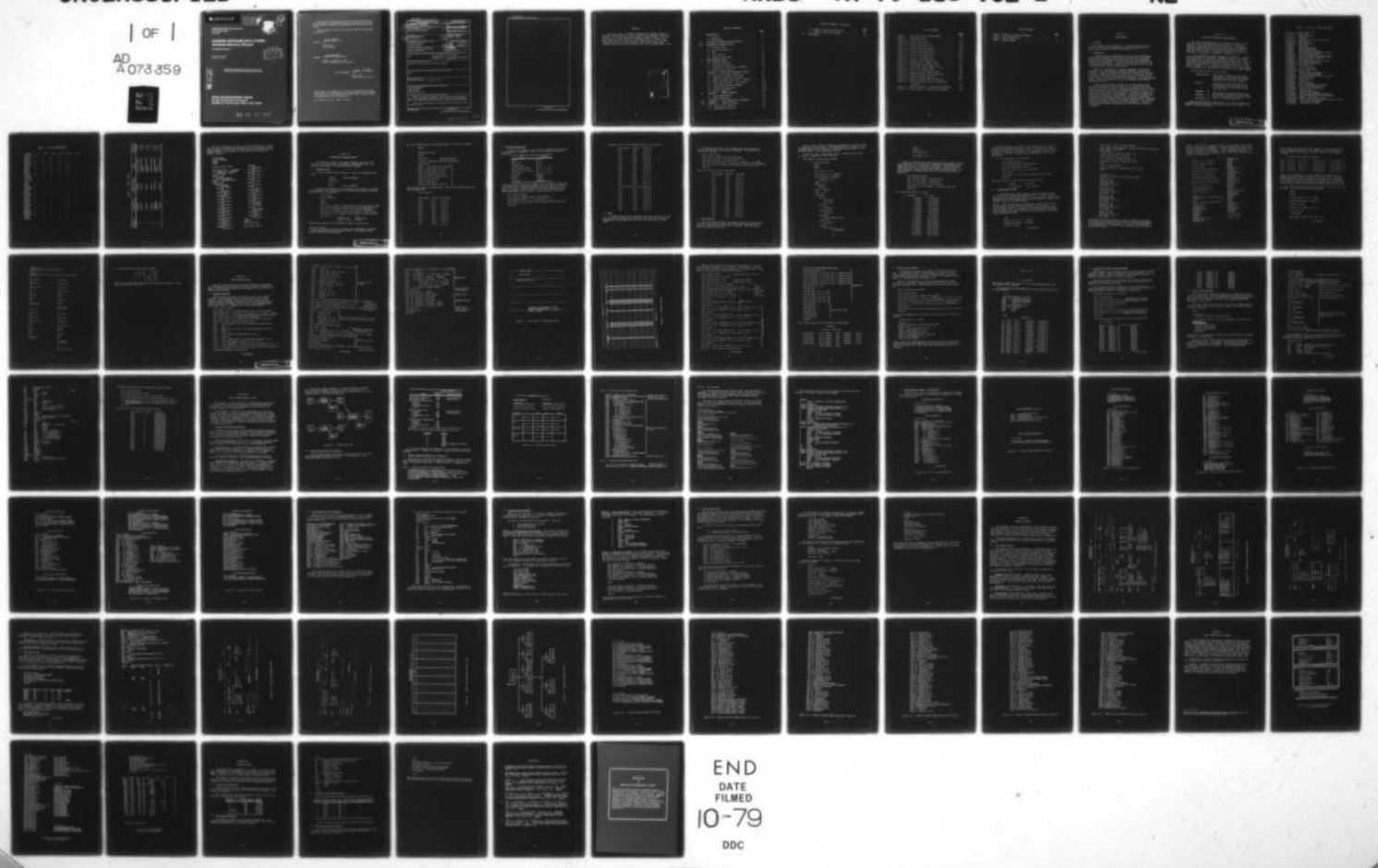
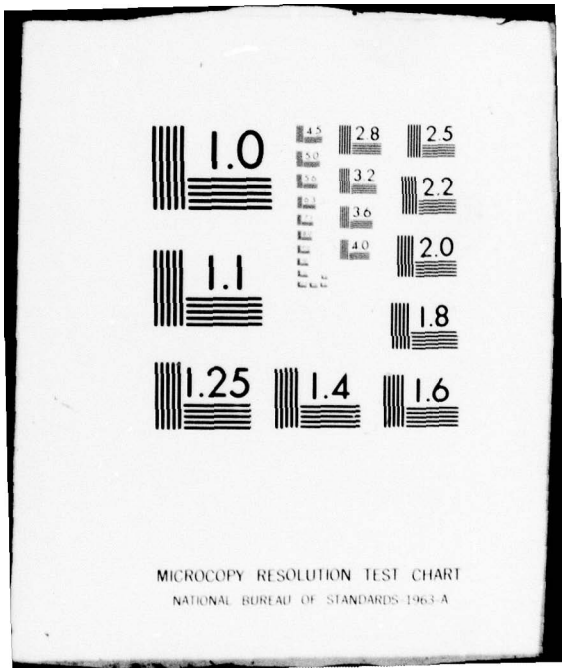
AD-A073 359 IIT RESEARCH INST CHICAGO IL F/G 9/2
BASELINE SOFTWARE DATA SYSTEM. VOLUME II. DATABASE REFERENCE MA--ETC(U)
JUL 79 L M DUVALL, C CURTIS F30602-77-C-0052
UNCLASSIFIED RADC -TR-79-185-VOL-2 NL



| OF |
AD
A073 359

END
DATE
FILMED
10-79
DDC


1.0
1.1
1.25
1.4
1.6
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.8
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A073359

LEVEL II

12

RADC-TR-79-185, Vol II (of two)
Final Technical Report
July 1979

BASELINE SOFTWARE DATA SYSTEM
Database Reference Manual

IIT Research Institute

Lorraine M. Duvall
Christine Curtis

DDC
RECEIVED
AUG 31 1979
C

DDC FILE COPY

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 08 31 007

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-185, Vol II (of two) has been reviewed and is approved for publication.

APPROVED: John Palaimo

JOHN PALAIMO
Project Engineer

APPROVED: Wendall C. Bauman

WENDALL C. BAUMAN, Col, USAF
Chief, Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-79-185, Vol II (of two)	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): BASELINE SOFTWARE DATA SYSTEM. Volume II. Database Reference Manual.	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. Feb 77 - Aug 78
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Lorraine M. Duvall, Christine Curtis	8. CONTRACT OR GRANT NUMBER(s): F30602-77-C-0052
9. PERFORMING ORGANIZATION NAME AND ADDRESS: IIT Research Institute, 10 West 35th Street, Chicago IL 60616	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63728F, 55500027, 08
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB NY 13441	12. REPORT DATE: Jul 1979; 13. NUMBER OF PAGES: 89
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS. (of this report): UNCLASSIFIED; 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John Palaimo (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

database management
data definition
data retrieval
software failure data

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Volume I of this report provides a feature evaluation of the Management Data Query System (MDQS), a discussion of the contents of the Baseline databases, and a summary of the data required for software reliability modelling.

Volume II is a reference guide for defining and retrieving data from the Baseline databases.

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

175 350

PREFACE

This final report, BASELINE SOFTWARE DATA SYSTEM, Volume II, Database Reference Manual, was prepared by IIT Research Institute, Chicago, IL, as part of Contract Number F30602-77-0052. The work was sponsored by the Rome Air Development Center, Griffiss Air Force Base, New York, with Mr. John Palaimo serving as the RADC Technical Monitor for this program. This report covers work conducted during the period from February 1977 through August 1978.

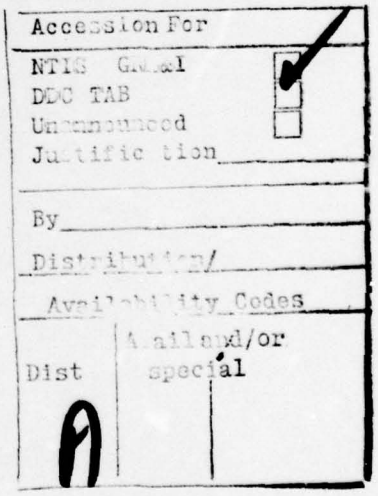


TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
	1.1 Purpose	1
	1.2 Background	1
II.	HISTORICAL DATABASE CHARACTERISTICS	3
III.	ELEMENTARY PROCEDURE USAGE	9
	3.1 A Simple Query	9
	3.2 Qualified Retrieval	11
	3.3 Sort	12
	3.4 Count and Sum	13
	3.5 Conversational MDQS	16
IV.	MORE ADVANCED USAGE	23
	4.1 Formatted Reports	23
	4.2 Writing Data Subsets	30
	4.3 Interface to Applications Programs	32
	4.4 Table Lookup Facility	33
V.	APPENDIX A - SYSTEM ADMINISTRATOR'S GUIDE	A-1
	5.1 Historical Database Definition	A-1
	5.2 Naming Conventions for Datasets	A-2
	5.3 Sample Database Definition - Project 1	A-3
	5.4 Definition Listings - All Projects	A-8
	5.5 A Decoding Subroutine Example	A-16
	5.6 Database Restructuring	A-18
	5.7 Privacy Subsystem	A-20
VI.	APPENDIX B - SUMMARY DATABASE	B-1
	6.1 Data Summarization	B-1
	6.2 MDQS Definitions	B-5
VII.	APPENDIX C - RADC PRODUCTIVITY DATABASE	C-1
VIII.	APPENDIX D - DATASET LOADING	D-1
	8.1 Creating the Permfiles	D-1
	8.2 Concatenating Dataset	D-1

TABLE OF CONTENTS (CONTINUED)

		Page
	8.3 Magnetic Tape Characteristics.....	D-2
	8.4 Magnetic Tape Data Loading.....	D-2
IX.	REFERENCES.....	37

LIST OF FIGURES

		Page
Figure 1	Cover Page for Formatted Report.....	26
Figure 2	Formatted Report.....	27
Figure A-1	Translation Flow.....	A-2
Figure A-2	Dataset Naming Form.....	A-4
Figure A-3	Project One Definitions.....	A-8
Figure A-4	Project Two Definitions.....	A-10
Figure A-5	Project Three Definitions.....	A-12
Figure A-6	Project Four Definitions.....	A-13
Figure A-7	Project Five Definitions.....	A-14
Figure A-8	Project Six Definitions.....	A-15
Figure B-1	Component Data Summary Form.....	B-2
Figure B-2	Technology Data Summary Form.....	B-3
Figure B-3	Instructions Data Summary Form.....	B-4
Figure B-4	Errors Data Summary Form.....	B-7
Figure B-5	Corrections Data Summary Form.....	B-8
Figure B-6	Component-Module Data Summary Form.....	B-9
Figure B-7	Summary Database Entries.....	B-10
Figure B-8	Summary Database MDQS Definitions.....	B-11
Figure C-1	Data-Item Descriptions - Productivity Database.....	C-2
Figure C-2	MDQS Definitions - Productivity Database....	C-3
Figure C-3	Query Example - Productivity Database.....	C-4

LIST OF TABLES

		Page
Table 1	Glossary of Data-Item Names.....	4
Table 2	Data-Item/Projects.....	5
Table 3	Dataset Names.....	6

Section I

INTRODUCTION

1.1 Purpose

The purpose of this manual is to provide the user of the Baseline Software Data System with a general capability for retrieving information from the databases.

1.2 Background

The databases are implemented on the Rome Air Development Center (RADC) Honeywell 6180 Computer System using the General Comprehensive Operating Supervisor (GCOS) and the Management Data Query System (MDQS). The reader of this guide should have a basic understanding of the use of GCOS for both the batch and timesharing environments. However, it is expected that the user of this guide has no knowledge of MDQS.

MDQS is a comprehensive database management system which provides tools for defining, loading, updating, and querying databases. This Reference Manual does not discuss all the capabilities of MDQS but contains a discussion of the basic features as applied to the Baseline Databases, and will help the novice user to query the data and write simple reports. For more advanced usage, the user is referred to the two Honeywell MDQS Manuals (references 4 and 5).

Section II introduces the user to the attributes of the Historical Database so that queries can be easily formulated. Sections III and IV provide a progressive technique for writing queries and reports as well as an explanation of the user's method of interfacing to application programs. Appendix A presents a step-by-step procedure for defining the Historical Database and definition listings for that database. The design and MDQS definitions for the Summary Database are contained in Appendix B, including the data summary forms used to summarize the information from the Historical Database. The MDQS definitions for the RADC Productivity Database are presented in Appendix C. Appendix D contains a method for loading the data from magnetic tape into a permfile.

Section II

HISTORICAL DATABASE CHARACTERISTICS

The Historical Database for the Baseline S/W Data System consists of six distinct sequential datasets containing a total of 31,912 eighty-four character records. The datasets represent software problem reports (SPR), module information, and run analysis reports on the software development of six projects. Each project (designated as Project 1 through 6) is discussed in references 3 through 8, respectively.

To use the MDQS procedure language for retrievals, unique names are needed to designate a database reference, an application definition, an entry, and the relevant data items. The entry names and the data-item names are the same for all datasets. The data-item names are listed and defined in Table 1; a designation of which datasets contain the specific data item is contained in Table 2. A "1" denotes that the data is in the dataset, a "3" indicates that information is included in the final report.

The following are valid entry names:

PROBLEM-REPORT - This entry normally includes such information as the date the problem was discovered and fixed, the module affected, and the error category.

MODULE - This entry contains descriptive information for each software module in the system including module name, type and size.

HARDWARE -
SOFTWARE -
TEST-CHAR - } These entries contain information on the hardware and software environment and the testing characteristics. Only Project 2 has these entries.

Table 3 contains, for each project, the project designation, database-reference name(s), application name, entry name(s), and number of record occurrences.

PRECEDING PAGE BLANK

TABLE 1: GLOSSARY OF DATA-ITEM NAMES

010	PROJ-ID	PROJECT IDENTIFICATION
020	PROJ-VERSION	PROJECT VERSION
030	PROJ-TYPE	PROJECT TYPE
040	SYS-ID	SYSTEM IDENTIFICATION
050	SYS-VERSION	SYSTEM VERSION
060	SYS-TYPE	SYSTEM TYPE
070	SSYS-ID	SUBSYSTEM OR FUNCTIONAL AREA IDENTIFICATION
080	SSYS-VERSION	SUBSYSTEM VERSION
090	SSYS-TYPE	SUBSYSTEM TYPE
100	MOD-ID	MODULE IDENTIFICATION
110	MOD-VERSION	MODULE VERSION
120	MOD-TYPE	MODULE TYPE
130	COMP-ID	COMPUTER IDENTIFICATION
140	COMP-OM	COMPUTER OPERATING MODE
150	COMP-RATE	COMPUTER PROCESSING RATE
160	COMP-OS	COMPUTER OPERATING SYSTEM TYPE
170	TECH-ID	IDENTIFICATION OF THE CONSTRUCTION TECHNOLOGY
180	COMPL-ID	TYPE OF COMPLEXITY MEASURE USED
190	COMPLEXITY	THE COMPLEXITY MEASURE VALUE
200	CONST-TYPE	CONSTITUENT TYPE(EX. JOVIAL,ASSEMBLY LANGUAGE)
210	NUM-OCCUR	NUMBER OF OCCURRENCES OF CONSTITUENT TYPE
220	PHASE	PHASE IN WHICH ACTION OCCURRED
230	NUM-RUNS-TOT	TOTAL NUMBER OF RUNS
235	TEST-PER	THE PERIOD IN WHICH THE TEST WAS PERFORMED
240	NUM-RUNS-OK	TOTAL NUMBER OF CORRECT RUNS
250	AHRS-PER-TEST	AVERAGE NUMBER OF HOURS PER TEST
260	TEST-ID	TEST IDENTIFICATION
270	TEST-TYPE	TYPE OF TEST
280	DATE-RUN	DATE THE TEST WAS RUN
290	STRESS-TYPE	TYPE OF STRESS APPLIED
300	STRESS-MEAS	AMOUNT OF STRESS APPLIED
310	TEST-RESULT	RESULT OF TEST
315	NUM-ERR	NUMBER OF ERRORS DISCOVERED PER TEST
320	SPR-NUM	SOFTWARE PROBLEM REPORT NUMBER
330	DATE-OPEN	DATE THE PROBLEM WAS REPORTED
340	MOD-SOURCE	THE MODULE ID WHERE THE PROBLEM WAS MANIFESTED
350	ERR-CAT-TYPE	ERROR CATEGORY TYPE
360	ERROR-CAT	ERROR CATEGORY CODE
370	SEV-TYPE	SEVERITY TYPE
380	SEVERITY	SEVERITY
390	TYPE-TERM	TYPE OF TERMINATION
400	HRS-TO-DISC	HOURS TO DISCOVERY
405	WORK-CAT	THE TYPE OF DEVELOPMENT TASK PERFORMED
410	SMN-NUM	SOFTWARE MODIFICATION NOTICE NUMBER
420	MOD-CHANGED	THE ID OF THE CHANGED MODULE
430	MOD-CH-VERS	THE VERSION OF THE CHANGED MODULE
440	COR-TYPE	CORRECTION TYPE
450	COR-MECH	CORRECTION MECHANISM
455	ACT-CAT	THE TYPE OF TEST PERFORMED
460	DATE-BEGUN	DATE WHEN PROBLEM SOLUTION WAS INITIATED
470	DATE-CLOSE	DATED WHEN PROBLEM WAS REPORTED TO BE CLOSED
480	DAYS-OPEN	NUMBER OF DAYS BETWEEN DATE OPEN AND DATE CLOSE
490	HHRS-TO-FIX	HUNDRETHS OF HOURS TO FIX
500	NUM-CHANGED	NUMBER OF SOURCE STATEMENTS CHANGED
510	CODE-CONT	A CODE THAT INDICATES AN SPR DOCUMENTS MORE THAN 1 PROBLEM
520	PROB-DESC	A DESCRIPTION OF THE PROBLEM
530	CORR-DESC	A DESCRIPTION OF THE CORRECTION
540	ERROR-DESC	A DESCRIPTION OF THE ERROR

*

TABLE 2: DATA-ITEM/PROJECTS

ATTRIBUTE	PROJ 1	PROJ 2	PROJ 3	PROJ 4	PROJ 5	PROJ 6
PROJ-ID	1	2	1	3	3	
PROJ-VERSION	1			1		
PROJ-TYPE	3	3	3	3	3	
SYS-ID	3	1	3	1		
SYS-VERSION		3		1		
SYS-TYPE	3	1	3	3		
SSYS-ID	3	1	3	1	1	1
SSYS-VERSION				1	1	
SSYS-TYPE	3			1		
MOD-ID	3	1	1		1	1
MOD-VERSION			1		2	
MOD-TYPE	3		1	1	2	
COMP-ID	3	1	3	3	3	
COMP-OM		3	3	3	3	
COMP-RATE		1	3	3	3	
COMP-OS		1	3			
TECH-ID	3	1	1	3	1	
COMPL-ID	3					
COMPLEXITY	3		1			
CONST-TYPE	3	1	1	3	1	
NUM-OCCUR	3	1	1	3	1	
NUM-RUNS-TOT		1				
NUM-RUNS-OK		1				
AHRS-PER-TEST		1				
PHASE		1	1	1	1	
TEST-PER	1		1			1
TEST-ID	1					
TEST-TYPE	3	3	3			
DATE-RUN						1
STRESS-TYPE						
STRESS-MEAS						1
TEST-RESULT						1
NUM-ERR						1
SPR-NUM	1	1	1	1	1	
DATE-OPEN	1	1	1		1	
MOD-SOURCE	1					
ERR-CAT-TYPE						
ERROR-CAT	1	1	1	1	1	1
SEV-TYPE						
SEVERITY	1		1			1
TYPE-TERM	1		1			
HRS-TO-DISC		1				
WORK-CAT						1
SMN-NUM	1		1	1		
MOD-CHANGED	1	1	1		1	
MOD-CH-VERS			1			
COR-TYPE	1	1	1		1	
COR-MECH		1				
ACT-CAT						1
DATE-BEGUN						
DATE-CLOSE	1	1	1	1	1	
DAYS-OPEN	1		1			
HHRS-TO-FIX		1				
NUM-CHANGED						1
CODE-CONT	1	1				
PROB-DESC					1	
CORR-DESC					1	
ERROR-DESC				1		

