SB21-0771-1

Field Developed Program

Query Facility For System/3

Program Number: 5798-ANB

Program Description/
Operations Manual

This manual describes the capabilities of the system and the programs. Discussion of design assumptions and potential modification areas are included. Record and file layouts are described and primary processing procedures specified. This manual is both a system description and an installation and operations reference document.

This is a reprint of SB21-0771-0 incorporating changes released in the following technical newsletter:

SN20-6137

IBM

PROGRAMMING SERVICES

During a specified number of months immediately following initial availability of each licensed program designated as the Programming Services Period, and referenced in the Supplement to Amended License Agreement for IBM Program Products, the Customer may submit documentation to a designated IBM location when he encounters a problem which his diagnosis indicates is caused by a licensed program error. During this period only, IBM through the program sponsor(s) will, without additional charge, respond to an error in the current unaltered release of the licensed program by issuing known error correction information to the Customer reporting the problem and/or issuing corrected code or notice of availability of corrected code. However, IBM does not guarantee service results or represent or warrant that all errors will be corrected. Any on-site programming services or assistance will be provided at a charge.

Program error correction requests and/or comments concerning the contents of this publication should be addressed to:

IBM Corporation
8712 West Dodge Road
Omaha, Nebraska 68114
Attn: Mr. Don Larimer

WARRANTY

EACH LICENSED PROGRAM IS DISTRIBUTED ON AN 'AS IS' BASIS WITHOUT WARRANTY OF ANY KIND EITHER EXPRESS OR IMPLIED.

Requests for copies of IBM publications should be made to your IBM Representative or to the IBM Branch Office serving your locality.

TABLE OF CONTENTS

FIGURES

INTRODUCTION

The purpose of this manual is to provide the user familiarity with the capabilities of the System/3 Query Facility and define the means by which a functioning system may be created. Described in detail are the Facility as a system, the operational aspects of the Facility, and a definition of each of the command statements that may be used to write a query. The necessary functions to tailor the programs to generate the system are enumerated. For information on how to modify the Facility to execute on a System/3 Model 6, refer to the Systems Guide.

SYSTEM OVERVIEW

The System/3 Query Facility is a tool that allows the user, programmer, and non-programmer alike to selectively access disk data files for preparing reports or gathering statistics. By using the functions provided in four simple ~commands and the functions of the System/3 Disk Sort, the user can rapidly formulate queries to the system and receive responses in the form of printed output without expending time and effort for writing and testing programs.

System Description

The System/3 Query Facility is comprised of two programs written in RPG II. Input to the first program is the 'SELECT' and 'SORT' commands. This program translates these statements into control statements that are compatible with the System/3 disk sort.

Upon processing these control statements, the sort utility then produces an output file of the records the user has selected, placing them in the defined sequence. This file then becomes input to the second program of the facility which processes the data and produces a report according to the definitions established by the 'PRINT' and 'COUNT' commands. Figure 4 presents samples of the control statements; Figure 5 illustrates how a query might be written and the resulting report.

The commands are designed to give a great deal of function to the user and demand little concern for rigid formatting. They are written in a manner that conveys meaning to the writer of a query, making a request simple and direct, and provides results in a minimum of time.

Programming Systems

The programs were written and tested under RPG II (5702-RG1, Version 5, Modification Level 0) for the System/3 Model 10 (disk). Instructions are provided for adaptation to a System/3 Model 6. In addition to the RPG II Compiler, the Disk Sort Program (5702-SM1) is required.

System Configuration

The object programs require 9216 bytes of memory. With a normal size core-resident system control program (approximately 3K bytes) a 12K central processor is required. The programs will execute on an otherwise minimum System/3 Model 10 disk configuration. Sufficient disk space should be available for sort work areas and output areas.

PROGRAM DESCRIPTIONS

The System/3 Query Facility is comprised of two RPG II programs. These two programs, in conjunction with the Disk Sort Program, process Query Facility commands which result in a printed report containing the information selected and sequenced by the user's commands. A discussion of the individual programs follows. (For a discussion of commands and their formats refer to page 11).

Program ANB01, the Sort Specification Generator

This program processes the user's 'SELECT' and 'SORT' statements. The 'SELECT' statement is optional; the 'SORT' statement is required. The order in which the two statements are input to the program is not important. Figure 1 illustrates the execution of program ANB01.

The OCL and data required for execution is:

```
    //  LOAD ANB01, F1
    //  RUN
    SELECT . . . . . . . . . . . .       (optional)
    . . . . . . . . . .   (2nd card of 'SELECT' statement is optional)
    SORT. . . . . . . . . . . . . . .
    /*
```

The OCL and data are placed in MFCU1. Blank cards should be placed in MFCU2 for punching the output cards. No printed output is produced.

The punched output consists of System/3 sort specifications which are supplied as input to the sort phase. (See the section "Operations Instructions").

The following limitations are placed on the "SELECT" and "SORT" statements:

a. The 'SELECT' statement may span only two cards.

b. A maximum of 10 "SELECT" relationships may be specified in the "SELECT" statement.

c. A maximum of 10 data element names may be specified in the 'SORT' statement.

If requirements are such that these limits must be exceeded, this must be planned for at generation time. Instructions for modification are included on page 20.

