BASELINE SOFTWARE DATA SYSTEM
Database Reference Manual

IIT Research Institute

Lorraine M. Duvall
Christine Curtis

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 PALAIAMO  
Project Engineer

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

FOR THE COMMANDER:  
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.
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.
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

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)

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3</td>
<td>Magnetic Tape Characteristics</td>
<td>D-2</td>
</tr>
<tr>
<td>8.4</td>
<td>Magnetic Tape Data Loading</td>
<td>D-2</td>
</tr>
<tr>
<td>IX.</td>
<td>REFERENCES</td>
<td>37</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

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
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 1</td>
<td>Glossary of Data-Item Names</td>
<td>4</td>
</tr>
<tr>
<td>Table 2</td>
<td>Data-Item/Projects</td>
<td>5</td>
</tr>
<tr>
<td>Table 3</td>
<td>Dataset Names</td>
<td>6</td>
</tr>
</tbody>
</table>
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 - These entries contain information on the hardware and software environment and the testing characteristics. Only Project 2 has these entries.

SOFTWARE -

TEST-CHAR -

Table 3 contains, for each project, the project designation, database-reference name(s), application name, entry name(s), and number of record occurrences.
# TABLE 1: GLOSSARY OF DATA-ITEM NAMES

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJ-ID</td>
<td>Project Identification</td>
</tr>
<tr>
<td>PROJ-VERSION</td>
<td>Project Version</td>
</tr>
<tr>
<td>PROJ-TYPE</td>
<td>Project Type</td>
</tr>
<tr>
<td>SYS-ID</td>
<td>System Identification</td>
</tr>
<tr>
<td>SYS-VERSION</td>
<td>System Version</td>
</tr>
<tr>
<td>SYS-TYPE</td>
<td>System Type</td>
</tr>
<tr>
<td>SSYS-ID</td>
<td>Subsystem or Functional Area Identification</td>
</tr>
<tr>
<td>SSYS-VERSION</td>
<td>Subsystem Version</td>
</tr>
<tr>
<td>SSYS-TYPE</td>
<td>Subsystem Type</td>
</tr>
<tr>
<td>MOD-ID</td>
<td>Module Identification</td>
</tr>
<tr>
<td>MOD-VERSION</td>
<td>Module Version</td>
</tr>
<tr>
<td>COMP-ID</td>
<td>Computer Identification</td>
</tr>
<tr>
<td>COMP-OM</td>
<td>Computer Operating Mode</td>
</tr>
<tr>
<td>COMP-RATE</td>
<td>Computer Processing Rate</td>
</tr>
<tr>
<td>COMP-OS</td>
<td>Computer Operating System Type</td>
</tr>
<tr>
<td>TECH-ID</td>
<td>Identification of the Construction Technology</td>
</tr>
<tr>
<td>COMPL-ID</td>
<td>Type of Complexity Measure Used</td>
</tr>
<tr>
<td>COMPLEXITY</td>
<td>The Complexity Measure Value</td>
</tr>
<tr>
<td>CONST-TYPE</td>
<td>Constituent Type (ex. JOVIAL, ASSEMBLY LANGUAGE)</td>
</tr>
<tr>
<td>NUM-OCCUR</td>
<td>Number of Occurrences of Constituent Type</td>
</tr>
<tr>
<td>PHASE</td>
<td>Phase in Which Action Occurred</td>
</tr>
<tr>
<td>NUM-RUNS-TOT</td>
<td>Total Number of Runs</td>
</tr>
<tr>
<td>TEST-PER</td>
<td>The Period in Which the Test Was Performed</td>
</tr>
<tr>
<td>NUM-RUNS-OK</td>
<td>Total Number of Correct Runs</td>
</tr>
<tr>
<td>AHRS-PER-TEST</td>
<td>Average Number of Hours Per Test</td>
</tr>
<tr>
<td>TEST-ID</td>
<td>Test Identification</td>
</tr>
<tr>
<td>TEST-TYPE</td>
<td>Type of Test</td>
</tr>
<tr>
<td>DATE-RUN</td>
<td>Date the Test Was Run</td>
</tr>
<tr>
<td>STRESS-TYPE</td>
<td>Type of Stress Applied</td>
</tr>
<tr>
<td>STRESS-MEAS</td>
<td>Amount of Stress Applied</td>
</tr>
<tr>
<td>TEST-RESULT</td>
<td>Result of Test</td>
</tr>
<tr>
<td>NUM-ERR</td>
<td>Number of Errors Discovered Per Test</td>
</tr>
<tr>
<td>SPR-NUM</td>
<td>Software Problem Report Number</td>
</tr>
<tr>
<td>DATE-OPEN</td>
<td>Date the Problem Was Reported</td>
</tr>
<tr>
<td>MOD-SOURCE</td>
<td>The Module ID Where the Problem Was Manifested</td>
</tr>
<tr>
<td>ERR-CAT-TYPE</td>
<td>Error Category Type</td>
</tr>
<tr>
<td>ERROR-CAT</td>
<td>Error Category Code</td>
</tr>
<tr>
<td>SEV-TYPE</td>
<td>Severity Type</td>
</tr>
<tr>
<td>SEV-TERMIN</td>
<td>Severity</td>
</tr>
<tr>
<td>MOD-CHANCED</td>
<td>The ID of the Changed Module</td>
</tr>
<tr>
<td>CORR-DESC</td>
<td>A Description of the Correction</td>
</tr>
<tr>
<td>ERROR-DESC</td>
<td>A Description of the Error</td>
</tr>
</tbody>
</table>

*
TABLE 2: DATA-ITEM/PROJECTS

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>PROJ 1</th>
<th>PROJ 2</th>
<th>PROJ 3</th>
<th>PROJ 4</th>
<th>PROJ 5</th>
<th>PROJ 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROJ-ID</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>PROJ-VERSION</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>PROJ-TYPE</td>
<td>3</td>
<td></td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>SYS-ID</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS-VERSION</td>
<td>3</td>
<td></td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS-TYPE</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS-ED</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SYS-VSERSION</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS-TYoe</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD-ID</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MOD-VSERSION</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>MOD-TYPE</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>CONP-ID</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CONF-VERS</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>CONF-OS</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TECH-ID</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>COMPL-ID</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COMPLEXITY</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>CONST-TYPE</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>NUM-OCUR</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>NUM-RUNS-OK</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMES-PER-TEST</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHASE</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TEST-PEK</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST-ID</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST-TYPE</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE-RUN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>STRESS-TYPE</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STRESS-MEAS</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RST chests</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEST-RESULT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUM-ERR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPK-NUM</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATE-OPEN</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>MOD-SOURCE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERR-CAT-TYPE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ERROR-CAT</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>SEVERITY</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TYPE-TERM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HAS-TODISC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WORK-CAT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNK-NUM</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD-CHANGED</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOD-CH-VERS</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COR-TYPE</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COR-MEG</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACT-CAT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE-BEGIN</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE-CLOSE</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATE-OPEN</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERRS-COMPLEX</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERRS-CHANGED</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CODE-CONT</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROB-DESC</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CORA-DESC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERROR-DESC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Project Designation</td>
<td>Database Reference Name</td>
<td>Application Definition Name</td>
<td>Entry Name</td>
<td>Number of Records</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-------------------------</td>
<td>------------------------------</td>
<td>---------------------</td>
<td>-------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>DBRTRWS</td>
<td>ADOTRWS</td>
<td>PROBLEM-REPORT</td>
<td>4,970</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DBRB1S1</td>
<td>ADOB1S1</td>
<td>PROBLEM-REPORT</td>
<td>2,036</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MODULE</td>
<td>69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>DBRSDS1</td>
<td>ADOSDS1</td>
<td>PROBLEM-REPORT</td>
<td>2,165</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MODULE</td>
<td>109</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>DBRAPS1</td>
<td>ADOAPS1</td>
<td>PROBLEM-REPORT</td>
<td>11,730</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>DBRSAFS1</td>
<td>(ADOSAFS1)</td>
<td>PROBLEM-REPORT</td>
<td>5,693</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>DBRSAFSN</td>
<td></td>
<td>MODULE</td>
<td>2,413</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>DBRSMTC</td>
<td>ADOSMTC</td>
<td>PROBLEM-REPORT</td>
<td>2,719</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
*ADFO