TABLE 3: DATASET NAMES

Project Designation	Database Reference Name	Application Definition Name	Entry Name	Number of Records
1	DBRTRWS	ADOTRWS	PROBLEM-REPORT	4,970
2	DBRB1S1	ADOB1S1	PROBLEM-REPORT MODULE	2,036 69
3	DBRSDS1	ADOSDS1	PROBLEM-REPORT MODULE	2,165 109
4	DBRAPS1	ADOAPS1	PROBLEM-REPORT	11,730
5	DBRSAFS1 DBRSAFSN	(ADOSAFS1)	PROBLEM-REPORT MODULE	5,693 2,413
6	DBRSMTC	ADOSMTC	PROBLEM-REPORT	2,719

The user also has the option, in GCOS Time Sharing, to determine the data item names and attributes of the particular dataset being queried by the use of the ADFQ (Application Definition File Query) Command. This feature is illustrated below using the Project 1 dataset.

```
SYSTEM ?MDQ
OLD OR NEW-NEW
READY
*ADFQ

BRIEF OR FULL  FULL

ADF OBJECT FILE -- ADQTRWS
DATA BASE REF -- DBRTRWS
    SEQUENTIAL FILE
ENTRY -- ALL
ENTRYS ARE
    PROBLEM-REPORT
ENTRY -- PROBLEM-REPORT
ITEM -- ALL
ITEMS ARE
    MOD-CHANGED
        A/N SIZE 7
    SPR-NUM
        A/N SIZE 4
    MOD-SOURCE
        A/N SIZE 7
    DATE-OPEN
        A/N SIZE 8
    MO-OPEN
        A/N SIZE 2
    DA-OPEN
        A/N SIZE 2
    YR-OPEN
        A/N SIZE 2
    SEVERITY
        A/N SIZE 1
    TEST-PER
        A/N SIZE 2
    TEST-ID
        A/N SIZE 8
    ERROR-CAT
        A/N SIZE 11
    MAJOR-CAT
        A/N SIZE 2
    MINOR-CAT
        A/N SIZE 3
    SMN-NUM
        A/N SIZE 6
    COR-TYPE
        A/N SIZE 6
    NEW-MOD
        A/N SIZE 1
    DOC-UP
        A/N SIZE 1
    COM-CH
        A/N SIZE 1
    DB-CH
        A/N SIZE 1
    EXPLAN
        A/N SIZE 2
    DATE-CLOSE
        A/N SIZE 8
    MO-CLOSE
        A/N SIZE 2
    DA-CLOSE
        A/N SIZE 2
    YR-CLOSE
        A/N SIZE 2
    PROJ-VERSION
        A/N SIZE 6
    PROJ-ID
        SCAN FIELD
        A/N SIZE 2
    CODE-CONT
        A/N SIZE 1
    DAYS-OPEN
        A/N SIZE 3
ITEM --
ENTRY --
DATA BASE REF --

ADF OBJECT FILE --
*
```

Section III

ELEMENTARY PROCEDURE USAGE

The elementary usage of the MDQS Language allows for the retrieval of instances of the database, sorting this data, performing computations, and printing simple reports.

3.1 A Simple Query

A simple procedure must contain at least the following statements:

```
INVOKE...        first statement
RETRIEVE...
PRINT...
END...           last statement
```

A simplified example is the retrieval of the first ten entries of the Project 1 data printing only the items MOD-CHANGED, SPR-NUM, and DATE-OPEN on a permfile.

```
SYSTEM ?MDQ
OLD OR NEW-NEW
READY
*AUTOX
0010 INVOKE ADOTRWS  ←Application Definition Object Name*
0020 RETRIEVE PROBLEM-REPORT FROM DBRTRWS ←Entry from Database
0030 ONLY 10 ENTRIES ←Limits retrievals to ten records
0040 PRINT ON FILE REPORT-1 MOD-CHANGED,SPR-NUM,DATE-OPEN
0050 END
                  ↑                    ↑
          Permfile to contain result   Items to be printed
```

The above procedure can be checked for errors and then run.

---

*It may be necessary to use the catalog name (BFCBMIS1) in addition to the Application Definition Object Name. This example would contain INVOKE BFCBMIS1/ADOTRWS.

PRECEDING PAGE BLANK

The RUNS command gives a continuous status of the job's progress.

```
*CHECK
 .....
PROCEDURE CHECKED
*RUNS
 .....
RUN-ID? CC                    -Entered by user.
$ IDENT?  BFCBMISI,C CURTIS,555008570052
SNUMB 6685T
6685T    GEIN EXECUTING @ 09.430
6685T-01 WAIT-PERIP @ 09.431
6685T-01 WAIT-CORE @ 09.431
6685T-01 EXECUTING @ 09.431
6685T-01 INITIALIZING @ 09.431
6685T OUTPUT STARTED
normal termination
6685T-01 PRG TERMIN @ 09.433
```

The resultant report is shown below. The column headings are generated automatically.

```
*LIST REPORT-1

MOD-CHANGED   SPR-NUM   DATE-OPEN
C106          0042      6-18-73
C102          0180      6-04-73
C102          0181      6-02-73
C105          0182      6-02-73
F403          0183      6-02-73
E112          0184      6-02-73
B103          0185      6-02-73
E113          0186      6-02-73
C503          0187      6-02-73
D/B           0188      6-04-73
```

3.2 Qualified Retrieval

A conditional expression may be used to qualify the retrieval of data during a procedure. Some of the allowable relational tests are:

Test	Operator
Equal	EQ or =
Less than	LT or <
Greater than	GT or >
Less than or equal	LE, or <=, or =<
Greater than or equal	GE, or >=, or =>
Not equal	NE, or ><, or <>
Present	PRESENT
Absent	ABSENT
Contains	CONTAINS

Using the previous Project 1 example, the retrieval is qualified to include only those problem-reports for the year 1973 and the month of October. Please note that DATE-OPEN consists of the three sub-items MO-OPEN, DA-OPEN, and YR-OPEN and can be referenced individually by item name or collectively as DATE-OPEN.

```
0010 INVOKE ADOTRWS
0020 RETRIEVE PROBLEM-REPORT FROM DBRTRWS
0030 WHERE YR-OPEN = "73" AND MO-OPEN = "10"
0040 PRINT ON FILE REPORT-1 MOD-CHANGED,SPR-NUM,DATE-OPEN
0050 END
```

A partial list of the resultant report is as follows.

MOD-CHANGED	SPR-NUM	DATE-OPEN
C104	4025	10-02-73
D/B	4027	10-05-73
C207	4028	10-05-73
A504	4029	10-08-73
C302	4030	10-09-73
A505	4031	10-08-73
A510	4031	10-08-73
A403	4033	10-09-73
D109	4034	10-10-73
A403	4035	10-11-73
B107	4036	10-11-73
B107	4037	10-11-73
D/B	4038	10-11-73
A404	4040	10-13-73
D/B	5123	10-01-73
G113	5124	10-02-73
H215	5125	10-02-73
H223	5125	10-03-73
H219	5126	10-03-73
D104	5127	10-04-73
H211	5128	10-04-73
G206	5129	10-04-73
H212	5130	10-04-73

3.3 Sort

A sorting process may be employed after the retrieval of the data to order the selected data items. The sort may be either in ascending or descending order and sorted on more than one data item.

In the previous report it is evident that the DATE-OPEN is not in order by day within the year and month. To correct this the SORT is employed.

```
0010 INVOKE ADOTRWS
0020 RETRIEVE PROBLEM-REPORT FROM DBRTRWS
0030 WHERE YR-OPEN = "73" AND MO-OPEN = "10"
0035 SORT PROBLEM-REPORT ON DA-OPEN IN ASCENDING ORDER
0040 PRINT ON FILE REPORT-1 MOD-CHANGED,SPR-NUM,DATE-OPEN
0050 END
```

← Sort Statement

Some of the sorted data is shown below.

MOD-CHANGED	SPR-NUM	DATE-OPEN
C101	4043	10-01-73
H215	5121	10-01-73
H211	5122	10-01-73
D/B	5123	10-01-73
H208	5194	10-01-73
C104	6451	10-01-73
H231	6452	10-01-73
E104	6455	10-01-73
E104	6457	10-01-73
A403	6462	10-01-73
D108	6464	10-01-73
D412	6467	10-01-73
D109	6468	10-01-73
E102	6522	10-01-73
E109	6843	10-01-73
C104	4025	10-02-73
C113	5124	10-02-73

3.4 Count and Sum

The COUNT function counts the number of times an item contains data other than blank and adds one to the count each time an expression containing this function is evaluated. The SUM function sums the contents of an item.

Both of these functions require the creation of a new variable to contain the sum or count. The mathematical new variable, $ new-variable-name, is initially set to zero and cannot exceed 21 significant digits unless previously defined.

Project 5, file 2 (the MODULE entry), contains some fields suitable to demonstrate these features.

The ADFQ for this dataset follows.

```
*SYSTEM MDQ NEW
READY
*ADFQ

BRIEF OR FULL  FULL

ADF OBJECT FILE — ADOSAFS1
DATA BASE REF — DBRSAFSN
    SEQUENTIAL FILE
ENTRY — ALL
ENTRYS ARE
    MODULE
ENTRY — MODULE
ITEM — ALL
ITEMS ARE
    MOD-ID
        SCAN FIELD
        A/N SIZE 8
    SSYS-ID
        A/N SIZE 1
    SSYS-VERSION
        A/N SIZE 7
    NUM-OCCUR
        COBOL NUMERIC SIZE 6
    CONST-TYPE
        A/N SIZE 7
    TECH-ID
        A/N SIZE 12
```

(continued)

```
ITEM —
ENTRY —
DATA BASE REF —
ADF OBJECT FILE —
*
```

Using the ADFQ and the definitions from the Glossary (Table 1) as a reference, it is possible to formulate a meaningful query employing both the SUM and COUNT functions. The following procedure was written to determine the number of modules and the total number of lines of code (NUM-OCCUR) for this project. In this procedure, a running total for ID-COUNT and TOT-LINES is generated.

```
0010 INVOKE ADOSAFS1
0020 RETRIEVE MODULE FROM DBRSAFSN
0040 LET $ID-COUNT = COUNT MOD-ID
0050 LET $TOT-LINES = SUM NUM-OCCUR
0055 PRINT ON FILE REPORT-4 $ID-COUNT, $TOT-LINES
0060 END
```

The resultant report is listed.

ID-COUNT	TOT-LINES
1.000000	228.000000
2.000000	506.000000
3.000000	540.000000
4.000000	668.000000
5.000000	674.000000
6.000000	688.000000
7.000000	702.000000
8.000000	1335.000000
9.000000	2354.000000
10.000000	2985.000000
11.000000	3549.000000
12.000000	3994.000000
13.000000	4564.000000
14.000000	4894.000000
15.000000	5607.000000

From the report it is obvious that the COUNT, SUM and PRINT statements are being executed for each record retrieval, and a running count and sum are being printed. To avoid this occurrence, the WRAP-UP statement can be used. In this case it causes the execution of the PRINT statement upon completion of all of the retrievals.

It is incorporated as shown below.

```
0010 INVOKE ADOSAFS1
0020 RETRIEVE MODULE FROM DBRSAFSN
0040 LET $ID-COUNT = MOD-ID
0050 LET $TOT-LINES = SUM NUM-OCCUR
0051 WRAP-UP
0055 PRINT ON FILE REPORT-4 $ID-COUNT,$TOT-LINES
0060 END
```

Now the resultant report contains only the totals.

```
ID-COUNT          TOT-LINES
2413.000000     317031.000000
```

3.5 Conversational MDQS

The Conversational MDQS Language (CMDQ) provides an interactive method of generating simple queries. To utilize this feature special permissions must be obtained from the organization responsible for the database administrator functions. Some of the relevant features are demonstrated.

BROWSE: Here the database is scanned for entries that meet the user-supplied criteria and are displayed on the terminal. In the following example, all Project 1 records that have a major error category of BB are requested and sorted by date-open while only the date-open, major and minor error categories, and number of days-open are printed.

```
 CMDQ
APPLICATION FILE IS - ADOTRWS
DATABASE REFERENCE IS DBRTRWS
FUNCTION - BROW
```

(continued)

```
ONLY ENTRY IN DBR IS PROBLEM-REPORT

DATA ITEMS TO BE DISPLAYED - MAJOR-CAT,MINOR-CAT,DATE-OPEN
More items ? DAYS-OPEN
More items ?

DISPLAY EVEN IF BLANK OR ZERO ? YES

PAUSE BETWEEN ENTRY INSTANCES ?

DATA ITEMS TO BE SUMMED OR COUNTED -

SELECTION CRITERIA - MAJOR-CAT = "BB"
More ?

LIMIT # SETS OF DATA TO -

SEQUENCE DATA ON - DA-OPEN,MO-OPEN, YR-OPEN
More ?

 .....

RUN-ID? CC
$ IDENT?  BFCBMIS1,C CURTIS,555008570052
SNUMB 1866T

MAJOR-CAT= BB
MINOR-CAT= 062
DATE-OPEN= 10-01-73
DAYS-OPEN=   0

MAJOR-CAT= BB
MINOR-CAT= 010
DATE-OPEN= 10-01-73
DAYS-OPEN    8

MAJOR-CAT= BB
MINOR-CAT= 060
DATE-OPEN= 11-01-73
DAYS-OPEN=  15

MAJOR-CAT= BB
MINOR-CAT= 061
DATE-OPEN= 11-01-73
DAYS-OPEN=   4

MAJOR-CAT= BB
MINOR-CAT= 062
DATE-OPEN= 11-01-73
DAYS-OPEN=   5
```

A carriage return was entered for null or negative responses. The selection criteria must be entered in the format acceptable to the procedures syntax analyser. If an error is made here the procedure will not execute and a message indicating syntax errors will appear.

CHOOSE: This function allows the user to choose data from a database and write it to a permfile. It may be qualified, sorted, etc., as per the interactive question. In this instance the user wishes to make a sub-file of the Project 1 data containing only the records with major error category of BB. Only five instances were chosen for convenience.

```
 CMDQ

APPLICATION FILE IS - ADOTRWS

DATABASE REFERENCE IS DBRTRWS

FUNCTION - CHOOSE

ONLY ENTRY IN DBR IS PROBLEM-REPORT

DATA ITEMS TO BE DISPLAYED - ALL

DISPLAY EVEN IF BLANK OR ZERO ? Y

PAUSE BETWEEN ENTRY INSTANCES ?

DATA ITEMS TO BE SUMMED OR COUNTED -

SELECTION CRITERIA - MAJOR-CAT = "BB"
More ?

LIMIT # SETS OF DATA TO - 5

OUTPUT CATALOG/FILE-STRING IS
- BFCBMIS1/CHOOSE

EXTRACT ALL INSTANCES ? Y

 .....
RUN-ID? CC
$ IDENT?  BFCBMIS1,C CURTIS,555008570052
SNUMB 2001T

2001T
TERMINATE PROCEDURE BY RESPONDING 'END' TO A
 NON DATA REQUEST

MOD-CHANGED= A313
SPR-NUM= 0194
MOD-SOURCE= A313
DATE-OPEN=  6-05-73
MO-OPEN=  6
DA-OPEN= 05
YR-OPEN= 73
SEVERITY= 2
TEST-PER=  V
TEST-ID=
ERROR-CAT= BB140M-0300
MAJOR-CAT= BB
MINOR-CAT= 140
SMN-NUM= M-0300
COR-TYPE=     X
NEW-MOD=
DOC-UP=
COM-CH=
DB-CH= X
EXPLAN=
DATE-CLOSE=  6-05-73
MO-CLOSE=  6
DA-CLOSE= 05
YR-CLOSE= 73
PROJ-VERSION= BLK2
PROJ-ID= 03
CODE-CONT=
DAYS-OPEN=    0
MOD-CHANGED= C211
SPR-NUM= 0214
MOD-SOURCE= C211
DATE-OPEN=  6-06-73
MO-OPEN=  6
DA-OPEN= 06
YR-OPEN= 73
SEVERITY=
TEST-PER=  V
TEST-ID=
ERROR-CAT= BB140M-0365
MAJOR-CAT= BB
MINOR-CAT= 140
SMN-NUM= M-0365
COR-TYPE= X
NEW-MOD= X
DOC-UP=
COM-CH=
DB-CH=
EXPLAN=
DATE-CLOSE=  6-08-73
MO-CLOSE=  6
DA-CLOSE= 08
YR-CLOSE= 73
PROJ-VERSION= BLK2
PROJ-ID= 03
CODE-CONT=
DAYS-OPEN=    2

ACTIVITY TERMINATED

FUNCTION - DONE

*
```

The response Y was given for yes answers. A CR was entered for null or negative responses. The permfile BFCBMIS1/CHOOSE was created automatically and has the same format as the original database.

A listing of the permfile follows.

```
A313    0194A313    6-05-732 V      BB140M-0300   X    6-05-73BLK2   03
F504    0195F504    6-04-73  V      BB170M-0301X       6-05-73BLK2   03
C210    0209C210    6-06-732 V      BB060M-0361X       6-08-73BLK2   03
G131    0211G131    6-06-73  V      BB060M-0371XX      6-11-73BLK2   03
C211    0214C211    6-06-73  V      BB140M-0365X       6-08-73BLK2   03
```

LOAD: A new sequential (or index sequential) database can be loaded with data supplied interactively by the user. The LOAD command queries the user for each item of data to be supplied. All the definitions must be previously defined for this command to function. In this case the file structure must be flat. The output database must be created as a permfile prior to execution.

Here the user wants to enter the Project 2 testing data interactively. Every data item will be supplied.

```
READY
*CMDQ

APPLICATION FILE IS - ADOB1S1

DATABASE REFERENCE IS DBRB1S1

FUNCTION - LOAD

ENTRY TO BE LOADED - TESTING

KEY ITEMS ARE
  TEST-CHAR

ADDITIONAL ITEMS TO BE LOADED - ALL
```

(continued)

```
 .....
RUN-ID? CC
$ IDENT?  BFCBMIS1,C CURTIS,555008570052
SNUMB 2215T

2215T
TERMINATE PROCEDURE BY RESPONDING 'END' TO A NON DATA REQUEST
TEST-CHAR?
-
5
NUM-RUNS-TOT?
-
179
NUM-RUNS-OK?
-
143
AHRS-PER-TEST?
-
  9
BLOCK?
-
1
TEST-PER?
-
IMCT
OKAY?
-
Y
ENTRY STORED
NEXT ENTRY
 ?
-
Y
TEST-CHAR?
-
5
NUM-RUNS-TOT?
-
403
NUM-RUNS-OK?
-
293
AHRS-PER-TEST?
-
  7
BLOCK?
-
0
TEST-PER?
-
IMCT
OKAY?
-
Y
ENTRY STORED
NEXT ENTRY
 ?
-
Y
TEST-CHAR?
-
5
NUM-RUNS-TOT?
-
  5
NUM-RUNS-OK?
-
  0
AHRS-PER-TEST?
-
248
BLOCK?1
-
1
TEST-PER?
-
 SVT
OKAY?
-
Y
ENTRY STORED
NEXT ENTRY
 ?
-
Y
TEST-CHAR?
-
5
NUM-RUNS-TOT?
-
  8
NUM-RUNS-OK?
-
  1
AHRS-PER-TEST?
-
320
BLOCK?
-
0
TEST-PER?
-
SVT
OKAY?
-
Y
ENTRY STORED
NEXT ENTRY
 ?
-
END

ACTIVITY TERMINATED
```

A list of the entered data is as follows.

```
5  179  143    9 1 IMCT
5  403  293    7 0 IMCT
5    5    0  248 1  SVT
5    8    1  320 0 SVT
```

Note that this was just an exercise and the actual Project 2 data was not entered in this way.

Section IV

MORE ADVANCED USAGE

Some of the more advanced features of MDQS are introduced here including the production of formatted reports, writing data subsets, interfacing to application programs, and the use of the Table Lookup facility.