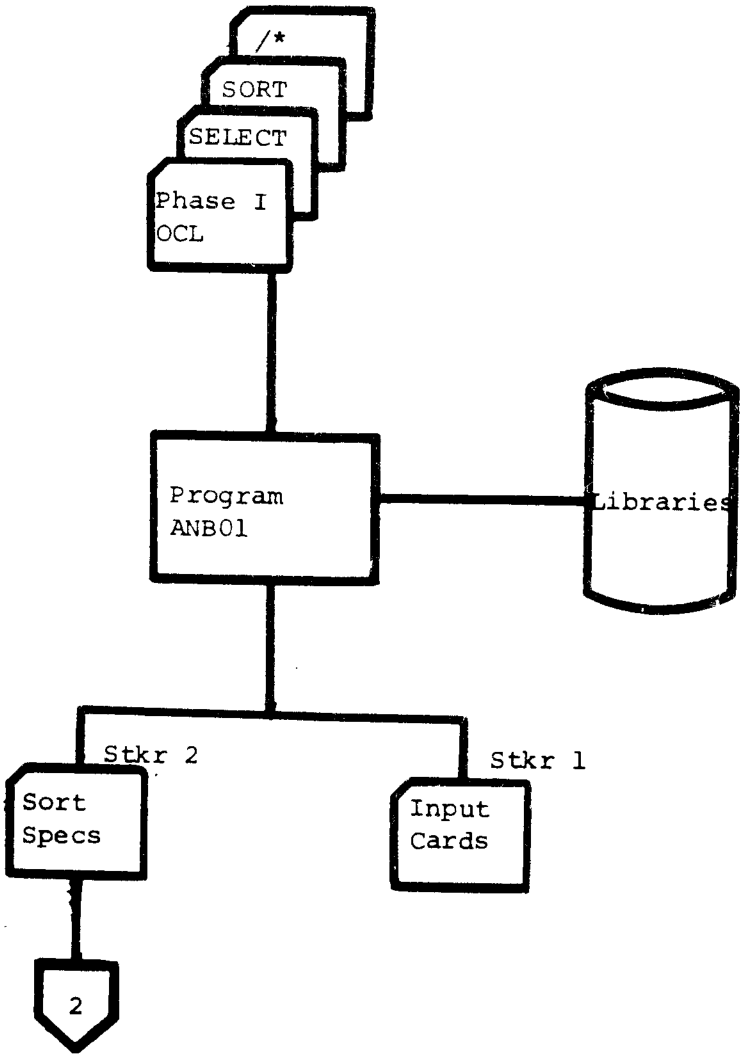


Figure 1 -- System/3 Query Facility Phase I

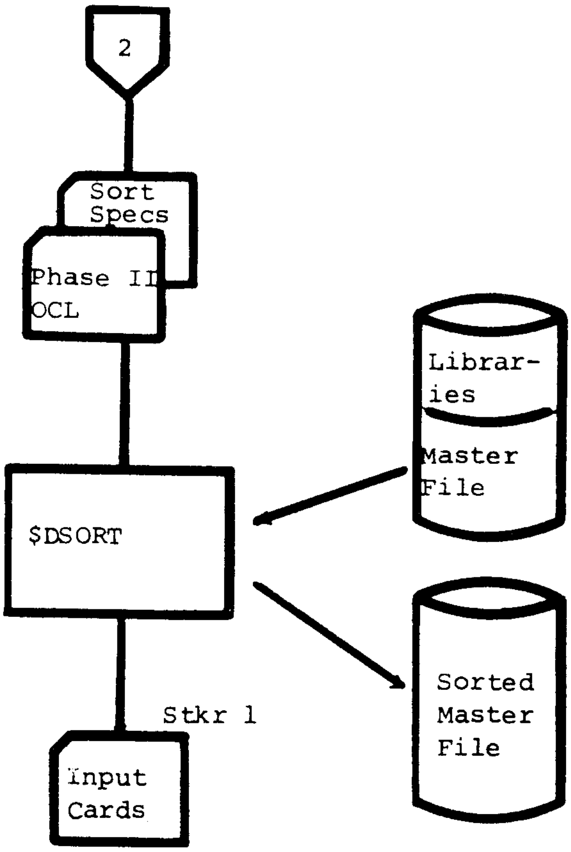


Figure 2 -- System/3 Query Facility Phase II

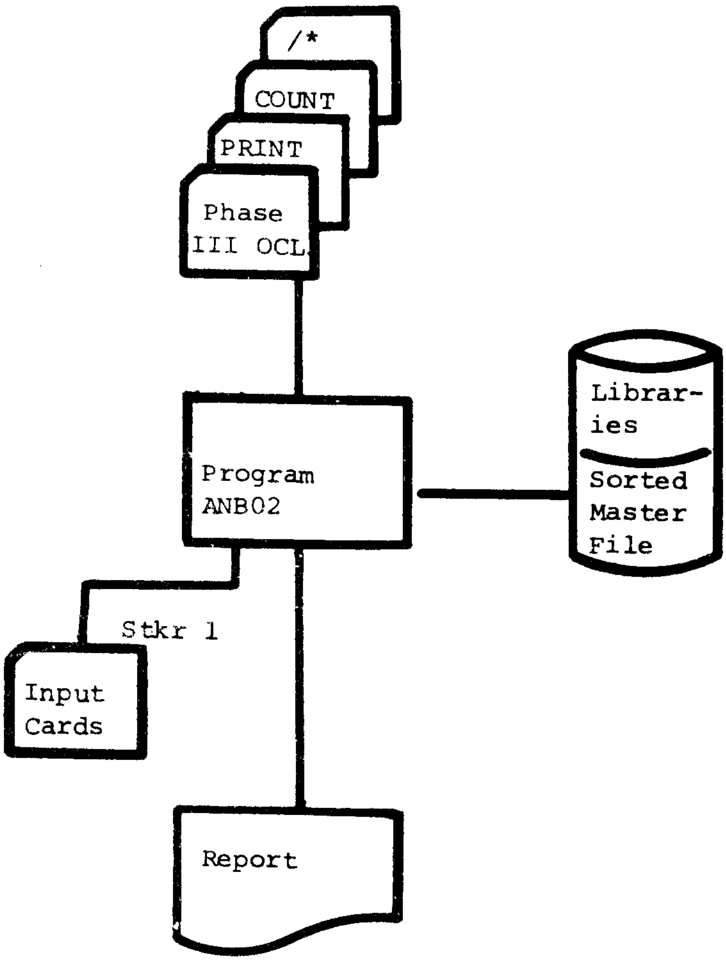


Figure 3 -- System/3 Query Facility Phase III

Intermediate Disk Sort Phase

Upon completion of program ANB01, the punched output is removed from Stacker 4 of the MFCU and placed directly behind the following OCL* in MFCU1:

```
//  LOAD  $DSORT, F1
//  FILE  NAME-INPUT, UNIT-uu, PACK-pppppp, LABEL-1....1
//  FILE  NAME-WORK, UNIT-uu, PACK-pppppp, TRACKS-nnn, RETAIN-S
//  FILE NAME-OUTPUT, UNIT-uu, PACK-pppppp, TRACKS-nnn, RETAIN-T, LABEL-S3QFOUT
//  RUN

    Cards produced by ANB01
```

Figure 2 illustrates the execution of the sort phase.

*Where uu, 1....1, pppppp, nnn, are the appropriate entries for unit, VTOC file label, volume label, and number of tracks respectively.

Program ANB02, the Report Program

This program processes the user's 'PRINT' and 'COUNT' statements. The 'PRINT' statement is required; the 'COUNT' statement is optional. The order in which they are input is not important. Figure 3 illustrates the flow for this program. The disk input file, which was produced by the sort phase, is interpreted for printing and tabulation via the 'PRINT' and'COUNT' statements.

The OCL and data required, placed in MFCU1, are as follows:

```
//  LOAD ANB02, F1
//  FILE NAME-INPUT, UNIT-uu*, PACK-pppppp*, LABEL-S3QFOUT
//  RUN
PRINT . . . . . . . . . . . . . . . . . .
COUNT . . . . . . . . . . . . . . . . . .    (optional)
/*
```

The report is produced on the system printer at this time. Figure 3 illustrates the execution of Phase III, program ANB02.

*uu - unit sorted master file resides on.
 pppppp - pack label for above unit.

Program ANB03, Sample File Creation Program

In order to run the sample program, a sample file must be created. The following OCL must be placed in MFCU1 along with the sample data in order to execute the sample file creation program:

```
//  LOAD ANB03, F1
//  FILE NAME-SAMPLE, UNIT-uu, PACK-ppppp, LABEL-SAMPLE, TRACKS-1, RETAIN-T
//  RUN

        Sample Data

/*
```

Where uu and pppppp are the appropriate unit and pack designations.

Modification Aids

There are two basic categories of modifications discussed elsewhere in the documentation. The first category involves tailoring the programs to the hardware configuration (System/3 Model 10) and specific master file of the user. This includes modifying for increased printer capacity, allowing for more information in control statements, and adjusting for the size of the Data Element Dictionary. These changes are presented in detail in the section on "Installation Instructions", pages 16 to 23 of this manual.

The second category which is a general discussion on how to modify these programs to execute on the System/3 Model 6 is contained in the Systems Guide.

```
SELECT   (SEX EQ F) AND (AGE LT 40) AND (CORNRY EQ +) AND (VENFIB EQ +)

SELECT   (ONHAND LT BAKORD) OR (BAKORD GT 0)

SELECT   (STATE EQ IA) AND (STATUS EQ D)

SORT     AGE, STATE, YEAR

SORT     ITEM, ONHAND

SORT     AMOUNT, DATE

PRINT    NAME, ADDR, CITY, AMOUNT, DUE

PRINT    ITEM, ONHAND, ONORD, BAKORD

PRINT    CUSTNO, CUSNAM, ADDRS, BALDUE

COUNT    AMOUNT

COUNT    BALUDUE, CURRNT, PASTDUE
```

Figure 4 -- Examples of Command Statements

The Problem: To determine all items that either have yielded a year-to-date margin of $1,000 or more, or, those which have sold more than 250 units. These items should be sorted by item class, and by the margin amount (descending) within that. The printed output should contain the item class, item description, item number, quantity sold, and the gross margin. The margin amount should be tabulated and printed at the end of the report. A count of records is automatic.

The Required Query Commands:

```
SELECT   (MARGIN GE 1000.00) OR (QUANTY GE 250)

SORT     ITMCLS, MARGIN-D

PRINT    ITMCLS, DESCR, ITEMNO, QUANTY, MARGIN

COUNT    MARGIN
```

The Resulting Report:

```
CLASS     DESCRIPTION      ITEM #      SOLD      GROSS MARGIN

A93       #13 GEAR         1468         12          2260.00
A93       #18 GEAR         1469         68          1108.00
A93       #24 GEAR         5201        252            12.50
B01       PINION            309          1          1000.00
B01       A-Y LEVER        2415        868           868.00


          RECORDS PRINTED = 5

          TOTAL MARGIN = 5248.50
```

Figure 5 -- A Sample Query and Resulting Report

SYSTEM/3 QUERY FACILITY COMMANDS

In order for the system to process a user's request, his problem must be expressed to the facility in the form of commands. This set of commands (a query) informs the system which specific records to select, the sequence in which to output the records, the data to print on the output report, and the numeric data fields for which totals are to be accumulated. A query can be comprised of four statement types. Their detailed descriptions and formats are found in the following pages.

'SORT' Control Statement

The 'SORT' statement is required for any query. This command defines how the records are to be sequenced by specifying the data element names to use for sequencing. The first data element name has the most significance in sorting; the last is least significant. All sequencing is assumed to be ascending; descending sequence for specific data elements can be specified (see below). The normal System/3 collating sequence is used, which, in general, treats blanks and special characters as lowest values, followed by alphabetic characters (A to Z), then by numerics (0 to 9). For more detail, refer to the Disk Sort Manual (SC21-7522).

The format is as follows:

SORT de_1, de_2, de_3, ... de_n

where de_n is a data element name. The characters 'SORT' must start in column 1 of the card and be followed by one or more blanks. Data element names must be separated by commas, and any number of blanks (0 up) may follow a comma. The program is written to allow a maximum of 10 data element names in the 'SORT' statement. As previously mentioned, a data element can be specified as a descending sort control field. This is accomplished by simply following the data element name in the 'SORT' command by the characters '-D'.