BRIEF OR FULL FULL

ADF OBJECT FILE -- ADOTRWS
DATA BASE REF -- DBRTRWS
SEQUENTIAL FILE
ENTRY -- ALL
ENTRIES 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
MOD-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 1
EXPLAN
    A/N SIZE 2
DATE-CLOSE
    A/N SIZE 8
MOD-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
    A/N SIZE 1
SCAN FIELD
    A/N SIZE 2
CODE-CONT
    A/N SIZE 2
DAYS-OPEN
    A/N SIZE 3
```

7
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 DBTRWS <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 Items to be contain result 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.*
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

<table>
<thead>
<tr>
<th>MOD-CHANGED</th>
<th>SPR-NUM</th>
<th>DATE-OPEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>C106</td>
<td>0042</td>
<td>6-18-73</td>
</tr>
<tr>
<td>C102</td>
<td>0180</td>
<td>6-04-73</td>
</tr>
<tr>
<td>C102</td>
<td>0181</td>
<td>6-02-73</td>
</tr>
<tr>
<td>C105</td>
<td>0182</td>
<td>6-02-73</td>
</tr>
<tr>
<td>F403</td>
<td>0183</td>
<td>6-02-73</td>
</tr>
<tr>
<td>E112</td>
<td>0184</td>
<td>6-02-73</td>
</tr>
<tr>
<td>B103</td>
<td>0185</td>
<td>6-02-73</td>
</tr>
<tr>
<td>E113</td>
<td>0186</td>
<td>6-02-73</td>
</tr>
<tr>
<td>C503</td>
<td>0187</td>
<td>6-02-73</td>
</tr>
<tr>
<td>D/B</td>
<td>0188</td>
<td>6-04-73</td>
</tr>
</tbody>
</table>
```
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:

<table>
<thead>
<tr>
<th>Test</th>
<th>Operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>EQ or =</td>
</tr>
<tr>
<td>Less than</td>
<td>LT or &lt;</td>
</tr>
<tr>
<td>Greater than</td>
<td>GT or &gt;</td>
</tr>
<tr>
<td>Less than or equal</td>
<td>LE, or &lt;=, or =&lt;</td>
</tr>
<tr>
<td>Greater than or equal</td>
<td>GE, or &gt;=, or =&gt;</td>
</tr>
<tr>
<td>Not equal</td>
<td>NE, or &lt;&gt;, or &lt;&gt;</td>
</tr>
<tr>
<td>Present</td>
<td>PRESENT</td>
</tr>
<tr>
<td>Absent</td>
<td>ABSENT</td>
</tr>
<tr>
<td>Contains</td>
<td>CONTAINS</td>
</tr>
</tbody>
</table>

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 DBTRWS
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.

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

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.

```plaintext
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
```

Some of the sorted data is shown below.

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

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.

```plaintext
%SYSTEM MDQ NEW
READY
%ADFQ
BRIEF OR FULL FULL
ADF OBJECT FILE — ADOSAFS1
DATA BASE REF — DBRSAFSN
SEQUENTIAL FILE
ENTRY — ALL
ENTRIES 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)
```
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 DBRSAFNS
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.

<table>
<thead>
<tr>
<th>ID-COUNT</th>
<th>TOT-LINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000000</td>
<td>228.000000</td>
</tr>
<tr>
<td>2.000000</td>
<td>506.000000</td>
</tr>
<tr>
<td>3.000000</td>
<td>540.000000</td>
</tr>
<tr>
<td>4.000000</td>
<td>668.000000</td>
</tr>
<tr>
<td>5.000000</td>
<td>674.000000</td>
</tr>
<tr>
<td>6.000000</td>
<td>688.000000</td>
</tr>
<tr>
<td>7.000000</td>
<td>702.000000</td>
</tr>
<tr>
<td>8.000000</td>
<td>1335.000000</td>
</tr>
<tr>
<td>9.000000</td>
<td>2354.000000</td>
</tr>
<tr>
<td>10.000000</td>
<td>2985.000000</td>
</tr>
<tr>
<td>11.000000</td>
<td>3549.000000</td>
</tr>
<tr>
<td>12.000000</td>
<td>3994.000000</td>
</tr>
<tr>
<td>13.000000</td>
<td>4564.000000</td>
</tr>
<tr>
<td>14.000000</td>
<td>4894.000000</td>
</tr>
<tr>
<td>15.000000</td>
<td>5607.000000</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>ID-COUNT</th>
<th>TOT-LINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>2413.000000</td>
<td>317031.000000</td>
</tr>
</tbody>
</table>

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)

16
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.

CMDO
APPLICATION FILE IS - ADOTORWS
DATABASE REFERENCE IS DBTRMS
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
- BFCBM1SI/CHOOSE

EXTRACT ALL INSTANCES ? Y

.....
RUN-ID? CC
$ IDENT? BFCBM1SI,C CURTIS,555008570052
SNMNB 2001T

2001T
TERMINATE PROCEDURE BY RESPONDING 'END' TO A
NULL 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= BB MAJOR-CAT= BB MINOR-CAT= 140

SNM-NUM= M-0300 COR-TYPE= X NBR-MOU= X DOC-UP=
CM-CH=
DB-CH= X EXPLAN=
DATE=CLOSE= 6-05-73 MOD=CLOSE= 6 DA=CLOSE= 05 YR=CLOSE= 73 PROJ-VERSION= BLK2 PROJ-ID= 03 CODE-COUNT= DAYS-OPEN= 0 MOD-CHANGED= (211)
SPR-NUM= 0214
MOD-SOURCE= U211
DAE=OPEN= 6-06-73
MOD=OPEN= 0 DA=OPEN= 06
YR=OPEN= 73
SEVERITY= TESI-PER= V
TEST-ID= ERROR-CAT= BB MAJOR-CAT= BB MINOR-CAT= 140
SNM-NUM= M-0305 COR-TYPE= X NBR-MOU= X DOC-UP=
CM-CH=
DB-CH=
EXPLAN=
DATE=CLOSE= 6-06-73
MOD=CLOSE= 0
DA=CLOSE= 08
YR=CLOSE= 73
PROJ-VERSION= BLK2
PROJ-ID= 03
CODE-COUNT=
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.

| A3 | 30 0194A313 | 6-05-732 V | BB140M-0300 X | 6-05-73BLK2 03 |
| F5 | 0195F504 | 6-04-73 V | BB170M-0301X | 6-05-73BLK2 03 |
| C2 | 0209C210 | 6-06-732 V | BB060M-0361X | 6-08-73BLK2 03 |
| G1 | 0211G131 | 6-06-73 V | BB060M-0371XX | 6-11-73BLK2 03 |
| C2 | 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
TESI-CHAR?
- 5
NUM-RUNS-TOT?
- 179
NUM-RUNS-OK?
- 143
AHRS-PER-TEST?
- 9
BLOCK?
- Y
TEST-PER?
- IMCT
OKAY?
- Y
ENTRY STORED
NEXT ENTRY
?
- Y
TEST-CHAR?
- 5
NUM-RUNS-TOT?
- 320
NUM-RUNS-OK?
- 293
AHRS-PER-TEST?
- 7
BLOCK?
- O
TEST-PER?
- IMCT
OKAY?
- Y
ENTRY STORED
NEXT ENTRY
?
- Y
TEST-CHAR?
- 5
NUM-RUNS-TOT?
- 320
NUM-RUNS-OK?
- 293
AHRS-PER-TEST?
- 7
BLOCK?
- O
TEST-PER?
- IMCT
OKAY?
- Y
ENTRY STORED
NEXT ENTRY
?
- Y
TEST-CHAR?
- 5
NUM-RUNS-TOT?
- 320
NUM-RUNS-OK?
- 293
AHRS-PER-TEST?
- 7
BLOCK?
- O
TEST-PER?
- IMCT
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
0030 COVER PAGE IS COV1
0040 PAGE HEADING IS HD1
0050 PAGE FOOTING IF FT1
0060 COV1, SPACE 15
0070 LINE "PROJECT 5 DATA" COL 60
0080 SPACE 2
0090 LINE "FOR 1974 ONLY" COL 60
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
0170 SPACE TOP

(continued)
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 — System variable in footer line.
0315   DEFINE $DIFF,$NODAYS2,$OPEN WITH PIC "999"
0316   DEFINE $AVEDAYS,$SAVEOPEN 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,
0350   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
0415   ONLY 45 ENTRIES
0416   WHEN A2
0420   SORT PROBLEM-REPORT ON MO-OPEN - 30 + DA-OPEN — Sort on date.
0430   IF $SW = 1 GO TO L1 — Branching logic.