4.1 Formatted Reports

MDQS provides relatively easy format control for printed reports. The following Project 5 report shows a cover page, horizontal page headings, new variable output format control, column control of headings and data, the use of system variables and the printing of a final or total line. Page numbering and lines/page control are defaulted to the system standard.

```
0010 INVOKE ADOSAFS1
0020 REPORT SAFS1 ON PRINTER◄Indicates report is to be printed.
0030 COVER PAGE IS COV1◄Names cover page section; causes auto. printing.
0040 PAGE HEADING IS HD1◄Names page heading section; causes printing
                         at top of each new page.
0050 PAGE FOOTING IF FT1◄Indicates a footing line for each page.
0060 COV1, SPACE 15◄Vertical and page control.
0070    LINE "PROJECT 5 DATA" COL 60◄Horizontal column control.
0080    SPACE 2
0090    LINE "FOR 1974 ONLY" COL 60◄Line indicates a new line.
0100    SPACE 3
0110    LINE "SORTED BY MONTH AND DAY" COL 55
0120    SPACE 20
0130    LINE "COMPUTATION OF DAYS BETWEEN FAILURES" COL 70
0140    LINE "AND NUMBER OF DAYS TO FIX" COL 70
0150    LINE "ASSUMING 365 DAYS/YEAR AND 30 DAYS/MONTH" COL 70
0160    LINE %DATE-TIME COL 70◄Print system variable for date and time.
0170SPACE TOP◄Spaces to new page top.
```

(continued)

PRECEDING PAGE BLANK

```
0180HD1. LINE "PROJECT 5 DATA - 1974" COL 60
0190     SPACE 3
0200     LINE "DAYS BETWEEN" COL 14,
0210     "DATE-OPEN" COL 39,
0220     "NO. OF DAYS" COL 61,
0230     "MODULE" COL 80,
0240     "ERROR CATEGORY" COL 99
0250     LINE "FAILURES" COL 16,
0260     "OPEN" COL 65,
0270     "ID" COL 82,
0280     "MAJOR" COL 100,
0290     "MINOR" COL 107
0300     SPACE 1
0310 FT1. LINE SPACE-NUMBER COL 65
0315 DEFINE $DIFF,$NODAYS2,$OPEN WITH PIC "999"
0316 DEFINE $AVEDAYS,$AVEOPEN WITH PIC "999"
0317 DEFINE $TOTDAYS,$TOTOPEN,$NOENT WITH PIC "99999"
0320 PRT. LINE $DIFF COL 19,
0330     DATE-OPEN COL 40,
0340     $OPEN COL 66,
350      MOD-CHANGED COL 79,
0360     MAJOR-CAT COL 103,
0370     MINOR-CAT COL 108,
0390 DEFINE $SW WITH PIC "9"
0391 PRT1. SPACE
0392 LINE "AVERAGE D.B.V" COL 2,$AVEDAYS COL 19 PIC "999
0393       "AVERAGE NO.D.O." COL 48,$AVEOPEN COL 66 PIC "999"
0400 A2. RETRIEVE PROBLEM-REPORT FROM DBRSAFS1
0410 WHERE YR-OPEN = 74
0415ONLY 45 ENTRIES
0416 WHEN A2
0420 SORT PROBLEM-REPORT ON MO-OPEN - 30 + DA-OPEN
0430 IF $SW = 1 GO TO L1
```

Horizontal column headings

System variable in footer line.

Defines new mathematical variables with Cobol-like picture desc.

Data line showing new variables and items from data with horizontal format control.

Final line of report.

Qualified retrieval.

Sort on date.

Branching logic.

(continued)

```
0440 LET $NODAYS2 = 30 + MO-OPEN - 30 + DA-OPEN
0450     LET $SW = 1
460 L1.LET $NODAYS1 - 30 + MO-OPEN + DA-OPEN
0470     LET $DIFF = $NODAYS2 - $NODAYS1
0480     LET $OPEN = (YR-CLOSE - 74) + 365
0490         + MO-CLOSE + 30 - 30 + DA-CLOSE
0500         - (MO-OPEN + 30 - 30 + DA-OPEN)
0502     LET $NOENT + COUNT DA-OPEN
0530 PRINT PRT
0540 LET $NODAYS2 = $NODAYS1
0541 HOLD WITHIN A2 $DIFF,$OPEN
0542 LET $TOTDAYS = SUM $DIFF
0543 LET $TOTOPEN = SUM $OPEN
0546 LET $AVEDAYS = $TOTDAYS / $NOENT
0547 LET $AVEOPEN = $TOTOPEN / $NOENT
0548 WRAP-UP
0549 PRINT PRT1
0550 END
```

Computations

Causes data line to be printed.

Control logic

Demonstrates sum.

Control logic

Causes final line to be printed.

PROJECT 5 DATA

FOR 1974 ONLY

SORTED BY MONTH AND DAY

COMPUTATION OF DAYS BETWEEN FAILURES
AND NUMBER OF DAYS TO FIX
ASSUMING 365 DAYS/YEAR AND 30 DAYS/MONTH
10/25/77 10.259

Figure 1: Cover Page for Formatted Report

PROJECT 5 DATA - 1974

DAYS BETWEEN FAILURES	DATE-OPEN	NO. OF DAYS OPEN	MODULE ID	ERROR CATEGORY MAJOR	MINOR
006	03/06/74	042	CODCHEK1	EK	030
002	03/08/74	007	CODIFACE	BB	040
004	03/12/74	021	CODBOSZZ	PP	020
000	03/12/74	322	CODPATCH	BA	160
006	03/18/74	027	CODBOSCD	BB	040
000	03/18/74	000	CODLIDBH	BB	070
002	03/20/74	002	CODMEMBH	DD	050
001	03/21/74	024	CODAPPDS	CC	030
008	03/29/74	334	CODIFACB	DD	050
003	04/02/74	031	CODLIMIT	BB	010
011	04/13/74	205	CODCHEK1	BK	020
003	04/22/74	282	CODBOSZZ	VV	020
002	04/24/74	119	COCCILZZ	VV	020
009	05/03/74	101	COAZHADB	BB	020
006	05/09/74	104	CODDMCHA	CC	010
008	05/13/74	035	CODBOSZZ	PP	020
004	05/17/74	037	COAZHADB	PP	020
000	05/17/74	127	COAZHADH	BU	030
007	05/24/74	114	CODZILBB	PP	020
004	05/28/74	020	CODCHEK1	PP	020
006	06/03/74	246	CODCHEK1	PP	010
008	06/07/74	237	COCCILZZ	VV	020
003	06/10/74	234	CODMEMBH	BB	070
001	06/11/74	206	COACILZZ	VV	020
001	06/12/74	094	COCOBCOH	PP	020
000	06/12/74	008	CODCHEK1	BU	030
003	06/19/74	038	COCOBCOH	BB	020
019	07/08/74	069	CODBOSZZ	PP	020
000	07/08/74	085	CODMEMBH	CC	161
001	07/09/74	041	CODBOSZZ	PP	020
006	07/15/74	062	CODMEMBH	PP	020
051	09/06/74	186	COAZHADB	BB	060
003	09/09/74	032	COAOIBBH	BB	010
001	09/10/74	153	CODBATHL	LL	060
006	09/16/74	023	CODLOKUP	BP	020
000	09/18/74	020	CODMEMH1	PP	020
000	09/18/74	143	CODMEMBB	DD	051
000	09/18/74	020	CODMEMBH	PP	020
003	09/20/74	048	CODMEMBH	BB	020
012	10/02/74	260	CODTIMER	AA	071
006	10/08/74	089	COACILZZ	VV	020
027	11/05/74	331	CODLIMIT	BB	070
001	11/06/74	226	CODCHEK1	BD	020
028	12/04/74	122	CDUSCOPE	VV	020
027	12/31/74	148	COAZHADB	LL	022

AVERAGE D.B.F 007 AVERAGE NO.D.O. 106

Figure 2: Formatted Report

Reports with format control can also be directed to a permfile for later printing on the teletype. The Project 3 "Errors by Test Period" report is one example of a teletype report. Note that in this report the titles are vertical.

```
0010 INVOKE ADOSDS1                    ←Indicates report to permfile.
0020 REPORT R6 ON FILE R65DS1
0030 PAGE HEADING IS HD1               ←Names page heading.
0040 PAGE FOOTING IS FT1               ←Names page footing.
0045 DEFINE $T WITH PIC "99999"
0046 DEFINE $T1,$T2,$T3,$T4 WITH PIC "9999"
0047 DEFINE $PER1,$PER2,$PER3,$PER4 WITH PIC "99.99"
0048 DEFINE $T5 WITH PIC "9999"
0049 DEFINE $PER5 WITH PIC "99.99"
0050 HD1. LINE "PROJECT 3" COL 38
0055 SPACE 1
0060 LINE "ERRORS BY TEST-PER" COL 34
0065 SPACE
0070 PRT. LINE "DEVELOPMENT" COL 2, "NO OF ERRORS" COL 20,
0080 $T1 COL 35, "PERCENT" COL 45, $PER1 COL 60
0090 SPACE 2
0100PRT2. LINE "VALIDATION" COL 2, "NO OF ERRORS" COL 20,
0110 $T2 COL 35, "PERCENT" COL 45, $PER2 COL 60
0115 SPACE 2
0120 PRT3. LINE "INTEGRATION" COL 2, "NO OF ERRORS" COL 20,
0130 $T3 COL 35, "PERCENT" COL 45, $PER3 COL 60
0135 SPACE 2
0140 PRT4. LINE "ACCEPTANCE" COL 2, "NO OF ERRORS" COL 20,
0150 $T4 COL 35, "PERCENT" COL 45, $PER4 COL 60
0155 SPACE 2
0156 PRT4A. LINE "OPERATIONAL" COL 2, "NO OF ERRORS" COL 20,
0157 $T5 COL 35, "PERCENT" COL 45, $PER5 COL 60
0158 SPACE 2
0160 PRT5. LINE "TOTAL ERRORS" COL 20, $T COL 35
0290 FT1. LINE %DATE COL 40            ←Page footing
```

Defines new variables and their size attributes.

Page heading

Vertical print lines

(continued)

```
0300 RETRIEVE PROBLEM-REPORT FROM DBRSDS1
0301 WHERE SPR-CHAR = "2"
0310 IF TEST-PER = "D" LET $T1 = $51 + 1 THEN GO TO FIN.
0320 IF TEST-PER = "V" LET $T2 = $T2 + 1 THEN GO TO FIN.
0340 IF TEST-PER = "I" LET $T3 = $T3 + 1 THEN GO TO FIN.
0350 IF TEST-PER = "A" LET $T4 = $T4 + 1 THEN GO TO FIN.
0355 IF TEST-PER = "0" LET $T5 = $T5 + 1 THEN GO TO FIN.
0440 FIN. LET $T = $T + 1
0450 WRAP-UP
0470 LET $PER2 = $T2 / $5 * 100.
0480 LET $PER1 = $T1 / $T * 100.
0490 LET $PER3 = $T3 / $T * 100.
0500 LET $PER4 = $T4 / $T * 100.
0505 LET $PER5 = $T5 / $T * 100.
0510 PRINT PRT
0520 PRINT PRT2
0530 PRINT PRT3
0T40PRINT PRT4
0545 PRINT PRT4A
0550 PRINT PRT5
0560END
```

Computations

Print report lines.

The result printed on the terminal is shown below.

PROJECT 3

ERRORS BY TEST-PER

DEVELOPMENT	NO OF ERRORS	0000	PERCENT	00.00
VALIDATION	NO OF ERRORS	0000	PERCENT	00.00
INTEGRATION	NO OF ERRORS	1984	PERCENT	91.64
ACCEPTANCE	NO OF ERRORS	0019	PERCENT	00.88
OPERATIONAL	NO OF ERRORS	0162	PERCENT	07.48
	TOTAL ERRORS	02165		

4.2 Writing Data Subsets

It is possible to create any number of subsets of the database. To demonstrate this, a data subset by a specific major error category was created using the Project 1 data. The WRITE statement was employed for this task. The WRITE statement writes data to a standard-system sequential file.

MDQS has the capability of accepting parameterized values as input to a procedure. Employing this facility for selecting the required major error category makes the procedure more universal.

This is demonstrated below.

```
0010 INVOKE ADOTRWS
0020 RETRIEVE PROBLEM-REPORT FROM DBRTRWS
0030 WHERE MAJOR-CAT = #MAJOR-CAT  <- Indicates a value will be inserted at run time.
0040 WRITE PROBLEM-REPORT TO BFCBMIS1/SHT-FILE  <- Writes sequential output file.
0050 LET $A = COUNT MAJOR-CAT  <- Counts the number of output records.
0060 WRAP-UP
0070PRINT ON FILE CNT-FILE $A  <- Prints the count on a file for printing.
0080 END
```

The run time parameter for the major error category is inserted as follows.

```
 RUNS #MAJOR-CAT = ("AA")
 .....
RUN-ID? CC
$ IDENT?  BFCBMIS1,C CURTIS,555008570052
SNUMB 7063T
7063T    GEIN EXECUTING @ 11.325
7063T-01 WAIT-PERIP @ 11.326
7063T-01 EXECUTING @ 11.328
7063T-01 INITIALIZING @ 11.327
7063T-01 RETRIEVEING SEQ    @ 11.330
7063T-01 PRG TERMIN @ 11.335
7063T OUTPUT STARTED
```

Error category AA Computational errors was chosen for this sample. Listing the count field shows the number of records written on the new file SHT-FILE. Note that this file need not be previously created.

```
LIST CNT-FILE

A
342.000000
```

The output record file, SHT-FILE, is a BCD permfile as was the input master file, DBNTRWS.

The following FORTRAN program was used to print the SHT-FILE on the terminal.

```
 LIST

0010*#RUN=(BCD)#SHT-FILE"28"
0020        DIMENSION IBUF(14)
0040   10   READ(28,END=30)IBUF
0050        WRITE(06,1000)IBUF
0051        GO TO 10
0055 1000   FORMAT(1X,14A6)
0060   30   CONTINUE
0070        STOP
0080        END


* RUN

                                MAJOR-CAT

C101   0197C101   6-05-73  V      AA040M-0332X        6-07-73BLK2  03
  2
F412   0198F412   6-05-73  V      AA040M-0306X        6-06-73BLK2  03
  1
C212   0234A203   6-07-73  V      AA050M-0362X        6-08-73BLK2  03
  1
C302   0241C302   6-07-73  V      AA020M-0412         6-13-73BLK2  03
  6
C302   0242C302   6-07-73  V      AA120M-0411X        6-13-73BLK2  03
  6
C102   0255C102   6-08-73  V      AAD20M-0351X        6-08-73BLK2  03
  0
B205   02598205   6-07-73  V      AA040M-0355X        6-08-73BLK2  03
  1
B104   02618104   6-06-732 V      AA040M-0357X        6-08-73BLK2  03
  2
F301   0303F301   6-11-73  V      AA010M-0378X        6-11-73BLK2  03
  0
F407   0350F407   6-13-731 V      AAD90M-0492  XX     6-19-73BLK2  03
  6
B105   03658105   6-13-73  V      AA030M-0415X        6-14-73BLK2  03
  1
B205   0373B204   6-14-731 V      AA040M-0459X        6-15-73BLK2  03
  1
A208   0381A208   6-14-731 V      AA120M-0616    EX   6-26-73BLK2  03
 12
C102   0415C102   6-15-731 V      AAD80M-0487X        6-15-73BLK2  03
  0
G107   0430G107   6-18-73  V      AA090M-081QX        7-06-73BLK2  03
 18
C102   0438C102   6-18-731 V      AA110M-
```

4.3 Interface to Applications Programs

MDQS procedures may reference user written programs in COBOL, FORTRAN, or GMAP. These user programs must be stored in a User Subroutine Library which is created in the Perform Subsystem. The method for this process is shown in Appendix A.

The MDQS procedure must reference the library in which the program is stored and the name of the program (Program-ID) from where the return value will come.

The following procedure references a previously written COBOL program to decode the complexity value in the Project 3 data. The actual COBOL listing is contained in Appendix A.

The MDQS procedure is as follows.

```
0020 INVOKE ADOSDS1
0030 LIBRARY BFCBMIS1/USLSDS1 <- Specifies the library where program resides.
0050 RETRIEVE MODULE FROM DBRSDS1
0060 WHERE MOD-CHAR CONTAINS "1"
0065 ONLY 25 ENTRIES
0070 PRINT ON FILE LIB-REP PROJ-ID,MOD-ID,MOD-VERSION
0071 COMPLEXITY USE CODE-ANS <- Names the program that performs the decoding.
0080 END
```

The following shows the result with the decoded value.

```
 LIST LIB-REP

PROJ-ID  MOD-ID    MOD-VERSION  COMPLEXITY
RAY01    PROG001   1E           MEDIUM
RAY01    PROG002   OK           MEDIUM
RAY01    PROG005   OC           MEDIUM
RAY01    PROG006   OB           COMPLEX
RAY01    PROG007   2K           MEDIUM
RAY01    PROG008   0B           MEDIUM
RAY01    PROG009   2J           MEDIUM
RAY01    PROG011   4D           MEDIUM
RAY01    PROG012   2G           SIMPLE
RAY01    PROG013   2D           MEDIUM
RAY01    PROG014   OC           MEDIUM
RAY01    PROG015   1B           MEDIUM
RAY01    PROG016   1C           SIMPLE
RAY01    PROG017   3C           MEDIUM
RAY01    PROG018   0A           MEDIUM
```

(continued)

```
RAY01     PROG019   0A          SIMPLE
RAY01     PROG020   2B          MEDIUM
RAY01     PROG021   2A          MEDIUM
RAY01     PROG022   OE          MEDIUM
RAY01     PROG024   1B          COMPLEX
RAY01     PROG025   4D          MEDIUM
RAY01     PROG026   2C          MEDIUM
RAY01     PROG027   2G          MEDIUM
RAY01     PROG028   3F          MEDIUM
RAY01     PROG029   2A          MEDIUM
```

4.4 Table Lookup Facility

The table lookup facility allows for the translation of the value of a data-item. The table generation as well as the linkage to the system is entered in the interactive perform subsystem. An outline of the procedure follows. For greater detail see reference 1, page C-25.

The first step in the process is to create a random file in which the table lookup process is to reside. The access function is used as follows.

```
*ACCE

FUNCTION? CF
   CATALOG STRUCTURE TO WORKING LEVEL?

   FILE NAME,SIZE(IN LLINKS),MAX SIZE,MODE? TLUTEST,20,20,R
PASSWORD?
[illegible]
   LOGICAL RECORD SIZE?
   GENERAL PERMISSIONS?
   SPECIFIC PERMISSIONS?
   ACCESS FILE?
 SUCCESSFUL.
```

Note that it is necessary to create the random file with the same maximum and minimum sizes.