The following are examples of 'SORT' statements:

```
SORT    STATE, CITY, NAME
SORT    VENDOR, ITEM
SORT    SALES-D
SORT    AGE, IQ-D, GRADE
```

'SELECT' Control Statement

The 'SELECT' statement gives the user the ability to extract records from his data file using the power of Boolean logic. Records that meet the criteria specified in the 'SELECT' statement appear in the output report;

all others are ignored. This gives the user the ability to retrieve categories of data from his files (such as all the customers in New York or all back-ordered items). It is also possible for the user to explore his files seeking unique relationships and correlations. This technique may be used to discover trends or other patterns not readily provided by other programming techniques.

The 'SELECT' statement is comprised of a series of Boolean relationships that are linked together by the 'AND' or 'OR' logical operators. The Boolean relationships are simple comparisons between two data elements or between a data element and a constant. The relationship is enclosed in parentheses. To form the relationship, the two data element names (or data element and constant) are separated by the comparative operators EQ (equal), NE (not equal), LT (less than), GT (greater than), LE (less than or equal), GE (greater than or equal). The first name in the relationship *must* be a data element name; the second may be a constant or a data element name (if it is not found in the Data Element Dictionary by the program, it is assumed to be a constant). Alphabetic constants may be from 1 to 20 characters in length. Numeric constants can be from 1 to 15 digits; in addition, a minus sign may preceed the numeric constant and a decimal point may be inserted. Some examples of these relationships are:

(AGE GT 20)		d.e. - constant
(AGE GT 20.0)	equivalent to above	d.e. - constant
(NAME EQ JONES)		d.e. - constant
(NAME NE)		d.e. - constant (blanks)
(NAME GT AAAAA)		d.e. - constant
(TEMP LT -10)		d.e. - constant
(ONHAND GE ONORDR)		d.e. - d.e.

To construct a 'SELECT' statement, the Boolean relationships are linked together by the linkage operators 'AND' or 'OR' to define the total criteria by which to extract the records. The format is as follows:

SELECT (Rel$_1$) AND/OR (Rel$_2$) AND/OR . . . . (Rel$_n$)

The characters 'SELECT' must begin in column 1 of the first card (two cards are permitted) followed by one or more blanks. The 'AND' and 'OR' operators may be preceeded or followed by one or more blanks. If 2 cards are used for the statement, the data in the first card must end with a right parenthesis (')') and the second card must begin with the operator 'AND' or 'OR' in column '1'.

It is important to recognize how the program groups the logical relationships according to the placement of the 'AND' and 'OR' operators. For a record to be selected when two or more relationships are linked by 'AND', it must meet the criteria of all the relationships. For example:

SELECT (AGE GT 20) AND (SEX EQ F)

states that the selected records must be for individuals who are both female and over the age of 20. But if

SELECT (AGE GT 20) OR (SEX EQ F)

is specified, all individuals over the age of 20 will be selected along with all females; the 'OR' operator acts as a logical separator to allow for selection of records meeting any one of several relationships. The use of 'OR' as a separator becomes more complex when linking several groups of 'AND' relationships. For example:

SELECT (AGE EQ 10) AND (SEX EQ F) OR (AGE EQ 12) AND (SEX EQ F)

will extract all records for both 10 and 12 year old females. However, if the statement were written as:

SELECT (SEX EQ F) AND (AGE EQ 10) OR (AGE EQ 12)

would select all 10 year old females and all 12 year olds, regardless of sex; the 'OR' operator has been used to group the relationships. A way to view the affect of the 'OR' operator is to replace it with brackets; this is a convenient way to desk check the 'SELECT' portion of a query. In general, the procedure is:

SELECT [(REL$_1$) AND (Rel$_2$) AND (Rel$_3$) OR (Rel$_4$)] OR [(Rel$_5$) AND (Rel$_6$)]

can be handwritten as:

SELECT [(Rel$_1$) AND (Rel$_2$) and (Rel$_3$)] line 1

or [(Rel$_4$)] line 2

or [(Rel$_5$) AND (Rel$_6$)] line 3

If the record meets either the bracketed criteria of line 1, line 2, OR line 3, it will be selected.

In summary, the 'SELECT' statement can be invoked in a simple fashion, or can be used to pose a relatively complex query to the system.

A few examples of 'SELECT' statements follow:

```
    SELECT (ITEM GT 6100) AND (VENDOR EQ 5621) OR (ITEM LT 5000)
    AND (ITEM GT 4000) AND (VENDOR EQ 4802)

    SELECT (STATE EQ NY)

    SELECT (DATE EQ 060172) AND (INVCE EQ 232)

    SELECT (ONHAND EQ 0) OR (SALES LT 2500.00)
```

'PRINT' Control Statement

The 'PRINT' statement is required for any query. The operands are a series of data element names separated by commas. This statement causes the specified data elements to be edited and printed for each record encountered. It also extracts the specified heading for the data element from the Data Element Dictionary and prints it at the top of each page. In addition to data elements, the operand 'NO' may be specified to suppress the printing of any information for each record encountered. This may be used in conjunction with the 'COUNT' statement (and the automatic record count provided) to obtain total information without printing the detail data, as the totals will print even with the presence of the 'NO' operand.

A 2-digit number may also be inserted as an operand to cause additional lateral spacing between fields when they print on the report. The number corresponds to the desired number of blank spaces to be inserted.

The general format of the 'PRINT' statement is:

PRINT de_1, de_2, de_3, nn, de_4, ... de_n, NO

where de_n is a data element name; nn is a 2-digit space control number; and 'NO' is a suppress print request. 'PRINT' must begin in column 1, followed by 1 or more blanks. Blanks may be inserted following the commas if desired. A maximum of 15 data element names may be specified; the use of numerics or the 'NO' operand each counts as a data element name in relation to the maximum of 15. An error will occur if the number of print positions required to satisfy the print command exceeds the size of the system printer.

Examples are:

```
    PRINT    NAME, ADDRS, AGE, AMOUNT

    PRINT    ITEM, 40, SALES, 20, GROSS
```

```
PRINT    LOCATION

PRINT    AMOUNT, BAL, AGE, NO
```

'COUNT' Control Statement

The 'COUNT' control statement directs the Query Facility to accumulate totals of specified numeric data elements and print the results at the end of the report. Any data element names specified in the 'COUNT' statement must also appear in the 'PRINT' statement; if they do not, they are ignored. A record count is automatically provided.

The general format is:

COUNT $de_1, de_2, de_3, de_4, \ldots \; de_n$

Where de_n is a numeric data element name that has also been specified in the 'PRINT' statement. The characters 'COUNT' followed by at least one blank must begin in column 1.

Examples are:

```
COUNT    AMOUNT, BAL, MARGIN

COUNT    AGE

COUNT    QUANTY, COST
```

INSTALLATION INSTRUCTIONS

The following pages describe the step-by-step procedures that must be followed to generate a viable System/3 Query Facility to process your master file. If these programs will be used in conjunction with several files, the Facility must be generated separately for each one and the programs cataloged with unique names (column 75-80 of the RPG II header card) in your library.

The card deck as received from P.I.D. is comprised of three RPG II source decks and one deck of sample data. The order is as follows:

Program ANB01
Program ANB02
Program ANB03
Sample Data

It is recommended that the entire deck be reproduced for back-up purposes. The user should review the sections on installation and operations before attempting to generate the system. It would be quite beneficial to devote the small amount of effort necessary to generate the sample system and execute the sample program before proceeding with generating a "live" system. This is described in the section "Sample Problem Input and Output".

Steps for System Generation

The following functions must be performed in generating the System/3 Query Facility:

1. Obtain a current version of the record layout description for the master file.

2. Create a Data Element Dictionary according to the instructions in the following section 'The Data Element Dictionary'.

3. Replace the sample Data Element Dictionary in the two programs ANB01 and ANB02 with the one created in Step 2. (This is found at the end of each program deck between the '**' and '/*' cards.)

4. After obtaining a source listing of each program, perform the required modifications and review the potential modifications to the source programs as defined in the section "Tailoring the Source Programs".

5. Compile and catalog the two programs ANB01 and ANB02.

6. Create the OCL necessary for executing the facility. This is described in the section entitled "Program Description".

7. (Optional) Catalog the OCL in the Procedure Library of the systems pack if you desire to use the 'CALL' functions of System/3.

8. Exercise the facility enough times to have used every data element name in a 'PRINT' command and every numeric data element name in a 'COUNT' command, checking the output for accuracy in content, format, and heading information. This will help verify the accuracy of the Data Element Dictionary.