(continued)
0440 LET $NODAYS2 = 30 + MO-OPEN - 30 + DA-OPEN
0450 LET $SW = 1
460 LET $NODAYS1 = 30 + MO-OPEN + DA-OPEN
0470 LET $DIFF = $NODAYS2 - $NODAYS1
0480 LET $OPEN = (YР-CLOSE - 74) + 365
0490 + MO-CLOSE + 30 - 30 + DA-CLOSE
0500 - (MO-OPEN + 30 - 30 + DA-OPEN)
0502 LET $NOENT + COUNT DA-OPEN

Computations

0480 LET $OPEN = (YR-CLOSE - 74) + 365
0490 + MO-CLOSE + 30 - 30 + DA-CLOSE
0500 - (MO-OPEN + 30 - 30 + DA-OPEN)

Causes data line to be printed.

0502 LET $NOENT + COUNT DA-OPEN

Control logic

0440 LET $NODAYS2 = 30 + MO-OPEN - 30 + DA-OPEN
0450 LET $SW = 1

Control logic

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

Demonstrates sum.
Figure 1: Cover Page for Formatted Report
### PROJECT B DATA - 1976

<table>
<thead>
<tr>
<th>DATE BETWEEN</th>
<th>DATE-OPEN</th>
<th>NO. OF DATE</th>
<th>MODEL</th>
<th>ERROR CATEGORY</th>
<th>MAJOR.Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>006</td>
<td>03/04/76</td>
<td>003</td>
<td>CSG8EX</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>007</td>
<td>03/04/76</td>
<td>007</td>
<td>CSG8EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>009</td>
<td>03/12/76</td>
<td>022</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>011</td>
<td>03/18/76</td>
<td>027</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>003</td>
<td>03/28/76</td>
<td>002</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>003</td>
<td>03/31/76</td>
<td>026</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>003</td>
<td>04/02/76</td>
<td>031</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>017</td>
<td>04/02/76</td>
<td>028</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>003</td>
<td>04/22/76</td>
<td>019</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>009</td>
<td>05/03/76</td>
<td>016</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>006</td>
<td>05/12/76</td>
<td>035</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>009</td>
<td>05/12/76</td>
<td>016</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>05/17/76</td>
<td>137</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>05/24/76</td>
<td>127</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>06/02/76</td>
<td>115</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>06/02/76</td>
<td>127</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>06/11/76</td>
<td>225</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>06/12/76</td>
<td>234</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>008</td>
<td>06/12/76</td>
<td>234</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>06/12/76</td>
<td>234</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>010</td>
<td>06/30/76</td>
<td>225</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>006</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>006</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>006</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
<tr>
<td>004</td>
<td>07/01/76</td>
<td>120</td>
<td>CSG97EH</td>
<td>SX</td>
<td>001</td>
</tr>
</tbody>
</table>

Figure 2: Formatted Report
Reports with format control can also be directed to a perm-
file 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

(continued)
0300 RETRIEVE PROBLEM-REPORT FROM DBRDS1
0301 WHERE SPR-CHAR = "2"
0310 IF TEST-PER = "D" LET $T1 = $T1 + 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 = "O" 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
0540 PRINT PRT4
0545 PRINT PRT4A
0550 PRINT PRT5
0560 END

The result printed on the terminal is shown below.

PROJECT 3
ERRORS BY TEST-PER

<table>
<thead>
<tr>
<th>DEVELOPMENT</th>
<th>NO OF ERRORS</th>
<th>PERCENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>VALIDATION</td>
<td>0000</td>
<td>00.00</td>
</tr>
<tr>
<td>INTEGRATION</td>
<td>1984</td>
<td>91.64</td>
</tr>
<tr>
<td>ACCEPTANCE</td>
<td>0019</td>
<td>00.88</td>
</tr>
<tr>
<td>OPERATIONAL</td>
<td>0162</td>
<td>07.48</td>
</tr>
</tbody>
</table>

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
0070 PRINT 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.
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