The actual table lookup values are entered interactively in the Perform Subsystem. The Project 1 data item TEST-PER was chosen to demonstrate the decoding. The interactive process for entering the values is as follows. The user responses are annotated.

```
SYSTEM PERFORM
FUNCTION? BUILD              ← Indicates a table lookup is to be built.
PROG. NAME? TLU
$ IDENT?BFCBMIS1,C CURTIS,555008570052 -User's ident.
PROCEDURAL REF. NAME-ABBR-TEST ← Name used in procedure to
                                 reference this table.
NAME ON LIBRARY- TESTLU      ←Name for internal library bookkeeping.
INDEX OR LOOKUP- LOOK        ←The process will be a lookup.
INPUT SIZE- 2                ←Size of value to be decoded.
OUTPUT SIZE- 12              ←Size of output value.
CORE OR FILE- CORE           ←Table will be core contained.
TABLE SOURCE ON FILE? NO     ←Indicates interactive entry.
IN-  D
OUT- DEVELOPMENT
IN-  V
OUT- VALIDATION
IN-  A
OUT- ACCEPTANCE              } Input and output pair values.
IN-  I                       } (Input is two characters, right-
OUT- INTEGRATION             } hand justified.)
IN- OD
OUT- OPERATIONALD
IN-
DEFAULT VALUE- ERROR
TLU LIBRARY FILE- BFCBMIS1/TLUTEST ← Library which is to
                                     contain this table.
```

The following batch run is created as a result of the above interactive process. This must be run successfully before a procedure to use the table is initiated.

```
FUNCTION? LIST

10$       IDENT   BFCBMIS1,C CURTIS,555008570052
20$       FILEDIT SOURCE,OBJECT,INITIALIZE
30$       FILE    R*,A1S,20L
40$       DATA    *C,,COPY
50$       INCLUDE SOURCE
60$       SYSLD   CATALOG=DMSCT1,RELOC,MASTER
```

(continued)

```
70$       OPTION NOSETU,NOFCB
80$       LOWLOAD
90$       GMAP                                          TESTLU
100        LBL      TESTLU
110        SYMDEF   TESTLU
120        ERLK
130        ORG      *-2
140TESTLU  NULL
150        BCI      5,ABBR-TEST
160        BCI      1,TESTLU
170        ZERO     1024
180        VFD      1/1,3/4,14/2,12/2,6/0
190        VFD      1/1,3/7,14/2,12/12,6/0
200        OCT      555555555555,0
210        END
220$       EXECUTE
230$       ENDLD
240$       SYSLD    CATALOG=TESTLU,RELOC,MASTER
250$       OPTION   NOSETU,NOFCB
260$       LOWLOAD
270$       GMAP                                          TESTLU
280        LBL      TESTLU
290        TTL      ABBR-TEST    TLU ROUTINE
300        LODM     TLUMAC
310        TLU      TESTLU
           320      STRTBL   2,12
330        PAIR     ( D),(DEVELOPMENT )
340        PAIR     ( V),(VALIDATION  )
350        PAIR     ( A),(ACCEPTANCE  )
360        PAIR     ( I),(INTEGRATION )
370        PAIR     (OD),(OPERATIONALD)
380        ENDTBL
390        LOOKUP   BFCBMIS1,
400        INOPE    L,(ERROR       )
410        END
420$       EXECUTE
430$       ENDLD
440$       ENDEDIT
450$       ENDCOPY
460$       SYSEDIT
470ENDFILE/TESTLU
480$       FILE     R*,AIR,20L
490$       PRMFL    Q*,W,R,BFCBMIS1/TLUTEST
500$       ENDJOB
```

A procedure demonstrating the decoding process follows.

```
0010 INVOKE ADOTRWS
0020LIBRARY BFCBMIS1/TLUTEST
0030 RETRIEVE PROBLEM-REPORT FROM DBRTRWS
0045 SORT PROBLEM-REPORT ON YR-OPEN,MO-OPEN,DA-OPEN
0050 PRINT ON FILE TLU-LST MOD-CHANGED,SEVERITY,TEST-PER
     TLU ABBR-TEST
0060 END
```

← Indicates use of table lookup for output.

A partial list of the decoded output follows.

MOD-CHANGED	SEVERITY	TEST-PER
C101		INTEGRATION
H215	2	INTEGRATION
H211	2	INTEGRATION
D/B	1	INTEGRATION
H208		OPERATIONALD
C104	1	INTEGRATION
H231	1	INTEGRATION
E104	2	INTEGRATION
E104	2	INTEGRATION
A403	3	INTEGRATION
D108	2	INTEGRATION
D412	3	INTEGRATION
D109	2	INTEGRATION
E102	2	INTEGRATION
E109	2	OPERATIONALD
C104	4	INTEGRATION
C113	2	INTEGRATION
H215	2	INTEGRATION
A104	1	VALIDATION
F506	2	VALIDATION
A309		VALIDATION
C106		VALIDATION

APPENDIX A

SYSTEM ADMINISTRATOR'S GUIDE

The purpose of this appendix is to provide an overall view of the database administrator functions for the Baseline S/W Data System. More detailed procedures are contained in the Honeywell Manual (see reference 2).

This appendix contains a brief description of steps that must be performed to define the Historical Database, the naming conventions that have been established for the datasets, a step-by-step procedure for defining the data using one of the Baseline datasets, definition listings for all of the datasets, an overview and example of establishing a subroutine to decode values in a dataset, and examples of database restructuring and the use of the Privacy Subsystem.

5.1 Historical Database Definition

The Historical Database for the Baseline S/W Data System is made up of six datasets representing error and module information for six software development projects. Each dataset is defined as a separate database using the MDQS Definition Languages—Directory, Data, and Application.

Directory Definition. The Directory Definition Language (DIR) defines the name of the database; i.e., its database reference, and the names of the files associated with the database.

Data Definition. The Data Definition Language (DDL) defines the structure of the database as it appears on the external medium, the attributes assigned to each data item (length, date type, etc.), and the relation among elements of the data.

The Directory and Data Definition Languages are used to build a complete definition of the database called the schema.

Application Definition. The Application Definition Language (ADF) defines the records and the data items for those records that are to be accessible by MDQS procedures and defines the access path to be used to retrieve each of the records that participates in the application entry. An application entry (or subschema) is a subsetted user's view of the databases. There may be multiple application entries.

For each of these languages, a source code must be written and translated by MDQS into an object form (see Figure A-1). The job control language necessary for the translation was written interactively in the Perform Subsystem.

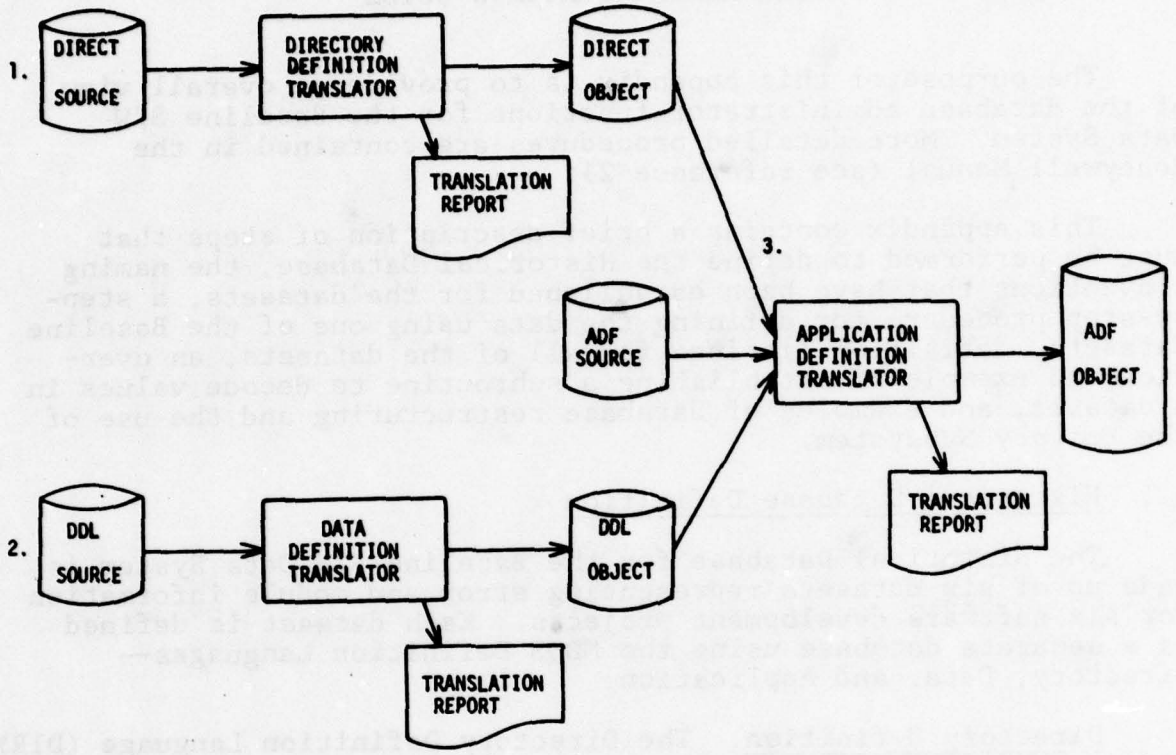


Figure A-1: Translation Flow

5.2 Naming Conventions for Datasets

The following naming convention was established to facilitate module recognition within the system. Each dataset has the following names established for it.

NAME	SYSTEM ACRONYM PREFIX	PROJECT SUFFIX
● Database Reference	DBR	Established for each project.
● Database Name	DBN	
● Directory Definition		
• Source	DIS	
• Object	DIO	
• Translation Rpt	TDI	
• JCL	JDI	
● Data Definition		Established for each project.
• Source	DDS	
• Object	DDO	
• Translation Rpt	TDD	
• JCL	JDD	
● Application Definition		
• Source	ADS	
• Object	ADO	
• Translation Rpt	TAD	
• JCL	JAD	

The Project Suffixes were established as follows.

PROJECT	SUFFIX
1	TRWS
2	B1S1
3	SDS1
4	APS1
5	SAFSI & SAFSN (two files)
6	SMTC

A form was designed and completed for each dataset to provide a naming control. Figure A-2 contains a copy of the form used for project 1.

5.3 Sample Database Definition - Project 1

The following annotated listings for Project 1 show the source code necessary for each step of database definition as well as the use of the Perform Subsystem to create the JCL to translate the code.

Step 1. Directory Definition Source Code

```
10 DATA-BASE-REFERENCE IS DBRTRWS◄Defines the database reference.
20 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNTRWS  ◄Location & name of data.
30 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSTRWS ◄Data definition source
40 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOTRWS ◄  & object.
50 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/ULSTRWS◄User library name.
60 END DBRTRWS
```

Data Set Name PROJECT 1

Data-Base-Reference (Same as Schema Name)	Data-Base Name
DBRTRWS	DBNTRWS

DEFINITION	JCL	SOURCE	OBJECT	TRANS
Directory	JDITRWS	DISTRWS	DIØTRWS	TDITRWS
Data	JDDTRWS	DDSTRWS	DDØTRWS	TDDTRWS
Application	JADTRWS	ADSTRWS	ADØTRWS	TADTRWS
User-Subr Lib		ULSTRWS	ULØTRWS	

Figure A-2: Dataset Naming Form

Step 2. Data Definition Source Code.

```
0020:MD  DBRTRWS:FILE IS SEQUENTIAL.                    -Defines file structure.
0020:01  PROBLEM-REPORT;TYPE IS "03" IN PROJ-ID         -Defines entry name.
0030::;RETRIEVAL VIA SCAN.                              -Defines retrieval method.
0060:     02:MOD-CHANGED:PIC X(7).
0070:     02:SPR-NUM PIC X(4) JUSTIFIED RIGHT.
0080:     02:MOD-SOURCE:PIC X(7).
0090:     02:DATE-OPEN.
0100:       03   MO-OPEN:PIC XX.
0110:       03   FILLER:PIC X.
0120:       03 DA-OPEN:PIC XX.
0130:       03   FILLER: PIC X.
0140:       03   YR-OPEN:PIC XX.
0150:     02:SEVERITY:PIC X.
0160:     02:TEST-PER:PIC XX JUSTIFIED RIGHT.
0170:     02:TEST-ID:PIC X(8).
0180:   02:ERROR-CAT.
0182:     03:MAJOR-CAT:PIC X(2).
0183:     03:MINOR-CAT:PIC X(3).
0192:     03:SMN-NUM:PIC X(6) JUSTIFIED RIGHT.
0195:   02:COR-TYPE.                                    }Cobol-like description
0200:     03:NEW-MOD:PIC X.                             of data.
0210:     03:DOC-UP:PIC X.
0220:     03:COM-CH:PIC X.
0230:     03:DB-CH:PIC X.
0240:     03:EXPLAN:PIC XX.
0250:     02:DATE-CLOSE.
0260:       03 MO-CLOSE:PIC XX.
0270:       03 FILLER PIC X.
0280:       03 DA-CLOSE:PIC XX.
0290:       03 FILLER:PIC X.
0300:       03 YR-CLOSE:PIC XX.
0310:     02:PROJ-VERSION:PIC X(6).
0320:   02:PROJ-ID:PIC XX.
0330:     02:CODE-CONT:PIC X.
0340:     02:DAYS-OPEN:PIC XXX JUSTIFIED RIGHT.
0341:     02:FILLER PIC X(10).
0350:98:SYSTEM DETAIL;SCAN ON PROJ-ID.                  -Defines record type.
```

Step 3. Application Definition Code.

```
0010 DATA-BASE IS DBRTRWS IN BFCBMIS1/DIOTRWS      -Defines location of
                                                    data to be used.
0030 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT -Defines entry record.
```

Step 4. JCL Creation.

The Perform Subsystem was used to create the job control language to translate the above source code. The user's responses following the dash and blank responses are carriage returns. The permfiles have no catalog names and no passwords hence a carriage return was entered.

Note that this sample shows all the JCL in one job stream but it can be done separately for each step. In the following example, all the translation reports are saved on permfile for easier verification.

```
SYSTEM ?PERFORM
FUNCTION? BUILD DBAJCL
$ IDENT?BECBMIS1,C CURTIS,555008570052
DBAJCL FUNCTION- LIST,DIRJCL,DDLJCL,ADFJCL,DFAJCL? DIRJCL

*MDQS DIRECTORY CREATION*

SOURCE DIRECTORY DEFINITION ON PERM FILE? YES
CATALOG NAME OF SOURCE DIR-
SUBCATALOG NAME OF SOURCE DIR-
FILE NAME OF SOURCE DIR- DISTRWS
PASSWD-
[redacted]
CREATE OBJECT DIR FILE? YES
LOG-ON PASSWORD-
PASSWD-
[redacted]
CATALOG NAME OF OBJECT DIR-
SUBCATALOG NAME OF OBJECT DIR-
FILE NAME OF OBJECT DIR- DIOTRWS
PASSWD-
[redacted]
MAX FILE SIZE IN LLINKS- 10
TRANSLATION REPORT TO PERM FILE? YES
CATALOG NAME OF TRANSLATION REPORT-
SUBCATALOG NAME OF TRANSLATION REPORT-
TRANSLATION REPORT FILE NAME- TDITRWS
PASSWD-
[redacted]
CREATE TRANSLATION REPORT FILE? YES
MAX FILE SIZE IN LLINKS- 10

FOLLOW WITH DDLJCL? YES


*MDQS DATA DEFINITION*

SOURCE DATA DEFINITION ON PERM FILE? YES
CATALOG NAME OF SOURCE DDL-
SUBCATALOG NAME OF SOURCE DDL-
FILE NAME OF SOURCE DDL- DDSTRWS
PASSWD-
[redacted]
CREATE OBJECT DDL FILE? YES
CATALOG NAME OF OBJECT DDL-
SUBCATALOG NAME OF OBJECT DDL-
FILE NAME OF OBJECT DDL- DDOTRWS
PASSWD-
[redacted]
MAX FILE SIZE IN LLINKS- 20
TRANSLATION REPORT TO PERM FILE? YES
CATALOG NAME OF TRANSLATION REPORT-
SUBCATALOG NAME OF TRANSLATION REPORT-
FILE NAME OF TRANSLATION REPORT- TDDTRWS
PASSWD-
[redacted]
CREATE TRANSLATION REPORT FILE? YES
MAX FILE SIZE IN LLINKS- 20

PRINT UTILITY REPORT OF OBJECT DDL? YES

FOLLOW WITH ADFJCL?
                  YES

*MDQS APPLICATION DEFINITION*

SOURCE APPLICATION DEFINITION ON PERM FILE? YES
CATALOG NAME OF SOURCE ADF-
SUBCATALOG NAME OF SOURCE ADF-
FILE NAME OF SOURCE ADF- ADSTRWS
PASSWD-
[redacted]
CREATE OBJECT ADF FILE? YES
CATALOG NAME OF OBJECT ADF-
SUBCATALOG NAME OF OBJECT ADF-
FILE NAME OF OBJECT ADF- ADOTRWS
PASSWD-
[redacted]
MAX FILE SIZE IN LLINKS- 20
TRANSLATION REPORT TO PERM FILE? YES
CATALOG NAME OF TRANSLATION REPORT-
SUBCATALOG NAME OF TRANSLATION REPORT-
FILE NAME OF TRANSLATION REPORT- TADTRWS
PASSWD-
[redacted]
CREATE TRANSLATION REPORT FILE? YES
MAX FILE SIZE IN LLINKS- 10
```

The following listing is the resultant job stream that was created by the above interactive procedure.

```
##NORM
$       IDENT   BFCBMIS1,C CURTIS,555008570052
$       FILSYS
USERID BFCBMIS1$P
FCREAT BFCBMIS1/DIOTRWS,MODE/RAND/,BLOCKS/10,10/
FCREAT BFCBMIS1/TDITRWS,MODE/SEQ/,BLOCKS/1,10/
$       PROGRAM DIRXLT,DUMP
$       LIMITS  10,10K
$       PRMFL   TR,R/W,S,BFCBMIS1/TDITRWS
$       PRMFL   DF,R/W,R,BFCBMIS1/DIOTRWS
$       DATA    I*
$       SELECTA BFCBMIS1/DISTRWS
$       FILSYS
USERID BFCBMIS1$P
FCREAT/IDS/  BFCBMIS1/DDOTRWS,BASESIZE/20/,RNG/1,20/.
             INVENTORY/NO/,BLOCKS/20/
FCREAT BFCBMIS1/TDDTRWS,MODE/SEQ/,BLOCKS/1,10/
$       PROGRAM DDLXLT,DUMP
$       LIMITS  10,27K
$       PRMFL   TR,R/W,S,BFCBMIS1/TDDTRWS
$       PRMFL   *3,R/W,R,BFCBMIS1/DDOTRWS
$       DATA    *S
$       SELECTA BFCBMIS1/DDSTRWS
$       PROGRAM QUTD
$       LIMITS  ,20K
$       PRMFL   A1,R,R,BFCBMIS1/DDOTRWS
$       DATA    I*
IDS     PRINT
$       FILSYS
USERID BFCBMIS1$P
FCREAT BFCBMIS1/ADOTRWS,MODE/RAND/,BLOCKS/1,20/
FCREAT BFCBMIS1/TADTRWS,MODE/SEQ/,BLOCKS/1,10/
$       PROGRAM ADFXLT,DUMP
$       LIMITS  10,41K
$       PRMFL   AF,R/W,R,BFCBMIS1/ADOTRWS
$       PRMFL   TR,W,S,BFCBMIS1/TADTRWS
$       DATA    I*
OBJECT ADF IS BFCBMIS1/ADOTRWS
SOURCE ADF IS BFCBMIS1/ADSTRWS
$       SELECTA BFCBMIS1/ADSTRWS
$       ENDJOB
```

5.4 Definition Listings - All Projects

The Directory, Data, and Application Definition listings for Projects 1 through 6 are contained in Figures A-3 through A-8, respectively.