THE DATA ELEMENT DICTIONARY

Each program of the Query Facility contains a set of table records called the Data Element Dictionary. Each record describes one of the data elements (fields) found in the record of the master file that the programs are generated to communicate with. The programs store the Data Element records in the form of alternating tables. The first portion of the table record, the name of the data element, is one table; the latter portion, the descriptive parameters, comprise the alternate table.

The creation of the Data Element Dictionary is the most important function that must be performed to generate the system. The information contained in it is a permanent "map" of the input records and the data element names are the communication link between the user and the Query Facility. The parameters should be completely checked for accuracy. The names should be unique and carry meaning to any potential user.

Creating the Data Element Dictionary

The entries for the Data Element Dictionary are punched into cards and included in the source decks for the Query Facility; these cards should be inserted in place of the sample Data Element Dictionary found at the end of programs ANB01 and ANB02 between the '**' and '/*' cards. Most of the information can be taken directly from the record layout form for the master file when creating the record for each data element (or field). Two parameters that need additional consideration are the data element name and the data element heading. The name should be selected to convey meaning to the users of the facility. The headings should also be uniquely descriptive of each field, as the heading will print on each page of the report produced. The number of characters in the heading should be somewhat related to the print length of the field in order to achieve a report pleasing to the eye.

Each data element is described in a single card that adheres to the following format:

Card Columns — Contents

1-6 — Data Element Name -- From one to six characters. The first character must be alphabetic; the name may contain no special characters or imbedded blanks; each name must be unique and should convey meaning to the user.

7-9 — Data Element Starting Position -- Three-digit numbers defining the starting position of the data element in the master record.

10-12 — Data Element Ending Position -- Three-digit number defining the ending position of the data element in the master record.

Card Columns — Contents

13-13 — Data Element Format -- One character defining the format of the field in the master record: "C" for alphameric fields, "P" for packed-decimal fields, "U" for zoned-decimal (unpacked) fields, and (if the supplemental routine to interpret binary data has been inserted in the source code) "B" for binary formatted numeric fields.

14-14 — Decimals -- One digit number defining the number of digits that exist to the right of the implied decimal point. This entry is valid only for numeric fields and must otherwise be blank. Two additional options are provided for numbers with 0 decimals: First, a negative '1' (J) may be specified in column 14 to eliminate the insertion of commas and a decimal point into the edited field (this would be used for fields like item number, etc.). For numeric date fields, a negative 2 (K) may be entered in column 14 to provide slashes in the edited field (e.g. mm/dd/yy).

15-16 — Data Element Print Positions -- Two-digit number defining the number of print positions required for the data element and its heading in the reports generated by the Query Facility. This is normally the larger of either the number of characters in the heading or the number of characters in the field (as it is printed). The print length for numeric fields should be computed as follows:

Add to the number of digits in the field the following:

2 -- if number of digits are from 1-3
3 -- if number of digits are from 4-6
4 -- if number of digits are from 6-15

The entry in columns 15-16 can be greater or less than the "normal" entry. A greater number can waste print space and a smaller number could cause truncation of either heading or data.

17-31 — Data Element Heading -- Any valid System/3 characters. Normally this entry is left-adjusted but it may enhance the printed output in certain cases to pad with blanks on the left. If padding is done, this must be considered when making the Print Positions entry in columns 15-16.

After the Data Element Dictionary is created, it is important to make a list of the data elements and their descriptions available to all users of the Query Facility. This is important as it is the single link of communication between the user and his master file when using the Facility. For an example, see Figure 7.

TAILORING THE SOURCE PROGRAMS

The source programs ANB01 and ANB02 must be adapted to the particular situation for which they will be used. Part of system generation is making the appropriate changes to the RPG II code before compiling. The required modifications are those related to the record size of the master file and the number of data elements in the Data Element Dictionary; the creation of the Data Element Dictionary is also mandatory. Additional modifications can be made to allow for an increased number of print positions and for allowing more information to be contained in the control statements. These are described in detail in the following sections.

Statements in the source programs that require modification are flagged with a modification code in columns 90-95. The action that is required for each statement is described in detail on pages 21-23.

In addition, the user may have a need to process numeric data that is contained in his master file records in binary format. This is described in the section "Processing Binary Fields".

Modifications to Program ANB01 to Tailor the System

Modification Code — Description

104* Columns 37-39 contain the number of data elements defined in the Data Element Dictionary (table 'TABFLD' and its alternate 'TABPAR').

113* Column 51 = 2 if master file record length from 10 to 99 bytes; column 51 = 3 if length from 100-999 bytes.

116 Columns 37-39 contain the number of bytes allowed to contain the 'SELECT' statement. The entry is a multiple of 96 corresponding to the arbitrary maximum number of 96 byte records allowed for the 'SELECT' statement. The program allows the "SELECT" statement to span multiple records and currently is set to 2x96 = 192.

117 Columns 37-39 contain an entry that corresponds to the maximum number of Boolean relationships that can be specified in the "SELECT" statement. This is related to the entry made for 116 in the sense that the more records that can be spanned by the 'SELECT' statement, the more field-field or field-literal relationships can exist. This must be between 10 and 99.

118 Columns 33-34 correspond to the maximum number of 'SELECT' relationships allowed (agrees with 117).

119 Columns 37-39 contain the maximum number of data elements that can be specified in the 'SORT' statement. This must be between 10 and 99.

120 Columns 33-34 correspond to the number of data elements that can be specified in the 'SORT' statement (see 119).

121 Columns 33-34 correspond to the maximum size of the 'SELECT' statement (agrees with 116).

123* Columns 48-51 correspond to the record length of the master file (agrees with 207).

124* Columns 55-57 contain the record length of the master file.

*Note: Codes 104, 113, 123, and 124 are the only ones that need be considered to implement this program with the author's pre-defined limits. The limits are:

2 cards for the 'SELECT' statement
10 'SELECT' relationships
10 'SORT' data elements

Modifications to Program ANB02 to Tailor the System

Modification Code	Description
201	Columns 25-27 contain the number of print positions on the system printer.
202	Columns 37-39 contain the number of print positions on the system printer.
203	Columns 37-39 contain the record length of the master file (must agree with 207).
204*	Columns 37-39 contain the number of data elements defined in the Data Element Dictionary (table 'TABFLD' and its alternate 'TABPAR').
205	Columns 38-39 contain the maximum number of data elements that can be specified by the user in the 'PRINT' statement. This number must be greater than 10, but its upper limit is restricted by the maximum number of fields that can be fit into one line of print, considering print positions required to be the number specified in columns 16-17 of the Data Element Dictionary record.
206	Columns 37-39 contain the maximum number of data elements that can be specified by the user in the 'COUNT' statement. This number must be greater than 10 but its upper limit is bounded by the number of data elements defined (See 204). Also, consider the impact on core requirements if you choose an arbitrarily large number.
207*	Columns 25-27 contain the record length of the master file.
208*	Columns 48-51 contain the record length of the master file (must agree with 207).
209	Column 51, array index field length, must be 2 for 96 print positions or 3 for 120 or 132 print positions.
210	Columns 33-35 contain the number of print positions (agrees with 202).
211*	Columns 33-34 contain the number of data elements defined in the 'PRINT' statement (agrees with 205).

Modification Code	Description
212	Columns 33-34 contain the number of data elements that can be specified in the 'COUNT' statement (agrees with 206).
213*	Column 51, array index field length, must equal 2 for a master file record length of 10-99 or 3 for a record length of 100-999. Consider your entry for 207.
214	Columns 33-34 must be one greater than the number of data element names allowed in the 'COUNT' statement (if your entry for 206 was 12, then this must be 13).
215	Columns 41-43 contain the number of print positions on the system printer (must agree with 202).

*Note: If you wish to implement your system with the limits specified by the author, then only the modifications associated with codes 203, 204, 207, 208, and 213 must be made (these relate to the size of the master record). The limits are:

96 print positions
15 data elements allowed in the 'PRINT' statement
10 data elements allowed in the 'COUNT' statement

OPTIONAL MODIFICATIONS

The following paragraphs discuss implementation of some optional functions in the Query Facility. They have not been subjected to any formal testing and are provided only as a guide. The RPG II code in the accompanying illustrations may be helpful to those wishing to expand the function of the Facility. With one exception, the use of these options may either increase core requirements or cause excessive overlays on a minimum configuration.

Sub-Totals (L1 Totals)

In some cases it could be beneficial to obtain group sub-totals for the numeric fields specified in the 'COUNT' statement, as well as overall totals. For example, an inventory query might be written to provide sub-totals for each vendor as well as final totals for all vendors. The RPG II statements in figure 14 provide a means to accomplish this function.