<table>
<thead>
<tr>
<th>MAJOR-CAT</th>
<th>%208</th>
<th>%201</th>
<th>%203</th>
<th>%207</th>
</tr>
</thead>
<tbody>
<tr>
<td>C101</td>
<td>0197C101</td>
<td>6-05-73 V</td>
<td>AAO4OM-032X</td>
<td>6-07-73BLK2 03</td>
</tr>
<tr>
<td>C101</td>
<td>0198F412</td>
<td>6-05-73 V</td>
<td>AAO4OM-030X</td>
<td>6-06-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0234A203</td>
<td>6-07-73 V</td>
<td>AAO5OM-0362X</td>
<td>6-08-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0241C302</td>
<td>6-07-73 V</td>
<td>AAO2OM-0412</td>
<td>6-13-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0242C302</td>
<td>6-07-73 V</td>
<td>AAO2OM-0412</td>
<td>6-13-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0256C102</td>
<td>6-08-73 V</td>
<td>AAO2OM-0351X</td>
<td>6-08-73BLK2 03</td>
</tr>
<tr>
<td>B205</td>
<td>0259B205</td>
<td>6-07-73 V</td>
<td>AAO4OM-0355X</td>
<td>6-08-73BLK2 03</td>
</tr>
<tr>
<td>B104</td>
<td>0260B104</td>
<td>6-06-7312 V</td>
<td>AAO4OM-0357X</td>
<td>6-08-73BLK2 03</td>
</tr>
<tr>
<td>B205</td>
<td>0305F301</td>
<td>6-11-73 V</td>
<td>AAO1OM-0378X</td>
<td>6-11-73BLK2 03</td>
</tr>
<tr>
<td>B205</td>
<td>0350F407</td>
<td>6-13-7311 V</td>
<td>AAO8OM-0492 XX</td>
<td>6-19-73BLK2 03</td>
</tr>
<tr>
<td>B205</td>
<td>0368B105</td>
<td>6-13-73 V</td>
<td>AAO3OM-0415X</td>
<td>6-14-73BLK2 03</td>
</tr>
<tr>
<td>B205</td>
<td>0373B204</td>
<td>6-14-7311 V</td>
<td>AAO4OM-0459X</td>
<td>6-15-73BLK2 03</td>
</tr>
<tr>
<td>A208</td>
<td>0381A208</td>
<td>6-14-7311 V</td>
<td>AAO1OM-0616 EX</td>
<td>6-26-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0415C102</td>
<td>6-19-7311 V</td>
<td>AAO8OM-0487X</td>
<td>6-15-73BLK2 03</td>
</tr>
<tr>
<td>G107</td>
<td>0430G107</td>
<td>6-18-73 V</td>
<td>AAO9OM-0810X</td>
<td>7-06-73BLK2 03</td>
</tr>
<tr>
<td>C102</td>
<td>0438C102</td>
<td>6-18-7311 V</td>
<td>AAO1OM-</td>
<td></td>
</tr>
</tbody>
</table>
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 PROGO01 1E MEDIUM
RAY01 PROGO02 0K MEDIUM
RAY01 PROGO05 0C MEDIUM
RAY01 PROGO06 0B COMPLEX
RAY01 PROGO07 2K MEDIUM
RAY01 PROGO08 0B MEDIUM
RAY01 PROGO09 2J MEDIUM
RAY01 PROGO11 4D MEDIUM
RAY01 PROGO12 2G SIMPLE
RAY01 PROGO13 2D MEDIUM
RAY01 PROGO14 0C MEDIUM
RAY01 PROGO15 1B MEDIUM
RAY01 PROGO16 1C SIMPLE
RAY01 PROGO17 3C MEDIUM
RAY01 PROGO18 0A MEDIUM
```

(continued)
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?
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
PROG. NAME? TLU
$ IDENT?BFCBMIS1,C CURTIS,555008570052 -User's ident.
PROCEDURAL REF. NAME-ABBR-TEST
NAME ON LIBRARY- TESTLU
INDEX OR LOOKUP- LOOK
INPUT SIZE- 2
OUTPUT SIZE- 12
CORE OR FILE- CORE
TABLE SOURCE ON FILE? NO
IN- D
OUT- DEVELOPMENT
IN- V
OUT- VALIDATION
IN- A
OUT- ACCEPTANCE
IN- I
OUT- INTEGRATION
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$ FILEEDIT SOURCE,OBJECT,INITIALIZE
30$ FILE RH,A1S,20L
40$ DATA HC,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
140 TESTLU 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 STARTBL 2, 12
330 PAIR (D), (DEVELOPMENT)
340 PAIR (V), (VALIDATION)
350 PAIR (A), (ACCEPTANCE)
360 PAIR (I), (INTEGRATION)
370 PAIR (O), (OPERATIONAL)
380 ENDTBL
390 LOOKUP BFCBMIS1,
400 INOP L, (ERROR)
410 END
420$ EXECUTE
430$ ENDLD
440$ ENDEDIT
450$ ENDCOPY
460$ SYSEEDIT
470 ENDFILE/TESTLU
480$ FILE R*, AIR, 20L
490$ PRMFL Q*, W, R, BFCBMIS1/TLU TEST
500$ ENDO
A procedure demonstrating the decoding process follows.

0010 INVOKE ADOTRWS
0020 LIBRARY 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** Indicates use of table lookup for output.

0060 END

A partial list of the decoded output follows.

<table>
<thead>
<tr>
<th>MOD-CHANGED</th>
<th>SEVERITY</th>
<th>TEST-PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>C101</td>
<td></td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>H215</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>H211</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>D/B</td>
<td>1</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>H208</td>
<td></td>
<td>OPERATIONALD</td>
</tr>
<tr>
<td>C104</td>
<td>1</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>H231</td>
<td>1</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>E104</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>E104</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>A403</td>
<td>3</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>D108</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>D412</td>
<td>3</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>D109</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>E109</td>
<td>2</td>
<td>OPERATIONALD</td>
</tr>
<tr>
<td>E109</td>
<td>2</td>
<td>OPERATIONALD</td>
</tr>
<tr>
<td>C104</td>
<td>4</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>C113</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>H215</td>
<td>2</td>
<td>INTEGRATION</td>
</tr>
<tr>
<td>A104</td>
<td>1</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>F506</td>
<td>2</td>
<td>VALIDATION</td>
</tr>
<tr>
<td>A303</td>
<td></td>
<td>VALIDATION</td>
</tr>
<tr>
<td>C106</td>
<td></td>
<td>VALIDATION</td>
</tr>
</tbody>
</table>
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.

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.

![Translation Flow Diagram]

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.
The Project Suffixes were established as follows.

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>SUFIX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TRWS</td>
</tr>
<tr>
<td>2</td>
<td>BS1S1</td>
</tr>
<tr>
<td>3</td>
<td>SDS1</td>
</tr>
<tr>
<td>4</td>
<td>APS1</td>
</tr>
<tr>
<td>5</td>
<td>SAFS1 &amp; SAFSN (two files)</td>
</tr>
<tr>
<td>6</td>
<td>SMTC</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>DEFINITION</th>
<th>JCL</th>
<th>SOURCE</th>
<th>OBJECT</th>
<th>TRANS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directory</td>
<td>JDITRWS</td>
<td>DISTRWS</td>
<td>DIOTRWS</td>
<td>TDIITRWS</td>
</tr>
<tr>
<td>Data</td>
<td>JDDTRWS</td>
<td>DDSTRWS</td>
<td>DDOTRWS</td>
<td>TDITRWS</td>
</tr>
<tr>
<td>Application</td>
<td>JADTRWS</td>
<td>ADSTRWS</td>
<td>ADOTRWS</td>
<td>TADTRWS</td>
</tr>
<tr>
<td>User-Subr Lib</td>
<td>ULSTRWS</td>
<td>ULOTRWS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure A-2: Dataset Naming Form
Step 2. Data Definition Source Code.

```
0020:MD DBTRWS:FILE IS SEQUENTIAL.          -Defines file structure.
0020:01 PROBLEM-REPORT;TYPE IS "03" IN PROJ-ID
0030:::RETRIEVAL VIA SCAN.                  -Defines retrieval method.
0060:02:MOD-CHANGED:PIC X(7).             -Defines entry name.
0070:02:SPR-NUM:PIC X(4) JUSTIFIED RIGHT. -Defines retrieval method.
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-VAL:PIC XX JUSTIFIED RIGHT. -Cobol-like description of data.
0170: 02:TEST-IDENT:PIC X(8).
0180: 02:ERROR-CAT.
0182: 03:MAJOR-CAT:PIC X(2).
0183: 03:MINOR-CAT:PIC X(3).
0192: 03:SMN:PIC X(6) JUSTIFIED RIGHT.     -Defines record type.
0195: 02:COR-TYPE.
0200: 03:NEW:PIC X.
0210: 03:DOC:PIC X.
0220: 03:COM:PIC X.
0230: 03:DBM: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:PIC X.
0340: 02:DAYS-OPEN:PIC XXX JUSTIFIED RIGHT.     -Defines location of data to be used.
0341: 02:FILLER PIC X(10).
0350:98:SYSTEM DETAIL;SCAN ON PROJ-ID. -Defines entry record.
```

Step 3. Application Definition Code.

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

A-5
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.

```jcl
SYSTMP?PERFORM
FUNCTION? BUILDDBAJCL
IDENTIFICATION. CURTIS.055006570052
DBAJCL FUNCTION= LIST,DIRJCL,DDLJCL,ADFJCL,DAJCL,DIRJCL

*MGDS DIRECTORY DEFINITION*
SOURCE DIRECTORY DEFINITION ON PERM FILE? YES
CATALOG NAME OF SOURCE DIR-
SUBCATALOG NAME OF SOURCE DIR-
FILE NAME OF SOURCE DIR- DISIRMS
PASSWD-
*C:B*
CREATE OBJECT DIR FILE? YES
LOG-ON PASSWORD-
PASSWD-
*C:B*
CATALOG NAME OF OBJECT DIR-
SUBCATALOG NAME OF OBJECT Dir-
FILE NAME OF OBJECT DIR - DISIRMS
PASSWD-
*C:B*
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- TDIRSMS
PASSWD-
*C:B*
CREATE TRANSLATION REPORT FILE? YES
MAX FILE SIZE IN LLINKS- 20
FOLLOW WITH DOLJCL? YES

*MGDS DATA DEFINITION*
SOURCE DATA DEFINITION ON PERM FILE? YES
CATALOG NAME OF SOURCE DDL-
SUBCATALOG NAME OF SOURCE DDL-
FILE NAME OF SOURCE DDL- DISIRMS
PASSWD-
*C:B*
CREATE OBJECT DDL FILE? YES
CATALOG NAME OF OBJECT DDL-
SUBCATALOG NAME OF OBJECT DDL-
FILE NAME OF OBJECT DDL- DISIRMS
PASSWD-
*C:B*
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- TDIRSMS
```

A-6
The following listing is the resultant job stream that was created by the above interactive procedure.