DIRECTORY DEFINITION

```
*LIST DISTRWS

10 DATA-BASE-REFERENCE IS DBRTRWS
20 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNTRWS
30 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSTRWS
40 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOTRWS
50 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/ULSTRWS
60 END DBRTRWS
```

DATA DEFINITION

```
*LIST DDSTRWS

0020:MD  DBRTRWS:FILE IS SEQUENTIAL.
0020:01  PROBLEM-REPORT:TYPE IS "03" IN PROJ-ID
0030:::RETRIEVAL VIA SCAN.
0060:      02:MOD-CHANGED:PIC X(7).
0070:      02:SPR-NUM PIC X(4) JUSTIFIED RIGHT.
0080:      02:MOD-SOURCE:PIC X(7).
0090:      02:DATE-OPEN.
0100:        03   MO-OPEN:PIC XX.
0110:        03   FILLER:PIC X.
0120:        03 DA-OPEN:PIC XX.
0130:        03   FILLER: PIC X.
0140:        03   YR-OPEN:PIC XX.
0150:      02:SEVERITY:PIC X.
0160:      02:TEST-PER:PIC XX JUSTIFIED RIGHT.
0170:      02:TEST-ID:PIC X(8).
0180:    02:ERROR-CAT.
0182:      03:MAJOR-CAT:PIC X(2).
0183:      03:MINOR-CAT:PIC X(3).
0192:      03:SMN-NUM:PIC X(6) JUSTIFIED RIGHT.
0195:    02:COR-TYPE.
0200:      03:NEW-MOD:PIC X.
0210:      03:DOC-UP:PIC X.
0220:      03:COM-CH:PIC X.
0230:      03:DB-CH:PIC X.
0240:      03:EXPLAN:PIC XX.
0250:      02:DATE-CLOSE.
0260:        03 MO-CLOSE:PIC XX.
0270:        03 FILLER PIC X.
0280:        03 DA-CLOSE:PIC XX.
0290:        03 FILLER:PIC X.
0300:        03 YR-CLOSE:PIC XX.
```

(continued)

Figure A-3: Project One Definitions

DATA DEFINITION (cont'd)

```
0310$    02$PROJ-VERSION$PIC X(6).
0320$  02$PROJ-ID$PIC XX.
0330$    02$CODE-CONT$PIC X.
0340$    02$DAYS-OPEN$PIC XXX JUSTIFIED RIGHT.
0341$    02$FILLER PIC X(10).
0350$98$SYSTEM DETAIL$SCAN ON PROJ-ID.
```

APPLICATION DEFINITION

```
*LIST ADSTRWS

0010 DATA-BASE IS DBRTRWS IN BFCBMIS1/DIOTRWS
0030 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
```

Figure A-3: Project One Definitions (Cont'd)

DIRECTORY DEFINITION

```
*LIST DISB1S1

10 DATA-BASE-REFERENCE IS DBRB1S1
20 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNB1S1
30 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSB1S1
40 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOB1S1
50 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/ULSB1S1
60 END DBRB1S1
```

DATA DEFINITION

```
*LIST DDSB1S1

0020:MD  DBRB1S1;FILE IS SEQUENTIAL.
0030:01  PROBLEM-REPORT; TYPE IS "1" IN SPR-CHAR
0040::;RETRIEVAL VIA SCAN.
0050:  02:SPR-CHAR:PICTURE X.
0060:  02:SSYS-ID:PIC X.
0070:  02 :FILLER:PIC X.
0080:  02:DATE-OPEN.
0090:    03:MO-OPEN:PIC 99.
0100:    03:DA-OPEN:PIC 99.
0105:    03:YR-OPEN:PIC 99.
0110:  02:DATE-CLOSE.
0120:    03:MO-CLOSE:PIC 99.
0130:    03:DA-CLOSE:PIC 99.
0140:    03:YR-CLOSE:PIC 99.
0145:  02:ERROR-CAT.
0150:    03:MAJOR-CAT:PIC X(2).
0160:    03:MINOR-CAT:PIC X(3).
0170:  02:FILLER:PIC X.
0180:  02:CODE-CONT:PIC 9.
0190:  02:FILLER:PIC X.
0200:  02:COR-TYPE:PIC X.
0210:  02:FILLER:PIC X.
0220:  02:COR-MECH:PIC X.
0230:  02:FILLER:PIC X.
0240:  02:PHASE:PIC X.
0250:  02:FILLER:PIC X.
0260:  02:TYPE-TERM:PIC X.
0270:  02:FILLER:PIC X(2).
0280:  02:HRS-TO-DISC:PIC X(5);JUSTIFIED RIGHT.
0290:  02:FILLER:PIC X(3).
0300:  02:HHRS-TO-FIX  PIC 9(5).
0310:  02:FILLER:PIC X(3).
0320:  02:SPR-NUM:PIC X(3).
0330:  02:FILLER:PIC X(2).
0340:  02:MOD-CHANGED:PIC X(27).
0341:  02:FILLER PIC X(4).
0345:98:SYSTEM DETAIL;SCAN ON SPR-CHAR.
0350:01  MODULE;TYPE IS "2" IN MOD-CHAR
0360::;RETRIEVAL VIA SCAN.
0370:  02:MOD-CHAR:PIC X.
0390:  02:FUNC-AREA2:PIC X.
0400:  02:FILLER:PIC X.
0410:  02:MOD-ID:PIC X(4).
0420:  02:FILLER:PIC X.
```

Figure A-4: Project Two Definitions

DATA DEFINITION (cont'd)

```
0430:  02:MOD-LANGJ:PIC X.
0440:  02:MOD-SIZEJ PIC 9(5).
0450:  02:FILLER:PIC X.
0460:  02:MOD-LANGA:PIC X.
0470:  02:MOD-SIZEA PIC 9(5).
0471:  02:FILLER PIC X(63).
0475:98:SYSTEM DETAIL;SCAN ON MOD-CHAR.
0480:01  HARDWARE;TYPE IS "3" IN HARD-CHAR
0490::;RETRIEVAL VIA SCAN;
0500:  02:HARD-CHAR:PIC X.
0510:  02:FILLER:PIC X.
0520:  02:COMP-ID:PIC X(13).
0530:  02:FILLER:PIC X.
0540:  02:COMP-RATE:PIC X(7).
0550:  02:FILLER:PIC X(4).
0560:  02:COMP-OS:PIC X(13).
0561:  02:FILLER PIC X(44).
0565:98:SYSTEM DETAIL;SCAN ON HARD-CHAR.
0570:01  SOFTWARE;TYPE IS "4" IN SYS-ID
0580::;RETRIEVAL VIA SCAN.
0590:  02:SYS-ID:PIC X.
0600:  02:SSYS-ID:PIC X.
0610:  02:FILLER:PIC X(9).
0620:  02:TECH-ID:PIC X(11).
0630:  02:FILLER:PIC X.
0640:  02:SOFT-LANGJ:PIC X.
0650:  02:SOFT-SIZEJ:PIC X(5) ;JUSTIFIED RIGHT.
0660:  02:FILLER:PIC X.
0670:  02:SOFT-LANGA;PIC X.
0680:  02:SOFT-SIZEA:PIC X(5) ;JUSTIFIED RIGHT.
0685:  02:FILLER PIC X(48).
0686:98:SYSTEM DETAIL;SCAN ON SYS-ID.
0690:01  TESTING;TYPE IS "5" IN TEST-CHAR
0700::;RETRIEVAL VIA SCAN.
0710:  02:TEST-CHAR:PIC X.
0720:  02:FILLER:PIC X(2).
0730:  02:NUM-RUNS-TOT:PIC X(3) ;JUSTIFIED RIGHT.
0740:  02:FILLER:PIC X(2).
0750:  02:NUM-RUNS-OK:PIC X(3) ;JUSTIFIED RIGHT.
0760:  02:FILLER:PIC X(2).
0770:  02:AHRS-PER-TEST:PIC X(3) ;JUSTIFIED RIGHT.
0780:  02:FILLER:PIC X.
0790:  02:BLOCK;PIC X.
0800:  02:FILLER:PIC X.
0810:  02:TEST-PER:PIC X(4) ;JUSTIFIED RIGHT.
0811:  02:FILLER PIC X(61).
0815:98:SYSTEM DETAIL;SCAN ON TEST-CHAR.
```

APPLICATION DEFINITION

```
*LIST ADS8ISI

10DATA-BASE IS DBRB1SI IN BFCWIISI/DIORISI
20ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
30ENTRY MODULE IS RECORD MODULE
40ENTRY HARDWARE IS RECORD HARDWARE
50ENTRY SOFTWARE IS RECORD SOFTWARE
60ENTRY TESTING IS RECORD TESTING
```

Figure A-4: Project Two Definitions (Cont'd)

DIRECTORY DEFINITION

```
*LIST DISSDS1

10 DATA-BASE-REFERENCE IS DBRSDS1
20 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSDS1
30 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSDS1
40 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSDS1
50 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/USLSDS1
60 END DBRSDS1
```

DATA DEFINITION

```
*LIST DDSSDS1

20:00  DBRSDS1;FILE IS SEQUENTIAL.
30:01  MODULE;TYPE IS "1" IN MOD-CHAR;
40::RETRIEVAL VIA SCAN.
50:     02:MOD-CHAR:PIC X.
60:   02:PROJ-ID:PIC X(5).
70:   02:PROJ-CODE:PIC XX.
80:   02:MOD-ID:PIC X(7).
90:   02:MOD-VERSION:PIC XX.
100:   02:MOD-TYPE:PIC X.
110:   02:COMPLEXITY:PIC X.
120:   02:CONST-TYPE:PIC X.
130:   02:NUM-OCCUR-S:PIC X(5);JUSTIFIED RIGHT.
140:   02:NUM-OCCUR-O:PIC X(5);JUSTIFIED RIGHT.
150:   02:TECH-ID:PIC 9.
151:   02:FILLER PIC X(53).
160:98:SYSTEM DETAIL;SCAN ON MOD-CHAR.
170:01  PROBLEM-REPORT;TYPE IS "2" IN SPR-CHAR;
180:.RETRIEVAL VIA SCAN.
190:   02:SPR-CHAR:PIC X.
200:   02:PROJ-ID:PIC X(5).
210:   02:PROJ-CODE:PIC XX.
220:   02:SPR-NUM:PIC X(4);JUSTIFIED RIGHT.
230:   02:MOD-CHANGED:PIC X(7).
240:   02:MOD-CH-VERS:PIC XX.
250:   02:DATE-OPEN.
260:     03:MO-OPEN:PIC XX.
270:     03:FILLER: PIC X.
280:     03:DA-OPEN:PIC XX.
290:     03:FILLER:PIC X.
300:     03:YR-OPEN:PIC XX.
310:   02:TYPE-TERM:PIC X.
320:   02:SEVERITY:PIC X.
330:   02:TEST-PER:PIC X.
340:   02:ERROR-CAT.
350:     03:MAJOR-CAT:PIC XX.
360:     03:MINOR-CAT:PIC XXX.
370:   02:SMN-NUM:PIC X(4).
380:   02:COR-TYPE:PIC X(5).
390:   02:DATE-CLOSE.
400:     03:MO-CLOSE:PIC XX.
410:     03:FILLER:PIC X.
420:     03:DA-CLOSE:PIC XX.
430:     03:FILLER:PIC X.
440:     03:YR-CLOSE:PIC XX.
450:   02:DAYS-OPEN:PIC XXX;JUSTIFIED RIGHT.
451:   02:FILLER PIC X(27).
460:98:SYSTEM DETAIL; SCAN ON SPR-CHAR.
```

APPLICATION DEFINITION

```
*LIST ADSSDS1

10 DATA-BASE IS DBRSDS1 IN BFCBMIS1/DIOSDS1
20 ENTRY MODULE IS RECORD MODULE
30 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
```

Figure A-5: Project Three Definitions

DIRECTORY DEFINITION

```
*LIST DISAPS1
010 DATA-BASE-REFERENCE IS DBRAPS1
020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNAPOS
025   ON TAPE 44284
030 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSAPS1
040 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOAPS1
050 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/USLAPS1
060 END DBRAPS1
```

DATA DEFINITION

```
*LIST DDSAPS1
020:MD  DBRAPS1;FILE IS SEQUENTIAL.
030:01  PROBLEM-REPORT;RETRIEVAL VIA SCAN.
040:   02:SMN-NUM:PIC X(4).
050:   02:DATE-CLOSE.
060:     03:YR-CLOSE:PIC XX.
070:     03:MO-CLOSE:PIC XX.
080:     03:DA-CLOSE:PIC XX.
090:   02:SSYS-TYPE:PIC X.
100:   02:SSYS-VERSION:PIC XXX.
110:   02:SPR-NUM:PIC X(6).
120:   02:FUNC-AREA:PIC X.
130:   02:ERROR-CAT.
140:     03:MAJOR-CAT:PIC X.
150:     03:MINOR-CAT:PIC XXX.
160:   02:ERROR-DESC:PIC X(50).
170:   02:SSYS-ID.
180:     03:SYS-ID:PIC X.
190:     03:SYS-VERSION:PIC XX.
200:   02:PHASE:PIC X.
201:   02:FILLER PIC X(5).
210:98:SYSTEM DETAIL;SCAN ON SMN-NUM.
```

APPLICATION DEFINITION

```
*LIST ADSAPS1
010 DATA-BASE IS DBRAPS1 IN BFCBMIS1/DIOAPS1
020 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
```

Figure A-6: Project Four Definitions

DIRECTORY DEFINITION

```
*LIST DISSAFS1
0010 DATA-BASE-REFERENCE IS DBRSAF61
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSAFS1
0030 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSAFS1
0040 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSAFS1
0050 END DBRSAFS1
0060 DATA-BASE-REFERENCE IS DBRSAFSN
0070 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSAFSN
0080 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSAFSN
0090 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSAFSN
0100 END DBRSAFSN
```

DATA DEFINITION

```
*LIST DDSSAFS1
0120:MD  DBRSAFS1;FILE IS SEQUENTIAL.
0130:01  PROBLEM-REPORT;RETRIEVAL VIA SCAN.
0140:    02  SPR-NUM PIC X(7).
0150:    02  FILLER PIC X.
0160:    02  DATE-OPEN.
0170:    03  MO-OPEN PIC 99.
0180:      03  FILLER PIC X.
0190:      03  DA-OPEN PIC 99.
0200:      03  FILLER PIC X.
0210:      03  YR-OPEN PIC 99.
0220:    02  DATE-CLOSE.
0230:      03  MO-CLOSE PIC 99.
0240:      03  FILLER PIC X.
0250:      03  DA-CLOSE PIC 99.
0260:      03 FILLER PIC X.
0270:      03  YR-CLOSE PIC 99.
0280:    02  PHASE PIC X(12).
0290:    02  FILLER PIC XXX.
0300:    02  MOD-CHANGED.
0305:      03  SSYS-ID PIC X.
0306:      03  SSYS-VERSION PIC X(7).
0310:    02  COR-TYPE PIC X(13).
0320:    02  FILLER PIC XX.
0330:    02  ERROR-CAT.
0340:      03  MAJOR-CAT PIC XX.
0350:      03  MINOR-CAT  PIC XXX.
0351:    02:FILLER PIC X(17).
0360:98 SYSTEM DETAIL; SCAN ON MOD-CHANGED.
```

```
*LIST DDSSAFSN
0380:MD  DBRSAFSN;FILE IS SEQUENTIAL.
0390:01  MODULE;RETRIEVAL VIA SCAN.
0410:    02  MOD-ID.
0411:      03  SSYS-ID PIC X.
0412:      03  SSYS-VERSION PIC X(7).
0430:    02  NUM-OCCUR PIC 9(6).
0450:    02  CONST-TYPE PIC X(7).
0470:    02  TECH-ID  PIC  X(12).
0480:98:SYSTEM DETAIL;SCAN ON MOD-ID.
```

APPLICATION DEFINITION

```
*LIST ADSSAFS1
0490 DATA-BASE IS DBRSAFS1 IN BFCBMIS1/DIOSAFS1
0500ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
0510 DATA-BASE IS DBRSAFSN IN BFCBMIS1/DIOSAFS1
0530 ENTRY MODULE IS RECORD MODULE
```

Figure A-7: Project Five Definitions

DIRECTORY DEFINITION

```
*LIST DISSMTC
0010 DATA-BASE-REFERENCE IS DBRSMTC
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSMTC
0030 ON TAPE 44916
0040 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSMTC
0050 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSMTC
0060 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/TLUSMTC
0070 END DBRSMTC
```

DATA DEFINITION

```
*LIST DDSSMTC
0090:MD  DBRSMTC;FILE IS SEQUENTIAL.
0100:01  PROBLEM-REPORT;RETRIEVAL VIA SCAN.
0110:02:SSYS-ID PIC XXX.
0120:02:MOD-ID PIC X(16).
0130:02:DATE-RUN.
0140:03:MO-RUN  PIC XX.
0150:03:DAY-RUN  PIC XX.
0160:03:YR-RUN PIC X.
0165:02:TME PIC X(4).
0170:02: STRESS-MEAS PIC X(6).
0180:02:TEST-RESULT PIC X.
0190:02:WORK-CAT PIC X.
0200:02:TEST-PER PIC X.
0210:02:ACT-CAT PIC X.
0220:02:NUM-CHANGED PIC X.
0230:02:SEVERITY PIC X.
0240:02:ERROR-CAT.
0250:03:MAJOR-CAT PIC X.
0260:03:MINOR-CAT PIC X.
0270:02:NUM-ERR PIC 9.
0280:02:FILLER PIC X(5).
0290:98;SYSTEM DETAIL;SCAN ON SEVERITY.
```

APPLICATION DEFINITION

```
*LIST ADSSMTC
0110 DATA-BASE IS DBRSMTC IN BFCBMIS1/DIOSMTC
0130 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
```

Figure A-8: Project Six Definitions

5.5 A Decoding Subroutine Example

MDQS allows the user to write subroutines in COBOL, FORTRAN or GMAP that can be called in an MDQS procedure. The following is an example of a COBOL subroutine used to decode values in a dataset for output.

```
0030$:COBOL:LS1OU,ON6,NCOMDK,NDECK
0040:IDENTIFICATION DIVISION.
0050:PROGRAM-ID. CODE-ANS.
0060:AUTHOR. CC.
0070:DATE-WRITTEN.
0080:REMARKS. SUBROUTINE TEST.
0090:ENVIRONMENT DIVISION.
0100:SOURCE-COMPUTER.
0110:OBJECT-COMPUTER.
0120:DATA DIVISION.
0130:WORKING-STORAGE SECTION.
0140:77  IN-CODE PIC X.
0150:77  IN-SCHAR PIC 9(6) COMP-1.
0160:77  IN-NCHAR PIC 9(6) COMP-1.
0170:77  OUT-VAL PIC X(7).
0180:77  OUT-SPOS PIC 9(6) COMP-1.
0190:77  OUT-STATUS PIC 9(6) VALUE 0.
0200:77  IN-DESC PIC 9(6).
0210:01  COMPLEX-TABLE.
0220::02  COMPLEX-DATA.
0230::   03 CODE-01 PIC X VALUE "S".
0240::   03 VAL-01 PIC X(7) VALUE "SIMPLE ".
0250::   03 CODE-02 PIC X VALUE "M".
0260::   03 VAL-2 PIC X(7) VALUE "MEDIUM ".
0270::   03 CODE-03 PIC X VALUE "C".
0280::   03 VAL-03 PIC X(7) VALUE "COMPLEX".
0290::02 COMPLEX-VAL REDEFINES COMPLEX-DATA
0300::    OCCURS 3 TIMES INDEXED BY II.
0310::   03 CODE PIC X.
0320::   03 VAL PIC X(7).
0330:PROCEDURE DIVISION.
0340:START.
0350::ENTER LINKAGE MODE.
0360::ENTRY POINT IS VALUES
0370::USING IN-CODE,IN-SCHAR,IN-NCHAR
0380::GIVING OUT-VAL,OUT-SPOS,OUT-STATUS.
0390::ENTER COBOL.
0400::   SET II TO 1.
0420::   SEARCH COMPLEX-VAL VARYING II
0430::     WHEN IN-CODE EQUALS CODE (II)
0440:::MOVE VAL (II) TO OUT-VAL AND
0450:::GO TO RET-ARGS.
0460::   MOVE "BAD COD" TO OUT-VAL.
0470::RET-ARGS.
0480::   EXIT VALUES.
0500::END PROGRAM.
0510$:ENDJOB
```

This subroutine must be compiled with the indicated options. The resulting compiler output is then used to create the linkage to MDQS. This is done interactively in the Perform Subsystem. For a detailed description see DD92, Appendix C.