Once the modification has been made to program ANB02, the user may request sub-totals in his query by the addition of a statement with 'BREAK' in columns 1-5. The statement contains no other information. The sub-totals are printed whenever a change in content of the first operand of the 'PRINT' statement is sensed. It is analogous to specifying an RPG II L1 break on the first 'PRINT' operand. If the 'BREAK' statement is not included, the sub-totals are not produced.

For example:

```
     PRINT      VENDOR, ITEM, COST, SELL
     COUNT      COST, SELL
     BREAK
```

would cause sub-totals of the COST and SELL fields to be printed for each vendor group and, as expected, final totals will appear at the end of the report. The sequence of the 'BREAK' statement is not important.

Calculations

This modification will allow the user to specify two of the numeric fields in a record to be used as operands in a calculation. The result of the calculation will be printed along with the other data fields included in the report. The operations included in this function are addition, subtraction, and multiplication.

An additional command is used to invoke this function. The format is:

Columns	1-7	'RESULT='	
Columns	8-13	Operand 1	
Columns	14	Operation Code:	'+', '-', or 'X'
Columns	15-20	Operand 2	

Examples are:

```
RESULT=QTY   XCOST
RESULT=GROSS -NET
RESULT=VALUEA+VALUEB
```

When the report is produced, the result field prints as the right most field in each printed line. The operand 'RESULT' is not allowed in the 'PRINT' statement of a query, but should be used as an operand in a 'COUNT' statement if totals of the result field are desired.

The RPG II code in figure 15 should be used as a guide for changing the print program ANB02. In addition an entry in the D.E.D. of that program must be made for the field 'RESULT' as follows:

```
RESULT000000U520RESULT FIELD
```

to properly define the field.

Program Size Reduction

In certain exceptional situations it may be desirable to reduce the size of the print program ANB02 or to eliminate excessive overlaying. One means to accomplish this is to modify the program so that the 'PRINT' and 'COUNT' statements are coded in fixed format instead of the free format described earlier in this manual. This simply means that the user must punch the operands in specific columns of the comman statements. The logic for scanning the free-form statements can then be removed from the program, reducing the size of the program.

The modifications for this are illustrated in figure 16. When these changes are made, the user must now code the operands of his 'PRINT' or 'COUNT' statements in the fixed positions of columns 7, 13, 19, 25, 28 .... etc. The operands are not separated by commas as in the past. For example, statement previously coded as:

```
PRINT   NAME, AGE, AMOUNT, PRCNT
```

would now be written as:

```
PRINT NAME  AGE   AMOUNTPRCNT
```

All other coding rules remain unchanged.

Catalogued Sort Specifications

With the announcement of Version 7 of the SCP for System/3 we can eliminate the punching of sort specifications in Phase I of processing a query. Instead, program ANB01 could be modified (by changing the device designation in the file description of the output file) to "punch" the sort specs on disk. A new job step, Phase 1A, could be a $MAINT job to catalog these statements in the source library. By adding a '// SOURCE-...' statement behind the OCL for Phase II, the sort will extract the specifications from the library instead of MFCU1.

Implementing this concept would increase the execution time of a query by a minute or so but would eliminate the inconvenience to the operator of having to move punched cards from stacker 4 to MFCU1 before execution of Phase II.

Processing Binary Formatted Numeric Data

The RPG II Statements in Figure 6 are a suggested means to allow the user to access binary fields in the master record. This coding converts the binary field to internal zoned-decimal format so that the existing routines in program ANB02 can print and tabulate the data. The length of the fied in the record is not restricted, except that its equivalent number of decimal digits does not exceed 15; however, if the field is created by an RPG II program, the binary field must be either 2 or 4 bytes long. The binary formatted data elements may not be specified in the 'SORT' and 'SELECT' statements; they are valid operands in the 'PRINT' and 'COUNT' statements.

OPERATIONS INSTRUCTIONS

User Prepared Input

There are three types of input information that must be supplied to the system upon execution. First, the user's master file. It is assumed that users will have existing application programs that create and maintain this file. There are no restrictions with regard to the file, except that the fields must be in standard System/3 formats, the record length from 10 to 999 bytes, and the organization either indexed or sequential.

Second, the OCL required for execution. This is described in the section "Program Description". Third, the Query Facility Control Statements. These are described in detail in the section "Control Statements".

Procedures

A query is processed in three phases. The first phase, using program ANB01, is the process of interpreting the 'SELECT' and 'SORT' statements and converting them into System/3 Disk Sort Specifications. Phase II is the execution of the Disk Sort Program which processes the master file, selecting and sorting data according to the specifications produced in Phase I. Phase III, through the interpretation of the 'PRINT' and 'COUNT' statements, produces the final output, the printed report.

These steps are required for processing a query:

a. Punch the control statements into 96-column cards.

b. Process the 'SELECT' (optional) and 'SORT' statements with program ANB01. The statements, preceded by the OCL, are placed in MFCU1 for execution as described in the "Program Description" section and Figure 1. Blanks are placed in MFCU2 for punching.

c. Clear the MFCU by pressing NPRO.

d. Remove the punched output from Stacker 4 and place behind the OCL for Phase II (sort phase) in MFCU1. Then execute Phase II. Refer to Figure 2.

e. Place the OCL for Phase III (report creation) in MFCU 1 followed by the 'PRINT' and 'COUNT' (optional) statements as described in the "Program Description" section. The printed report is then produced.

f. To conserve disk space, the file "S3QFOUT" should be deleted. (If a location parameter is defined in the OCL for the OUTPUT file of Phase II, and "RETAIN"-T is specified, the file can automatically be overlayed following an 'LL' halt during Phase II execution.) To delete the file, use these OCL and control statements:

```
// LOAD  $DELET,F1
// RUN
// SCRATCH UNIT-uu, PACK-pppppp, LABEL-S3QFOUT
// END
```

Errors

Error conditions encountered by the system are displayed in the form of halt indicators on the console. Normal response to these halts, after noting the cause, is to immediately cancel the job. The control statements should be corrected and resubmitted. Some rare errors, however, will result in abnormal termination of Phase II. These errors are incapable of being trapped by ANB01 but the sort finds the output of the program in Phase I unintelligible.

These halts may occur:

Halt	Program	Description
H1	ANB01	Too many select records.
H2	ANB01	Invalid data element name in 'SORT' statement.
H3	ANB01	Invalid data element name in 'SELECT' statement ('A' operand).
H4	ANB01	Multiple card 'SELECT' statement; first card is not the 'SELECT' card.
H5	ANB01	Invalid card type (does not begin with 'SELECT', 'AND', 'OR', or 'SORT').
H6	ANB01	Data Element Dictionary starting entry greater than end position entry for a data element.
H7	ANB01	No 'SORT' specifications encountered.
H8	ANB01	Numeric field defined in Data Element Dictionary with more than 15 digits.

Halt	Program	Description
H1	ANB02	Invalid data element name in 'PRINT' statement.
H2	ANB02	Invalid data element name in 'PRINT' statement.
H3	ANB02	Too many data elements specified in 'PRINT' statement.
H4	ANB02	Too many data elements specified in 'COUNT' statement.
H5	ANB02	Data element has an invalid format code in Data Element Dictionary.
H6	ANB02	Print line length exceeded.
H8	ANB02	Invalid command type.
14	ANB02	Numeric data element defined with more than 15 digits.

SYSTEM CAPACITIES AND LIMITATIONS

The following restrictions apply to the system in its unmodified form:

a. Object programs require 9K bytes of main storage.

b. Output printer requires 96 print positions.

c. Master file for which the facility is generated must be indexed or sequential in organization. The record length must be between 10 and 999 bytes long.

d. Acceptable data formats are alphanumeric; packed decimal, and zoned decimal.

e. Maximum of 2 cards to contain a 'SELECT' statement; all other statement types are restricted to one card.

f. Maximum of 10 comparative relationships in the 'SELECT' statement.

g. Maximum of 10 Data Element names can be specified in the 'SORT' and 'COUNT' statements.

h. Maximum of 15 Data Element names can be specified in the 'PRINT' statement.

Throughput

The user of the Query Facility will experience a wide variation of print speeds while executing the report program, ANB02. The differences are caused by the variation in data fields being printed from one job to another: A print request for 2 or 3 alphabetic fields will experience higher print speeds than a request that prints several fields that were originally packed data. Numeric fields, especially packed format, require more attention by the program because of editing and tabulating requirements.

SAMPLE PROBLEM INPUT AND OUTPUT

The sample problem is provided to demonstrate the functions of the System/3 Query Facility to the user. It is recommended that the sample file be created and the sample queries be executed prior to generating the system for use with live data files.