```plaintext
#NORM
$ IDENT BFCBMIS1.C CURTIS,555008570052
$ FILSYS
USERID BFCBMIS1$P
FCREAT BFCBMIS1/DIOTRWS,MODE/RAND/,BLOCKS/10,10/
FCREAT BFCBMIS1/TDIIRWS,MODE/SEQ/,BLOCKS/1,10/
$ PROGRAM DIRXLT,DUMP
$ LIMITS 10,10K
$ PRMFL TR,R/W,S,BFCBMIS1/TDIIRWS
$ PRMFL DF,R/W,R,BFCBMIS1/DIOTRWS
$ DATA I*
$ SELECTA BFCBMIS1/DISTRWS
$ FILSYS
USERID BFCBMIS1$P
FCREAT/IDS/ BFCBMIS1/DDOSTRWS,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/DDOSTRWS
$ DATA I*
$ SELECTA BFCBMIS1/DDSTRWS
$ PROGRAM QUTD
$ LIMITS ,20K
$ PRMFL A1,R,R,BFCBMIS1/DDOSTRWS
$ 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 DISIRWS

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

DATA DEFINITION

*LIST DDSTRWS

0020*MD DBRTRWS*FILE IS SEQUENTIAL.
0020*100 PROBLEM-REPORT-TYPE IS "03" IN PROJ-ID
0030*100 RETRIEVAL VIA SCAN.
0060* 02#MOD-CHANGED*PIC X(7).
0070* Q2*SPR-NUM PIC X(4) JUSTIFIED RIGHT.
0080* 02#MOD-SOURCE*PIC X(7).
0090* 02#DATE-OPEN.
0100* 03#MOD-OPEN*PIC X.
0110* 03 FILLER PIC X.
0120* 03 DA-OPEN*PIC X.
0130* 03 FILLER PIC X.
0140* 03 YR-OPEN PIC X.
0150* 02#SEVERITY*PIC X.
0160* 02#TEST-PER*PIC XX JUSTIFIED RIGHT.
0170* 02#TEST-ID*PIC X.
0180* 02#ERROR-CAT.
0182* 03 MAJOR-CAT*PIC X.
0183* 03 MINOR-CAT*PIC X.
0192* 03#SMN-NUM*PIC X JUSTIFIED RIGHT.
0195* 02#COR-TYPE.
0200* 03#NEW-MOD*PIC X.
0210* 03#NEW-UP*PIC X.
0220* 03#CH-CH*PIC X.
0230* 03#DB-CH*PIC X.
0240* 03#EXPLAN*PIC XX.
0250* 02#DATE-CLOSE.
0260* 03#MOD-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:  08*SYSTEM DETAIL*SCAN ON PROJ-ID.

APPLICATION DEFINITION

*LIST ADSTIRMS

0010 DATA-BASE IS DBGTRMS IN BFCBMIS/DIOIRMS
0030 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT

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

LIST DISBISI

10 DATA-REFERENCE IS DISBISI
20 SEQUENTIAL-DATA-BASE IS BFCMISI/DBNISI
30 DEFINITION-SOURCE-FILE IS BFCMISI/DBNISI
40 DEFINITION-OBJECT-FILE IS BFCMISI/DBNISI
50 USER-SUBROUTINE-LIBRARY IS BFCMISI/ILSISI
60 END DBNISI

DATA DEFINITION

LIST DOSBISI

0020:MD DISBISI;FILE IS SEQUENTIAL.
0030:01 PROBLEM-REPORT; TYPE IS "1" IN SPR-CHAR
0040:01 RETRIEVAL VIA SCAN.
0050:02 SPR-CHAR:PICTURE X.
0060:02 SYS-ID:PIC X.
0070:02 FILLER:PIC X.
0080:02 DATE-OPEN.
0110:02 DATE-CLOSE.
0120:03 HO-CLOSE:PIC 99.
0130:03 DA-CLOSE:PIC 99.
0140:03 TR-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-CNT: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.
0280:02 HRS-TO-TERM:PIC X(27).
0290:02 FILLER:PIC X(3).
0300:02 HRS-TO-FIX:PIC 9(5).
0310:02 FILLER:PIC X(3).
0320:02 FILLER:PIC X(3).
0330:02 FILLER:PIC X(2).
0340:02 MOD-CHANGED:PIC X(27).
0341:02 FILLER:PIC X(4).
0345:02 SYSTEM-DETAIL:SCAN ON SPR-CHAR.
0350:01 MODULE;TYPE IS "2" IN MOD-CHAR
0360:01 RETRIEVAL VIA SCAN.
0370:02 MOD-CHAR:PIC X.
0390:02 FILLER:PIC X.
0400:02 FILLER:PIC X.
0410:02 MOD-ID:PIC X(4).
0410:02 FILLER:PIC X.

Figure A-4: Project Two Definitions
DATA DEFINITION (cont'd)

0430: 02:MOD-LANG:PIC X.
0440: 02:MOD-SIZE:PIC 9(5).
0450: 02:FILLER:PIC X.
0460: 02:MOD-LANG:PIC X.
0470: 02:MOD-SIZE:PIC 9(5).
0471: 02:FILLER:PIC X(63).
0473:98:SYSTEM DETAIL:SCAN ON MOD-CHAR.
0480:01 HARDWARE;TYPE IS "3" IN HARD-CHAR
0490:::RETRIEVAL VIA SCAN;
0500: 02:NARD-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(6).
0560: 02:COMP-OS:PIC X(13).
0561: 02:FILLER:PIC X(64).
0563:98: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:SYS-ID:PIC X.
0610: 02:FILLER:PIC X(9).
0620: 02:TECH-ID:PIC X(11).
0630: 02:FILLER:PIC X.
0640: 02:SOFT-LANG:PIC X.
0650: 02:SOFT-SIZE:PIC X(5);JUSTIFIED RIGHT.
0660: 02:FILLER:PIC X.
0670: 02:SOFT-LANG:PIC X.
0680: 02:SOFT-SIZE:PIC X(5);JUSTIFIED RIGHT.
0690: 02:FILLER:PIC X(68).
0685:98: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-OR: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-NAME IS DBMISI IN MFCWISIS/DBMISI
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 DBRSSDS1
20 SEQUENTIAL-DATA-BASE IS BFCBMI/SDBRSSDS1
30 DEFINITION-SOURCE-FILE IS BFCBMI/DDRSSDS1
40 DEFINITION-OBJECT-FILE IS BFCBMI/DDRSSDS1
50 USER-SUBROUTINE-LIBRARY IS BFCBMI/ULRSSDS1
60 END DBRSSDS1