The output of this interactive process is as follows.

```
.*RELE VALUES-C
FILE RELEASED-VALUES-C
*LIST TESTLIB
028-READ LINKED FILES ONLY WITH THIS COMMAND
SYSTEM ?CARD NEW
READY
.*LIST JLIBSDS1

10$       IDENT    BFCBMIS1,C CURTIS,555008570052
20$       FILEDIT  INITIALIZE,SOURCE,OBJECT
30$       FILE     K*,A1R,10L
40$       FILE     R*,A2C,10L
50$       DATA     *C,,COPY
60$       INCLUDE  SOURCE
70$       SYSLD    CATALOG=DMSCT1,RELOC,MASTER
80$       OPTION   NOSETU,NOFCB
90$       LOWLOAD
100$       GMAP     COMDK
110        SYMDEF   VALUES
120        LBL      VALUES
130        TTL      CATALOG=VALUES
140        ERLK
150        ORG *-2
160VALUES  NULL
170F       BOOL 1
180S       BOOL 0
190        TTLS     CODE-ANS
200        REM
210        BCI      5,CODE-ANS
220        BCI      1,VALUES
230        VFD      1/0,17/4056,018/22
240        VFD      1/1,1/S,1/0,1/0,12/01,2/6,12/0001,6/000
250        VFD      1/1,1/S,1/1,1/1,12/01,2/6,12/0007,6/000
260        OCT      555555555555
270        END
280$       EXECUTE
290$       ENDLD
300$       SYSLD    CATALOG=VALUES,RELOC,MASTER
310$       OPTION   NOSETU,NOFCB
320$       LOWLOAD
330$       SELECTA  BFCBMIS1/COMPLE
340$       EXECUTE
350$       ENDLD
360$       ENDEDIT
370$       ENDCOPY
380$       SYSEDIT  INITIALIZE
390$       FILE     R*,A2R,10L
400$       PRMFL    Q*,W,R,BFCBMIS1/USLSDS1
410$       ENDJOB
```

When the previous job is run successfully, the linkage is established and the user can then reference the program in the manner indicated in Section 4.3 of this report.

5.6 Database Restructuring

MDQS has the capability to allow the database Administrator to restructure the database; i.e., modify, rearrange, add or delete certain elements of the data.

The task is accomplished in three phases. They are:

1. Data Definition Restructure
2. COBOL Translation
3. Data Restructuring

Phase 1. Data Definition Restructure. The Project 5 module data was chosen to demonstrate this function due to the fact that it contains many unused character positions,* thus wasting permfile space. A new data definition was written eliminating the filler.

```
0380:MD  DBRSAFSN;FILE IS SEQUENTIAL.
0390:01  MODULE;RETRIEVAL VIA SCAN.
0410:  02  MOD-ID.
0411:    03  SSYS-ID PIC X.
0412:    03  SSYS-VERSION PIC X(7).
0430:  02  NUM-OCCUR PIC 9(6).
0450:  02  CONST-TYPE PIC X(7).
0470:  02  TECH-ID  PIC  X(12).
0480:98:SYSTEM DETAIL;SCAN ON MOD-ID.
READY
```

This was translated in the manner previously described in Step 4 of this appendix yielding the usual object module.

A sequential permfile was then created using access to contain the new database. A catalogue of its current attributes follows.

```
FILE NAME-DBNSAFSN
ORIGINATOR-BFCRMIS1
DATE CREATED-110277
DATE CHANGED-110277(11.400)
LAST DATE ACCESSED-010578
NUMBER OF ACCESSES-9
MAX FILE SIZE-61 LLINKS
CURRENT FILE SIZE-61 LLINKS
FILE TYPE-LINKED
DEVICE-DP3
GENERAL PERMISSIONS-R,E
SPECIFIC PERMISSIONS-NONE
```

*This is the result of the Multics to GCOS transfer constraints.

Phase 2. COBOL Translation. The Perform Subsystem was used to generate the JCL necessary for the COBOL translation. See Section 6 of DD94, Revision 1, for details. The following job stream resulted.

```
$       IDENT   BFCBMIS1,C CURTIS,555008570052
$       PROGRAM REST
$       LIMITS  ,28K
$       DATA    F1,,COPY
$       SELECTA BFCBMIS1/DDSSAFS2
$       ENDCOPY
$       DATA    F2,,COPY
$       SELECTA BFCBMIS1/DDSSAFSN
$       ENDCOPY
$       FILE    F3,X1C,30L
$       COBOL   NLSTOU,NDECK
$       FILE    S*,X1R,30L
$       EXECUTE DUMP
$       LIMITS  ,32K
$       PRMFL   F4,R,S,BFCBMIS1/DBNSAFS2
$       PRMFL   F5,R/W,S,BFCBMIS1/DBNSAFSN
$       ENDJOB
```

Phase 3. Data Restructuring. The execution of the above JCL caused the actual Data Restructuring to take place. The Directory Definition and Application Definition were updated to reflect the new database. Its name was changed from DBNSAFS2 to DBNSAFSN and the new Data Definition files to DDSSAFSN, DDOSAFSN. These new files are listed below.

```
0010 DATA-BASE-REFERENCE IS DBRSAFS1
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSAFS1
0030 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSAFS1
0040 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSAFS1
0050 END DBRSAFS1
0060 DATA-BASE-REFERENCE IS DBRSAFSN
0070 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSAFSN
0080 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSAFSN
0090 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSAFSN
0100 END DBRSAFSN

READY

0490 DATA-BASE IS DBRSAFS1 IN BFCBMIS1/DIOSAFS1
0500ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
0510 DATA-BASE IS DBRSAFSN IN BFCBMIS1/DIOSAFS1
0530 ENTRY MODULE IS RECORD MODULE

READY
```

This database restructuring resulted in a permfile savings of approximately 90 little links.

5.7 Privacy Subsystem

The Privacy Subsystem provides protection from MDQS procedures accessing elements of data that have been specified as locked. The Privacy Subsystem is used to verify that a particular USERID has permission to obtain the locked portion of data. The Privacy Subsystem compares the locks and keys and if a match is found permission is granted to the procedure. If a match is not found a Privacy Breach is returned.

To enable the Privacy Subsystem the database administrator must:

1. Assign the locks and keys
2. Build and maintain the Privacy File

To demonstrate this capability, a lock (named LCK1) was placed on the hardware records in the Project 2 data. The lock on the data is set in the data definition as follows (only the hardware record, the portion changed, is shown).

```
0480:01  HARDWARE;TYPE IS "3" IN HARD-CHAR
0490::;RETRIEVAL VIA SCAN;
0495::PRIVACY LOCK FOR READ,WRITE IS "LCK1".
0500:  02:HARD-CHAR:PIC X.
0510:  02:FILLER:PIC X.
0520:  02:COMP-ID:PIC X(13).
0530:  02:FILLER:PIC X.
0540:  02:COMP-RATE:PIC X(7).
0550:  02:FILLER:PIC X(4).
0560:  02:COMP-OS:PIC X(13).
0561:  02:FILLER PIC X(44).
0565:98:SYSTEM DETAIL;SCAN ON HARD-CHAR.
```

The Directory Definition must be updated to reflect the Privacy file which will be created.

```
10 DATA-BASE-REFERENCE IS DBRB1S1
20 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNB1S1
30 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSB1S1
40 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOB1S1
50 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/ULSB1S1
55 PRIVACY-FILE IS BFCBMIS1/PRIVACY
60 END DBRB1S1
```

The Application Definition needs no source changes. The three files DISB1S1, DDSB1S1, and ADSB1S1 are then retranslated as in Step 4 of this appendix.

The Privacy file is created using Access. It must be random, have at least three little links of space and have general read permission granted. A catalogue of this file follows.

```
FILE NAME-PRIVACY
ORIGINATOR-BFCBMIS1
DATE CREATED-111577
DATE CHANGED-111577(10.435)
LAST DATE ACCESSED-111677
NUMBER OF ACCESSES-6
MAX FILE SIZE-10 LLINKS
CURRENT FILE SIZE-10 LLINKS
FILE TYPE-RANDOM
DEVICE-DP4
GENERAL PERMISSIONS-4,3
SPECIFIC PERMISSIONS-NONE
```

The data is entered into the Privacy file by the interactive use of the MDQS PRIV command as illustrated below.

```
*PRIV
PRIVACY FILE ? BFCBMIS1/PRIVACY
USERIDS - BFCBMIS1
ROSTERS FOR bfcbmis1 - ROST1
KEYS FOR rost1 - LCK1

FUNCTION - DONE
```

The following is an example of accessing this data using Conversational MDQS.

```
*CMDQ
APPLICATION FILE IS - ADOB1S1
DATABASE REFERENCE IS DBRB1S1
FUNCTION - BROW
ENTRY IS -HARDWARE
DATA ITEMS TO BE DISPLAYED - ALL
DISPLAY EVEN IF BLANK OR ZERO ? YES
PAUSE BETWEEN ENTRY INSTANCES ?
DATA ITEMS TO BE SUMMED OR COUNTED -
SELECTION CRITERIA -
LIMIT # SETS OF DATA TO -
SEQUENCE DATA ON -
 .....
```

(continued)

```
RUN-ID?  CC
$ IDENT?  BFCBMIS1,C CURTIS,555008570052
SNUMB 2159T

2159T

HARD-CHAR= 3
COMP-ID= SKC2070
COMP-RATE= 500KOPS
COMP-OS= NOT APPLICABL

HARD-CHAR= 3
COMP-ID= HARRIS 6024/5
COMP-RATE= 500KOPS
COMP-OS= NOT APPLICABL

ACTIVITY TERMINATED

FUNCTION - DONE
```

Note that the initial attempt to access this data was unsuccessful and resulted in an I/O Busy Error on the Privacy file. It was necessary for the facility personnel to "unbusy" the file before successful access could be accomplished.

APPENDIX B

SUMMARY DATABASE

This appendix contains a description of the Summary Database that was generated from the Historical Database and data contained in the final reports for the six datasets. This Summary Database was designed so that queries could be formulated across projects. Data summary forms were designed and used to record project descriptive information and to specify summarization requirements.

Included in Appendix B is a discussion of each Data Summary Form, the MDQS Definitions for defining the database, and sample queries.

6.1 Data Summarization

Data summary forms were developed to record information from the technical reports for the six datasets in the Historical Database and to provide summarization requirements to convert the data from the datasets into the format required for the Summary Database. Each form contains eight fields that provide a basis for defining a unique key for each record occurrence within the Summary Database. This key identifies the applicable project, system, functional group, and module that applies to the component information recorded. Also included in this key is information concerning the level of summarization and the record type which indicates the format of the data.

In addition to the key data, the following information is recorded on each form.

Component (see Figure B-1). Component name, type, and description; developer, contract number, and data source; the number of systems, functional groups and modules; contract type and standards applied; the purpose of the data collection and the procedures used; the priorities and constraints of the product development.

Technology (see Figure B-2). The phase, reporting level and the applicable dates; the technology utilized, the name of the tool used, and the percentage of usage.

Instructions (see Figure B-3). The phase, reporting level and the applicable dates; the programming language used; the number of source instructions, object words, and percent of usage; complexity type and measure; and the mode of construction.

DATA SUMMARY FORM - COMPONENT

PROJECT ID _____ (6) / SYSTEM ID _____ (4) / FUNC GP ID _____ (5) / MODULE ID _____ (9)

SUMMARIZATION LEVEL

P	Project	F	Funct Gp
S	System	M	Module

VERSION ID _____ (7)

RECORD TYPE 010 (3)

RECORD NUMBER _____ (3)

NAME _____ (12) TYPE _____ (3) DESCRIPTION _____ (28)

DEVELOPER _____ (11) CONTRACT NUM _____ (24)

DATA DATE (Initial) _____ (6) DATA DATE (Current) _____ (6)

DATA SOURCE (2)

01 RADC Purchased	[ ]	[ ]
[ ]	[ ]	[ ]

NUMBER SYSTEMS _____ (4) NUMBER FUNC GR _____ (4) NUMBER MODULES _____ (4)

CONTRACT TYPE (5 Max)

PRIM	CPFF	FFP
SUB	CDIF	FPIF
CO	COST	
INT	TM	

(4)

STANDARDS (5 Max)

AF	DOD	OTHER	
3101	5010	8506	
483	4120	5233	
3751	490	4130	
8003	481	8040	
	100A	700	

(4)

COLLECTION PURPOSE (5 Max)

CO	Contract-During
CA	Contract-After
VI	Visibility
EP	Evaluate Pers.
EA	Evaluate App.

SM	Schedule Monitoring
CM	Cost Monitoring
QC	Quality Control

(2)

COLLECTION PROCEDURE (Max 5)

CM	Configuration Management
LI	Automated Library
IN	Internal Manual

(2)

PRIORITIES (5 Max)

PR	Processing Speed
CO	Core Utilization
SC	Schedule
CS	Cost
QU	Quality

(2)

CONSTRAINTS (5 Max)

HRDW	Hardware Limitations
LISC	Limited Schedule
LICM	Limited Computer Accessibility
LIFN	Limited Funding
LIST	Limited Staffing
LIMS	Limited Management Support
TRCM	Target Computer Different
UNST	Unstable Requirements
NEWA	New Application Area

(4)

DOCUMENT ACCESSION NUMBERS (Max 5) _____ (5) _____ _____ _____ _____

Figure B-1: Component Data Summary Form

DATA SUMMARY FORM - TECHNOLOGY

PROJECT ID _____ (6) / SYSTEM ID _____ (4) / FUNC GP ID _____ (5) / MODULE ID _____ (9)

SUMMARIZATION LEVEL

P	Project	F	Funct. Gp
S	System	M	Module
I			

VERSION ID _____ (7) RECORD TYPE 020 (3) RECORD NUMBER _____ (3)

PHASE

C	Conceptual
R	Requirements
D	Design
I	Implementation
T	Test
O	Operation

(2)

REPORTING LEVEL

MO	Monthly
PH	Phase
T	Total

(2)

BEGIN DATE _____ (6)

END DATE _____ (6)

TECHNOLOGY

CPT	Chief Programmer Team	PSL	Program Support Library	HOL	High Order Language
AUDT	Automated Design Tools	SIM	Simulation	PCOM	Pre-compiler
AURT	Automated Requirements Tools	STPR	Structured Programming	RUSE	Re-usable Code
HIPO	HIPO Design Aid	WALK	Walk-throughs	ASSP	Assertion Proofs
PDL	Process Design Language	CRPF	Critical Piece First	CSTA	Code Standards Auditor
STCH	Structure Charts	DBAN	Data Base Analyzer	COMP	Compatability Checker
TDD	Top-down Development	DDIC	Data Dictionary	INTT	Independent Test Team
MODE	Modular Decomposition	DOCG	Documentation Generator	PFAN	Program Flow Analyzer
___	_____	___	_____	___	_____
___	_____	___	_____	___	_____

TOOL NAME _____ (5) PERCENT OF USAGE _____ (2)

Figure B-2: Technology Data Summary Form

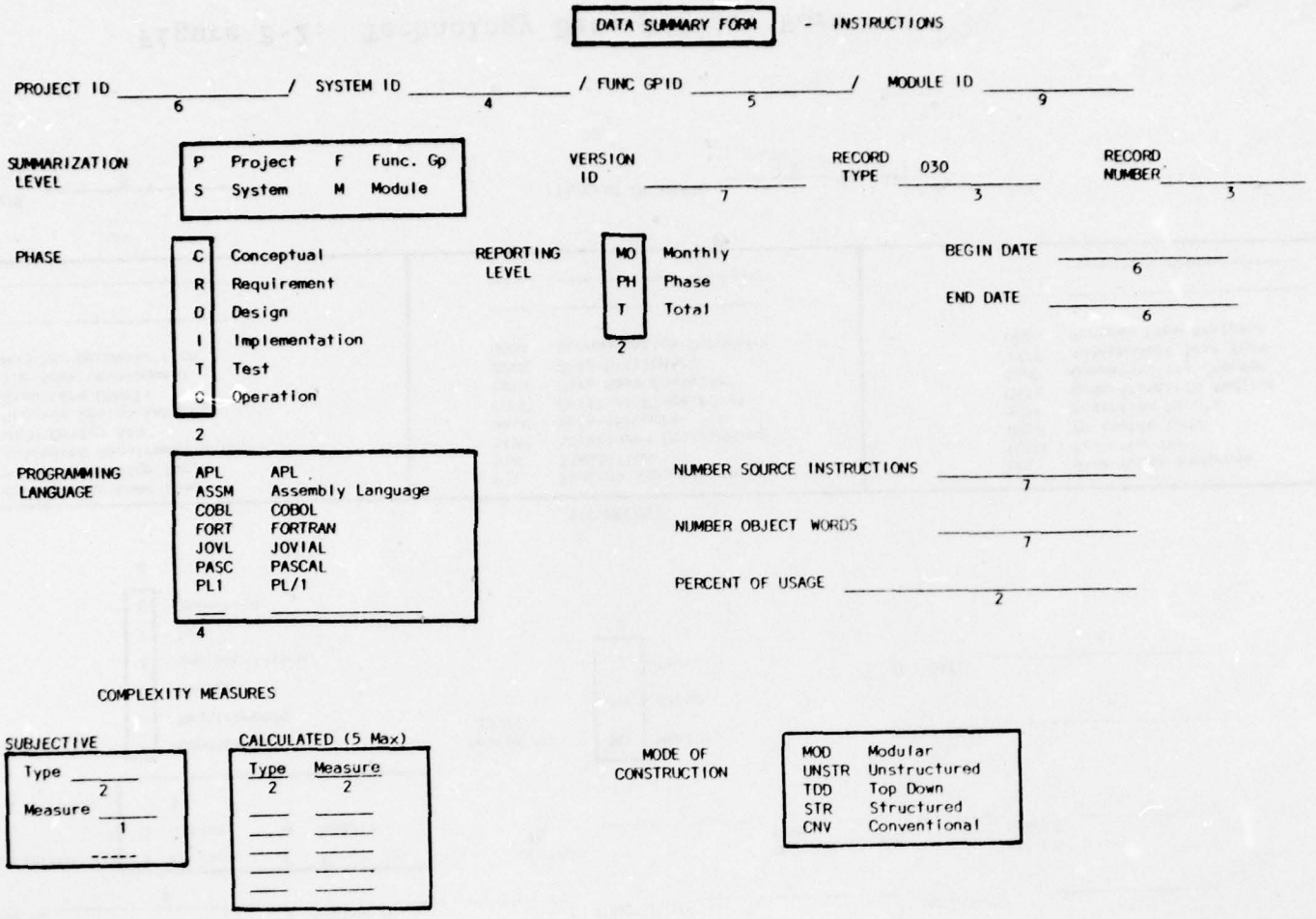


Figure B-3: Instructions Data Summary Form

Errors (see Figure B-4). The test period, reporting level and the applicable dates; the error category type, the error category, and the number of errors.

Corrections (see Figure B-5). The test period, reporting level and the applicable dates; the correction type, the average number of days open, and the number of errors.

Component-Module (see Figure B-6). This form is used to establish the key in a concise manner for any of the record types.

6.2 MDQS Definitions

Figure B-7 illustrates the entries within the Summary Database and contains the entry name, the record type designator (010-090) and the name of the MDQS chain. This database was defined using the MDQS Directory, Data and Application Definition Languages as an Index-Sequential File (see Figure B-8).