Generating the Sample Problem

In order to run the sample problems, a Query Facility must be generated for the sample file and the sample file must be created. The necessary modifications to the decks for programs ANB01 and ANB02 have already been made. The Data Element Dictionary has been created and inserted in the source decks. Figure 8 gives the record layout specifications for the sample file; it was from this that the Data Element Dictionary was created (Figure 9) and punched into 96 column cards. Figure 7 illustrates a sample of a user's guide for the sample file. It is this document by which the user bases his queries.

These are the steps necessary to generate the sample Facility and execute the sample queries:

1. Compile and catalog programs ANB01, ANB02, and ANB03.

2. Execute program ANB03 using the sample data found in the distribution deck. Reference the section on "Program Descriptions", page 8.

3. Execute program ANB01 using the OCL defined on page 3 and the following query statements:

```
SELECT  (VENDOR EQ 357) AND (ONHAND LT BAKORD)

SORT    ONHAND, ITEM

/*
```

4. Execute the sort phase using the OCL defined on page 7.

5. Execute program ANB02 using the OCL defined on page 7 and the following query statements:

```
PRINT  ITEM, DESCR, CLASS, ONHAND, ONORDR, BAKORD

COUNT  BAKORD

/*
```

6. Rerun Step 3 using the following query command:

```
SORT  SALES

/*
```

7. Rerun Step 4.

8. Rerun Step 5, using the following query commands:

```
PRINT  ITEM, DESCR, VNAME, COST, SELL, SALES, MARGIN, QTYSLD

COUNT  SALES, MARGIN

/*
```

9. Write and execute sample queries to gain familiarity with the facility.

MODIFICATIONS TO PROCESS BINARY FORMATTED FIELDS

INSERT THESE CARDS AFTER STATEMENT NUMBER 1700 IN PROGRAM ANB02.

```
CSR          FMT        COMP 'B'                         22  BINARY
CSR 22                  SETOF                      H5
CSR 22                  EXSR CVD
```

INSERT THIS SUBROUTINE AFTER STATEMENT NUMBER 3530 IN PROGRAM ANB02.

```
CSR          CVD        BEGSR
CSR                     Z-ADD0        NM
CSR                     TESTB'0'      AIN,B          55  NEGATIVE NO.
CSR                     BITOF'0'      AIN,B
C*CONVERT BINARY FIELD TO A NUMERIC FIELD, 'NM'
CSR          BITEST     TAG
CSR                     TESTB'0'      AIN,B          60
CSR                     TESTB'1'      AIN,B          61
CSR                     TESTB'2'      AIN,B          62
CSR                     TESTB'3'      AIN,B          63
CSR                     TESTB'4'      AIN,B          64
CSR                     TESTB'5'      AIN,B          65
CSR                     TESTB'6'      AIN,B          66
CSR                     TESTB'7'      AIN,B          67
CSR 67       1          ADD  NM       NM
CSR 66       2          ADD  NM       NM
CSR 65       4          ADD  NM       NM
CSR 64       8          ADD  NM       NM
CSR 63       16         ADD  NM       NM
CSR 62       32         ADD  NM       NM
CSR 61       64         ADD  NM       NM
CSR 60       128        ADD  NM       NM
CSR          1          ADD  B        B
CSR          B          COMP E                     50
CSRN50       NM         MULT 256      NM
CSRN50                  GOTO BITEST
CSR 55                  Z-SUBNM       NM
CSR                     ENDSR
```

Figure 6 -- Binary Routine

Data Element Name	Format	Heading	Print Positions	Description
ITEM	Numeric	'ITEM #'	6	Item Number
CLASS	Alpha	'CLASS'	5	Item Class
VENDOR	Numeric	'VENDR #'	8	Vendor Number
VNAME	Alpha	'VENDOR NAME'	11	Vendor Name
DESCR	Alpha	'DESCRIPTION'	11	Item Description
COST	Numeric	'COST'	8	Unit Cost of Item
SELL	Numeric	'SELL'	8	Unit Selling Price
SALES	Numeric	'YTD SALES'	9	Year-To-Date Sales ($)
QYTSLD	Numeric	'YTD QUANTITY'	12	Year-To-Date Sales - Units
MARGIN	Numeric	'YTD MARGIN'	11	Year-To-Date Gross Margin
ONHAND	Numeric	'ON-HAND'	11	Quantity On-Hand
ONORDR	Numeric	'ON ORDER'	11	Quantity On-Order
BAKORD	Numeric	'BACK ORDERED'	12	Quantity Back Ordered to Customers

Figure 7 -- Sample File, User's Data Element Dictionary Guide

Field Contents	Location	Format	# of Decimals
Item Number	1- 5	Zoned	0
Class of Item	6-10	Alpha	
Vendor Number	11-13	Zoned	0
Item Description	24-33	Alpha	
Vendor Name	14-23	Alpha	
Item Cost/Unit	34-36	Packed	3
Item Sell/Unit	37-39	Packed	2
Year-To-Date Gross Sales	40-43	Packed	2
Year-To-Date Quantity Sold	44-47	Packed	0
Year-To-Date Gross Margin	48-51	Packed	2
Quantity On-Hand	52-54	Packed	0
Quantity On-Order From Vendor	55-57	Packed	0
Quantity Back-Order to Customers	58-60	Packed	0

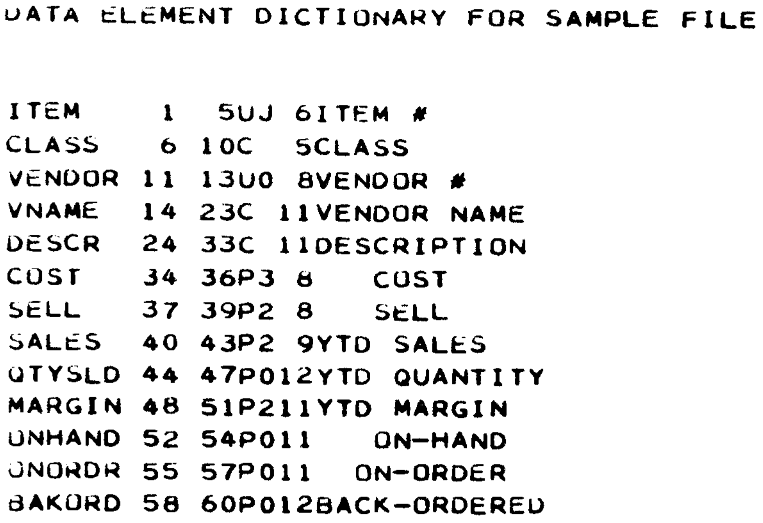
Record Length = 60 bytes

Figure 8 -- Sample File, Record Format Specifications

D.E. Name	Start Pos.	End Pos.	Format	Decimals	Print Pos.	Heading
1....6	7.9	10.12	13	14	15.16	17............31
ITEM	001	005	U	- 1	06	ITEM #
CLASS	006	010	C		05	CLASS
VENDOR	011	013	U	0	08	VENDOR #
DESCR	024	003	C		11	DESCRIPTION
COST	034	036	P	3	08	COST
SELL	037	039	P	2	08	SELL
SALES	040	043	P	2	09	YTD SALES
QTYSLD	044	047	P	0	12	YTD QUANTITY
MARGIN	048	051	P	2	11	YTD MARGIN
ONHAND	052	054	P	0	11	ON-HAND
ONORDR	055	057	P	0	11	ON-ORDER
BAKORD	058	060	P	0	12	BACK-ORDERED

These 13 cards comprise the Data Element Dictionary for the Sample File.