DATA DEFINITION

**LIST DISSDS1**
20:00 DBRSSDS1:FILE IS SEQUENTIAL.
30:01 MODULE;TYPE IS "1" IN MOD-CHAR;
40:00 RETRIEVAL VIA SCAN.
50:00 02:MOD-CHAR:PIC X.
60:00 02:PROJ-ID:PIC X(5).
70:00 02:PROJ-CODE:PIC XX.
80:00 02:MOD-ID:PIC X(7).
90:00 02:MOD-VERSION:PIC XX.
100:00 02:MOD-TYPE:PIC X.
110:00 02:COMPLEXITY:PIC X.
120:00 02:CONST-TYPE:PIC X.
130:00 02:NUM-OCURR-S:PIC X(5);JUSTIFIED RIGHT.
140:00 02:NUM-OCURR-O:PIC X(5);JUSTIFIED RIGHT.
150:00 02:TECH-ID:PIC 9.
151:00 02:FILLER PIC X(53).
160:98:SYSTEM DETAIL;SCAN ON MOD-CHAR.
170:01 PROBLEM-REPORT;TYPE IS "2" IN SPR-CHAR;
180:00 RETRIEVAL VIA SCAN.
190:00 02:SPR-CHAR:PIC X.
200:00 02:PROJ-ID:PIC X(5).
210:00 02:PROJ-CODE:PIC XX.
220:00 02:SPR-NU:PIC X(4);JUSTIFIED RIGHT.
230:00 02:MOD-CHAR:PIC X(7).
240:00 02:MOD-CH-VERS:PIC XX.
250:00 02:DATE-OPEN.
260:00 03:MOD-OPEN:PIC XX.
270:00 03:FILLER:PIC X.
280:00 03:DA-OPEN:PIC XX.
290:00 03:FILLER:PIC X.
300:00 03:YR-OPEN:PIC XX.
310:00 02:TYPE-TERM:PIC X.
320:00 02:SEVERITY:PIC X.
330:00 02:TEST-TERM:PIC X.
340:00 02:ERROR-CAT.
350:00 03:MAJOR-CAT:PIC XX.
360:00 03:MINOR-CAT:PIC XXX.
370:00 02:SHR-NU:PIC X(4).
380:00 02:CHAR-TYPE:PIC X(3).
390:00 02:DATE-CLOSE.
400:00 03:MOD-CLOSE:PIC XX.
410:00 03:FILLER:PIC X.
420:00 03:DA-CLOSE:PIC XX.
430:00 03:FILLER:PIC X.
440:00 03:YR-CLOSE:PIC XX.
450:00 02:DAT-OPEN:PIC XXX;JUSTIFIED RIGHT.
451:00 02:FILLER PIC X(27).
460:98:SYSTEM DETAIL; SCAN ON SPR-CHAR.

APPLICATION DEFINITION

**LIST ADSSDS1**
10 DATA-BASE IS DBRSSDS1 IN BFCBMI/ADRSSDS1
20 ENTRY MODULE IS RECORD MODULE.
30 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT

Figure A-5: Project Three Definitions
DIRECTORY DEFINITION

\%LIST DDSAPS1
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/DDOAPS1
020 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT

Figure A-6: Project Four Definitions
DIRECTORY DEFINITION

\%LIST DISSAFS1
0010 DATA-BASE-REFERENCE IS DBRSAFS1
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBSAFS1
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/DBSAFSN
0080 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSAFSN
0090 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSAFSN
0100 END DBRSAFSN

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

\%LIST DDSSAFSN
0380: MD DBRSAFSN;FILE IS SEQUENTIAL.
0390: PROBLEM-REPORT;RETRIEVAL VIA SCAN.
0410: MOD-ID.
0411: SSYS-ID PIC X.
0412: SSYS-VERSION PIC X(7).
0430: NUM-OCCUR PIC 9(6).
0450: CONST-TYPE PIC X(7).
0470: TECH-ID PIC X(12).
0480: SYSTEM DETAIL; SCAN ON MOD-CHANGED.

APPLICATION DEFINITION

\%LIST ADSSAFS1
0490 DATA-BASE IS DBRSAFS1 IN BFCBMIS1/DDOSAFS1
0500 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
0510 DATA-BASE IS DBRSAFSN IN BFCBMIS1/DDOSAFS1
0530 ENTRY MODULE IS RECORD MODULE

Figure A-7: Project Five Definitions

A-14
DIRECTORY DEFINITION

LIST DISSMT
0010 DATA-BASE-REFERENCE IS DBRSMTC
0020 SEQUENTIAL-DATA-BASE IS BFCBMIS1/DBNSMTC
0030 ON TAPE 44916
0040 DEFINITION-SOURCE-FILE IS BFCBMIS1/DDSSMT
0050 DEFINITION-OBJECT-FILE IS BFCBMIS1/DDOSMT
0060 USER-SUBROUTINE-LIBRARY IS BFCBMIS1/TLUSMT
0070 END DBRSMTC

DATA DEFINITION

LIST DDSSMT
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:TIME 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 ADSSMT
0110 DATA-BASE IS DBRSMTC IN BFCBMIS1/DIOSMT
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$:COBOL:LSIU,ON6,NCOMDK,NDECK
0040$: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".
```

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 TESLIB
028-READ LINKED FILES ONLY WITH THIS COMMAND
SYSTEM ?CARD NEW
READY
*LIST JLIBSDS1

10$ IDENT BFCBMIS1,C CURTIS,555008570052
20$ FILEEDIT INITIALIZE,SOURCE,OBJECT
30$ FILE R*,AIR,10L
40$ FILE R*,A2C,10L
50$ DATA *C,COPY
60$ INCLUDE SOURCE
70$ SYSLD CATALOG=DMSCST1,RELOC,MASER
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 BXNL X
180S BXNL 0
190 TTLS CODE-ANS
200 REM 5, CODE-ANS
210 BCI 5,VALUES
220 BCI 1,VALUES
230 VFD 1/0,17/4056,018/22
240 VFD 1/1,1/5,1/0,1/0,12/01,2/6,12/0001,6/000
250 VFD 1/1,1/5,1/1,1/1,12/01,2/6,12/0007,6/000
260 OCT 555555555555
270 END
280$ EXECUTE
290$ ENOLD
300$ SYSLD CATALOG=VALUES,RELOC,MASER
310$ OPTION NOSETU,NOFCB
320$ LOWLOAD
330$ SELECTA BFCBMIS1/COMPLE
340$ EXECUTE
350$ ENLD
360$ ENEDIT
370$ ENDCOPY
380$ SYSEDIT INITIALIZE
390$ FILE R*,A2R,10L
400$ PRMFL Q*,M,R,BFCBMIS1/USLSDS1
410$ ENDDO

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.

```
0380:01 MD BRSAFSPN; 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.
```

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=URNSAPSN
ORIGINATOR=RFC\AMIS1
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=NULL
```

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

A-18
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.

```plaintext
$ 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*,XIR,30L
$ EXECUTE DUMP
$ LIMITS ,32K
$ PRMFL F4,R,S,BFCBMIS1/DBNSAFS2
$ PRMFL F5,R/W,S,BFCBMIS1/DBNSAFSN
$ ENDDJ
```

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
0500 ENTRY PROBLEM-REPORT IS RECORD PROBLEM-REPORT
0510 DATA-BASE IS DBRSAFSN IN BFCBMIS1/DIOSAFSN
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-SD: 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.

A-20
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)
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.
Figure B-1: Component Data Summary Form
**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.

5.2 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
0030 PAGE 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
0070 P1. 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 COK-TYPE EQ "C"
0220 PRINT P3.
0230END

<table>
<thead>
<tr>
<th>MOD ID</th>
<th>INSTR TYPE</th>
<th># SOURCE</th>
<th># ERRORS</th>
<th>ERR CAT</th>
<th>#CORRECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAI</td>
<td>ASSM</td>
<td>2282</td>
<td></td>
<td>BB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>DD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>KK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>RR</td>
<td></td>
</tr>
<tr>
<td>AI</td>
<td>ASSM</td>
<td>2830</td>
<td></td>
<td>AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JOVL</td>
<td>7223</td>
<td></td>
<td>BB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>CC</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>20</td>
<td>DD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>FF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>GG</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>JJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>KK</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>LL</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18</td>
<td>MM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>NN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>PP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>7</td>
<td>RR</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>SS</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>TT</td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td>JOVL</td>
<td>6897</td>
<td></td>
<td>AA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ASSM</td>
<td>600</td>
<td></td>
<td>BB</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>77</td>
<td>DD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>36</td>
<td>FF</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>GG</td>
<td></td>
</tr>
</tbody>
</table>