Illustrated below is the MDQS Procedure Language results for querying the Summary Database at the component level and printing six fields of this entry.

```
010 INVOKE ADOSUM
020 RETRIEVE COMPONENT FROM DBRSUM
030 WHERE SUM-LEVEL EQ "P"
040 PRINT ON FILE SUMOUT
050 PROJECT-ID,COMP-TYPE,NUM-SYS,NUM-FG.NUM-MOD
055 ,STANDARDS
060 END
```

PROJECT-ID	COMP-TYPE	NUM-SYS	NUM-FG	NUM-MOD	STANDARDS
P00501	013	8	25	249	6147
P00502	003	2	7	69	
P00503	014	1	1	109	
P00504	015	16	0	0	
P00505	014	0	0	2413	
P00506	016	2	45	0	483MOD

A sequential transaction database using the same entries and data names as in the Index-sequential Summary Database was defined using the MDQS Languages. Below is an illustration of a query that retrieves and prints data from the COMPONENT, INSTRUCTIONS, ERRORS, and CORRECTIONS entities.

```
0010 INVOKE ADOTRAN
0020 REPORT A ON FILE INST-ERR
0030PAGE HEADING IS HD1.
```

(continued)

```
0040 HD1.  LINE "MOD ID" COL 1,
0050   "INSTR TYPE" COL 10,  "# SOURCE" COL 20,
0060   "# ERRORS" COL 29,"ERR CAT" COL 40,
0065   "#CORRECTIONS" COL 50
0070P1.  LINE MOD-ID COL 1 GROUP INDICATE 1,
0080   INST-TYPE COL 12, NUM-INST-S COL 20
0090 P2.  LINE NUM-ERRORS COL 30, MAJOR-CAT COL 43
0100 P3.  LINE NUM-COR COL 53
0110 L1.  RETRIEVE COMPONENT WHERE PROJECT-ID EQ "P00502"
0120   AND SUM-LEVEL EQ "M"
0130 WHEN L1.
0140     RETRIEVE INSTRUCTIONS
0150  PRINT P1.
0160  WHEN L1.
0170     RETRIEVE ERRORS WHERE MINOR-CAT EQ "TOT"
0190  PRINT P2.
0200  WHEN L1.
0210     RETRIEVE CORRECTIONS WHERE COR-TYPE EQ "T"
0220  PRINT P3.
0230END
```

MOD ID	INSTR TYPE	# SOURCE	# ERRORS	ERR CAT	#CORRECTIONS
AAI	ASSM	2282			
			2	BB	
			4	DD	
			1	KK	
			1	RR	
					8
AI	ASSM	2830			
	JOVL	7223			
			1	AA	
			44	BB	
			3	CC	
			20	DD	
			4	FF	
			4	GG	
			1	JJ	
			1	KK	
			3	LL	
			18	MM	
			4	NN	
			2	PP	
			7	RR	
			1	SS	
			3	TT	
					116
AM	JOVL	6897			
	ASSM	600			
			6	AA	
			77	BB	
			36	DD	
			1	FF	
			7	GG	

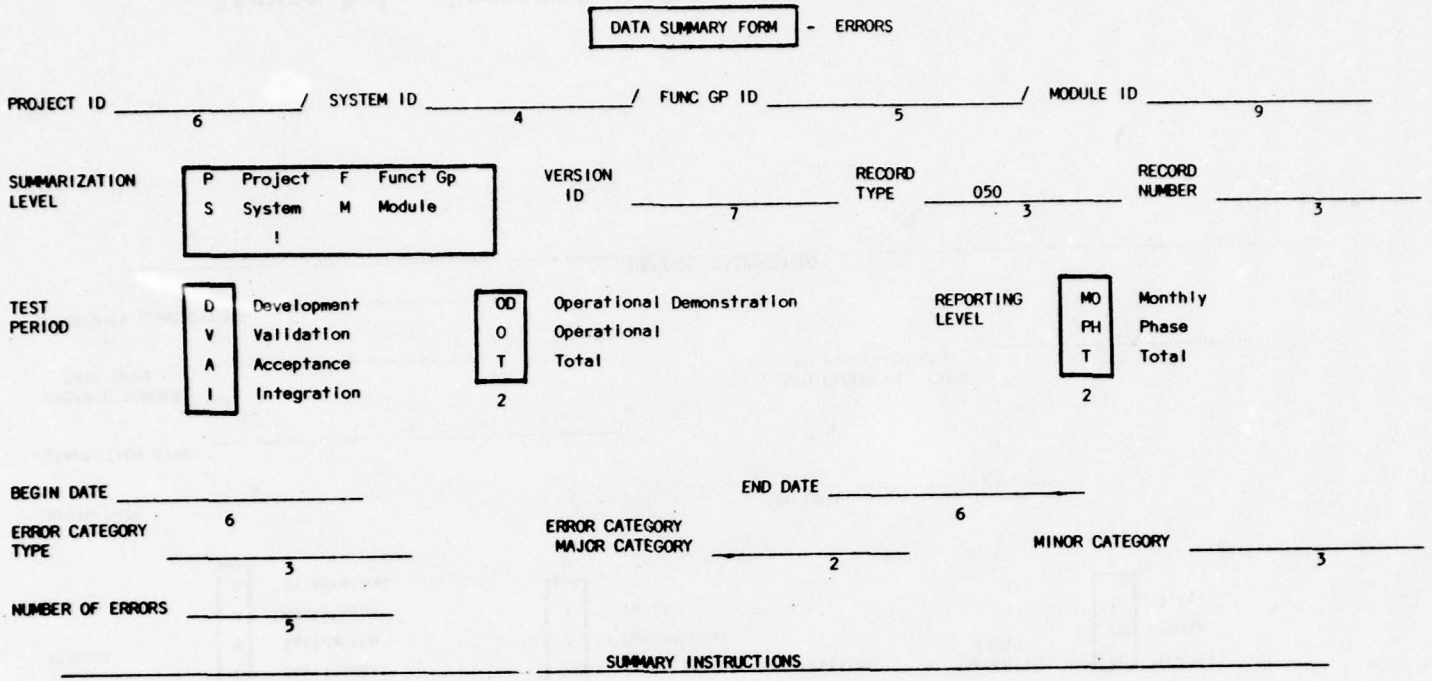


Figure B-4: Errors Data Summary Form

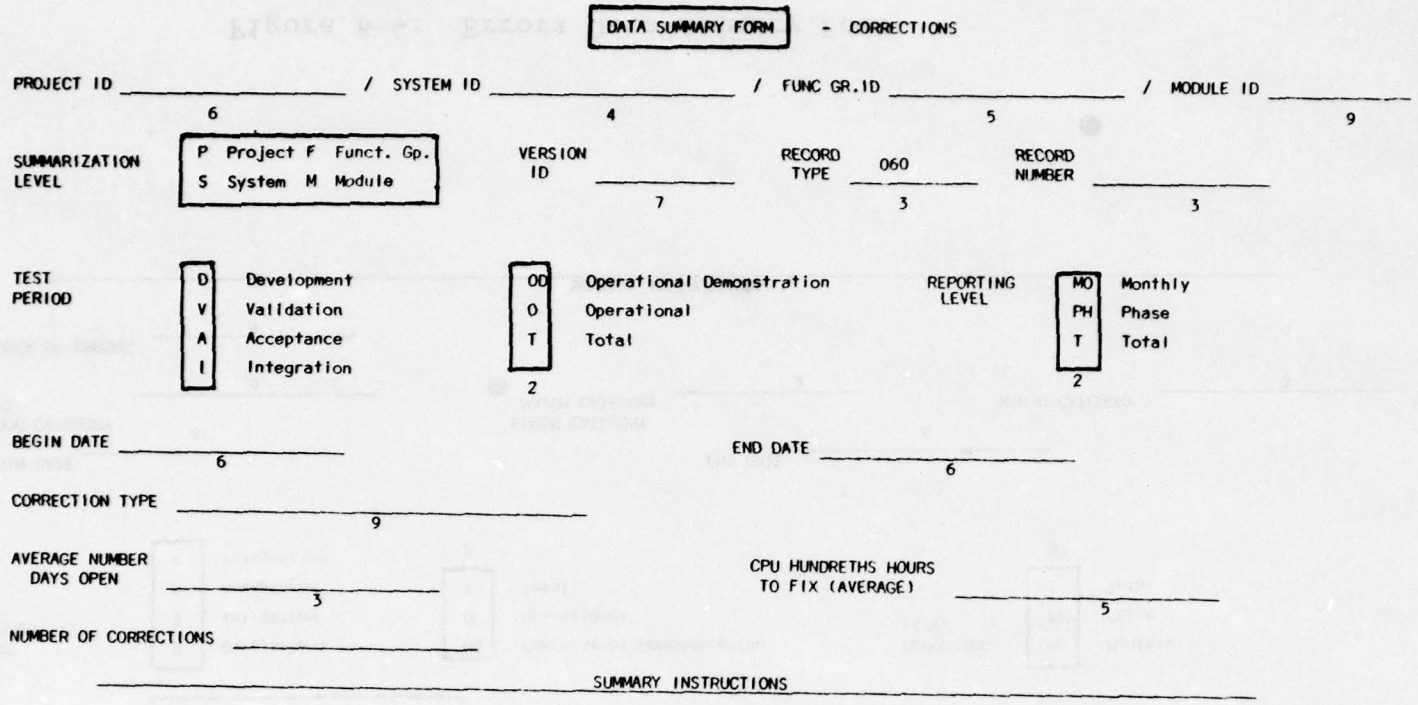


Figure B-5: Corrections Data Summary Form

DATA SUMMARY FORM - COMPONENT - MODULE SUMMARY

PROJECT ID 6	SYSTEM ID 4	FUNC GR ID 5	MODULE ID 9	SUMMARIZE LEVEL 1	VERSION ID 7	RECORD TYPE 3	REC # 3	NAME 12

Figure B-6: Component-Module Data Summary Form

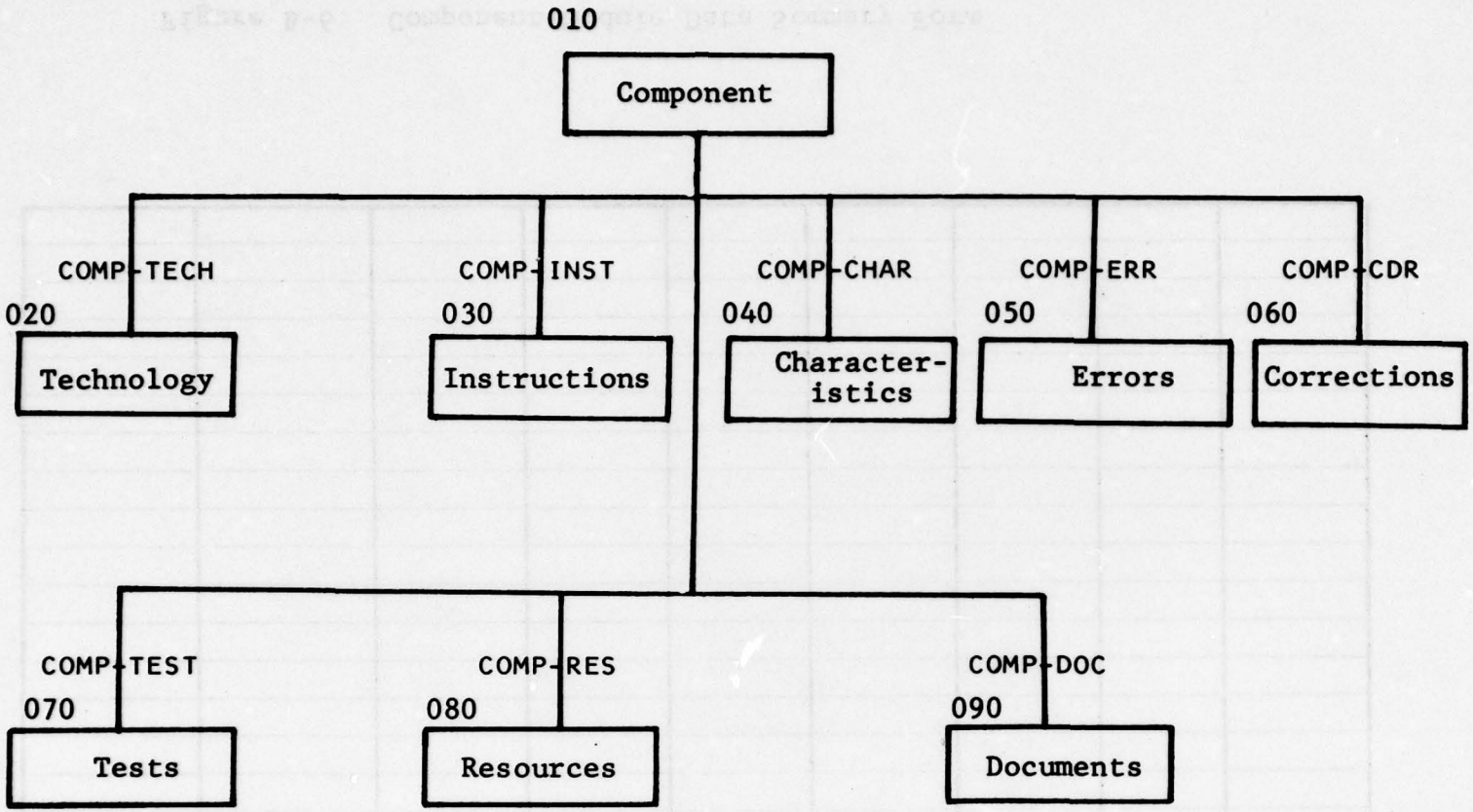


Figure B-7: Summary Database Entries

```
*LIST DISSUM
1130 DATA-BASE-REFERENCE IS DBRSUM
1140 ISP-DATA-BASE IS BFCBMIS1/DBNSUM
1150 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSUM
1160 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSUM
1170 ISP-INDEX-FILE IS BFCBMIS1/INDSUM
1180 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/USLHIST
1190 END DBRSUM
1200 DATA-BASE-REFERENCE IS DBRTRAN
1210 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNTRAN
1220 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSTRAN
1230 DEFINITION-OBJECT-FILE BFCBMIS1/DDOTRAN
1240 TRANSACTION-REJECT-FILE IS BFCBMIS1/TRAN-REJ
1250 END DBRTRAN
1260 DATA-BASE-REFERENCE IS DBRHTRI
1270 ISP-DATA-BASE IS BFCBMIS1/DBNHTRI
1280 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSHTRI
1290 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOHTRI
1300 ISP-INDEX-FILE IS BFCBMIS1/INDHTRI
1310 TRANSACTION-REJECT-FILE IS BFCBMIS1/TRJHTRI
1320 END DBRHTRI
1330 DATA-BASE-REFERENCE IS DBRHTRS
1340 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNHTRS
1350 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSHTRS
1360 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOHTRS
1370 TRANSACTION-REJECT-FILE IS BFCBMIS1/TRJHTRS
1380 END DBRHTRS


*LIST ADSSUM
010 DATA-BASE IS DBRSUM IN BFCBMIS1/DIOSUM
020 ENTRY COMPONENT IS RECORD COMPONENT VIA SEARCH
030 ENTRY ERRORS IS RECORD ERRORS THRU COMPONENT
040 ENTRY CORRECTIONS IS RECORD CORRECTIONS THRU COMPONENT
050 ENTRY INSTRUCTIONS IS RECORD INSTRUCTIONS THRU COMPONENT
```

Figure B-8: Summary Database MDQS Definitions

```
*LIST DDSSUM
10:MD DBRSUM;FILE IS INDEXED-SEQUENTIAL.
20:01  COMPONENT; TYPE IS "010" IN TYPE-010;
30: RETRIEVAL VIA SEARCH.
40:  02  COMP-REC-ID.
50:    03  COMP-ID.
60::04  PROJECT-ID PIC X(6).
70::04  SYS-ID PIC X(4).
80::04  FUNC-ID PIC X(5).
90::04  MOD-ID PIC X(9).
100:  03 COMP-SUM-ID.
110::04  SUM-LEVEL PIC X.
120::04  VERS-ID PIC X(7).
130::04  TYPE-010 PIC 9(3).
140::04  NUM-010 PIC 9(3).
150:02  NAME PIC X(12).
160:02  COMP-TYPE PIC X(3).
170:02  DESC PIC X(28).
180:02  DEVELOPER PIC X(11).
190:02  CONT-NUM PIC X(24).
200:02  DATA-DATE-I.
210:  03  MO-DA-I PIC 99.
220:  03  DA-DA-I PIC 99.
230:  03  YR-DA-I PIC 99.
240:02  DATA-DATE-C.
250:  03  MO-DA-C PIC 99.
260:  03  DA-DA-C PIC 99.
270:  03  YR-DA-C PIC 99.
280:02  DATA-SRC PIC XX.
290:02  NUM-SYS PIC 9(4).
300:02  NUM-FG PIC 9(4).
310:02  NUM-MOD PIC 9(4).
320:02  CONTRACT-TYPE PIC X(20).
330:02  STANDARDS PIC X(20).
340:02  COLL-PUR PIC X(10).
350:02  COLL-PROC PIC X(10).
360:02 PRIORITIES PIC X(10).
365:02  CONSTRAINTS PIC X(20).
370:02  DOC-ACC PIC X(25).
380:98 INDEX DETAIL;SEARCH ON COMP-REC-ID.
390:98  COMP-TECH MASTER;ORDER IS SORTED.
400:98 COMP-INST MASTER;ORDER IS SORTED.
410:98  COMP-CHAR MASTER;ORDER IS SORTED.
420:98 COMP-ERR MASTER;ORDER IS SORTED.
430:98 COMP-COR MASTER;ORDER IS SORTED.
440:98 COMP-TEST MASTER; ORDER IS SORTED.
450:98 COMP-RES MASTER; ORDER IS SORTED.
460:98  COMP-DOC MASTER;ORDER IS SORTED.
```

Figure B-8: Summary Database MDQS Definitions (Cont'd)

```
470:01  ERRORS;TYPE IS "050" IN TYPE-050;
480:  RETRIEVAL VIA COMP-ERR.
490:02  ERR-REC-ID.
500:  03  ERR-ID.
510::04  ERR-PID PIC X(6).
520::04  ERR-SID PIC X(4).
530::04  ERR-FID PIC X(5).
540::04  ERR-MID PIC X(9).
550:  03  ERR-SUM-ID.
560::04  ERR-SUM PIC X.
570::04  ERR-VID  PIC X(7).
580::04  TYPE-050 PIC 9(3).
590::04  NUM-050 PIC 9(3).
600:02  TEST-PER-E PIC XX.
610:02  REP-LEVEL-E PIC XX.
620:02  CALEN-PER-E.
630:  03  BEGIN-DATE-E.
640::04  MO-BEG-E PIC 99.
650::04  DA-BEG-E  PIC 99.
660::04  YR-BEG-E PIC 99.
670:  03  END-DATE-E.
680::04  MO-END-E PIC 99.
690::04  DA-END-E PIC 99.
700::04  YR-END-E PIC 99.
710:02 ERR-CAT-TYPE PIC X(3).
720:02  ERROR-CAT.
730:  03  MAJOR-CAT PIC XX.
740:  03  MINOR-CAT PIC X(3).
750:02  NUM-ERRORS PIC 9(5).
760:98  COMP-ERR DETAIL;
770  :ASCENDING KEY IS ERR-REC-ID.
780:01 CORRECTIONS;TYPE IS "060" IN TYPE-060;
790:  RETRIEVAL VIA COMP-COR.
800:02  COR-REC-ID.
810:  03  COR-ID.
820::04  COR-PID PIC X(6).
830::04  COR-SID PIC X(4).
840::04  COR-FID PIC X(5).
850::04 COR-MID PIC X(9).
860:  03  COR-SUM-ID.
870::04  COR-SUM PIC X.
880::04  COR-VID PIC X(7).
890::04  TYPE-060 PIC 9(3).
900::04 NUM-060 PIC 9(3).
910:02  TEST-PER-C PIC XX.
920:02  REP-LEVEL-C PIC XX.
930:02  CALEN-PER-C.
```

Figure B-8: Summary Database MDQS Definitions (Cont'd)

```
940:  03  BEGIN-DATE-C.
950::04  MO-BEG-C PIC 99.
960::04  DA-BEG-C PIC 99.
970::04  YR-BEG-C PIC 99.
980:  03  END-DATE-C.
990::04  MO-END-C PIC 99.
1000::04 DA-END-C PIC 99.
1010::04  YR-END-C PIC 99.
1020:02  COR-TYPE PIC X(9).
1030:02  DAYS-OPEN-AV-C PIC 9(3).
1040:02  HHRS-TO-FIX-AV PIC 9(5).
1050:02  NUM-COR PIC 9(5).
1060:98  COMP-COR DETAIL;
1070: ASCENDING KEY IS COR-REC-ID.
1080:01 TECHNOLOGY;TYPE IS "020" IN TYPE-020;
1090:  RETRIEVAL VIA COMP-TECH.
1100:02  TECH-REC-ID.
1110:  03  TECH-ID.
1120::04  TECH-PID PIC X(6).
1130::04  TECH-SID PIC X(4).
1140::04  TECH-FID PIC X(5).
1150::04 TECH-MID PIC X(9).
1160:  03  TECH-SUM-ID.
1170::04  TECH-SUM PIC X.
1180::04  TECH-VID PIC X(7).
1190::04  TYPE-020 PIC 9(3).
1200::04 NUM-020 PIC 9(3).
1210:02  PHASE-T PIC XX.
1220:02  REP-LEVEL-T PIC XX.
1221:02  CALEN-PER-T.
1222:  03  BEGIN-DATE-T.
1223::04  MO-BEG-T PIC 99.
1224::04  DA-BEG-T PIC 99.
1225::04  YR-BEG-T PIC 99.
1226:  03  END-DATE-T.
1227::04  MO-END-T PIC 99.
1228::04  DA-END-T PIC 99.
1229::04  YR-END-T PIC 99.
1230:02  TECH-ID PIC X(4).
1240:02 TECH-PER PIC 99.
1250:98  COMP-TECH DETAIL;
1260: ASCENDING KEY IS TECH-REC-ID.
1270:01 INSTRUCTIONS;TYPE IS "030" IN TYPE-030;
1280:  RETRIEVAL VIA COMP-INST.
1290:02  INST-REC-ID.
1300:  03  INST-ID.
1310::04  INST-PID PIC X(6).
```