Figure 9 -- Sample File, Data Element Dictionary



```
DATA ELEMENT DICTIONARY FOR SAMPLE FILE


ITEM     1  5DJ 6ITEM #
CLASS    6 10C  5CLASS
VENDOR  11 13D0 8VENDOR #
VNAME   14 23C 11VENDOR NAME
DESCR   24 33C 11DESCRIPTION
COST    34 36P3 8   COST
SELL    37 39P2 8   SELL
SALES   40 43P2 9YTD SALES
QTYSLD  44 47P012YTD QUANTITY
MARGIN  48 51P211YTD MARGIN
ONHAND  52 54P011   ON-HAND
ONORDR  55 57P011  ON-ORDER
BAKORD  58 60P012BACK-ORDERED
```

Figure 10 -- Sample File, Data Element Dictionary

DATA TO LOAD SAMPLE FILE, PROGRAM ANB03

```
10036AP-39357ROBERT'S  6IN WRENCH 2755   550331100   603164974  12 144   48JUN 119764325283319872
10037AP-39357ROBERT'S  7IN WRENCH 2900   680855440  1258490620 729   0    0AUG 133215557654288213
10038AP-39357ROBERT'S  8IN WRENCH 3080   725 66700    92 38364 305  12    0DEC  900012311640000097
10039AP-39357ROBERT'S  9IN WRENCH 3275   825376200   456 61266   0   0   37DEC  900003213647888851
15206AP-39357ROBERT'S  429 ASSBLY  290    42   462    11   203  36   0    0DEC  999336560231235 77
  529AP-39357ROBERT'S  434 ASSBLY  290    43371047  8629282612 5401000    0DEC  955555252525250 01
  968AR-39357ROBERT'S  407 ASSBLY  420    99 82071   829230019   0   0  103APR  198721231345545 58
72001AR-39357ROBERT'S  27#ROCKER  7560 1650118800    72 42861   0   0   12JAN 310000000000000000
72002AR-39357ROBERT'S  27#RKR/RED 7560 1650112200    65 40805   0  40    0NOV 153216321316464978
72003AR-39357ROBERT'S  28#ROCKER  9630 1785   357     K  180K  46   0    0NOV 150000000000000000
72008AR-39357ROBERT'S  28#RKR/RED 9630 1785  7140     4  1802  36   0    0JUL  29879876546543210
12911N004 357ROBERT'S  BOLT        260   72 11880   165  7212  96   0    0MAY 113210000003210000
12913N004 357ROBERT'S  BOLT        280   75     0     0     0   0   0    0MAY 18321222221033000
12914N004 357ROBERT'S  STUD RTNN 12080 3200160000    50 76211 146  12    0JUN 28321321321009 6797
12915N004 357ROBERT'S  STUD RTNRX12080 3200 32000    10 10500   8   0    1JUL 150000999900009999
12958N004 357ROBERT'S  THRST #SHR  690  179 50299   281 40300  13   0 1100JUL 159872132646056425
12962N004 357ROBERT'S  SLEEVE     1762  225 10350    46  7300  32   0    0JUL 189875552222000000
12975N004 357ROBERT'S  SLEEVE/ARM18500 3000   900     L  470L  12   0    4JUL 191111122222333330
12987N004 357ROBERT'S  SLEEVE/BRG28000 4600   460     J  260K   0   0    2JUL 191111122222333330
10612AP-39108A-Z INC.  LFT-HANDED62000 9900     0     0     0   0   0    0JUN 109879546513200132
10684AP-39108A-Z INC.  24 MONKEY  8500 1500  9000     6  3900  35   0    0JUN 109875551320013456
10692AP-39108A-Z INC.  5IN. LEFT  2000  500 36000    72 22600  12  60   24JUN 109879995552220000
10707AP-39108A-Z INC.  5IN. RIGHT 2500  500 54000   108 27000   0 144  144JUN 113210000321564654
85110STK  108A-Z INC.  AG-9 SPARK  410   102979200  96005856001024   0    0APR 013212226546546371
85208STK  108A-Z INC.  INLT GSKET  120    60 24120   402 19296 101   0    0APR 013692581472573 58
89621STK  108A-Z INC.  EX VALVE-823000 4500351000    78171600   0  48    0APR 013692581472573 58
89620STK  108A-Z INC.  EX VALVE-616000 3000105000    35 49000   0  64    0SEP 303692828258005025
82003STK  108A-Z INC.  CARBURETOR72000118501303 50   11 51150   4   0    0SEP 309731301031030001
48728STX  602R-U REDDY GLOVES-SZ8  500   99 59598   602 29498  78  96    0SEP 300000032100000321
48729STX  602R-U REDDY GLOVES-SZ9  500   99 64845   655 32095   2  96   64DEC  19879654643210003
48727STX  602R-U REDDY GLOVES-SZM  500   99 30393   307 15043  27   0    0JAN 9 321444432164120 0
46200STX  602R-U REDDY HARD HAT   4000  625 42500    68 15300  19   6    0FEB  632144443111122 22
    29CLNG  96NO PROPHETHAND SOAP  1250    0     0     0     0 103   0    0NOSALE      0       0
    30CLNG  96NO PROPHETFLOOR WAX  5500    0     0     0     0  27   0    0NOSALE      0       0
    31CLNG  96NO PROPHETDEGREASER 47500    0     0     0     0   2   0    0NOSALE      0       0
    35CLNG  96NO PROPHETSUDS-O     6080    0     0     0   021   2   0    0NOSALE      0       0
```

Figure 11 -- Sample Data, Program ANB03

```
ITEM #   DESCRIPTION   CLASS      ON-HAND      ON-ORDER   BACK-ORDERED

   968   407 ASSBLY    AR-39            0             0            103
 10039   9IN WRENCH    AP-39            0             0             87
 12987   SLEEVE/BRG    N004             0             0              2
 72001   27#ROCKER     AR-39            0             0             12
 10036   6IN WRENCH    AP-39           12           144             48
 12958   THRST WSHR    N004            13             0          1,100


   TOTAL OF   BAKORD                    1,352

      TOTAL RECORDS                     6
```

Figure 12 -- Sample Output, Steps 3, 4, & 5, Page 29

```
ITEM #  DESCRIPTION  VENDOR NAME     COST      SELL   YTD MARGIN  YTD SALES  YTD QUANTITY

12975   SLEEVE/ARM   ROBERT'S      18.500     30.00      47.03-      90.00-            3-
12987   SLEEVE/BRG   ROBERT'S      28.000     46.00      26.02-      46.00-            1-
72003   28#ROCKER    ROBERT'S       9.630     17.85      18.02-      35.70-            2-
12913   BOLT         ROBERT'S         280        75          0           0             0
10612   LFT-HANDED   A-Z INC.      62.000     99.00          0           0             0
   31   DEGREASER    NO PROPHET    47.500         0          0           0             0
   29   HAND SOAP    NO PROPHET     1.250         0          0           0             0
   30   FLOOR WAX    NO PROPHET     5.500         0          0           0             0
   35   SUDS-O       NO PROPHET     6.080         0          0           0             0
15206   429 ASSBLY   ROBERT'S         290        42       2.03        4.62            11
72008   28#RKR/RED   ROBERT'S       9.630     17.85      18.02       71.40             4
10684   24 MONKEY    A-Z INC.       8.500     15.00      39.00       90.00             6
12962   SLEEVE       ROBERT'S       1.762      2.25      73.00      103.50            46
12911   BOLT         ROBERT'S         260        72      72.12      118.80           165
85208   INLT GSKET   A-Z INC.         120        60     192.96      241.20           402
48727   GLOVES-SZM   R-U REDDY        500        99     150.43      303.93           307
12915   STUD RTNRX   ROBERT'S      12.080     32.00     105.00      320.00            10
10692   5IN. LEFT    A-Z INC.       2.000      5.00     226.00      360.00            72
46200   HARD HAT     R-U REDDY      4.000      6.25     153.00      425.00            68
12958   THRST WSHR   ROBERT'S         690      1.79     403.00      502.99           281
10707   5IN. RIGHT   A-Z INC.       2.500      5.00     270.00      540.00           108
48728   GLOVES-SZ8   R-U REDDY        500        99     294.98      595.98           602
48729   GLOVES-SZ9   R-U REDDY        500        99     320.95      648.45           655
10038   8IN WRENCH   ROBERT'S       3.080      7.25     383.64      667.00            92
  968   407 ASSBLY   ROBERT'S         420        99     389.63      820.71           829
89620   EX VALVE-6   A-Z INC.      16.000     30.00     490.00    1.050.00            35
72002   27#RKR/RED   ROBERT'S       7.560     16.50     408.06    1.122.00            68
72001   27#ROCKER    ROBERT'S       7.560     16.50     428.61    1.188.00            72
82003   CARBURETOR   A-Z INC.      72.000    118.50     511.50    1.303.50            11
12914   STUD RTNR    ROBERT'S      12.080     32.00     762.11    1.600.00            50
10036   6IN WRENCH   ROBERT'S       2.755      5.50   1.649.74    3.311.00           603
89621   EX VALVE-8   A-Z INC.      23.000     45.00   1.716.00    3.510.00            78
  529   434 ASSBLY   ROBERT'S         290        43   2.826.12    3.710.47         8.629
10039   9IN WRENCH   ROBERT'S       3.275      8.25     612.66    3.762.00           456
10037   7IN WRENCH   ROBERT'S       2.900      6.80   4.906.20    8.554.40         1.258
85110   AG-9 SPARK   A-Z INC.         410      1.02   5.856.00    9.792.00         9.600



  TOTAL OF    SALES             44.545.25

  TOTAL OF    MARGIN            23.169.69

     TOTAL RECORDS                  36
```

