Y>1999 THEN Y=Y-100
1910         Dt$=VAL$(D)&"/"&VAL$(M)&"/"&VAL$(Y-1900)
1920 !
1930 !       Insert leading zeros in month and day, if necessary,
1940 !       to get dd/mm/yy
1950         IF Dt$[2,2]="/" THEN Dt$="0"&Dt$
1960         IF Dt$[5,5]="/" THEN Dt$=Dt$[1,3]&"0"&Dt$[4]
1970         IF LEN(Dt$)=7 THEN Dt$=Dt$[1,6]&"0"&Dt$[7]
1980 End:    RETURN Dt$
1990         FNEND
```