Figure B-8: Summary Database MDQS Definitions (Cont'd)

```
1320::04  INST-SID PIC X(4).
1330::04  INST-FID PIC X(5).
1340::04 INST-MID PIC X(9).
1350:  03  INST-SUM-ID.
1360::04  INST-SUM PIC X.
1370::04  INST-VID PIC X(7).
1380::04  TYPE-030 PIC 9(3).
1390::04 NUM-030 PIC 9(3).
1400:02  PHASE-I PIC XX.
1410:02  REP-LEVEL-I PIC XX.
1411:02  CALEN-PER-I.
1412:  03  BEGIN-DATE-I.
1413::04  MO-BEG-I PIC 99.
1414::04  DA-BEG-I PIC 99.
1415::04  YR-BEG-I PIC 99.
1416:  03  END-DATE-I.
1417::04  MO-END-I PIC 99.
1418::04  DA-END-I PIC 99.
1419::04  YR-END-I PIC 99.
1420:02 INST-TYPE PIC X(4).
1430:02  NUM-INST-S PIC 9(7).
1440:02  NUM-INST-O PIC 9(7).
1450:02  INST-PER PIC 99.
1451:02  COMPLEXITY-S.
1452:  03  CPLXS-TYPE PIC 99.
1453:  03  CPLXS-MEAS PIC X.
1454:02  COMPLEXITY-C.
1455:  03  CPLXC-TYPE PIC 99 OCCURS 5 TIMES.
1456:  03  CPLXC-MEAS PIC 9(5) OCCURS 5 TIMES.
1457:02  MODE-CONSTR PIC 99.
1460:98  COMP-INST DETAIL;
1470: ASCENDING KEY IS INST-REC-ID.
1480:01 CHARACTERISTICS;TYPE IS "040" IN TYPE-040;
1490:  RETRIEVAL VIA COMP-CHAR.
1500:02  CHAR-REC-ID.
1510:  03  CHAR-ID.
1520::04  CHAR-PID PIC X(6).
1530::04  CHAR-SID PIC X(4).
1540::04  CHAR-FID PIC X(5).
1550::04 CHAR-MID PIC X(9).
1560:  03  CHAR-SUM-ID.
1570::04  CHAR-SUM PIC X.
1580::04  CHAR-VID PIC X(7).
1590::04  TYPE-040 PIC 9(3).
1600::04 NUM-040 PIC 9(3).
1610:98  COMP-CHAR DETAIL;
1620: ASCENDING KEY IS CHAR-REC-ID.
```

Figure B-8: Summary Database MDQS Definitions (Cont'd)

```
1630:01 TESTS;TYPE IS "070" IN TYPE-070;
1640:  RETRIEVAL VIA COMP-TEST.
1650:02  TEST-REC-ID.
1660:  03  TEST-ID.
1670::04  TEST-PID PIC X(6).
1680::04  TEST-SID PIC X(4).
1690::04  TEST-FID PIC X(5).
1700::04 TEST-MID PIC X(9).
1710:  03  TEST-SUM-ID.
1720::04  TEST-SUM PIC X.
1730::04  TEST-VID PIC X(7).
1740::04  TYPE-070 PIC 9(3).
1750::04 NUM-070 PIC 9(3).
1760:98  COMP-TEST DETAIL;
1770: ASCENDING KEY IS TEST-REC-ID.
1780:01 RESOURSES;TYPE IS "080" IN TYPE-080;
1790:  RETRIEVAL VIA COMP-RES.
1800:02  RES-REC-ID.
1810:  03  RES-ID.
1820::04  RES-PID PIC X(6).
1830::04  RES-SID PIC X(4).
1840::04  RES-FID PIC X(5).
1850::04 RES-MID PIC X(9).
1860:  03  RES-SUM-ID.
1870::04  RES-SUM PIC X.
1880::04  RES-VID PIC X(7).
1890::04  TYPE-080 PIC 9(3).
1900::04 NUM-080 PIC 9(3).
1910:98  COMP-RES DETAIL;
1920: ASCENDING KEY IS RES-REC-ID.
1930:01 DOCUMENTS;TYPE IS "090" IN TYPE-090;
1940:  RETRIEVAL VIA COMP-DOC.
1950:02  DOC-REC-ID.
1960:  03  DOC-ID.
1970::04  DOC-PID PIC X(6).
1980::04  DOC-SID PIC X(4).
1990::04  DOC-FID PIC X(5).
2000::04 DOC-MID PIC X(9).
2010:  03  DOC-SUM-ID.
2020::04  DOC-SUM PIC X.
2030::04  DOC-VID PIC X(7).
2040::04  TYPE-090 PIC 9(3).
2050::04 NUM-090 PIC 9(3).
2060:98  COMP-DOC DETAIL;
2070: ASCENDING KEY IS DOC-REC-ID.
```

Figure B-8: Summary Database MDQS Definitions (Cont'd)

APPENDIX C

RADC PRODUCTIVITY DATABASE

The RADC Productivity Database was defined and queried using MDQS.* This database contains summary information from over 400 projects and includes project and company name, the programming language used and the percent of utilization, the number of pages of documentation and the number of lines of source code delivered, the total number of man months and calendar months, the number of software problem reports, an SPR and productivity rate, and the type of technology used (i.e., structured code, top-down programming, etc.). Figure C-1 contains a list of the data items, the MDQS name, and the number of characters required for each value.

The Directory, Data and Application Definition listings for this database are contained in Figure C-2.

Figure C-3 contains a sample query illustrating the use of this database. Records are retrieved and printed where the number of delivered lines of source code (DSLOC) and the error rate (SPR-RATE) are present. A count of the total number of records that contain this information and the sum of the total number of delivered lines of source code are printed.

---

*Richard Nelson, Software Data Collection and Analysis, Rome, NY: Rome Air Development Center, September 1978.

REC-A			
	Seq. ID	ID-A	4
	Record Type	TYPE-A	1
	Reference Document	REF	3
	Project Name	PROG-NAME	28
	Company Name	COMPANY	11
	Prog. Lang. 1	LANG-1	11
	" " % utilization	PER-CENT-1	3

A-B Chain

REC-B			
	Seq. ID	ID-B	4
	Record Type	TYPE-B	1
	Prog. Lang. 2	LANG-2	12
	" " % utilization	PER-CENT-2	3
	Prog. Lang. 3	LANG-3	12
	" " % utilization	PER-CENT-3	3
	Program Design Lang.	PDL	6
	Design Lang. Rate	PDL-RATE	5
	Documentation	DOC	6
	Documentation Rate	DOC-RATE	6

B-C Chain

REC-C			
	Seq. ID	ID-C	4
	Record Type	TYPE-C	1
	Delivered Source Lines	DSLOC	7
	Project Effort (man months)	TMM	5
	" " (calendar months)	TM	3
	Productivity Rate	PROD	5
	Structured Code (%)	SC	3
	Top-Down Programming (%)	TDP	3
	Chief Programmer (%)	CPT	3
	Librarian or PSL (%)	LIB	3
	Code Reviews (%)	CR	3
	Number of Problem Reports	SPR	5
	Error Rate (per 100)	SPR-RATE	6
	Average # People	PEO	3

Derived Data

Productivity = PROD = DSLOC/TMM
Average People = PEO = TMM/TM
Error Rate = SPR-RATE = (SPR's/DSLOC) 100
Documentation Rate = DOC-RATE = DOC/DSLOC
Design Lang. - Rate = PDL/RATE = PDL/DSLOC

Figure C-1: Data-Item Descriptions - Productivity Database

```
LIST DDSDN

0050 :MD DBRDN;FILE IS SEQUENTIAL.
0060: 01 REC-A;TYPE IS "A" IN TYPE-A;
0070:  RETRIEVAL VIA SCAN.
0080:  02 FILLER PIC X(8).
0090:  02 ID-A PIC X(4).
0100:02  TYPE-A PIC X.
0110:02  REF PIC XXX.
0120:02 FILLER PIC X.
0130:02 PROG-NAME PIC X(28).
0140:02 FILLER PIC X.
0150:02 COMPANY PIC X(11).
0160:02  FILLER PIC X.
0170:02 LANG-1 PIC X(11).
0180:02 FILLER PIC X.
0190:02 PER-CENT-1 PIC XXX.
0200:02 FILLER PIC X(11).
0202: 98 REC-AREC-B MASTER;ORDER IS SORTED.
0210:98 SYSTEM DETAIL;SCAN ON TYPE-A.
0220:01 REC-B;TYPE IS "B" IN TYPE-B;
0230:RETRIEVAL VIA REC-AREC-B.
0240:02 FILLER PIC X(8).
0250:02 ID-B PIC X(4).
0260:02 TYPE-B PIC X.
0280:02 LANG-2 PIC X(12).
0290:02 FILLER PIC X.
0300:02 PER-CENT-2 PIC XXX.
0310:02 FILLER PIC X.
0320:02 LANG-3 PIC X(12).
0330:02 FILLER PIC X.
0340:02 PER-CENT-3 PIC XXX.
0350:02 FILLER PIC X.
0360:02 PDL PIC X(6).
0370:02 FILLER PIC X.
0380:02 PDL-RATE PIC X(5).
0390:02 FILLER PIC X.
0400:02 DOC PIC X(6).
0410:02 FILLER PIC X.
0420:02 DOC-RATE PIC X(6).
0430:02 FILLER PIC X(11).
0440: 98 REC-BREC-C MASTER;ORDER IS SORTED.
0441:98 REC-AREC-B DETAIL.
0450:01 REC-C;TYPE IS "C" IN TYPE-C;
0460:RETRIEVAL VIA REC-BREC-C.
0470:02 FILLER PIC X(8).
0480:02 ID-C PIC X(4).
0490:02 TYPE-C PIC X.
0500:02 DSLOC PIC X(7).
0510:02 FILLER PIC X.
0520:02 TMM PIC X(5).
0530:02 FILLER PIC X.
0540:02 TM PIC XXX.
0550:02 FILLER PIC X.
0560:02 PROD PIC X(5).
0570:02 FILLER PIC X.
0580:02 SC PIC XXX.
0590:02 FILLER PIC X.
0600:02 TDP PIC XXX.
0610:02 FILLER PIC X.
0620:02 CPT PIC XXX.
0630:02 FILLER PIC X.
0640:02 LIB PIC X(3).
0650:02 FILLER PIC X.
0660:02 CR PIC XXX.
0670:02 FILLER PIC X.
0680:02 SPR PIC X(5).
0690:02  FILLER PIC X.
0700:02 SPR-RATE PIC X(6).
0710:02 FILLER PIC X.
0720:02 PEO PIC XXX.
0730:02 FILLER PIC X(11).
0740:98 REC-CREC-D MASTER;ORDER IS SORTED.
0741:98 REC-BREC-C DETAIL.
0750:01 REC-D;TYPE IS "D" IN TYPE-D;RETRIEVAL VIA REC-CREC-D.
0760:02 FILLER PIC X(12).
0770:02 TYPE-D PIC X.
0780:02 FILLER PIC X(71).
0790:98 REC-CREC-D DETAIL.

LIST ADSDN

0250 DATA-BASE IS DBRDN IN BFCBMIS1/DIODN
0260 ENTRY X IS RECORD REC-A AND REC-B AND REC-C
0270ITEMS ARE:
0280RECORD REC-A ALL ITEMS
0281 PER-CENT-1 INPUT-EDITED
0290RECORD REC-B ALL ITEMS
0291 PER-CENT-2 INPUT-EDITED
0292 PER-CENT-3 INPUT-EDITED
0293 PDL INPUT-EDITED
0294 PDL-RATE INPUT-EDITED
0295 DOC INPUT-EDITED
0296 DOC-RATE INPUT-EDITED
0300RECORD REC-C ALL ITEMS
0310  DSLOC INPUT-EDITED
0311 TMM INPUT-EDITED
0312 TM INPUT-EDITED
0313 PROD INPUT-EDITED
0314 SC INPUT-EDITED
0315 TDP INPUT-EDITED
0316 CPT INPUT-EDITED
0317 LIB INPUT-EDITED
0318 CR INPUT-EDITED
0319 SPR INPUT-EDITED
0320 SPR-RATE INPUT-EDITED
0321 PEO INPUT-EDITED

LIST DISDN

0010 DATA-BASE-REFERENCE IS DBRDN
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNDN
0030 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSDN
0040 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDODN
```

Figure C-2: MDQS Definitions - Productivity Database

```
0070 INVOKE ADODN
0075 DEFINE $DSL WITH PIC "9(14)"
0080 RETRIEVE X FROM DBRDN
0090 WHERE DSLOC GE 0
0100 AND SPR-RATE GE 0
0110 SORT X ON DSLOC
0120 PRINT  ON FILE TESTDN FOR TTY
0130 DSLOC,SPR-RATE,SPR,ID-A,LANG-1,LANG-2,LANG-3
0140  LET $DSCOUNT = COUNT DSLOC
0150 LET $DSL = SUM DSLOC
0160 WRAP-UP
0170 PRINT ON FILE TESIDN FOR TTY $DSCOUNT,$DSL
0180 END
```

DSLOC	SPR-RATE	SPR	ID-A	LANG-1	LANG-2	LANG-3
115346	1.739	2006	85	JOVIAL J4	X	X
120000	1.697	2036	140	ASSY	JOVIAL J3B	X
136350	1.960	2673	81	CENTRAN	X	X
136689	1.584	2165	169	JOVIAL J3	ASSY	X
250000	.414	1036	184	CMS-2	X	X
300000	1.083	3250	175	FORTRAN IV	ASSY	X
14500	.593	86	88	ASSY	X	X
17500	.251	44	93	COBOL	X	X
18246	.016	3	89	COBOL	X	X
19045	1.691	322	206	COBOL	X	X
22075	2.818	622	82	CENTRAN	X	X
22816	1.043	238	83	CENTRAN	X	X
25900	.050	13	94	COBOL	X	X
27055	1.918	519	159	ASSY	X	X
31500	.984	310	190	X	X	X
32000	.522	167	192	X	X	X
32400	.660	214	77	COBOL	X	X
36000	.358	129	194	X	X	X
48000	.154	74	396	COBOL	X	X
54116	.819	443	207	COBOL	FORTRAN	ASSY
73000	.514	375	193	X	X	X
78640	2.195	1726	97	FORTRAN	ASSY	X
81650	.372	304	191	X	X	X
83324	.055	46	23	PL/1	ASSY	X
83866	13.985	11729	168	ASSY	X	X
96931	1.545	1498	84	JOVIAL J4	X	X
1000	6.600	66	96	PL/1	X	X
2280	7.851	179	205	COBOL	X	X
4023	8.476	341	87	JOVIAL J3	X	X
5100	.098	5	90	COBOL	X	X

DSCOUNT	DSL
30.000000	00000001969352

Figure C-3: Query Example - Productivity Database

APPENDIX D

DATASET LOADING

The purpose of this appendix is to assist the user in loading the datasets into permfile as is necessary for retrieval by MDQS. Currently all the datasets used in this project are available on magnetic tape in a format easily readable by GCOS.

This appendix contains the minimum sizes of permfiles necessary to contain the data, the tape characteristics, a method for loading the permfiles, and a method for concatenating datasets.

8.1 Creating the Permfiles

Before loading any data, the permfile into which the data is to be stored must be created using the GCOS Access command. All files are sequential.

The following are the minimum sizes in little links (11) of permfile necessary for each dataset.

PROJECT	LITTLE LINKS	WORDS
1	301	96,320
2	120	38,400
3	150	48,000
4	514	164,220
5 file 1	320	102,400
5 file 2	160	51,200
6	120	38,100

8.2 Concatenating Dataset

The Project 1 data is in two separate tape files. For easier data manipulation the files were appended. The following FORTRAN program was used to accomplish this.

```
*LIST

10*#RUNH=(BCD)*OP30"30";DBNTRWS"27";TRW-OP"28"
20        DATA ICOUNT,ICNT27,ICNT28/0,0,0/
30        DIMENSION IBUF(14)
40   20   READ(27,END-40)IBUF
50        WRITE(30)IBUF
60        ICOUNT - ICOUNT+1
70        ICNT27 = ICNT27+1
80        GO TO 20
90   40   READ(28,END=50)IBUF
100        WRITE(30)IBUF
110        ICOUNT = ICOUNT+1
120        ICNT28 = ICNT28+1
130        TO TO 40
230  50    WRITE(60,100)ICNT27,ICNT28,ICOUNT
240  100   FORMAT(1X,2I6)
250        STOP
260        END

ready
```

8.3 Magnetic Tape Characteristics

The tape characteristics for each project are as follows.

PROJECT	TAPE #	TRACK	BPI	FILE	SPECIAL CHARACTERISTICS
1	46608	9	800	1,3	
2	44624	7	800	1	
3	46608	9	800	2	*
4	44284	9	800	1	
5	44314	9	800	1	
6	44317	9	800	1	

```
*$:FFILE:LUD,NSTP,NOSRLS,MODBCD,MLTFIL,FIXLING14,BUFSIZ/1
```

8.4 Magnetic Tape Data Loading

The standard GCOS utility can be used to read each tape into permfile. For utility specifics see Honeywell Manual DD12. An example for Project 2 follows.

```
LIST

10##N
20$:IDENT:BFCBMIS1,C CURTIS,555008570052
30$:UTILITY:NDUMP
40$:TAPE9:27,A3D,,44624,,INPUT,,DEN8
55$:PRMFL:28,R/W,S,BFCBMIS1/DBNB1S1
60$:FUTIL:27,28,RWD/27/,COPY/1F/
140$:ENDJOB

ready
```

The user would alter the above CARDIN job to reflect the GCOS ID and permfile name as well as the tape specifics for each project.

REFERENCES

1. Management Data Query System/IV, User's Guide. Waltham, MA: Honeywell Information Systems, Incorporated, DD92, Revision 1, August, 1976.

2. Management Data Query System, Administrator's Guide. Waltham, MA: Honeywell Information Systems, Incorporated, DD94, Revision 1, August, 1976.

3. Thayer, T. A. (TRW Defense and Space Systems Group, Redondo Beach, CA). Software Reliability Study, Griffiss AFB, NY: Rome Air Development Center, RADC-TR-76-238, August, 1976. A030798.

4. Fries, M. J. (Boeing Aerospace Company, Seattle, WA). Software Error Data Acquisition. Griffiss AFB, NY: Rome Air Development Center, RADC-TR-77-130, April, 1977 A039916.

5. William, H. E., Jr., James, T. A., Beaureguard, A. A., Hilcoff, P. (Raytheon Company, Bedford, MA). Software Systems Reliability: A Raytheon Project History. Griffiss AFB, NY: Rome Air Development Center, RADC-TR-77-188, June, 1977 A040992.

6. Rye, P., Bamberger, F., Ostanek, W., Brodeur, N., Goode, J. (The Charles Stark Draper Laboratory, Incorporated, Cambridge, MA). Software Systems Development: A CSDL Project History. Griffiss AFB, NY: Rome Air Development Center, RADC-TR-77-213, June, 1977 A042186.

7. Baker, W. F. (IBM Corporation, Whippany, NJ). Software Data Collection and Analysis: A Real-Time System Project History. Griffiss AFB, NY: Rome Air Development Center, RADC-TR-77-192, June, 1977 A041644.

8. Hecht, H., Sturm, W. A., Trattner, S. (The Aerospace Corporation, El Segundo, CA). Reliability Measurement During Software Development. Hampton, VA: NASA Langley Research Center, NASA-CR-145205, September, 1977.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.