Figure 13 -- Sample Output, Steps 6, 7, & 8, Page 30

```
        ROUTINE FOR IMPLEMENTING OUTPUT OF SUB-TOTALS

    * THIS CODE USES INDICATORS 40 THRU 44


195E                     TOT2       10 15 0              SUB TOTAL ARRAY

241I        NS  41   1 CB   2 CR   3 CE

295C   41                SETON                     42
296C* INDICATOR 42 SETS THE PROGRAM IN A MODE TO TAKE SUB-TOTALS

401C                     Z-ADD1       F
402C   42                EXSR CHKLV1





403C                     SETON                     44    AVOIDS 1ST REC BRK
404C   40                EXSR TOTOUT                     OUTPUT SUBTOTALS
405C   40       TOT2     SUB  TOT2    TOT2               BLANK AFTER
406C                     SETOF                     40

641CLR 42                SETON                     40    LR L1
642CLR 40                EXSR TOTOUT                     OUTPUT LAST SBTOT
643CLR                   SETOF                     40

   C* THE FOLLOWING SUBROUTINE SHOULD BE PLACED AFTER ALL CALCS,PROGRAM ANB02
   CSR        CHKLV1    BEGSR
   CSR                  EXSR PARAMS
   CSR                  MOVE ' '     BLANKS 20            DEFINE BLANK FIELD
   CSR                  MOVE BLANKS  NEWL1  20           CURRENT CONTROL FIELD
   CSR        LVLOOP    TAG
   C* ISOLATE CONTENTS OF THIS RECORDS LEVEL FIELD
   CSR                  MOVELNEWL1   WORK19 19
   CSR                  MOVE WORK19  NEWL1
   CSR                  MOVELAIN,B   NEWL1               GET A BYTE
   CSR        1         ADD  B       B
   CSR        B         COMP E                  43       LAST BYTE
   CSRN43               GOTO LVLOOP                      NO, GET ANOTHER
   C* IS IT A LEVEL BREAK
   CSR 44     NEWL1     COMP OLDL1              4040     40 L1 BREAK
   CSR                  MOVE NEWL1   OLDL1  20
   CSR                  ENDSR


2315CSRN19 50  NM       ADD  TOT2,N  TOT2,N
3455CSR 40              Z-ADDTOT2,N  NM

36150      OR  1     40 45

36250                40              3 'SUB'
EPLACE STATEMENTS 3610 AND 3590 WITH THE FOLLOWING CARDS
36100      E  2      45N40
35900      E  23     LRN45N40
```

Figure 14--Modifications for Subtotals

```
* THESE CHANGES ALLOW THE USE OF A NEW CONTROL STATEMENT THAT DEFINES
* A CALCULATION TO BE PERFORMED ON TWO DATA ELEMENTS. THE RESULT
* IS PRINTED ON EACH LINE OF THE REPORT AS ANY OTHER FIELD.
* THE DATA ELEMENT NAME 'RESULT' MUST BE GENERATED IN THE D.E.D. AS FOLLOWS
*       STARTING IN COLUMN 1 RESULT000000U520RESULT FIELD
* THE FORMAT OF THE COMMAND IS
*                       COL. 1-7       'RESULT '
*                       COL. 8-13      NAME OF FIRST OPERAND
*                       COL. 14        OPERATION CODE--'+','-',OR 'X'
*                       COL. 15-20 NAME OF SECOND OPERAND
*
* EXAMPLES ARE         RESULT PERCNTXAMOUNT
*                      RESULT GROSS -NET
*                      RESULT VALUEA+VALUEB
*
* THE OPERANDS MUST ALSO BE AMONG THOSE SPECIFIED IN THE 'COUNT'
* STATEMENT.
* THE RESULT FIELD WILL REQUIRE 22 PRINT POSITIONS.
* THE OPERAND 'RESULT' SHOULD NEVER BE USED IN THE 'PRINT' STATEMENT.
* THE OPERAND 'RESULT' MAY BE USED IN THE 'COUNT' STATEMENT.
*
* THIS CODE USES INDICATORS 34 THRU 39.


 205ICONTROL NS   34    1 CR    2 CE    3 CS
 206I                                                 8   13 OP1NAM
 207I                                                14   14 OPCODE
 208I                                                15   20 OP2NAM

 298C    34                  SETON                          35    REMEMBER THIS RECD.

 465C    35 50               EXSR RSLTSB                          CALCULATE/PRINT RSLT


1455CSR 36 35    'RESULT'   LOKUPTABFLD    TABPAR          50

2301CSRN35                   GOTO NOCAL
2301CSRN19 50    FLD.F       COMP OP1NAM                   38
2302CSRN19 50    FLD.F       COMP OP2NAM                   37
2303CSRN19 50 38             Z-ADDNM        OP1     154      SAVE
2304CSRN19 50 37             Z-ADDNM        OP2     154      VALUES
2305CSRN19 50 38             Z-ADDD         OP1DEC   90      OF DATA AND
2306CSRN19 50 37             Z-ADDD         OP2DEC   90      NO. OF DECIMALS
2307CSR          NOCAL       TAG


2741CSR 36                   GOTO XYZ
```

...continued on next page

Figure 15--Modifications for Calculations

...continued from previous page

```
C* THESE SUBROUTINES SHOULD BE PLACED AT THE END OF THE EXISTING
C* SUBROUTINES IN THE CALCS OF PROGRAM ANB02.  THEY PERFORM THE CALCULATION
C* SPECIFIED IN THE 'RESULT ' CONTROL STATEMENT AND MOVES THE ANSWER TO
C* THE PRINT LINE.
CSR        RSLTSB    BEGSR
C* ADJUST DECIMALS SO RESULT IS O.K.
CSR                  Z-ADDOP1DEC   DECS    10
CSR                  EXSR PWR10
CSR        OP1       DIV  FCTR10   OP1
CSR                  Z-ADDOP2DEC   DECS
CSR                  EXSR PWR10
CSR        OP2       DIV  FCTR10   OP2
CSR        OPCODE    COMP '+'                  36 WHAT IS
CSR        OPCODE    COMP '-'                  37 THE OPERATION
CSR        OPCODE    COMP 'X'                  38
CSR 36     OP1       ADD  OP2      RESULT 155
CSR 37     OP1       SUB  OP2      RESULT
CSR 38     OP1       MULT OP2      RESULT
CSR        RESULT    MULT 100000   NM
CSR                  Z-ADD1        N
CSR        'RESULT'  LOKUPCNT,N                36
CSR 36     NM        ADD  TOT,N    TOT,N
C*R 36 42  NM        ADD  TOT2,N   TOT2,N
C* US THE ABOVE STATEMENT ONLY OF THE LEVEL BREAK MOD. IS USED...
CSR                  SETON                   3619
CSR                  EXSR PARAMS
CSR                  EXSR NUMRIC                 RSLT. FLD. TO PRNT LN
CSR                  SETOF                   3619
CSR                  SETON                   50
CSR                  ENDSR


CSR        PWR10     BEGSR
C* THIS ROUTINE RAISES '10' TO A POWER PLACED IN THE FIELD DECS AND
C* PLACES THE ANSWER IN THE FIELD FCTR10
CSR                  Z-ADD1        FCTR10 100
CSR        AGN10     TAG
CSR        DECS      SUB  1        DECS        3636
CSRN36     FCTR10    MULT 10       FCTR10
CSRN36               GOTO AGN10
CSR                  ENDSR

  * THIS ENTRY MUST BE MADE IN THE DATA ELEMENT DICTIONARY
RESULT000000U520RESULT FIELD
```

Figure 15--Modifications for Calculations

```
      * THE FOLLOWING PROGRAM CHANGES TO THE PRINT PROGRAM-ANB02-
      * CAN REDUCE CORE REQUIREMENTS AT THE COST OF ABANDONDING
      *  THE FREE-FORMAT CODING OF THE 'PRINT' AND 'COUNT' STATEMENTS.
      * THE OPERANDS MUST BE PUNCHED IN COLS. 7,13,19,25...ETC.
      *AND ARE NOT SEPARATED BY COMMAS.  ALL OTHER CODING RULES APPLY.
      * THIS IS A QUICK-AND-DIRTY WAY TO SAVE SOME CORE.
```

```
*******REPLACE STATEMENTS 210,220,230,240 WITH THE FOLLOWING
  210ICONTROL NS  01   1 CP   2 CR   3 CI
  220I                                      7  96 FLD
  230I        NS  02   1 CC   2 CO   3 CU
  240I                                      7  66 CNT
```

```
 REPLACE 410,2010
  410C                     Z-ADD1         F       20          INITIALIZE INDEX
 2010CSR         1         ADD  COUNT     COUNT   20
```

```
****** REMOVE STATEMENTS  140,380, AND 710 THRU 1280 INCLUSIVE
```

Figure 16--Modifications to Reduce Program size

SB21-0771-1

Query Facility For System/3
Program Description/Operations Manual

Printed in U.S.A. SB21-0771-1

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)