B-6
<table>
<thead>
<tr>
<th>PROJECT ID</th>
<th>SYSTEM ID</th>
<th>FUNC GP ID</th>
<th>MODULE ID</th>
<th>VERSION ID</th>
<th>RECORD TYPE</th>
<th>RECORD NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>4</td>
<td>5</td>
<td>9</td>
<td>1</td>
<td>050</td>
<td>3</td>
</tr>
</tbody>
</table>

**SUMMARIZATION LEVEL**

- P: Project
- F: Func Gp
- S: System
- M: Module

**TEST PERIOD**

- D: Development
- V: Validation
- A: Acceptance
- I: Integration

**ERROR CATEGORY TYPE**

- 6

**NUMBER OF ERRORS**

- 3

**ERROR CATEGORY**

- 2

**SUMMARY INSTRUCTIONS**

**REPORTING LEVEL**

- MD: Monthly
- PH: Phase
- T: Total

**Figure B-4: Errors Data Summary Form**
<table>
<thead>
<tr>
<th>PROJECT ID</th>
<th>SYSTEM ID</th>
<th>FUNC GR. ID</th>
<th>MODULE ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SUMMARY LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>P Project</td>
</tr>
<tr>
<td>S System</td>
</tr>
<tr>
<td>M Module</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VERSION ID</th>
<th>RECORD TYPE</th>
<th>RECORD NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>O60</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TEST PERIOD</th>
<th>REPORTING LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>D Development</td>
<td>MO Monthly</td>
</tr>
<tr>
<td>V Validation</td>
<td>PH Phase</td>
</tr>
<tr>
<td>A Acceptance</td>
<td>T Total</td>
</tr>
<tr>
<td>I Integration</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BEGIN DATE</th>
<th>END DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CORRECTION TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AVERAGE NUMBER DAYS OPEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CPU HUNDREDS HOURS TO FIX (AVERAGE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NUMBER OF CORRECTIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Figure B-5:** Corrections Data Summary Form
**LIST DISSUM**

1130 DATA-BASE-REFERENCE IS DISSUM
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 BFCBMIS1/DDOHTRI
1300 ISP-INDEX-FILE IS BFCBMIS1/INDHTRI
1310 TRANSACTION-REJECT-FILE IS BFCBMIS1/TRHTRI
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/TRHTRS
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.
230: 03 YR-DA-I PIC 99.
240: 02 DATA-DATE-C.
250: 03 MO-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)
Figure B-8: Summary Database MDQS Definitions (Cont'd)
940: 03 BEGIN-DATE-C.
950: 04 MO-REG-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.
1020: 02 COR-TYPE PIC X(3).
1030: 02 DAYS-OPEN-AV-C PIC 9(3).
1040: 02 HRS-TO-FIX-AV PIC 9(5).
1050: 02 NUM-COR PIC 9(5).
1060: 98 COMP-COR DETAIL;
1070: ASCENDING KEY IS COR-REC--.
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-NID 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--.
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)
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 RESOURCES; 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).
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.

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

<table>
<thead>
<tr>
<th>Derived Data</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Productivity</td>
<td>$PROD = \frac{DSLOC}{TMM}$</td>
</tr>
<tr>
<td>Average People</td>
<td>$PEO = \frac{TMM}{TM}$</td>
</tr>
<tr>
<td>Error Rate</td>
<td>$SPR-RATE = \frac{SPR's}{DSLOC} \times 100$</td>
</tr>
<tr>
<td>Documentation Rate</td>
<td>$DOC-RATE = \frac{DOC}{DSLOC}$</td>
</tr>
<tr>
<td>Design Lang. Rate</td>
<td>$PDL-RATE = \frac{PDL}{DSLOC}$</td>
</tr>
</tbody>
</table>

---

**Table: REC-A**

<table>
<thead>
<tr>
<th>ID</th>
<th>Record Type</th>
<th>REF</th>
<th>PROG-NAME</th>
<th>COMPANY</th>
<th>LANG-1</th>
<th>PER-CENT-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>11</td>
<td>11</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table: REC-B**

<table>
<thead>
<tr>
<th>ID</th>
<th>Record Type</th>
<th>LANG-2</th>
<th>PER-CENT-2</th>
<th>LANG-3</th>
<th>PER-CENT-3</th>
<th>PDL</th>
<th>PDL-RATE</th>
<th>DOC</th>
<th>DOC-RATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>12</td>
<td>12</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
</tbody>
</table>

**Table: REC-C**

<table>
<thead>
<tr>
<th>ID</th>
<th>Record Type</th>
<th>DSLOC</th>
<th>TMM</th>
<th>PROD</th>
<th>SC</th>
<th>CPT</th>
<th>LIB</th>
<th>CR</th>
<th>SPR</th>
<th>SPR-RATE</th>
<th>PEO</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>
LIST DDSDN

0050 DATA-BASE = DBRDN; FILE IS SEQUENTIAL.
0060: DATA = A; TYPE IS "A" IN TYPE-A;
0070: RETRIEVAL VIA SCAN.
0080: FILLER PIC X(8).
0090: FILLER PIC X(4).
0100: FILLER PIC X.
0110: FILLER PIC XXX.
0120: FILLER PIC X.
0130: FILLER PIC X(28).
0140: FILLER PIC X.
0150: FILLER PIC X.
0160: FILLER PIC X.
0170: FILLER PIC X(11).
0180: FILLER PIC X.
0190: FILLER PIC XXX.
0200: FILLER PIC X(11).
0210: 98 REC-AREC-B MASTERS; ORDER IS SORTED.
0210: 98 SYSTEM DETAILED SCAN ON TYPE-A.
0220: 01 REC-B; TYPE IS "B" IN TYPE-B;
0230: RETRIEVAL VIA REC-AREC-B.
0240: FILLER PIC X(8).
0250: ID-B PIC X(4).
0260: TYPE-B PIC X.
0280: LANG-2 PIC X(12).
0290: FILLER PIC X.
0300: PER-CENT-2 PIC XXX.
0310: FILLER PIC X.
0320: LANG-3 PIC X(12).
0330: FILLER PIC X.
0340: PER-CENT-3 PIC XXX.
0350: FILLER PIC X.
0360: PER-CENT-3 PIC X(6).
0370: FILLER PIC X.
0380: PER-CENT-3 PIC X(5).
0390: FILLER PIC X.
0400: DOC PIC X(6).
0410: FILLER PIC X.
0420: DOC-RATE PIC X(6).
0430: FILLER PIC X.
0440: DOC-RECT PIC X(11).
0450: 98 REC-BREC-C MASTERS; ORDER IS SORTED.
0460: RETRIEVAL VIA REC-BREC-C.
0470: FILLER PIC X(8).
0480: ID-C PIC X(4).
0490: TYPE-C PIC X.
0500: DSLOC PIC X(7).
0510: FILLER PIC X.
0520: TM PIC X(5).
0530: FILLER PIC X.
0540: TM PIC XXX.
0550: FILLER PIC X.
0560: FILLER PIC X.
0570: FILLER PIC X.
0580: SC PIC XXX.
0590: FILLER PIC X.
0600: TDP PIC XXX.
0610: FILLER PIC X.
0620: FILLER PIC X.
0630: FILLER PIC X.
0640: LIB PIC X(3).

LIST ADSDN

0230: DATA-BASE IS DBRDN IN BFCMIS1/DDSN.
0240: ENTRY X IS RECORD REC-A AND REC-B AND REC-C.
0250: ITEMS ARE:
0260: 02 RECORD REC-A ALL ITEMS.
0270: FILLER PIC X.
0280: PER-CENT-1 INPUT-EDITED.
0290: REC-C-REC-B ALL ITEMS.
0310: FILLER PIC X.
0320: PER-CENT-2 INPUT-EDITED.
0330: FILLER PIC X.
0340: PER-CENT-3 INPUT-EDITED.
0350: FILLER PIC X.
0360: PER-CENT-3 INPUT-EDITED.
0370: PER-CENT-3 INPUT-EDITED.
0380: FILLER PIC X.
0390: PER-CENT-3 INPUT-EDITED.
0400: FILLER PIC X.
0410: PER-CENT-3 INPUT-EDITED.
0420: FILLER PIC X.
0430: PER-CENT-3 INPUT-EDITED.
0440: FILLER PIC X.
0450: PER-CENT-3 INPUT-EDITED.
0460: PER-CENT-3 INPUT-EDITED.
0470: FILLER PIC X.
0480: PER-CENT-3 INPUT-EDITED.
0490: FILLER PIC X.
0500: PER-CENT-3 INPUT-EDITED.
0510: FILLER PIC X.
0520: PER-CENT-3 INPUT-EDITED.
0530: FILLER PIC X.
0540: TM PIC XXX.
0550: FILLER PIC X.
0560: FILLER PIC X.
0570: FILLER PIC X.
0580: SC PIC XXX.
0590: FILLER PIC X.
0600: TDP PIC XXX.
0610: FILLER PIC X.
0620: FILLER PIC X.
0630: FILLER PIC X.
0640: LIB PIC X(3).

LIST DISDN

0010 DATA-BASE = REFERENCE IS DBRDN.
0020 SEQUENTIAL-DBASE IS BFCMIS1/DDSN.
0030 DEFINITION-SRC-FILE IS BFCMIS1/DDSN.
0040 DEFINITION-OBJ-FILE IS BFCMIS1/DDSN.

Figure C-2: MDQS Definitions - Productivity Database

C-3
0070 INVOKE ADODN
0075 DEFINE $DSL WITH PIC "9(14)"
0080 RETRIEVE X FROM DBPDN
0090 WHERE DSLOC GE 0
0100 AND SPR-RATE GE 0
0110 SORT X ON DSLOC
0120 PRINT ON FILE TESTDN FOR ITY
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 TESTDN FOR ITY $DSCOUNT,$DSL
0180 END

<table>
<thead>
<tr>
<th>DSLOC</th>
<th>SPR-RATE</th>
<th>SPR-ID-A</th>
<th>LANG-1</th>
<th>LANG-2</th>
<th>LANG-3</th>
</tr>
</thead>
<tbody>
<tr>
<td>115346</td>
<td>1.739</td>
<td>2006</td>
<td>85</td>
<td>JOVIAL J4</td>
<td>X</td>
</tr>
<tr>
<td>120000</td>
<td>1.697</td>
<td>2036</td>
<td>140</td>
<td>ASSY</td>
<td>JOVIAL J3B</td>
</tr>
<tr>
<td>136350</td>
<td>1.980</td>
<td>2673</td>
<td>81</td>
<td>CENTRAN</td>
<td>X</td>
</tr>
<tr>
<td>136689</td>
<td>1.584</td>
<td>2165</td>
<td>169</td>
<td>ASSY</td>
<td>JOVIAL J3</td>
</tr>
<tr>
<td>250000</td>
<td>0.414</td>
<td>1036</td>
<td>184</td>
<td>CMS-2</td>
<td>X</td>
</tr>
<tr>
<td>300000</td>
<td>1.083</td>
<td>3250</td>
<td>175</td>
<td>FORTRAN IV</td>
<td>ASSY</td>
</tr>
<tr>
<td>14500</td>
<td>0.593</td>
<td>86</td>
<td>88</td>
<td>ASSY</td>
<td>X</td>
</tr>
<tr>
<td>17500</td>
<td>0.251</td>
<td>44</td>
<td>93</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>18246</td>
<td>0.16</td>
<td>3</td>
<td>89</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>19045</td>
<td>1.691</td>
<td>322</td>
<td>206</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>22075</td>
<td>2.818</td>
<td>622</td>
<td>82</td>
<td>CENTRAN</td>
<td>X</td>
</tr>
<tr>
<td>22816</td>
<td>1.043</td>
<td>238</td>
<td>83</td>
<td>CENTRAN</td>
<td>X</td>
</tr>
<tr>
<td>290000</td>
<td>0.050</td>
<td>13</td>
<td>94</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>27055</td>
<td>1.918</td>
<td>519</td>
<td>159</td>
<td>ASSY</td>
<td>X</td>
</tr>
<tr>
<td>302000</td>
<td>0.522</td>
<td>167</td>
<td>192</td>
<td>ASSY</td>
<td>X</td>
</tr>
<tr>
<td>32400</td>
<td>0.600</td>
<td>214</td>
<td>77</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>36000</td>
<td>0.154</td>
<td>74</td>
<td>396</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>48000</td>
<td>0.154</td>
<td>74</td>
<td>396</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>54116</td>
<td>0.819</td>
<td>443</td>
<td>207</td>
<td>FORTRAN</td>
<td>ASSY</td>
</tr>
<tr>
<td>73000</td>
<td>0.514</td>
<td>375</td>
<td>193</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>78440</td>
<td>2.195</td>
<td>1726</td>
<td>97</td>
<td>FORTRAN</td>
<td>ASSY</td>
</tr>
<tr>
<td>81650</td>
<td>0.372</td>
<td>304</td>
<td>191</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>83324</td>
<td>0.055</td>
<td>46</td>
<td>23</td>
<td>PL/1</td>
<td>ASSY</td>
</tr>
<tr>
<td>83866</td>
<td>13.995</td>
<td>11739</td>
<td>168</td>
<td>ASSY</td>
<td>X</td>
</tr>
<tr>
<td>92931</td>
<td>1.545</td>
<td>1468</td>
<td>84</td>
<td>JOVIAL J4</td>
<td>X</td>
</tr>
<tr>
<td>1000</td>
<td>6.600</td>
<td>66</td>
<td>96</td>
<td>PL/1</td>
<td>X</td>
</tr>
<tr>
<td>2280</td>
<td>7.851</td>
<td>179</td>
<td>205</td>
<td>CUBOL</td>
<td>X</td>
</tr>
<tr>
<td>4023</td>
<td>8.476</td>
<td>341</td>
<td>87</td>
<td>JOVIAL J3</td>
<td>X</td>
</tr>
<tr>
<td>5100</td>
<td>0.948</td>
<td>5</td>
<td>90</td>
<td>CUBOL</td>
<td>X</td>
</tr>
</tbody>
</table>

Figure C-3: Query Example - Productivity Database

C-4
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.

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>LITTLE LINKS</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>301</td>
<td>96,320</td>
</tr>
<tr>
<td>2</td>
<td>120</td>
<td>38,400</td>
</tr>
<tr>
<td>3</td>
<td>150</td>
<td>48,000</td>
</tr>
<tr>
<td>4</td>
<td>514</td>
<td>164,220</td>
</tr>
<tr>
<td>5 file 1</td>
<td>320</td>
<td>102,400</td>
</tr>
<tr>
<td>5 file 2</td>
<td>160</td>
<td>51,200</td>
</tr>
<tr>
<td>6</td>
<td>120</td>
<td>38,100</td>
</tr>
</tbody>
</table>

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,216)
250 STOP
260 END

ready

8.3 Magnetic Tape Characteristics

The tape characteristics for each project are as follows.

<table>
<thead>
<tr>
<th>PROJECT</th>
<th>TAPE #</th>
<th>TRACK</th>
<th>BPI</th>
<th>FILE</th>
<th>SPECIAL CHARACTERISTICS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>46608</td>
<td>9</td>
<td>800</td>
<td>1,3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>44624</td>
<td>7</td>
<td>800</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>46608</td>
<td>9</td>
<td>800</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>44284</td>
<td>9</td>
<td>800</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>44314</td>
<td>9</td>
<td>800</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>44317</td>
<td>9</td>
<td>800</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

"$: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##
20$:IDENT:BFCBMIS1,C CURTIS,555008570052
30$:UTILITY:NDUMP
40$:TAPE9:27,A3D,,44624,,INPUT,,DEN8
55$:PRMFL:28,R/W,S,BFCBMIS1/DBNBL1S1
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


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 (C3I) 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.