TWO RELATIONAL DBMS: A COMPARISON

by

GARY F. GARTEN

B.S., Kansas State University, 1978
B.A., Kansas State University, 1979
B.S., Kansas State University, 1979

AN ABSTRACT OF A MASTER'S DISSERTATION

submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE, COMPUTER SCIENCE

College of Arts and Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987
Acknowledgments

A great many people have contributed in the production of this paper. Unfortunately, only a very few of those will be here.

My wife Becky. She finished her Master's Report from Kansas State University a few months earlier and knows the work involved in researching and writing one.

My parents Fred and Roberta. They saw me through several years of college and the completion of three undergraduate and one graduate degree at Kansas State University. Thank you for all that you did for me.

My major professor, Dr. Elizabeth Unger, Kansas State University Department of Computing and Information Sciences.

The kind and very helpful people at ADR and IBM who provided technical help with the many questions and issues I raised.
Brent Ahsmuhs, a fellow K-State alumni, and his DP Department manager at Forrest T. Jones, KCMO for all of the help with the DB2 miniature system.

And my cat Genny, who sat beside the PC for hours at a time while I worked tirelessly at the word processor typing the text of the paper.

Thanks to all of you. This paper could not have been written without you!
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acknowledgments</td>
<td>i</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Figures</td>
<td>v</td>
</tr>
<tr>
<td>Chapter 1: Introduction</td>
<td>1-1</td>
</tr>
<tr>
<td>Chapter 2: DB2 and Datacom/DB details</td>
<td>2-1</td>
</tr>
<tr>
<td>Chapter 3: Mini-system product comparison</td>
<td>3-1</td>
</tr>
<tr>
<td>3.0 Introduction</td>
<td>3-1</td>
</tr>
<tr>
<td>3.1 Program Conversion Problems</td>
<td>3-28</td>
</tr>
<tr>
<td>3.2 Program Timing Comparisons</td>
<td>3-34</td>
</tr>
<tr>
<td>Chapter 4: Conclusions and Future Work</td>
<td>4-1</td>
</tr>
<tr>
<td>4.0 Conclusions</td>
<td>4-1</td>
</tr>
<tr>
<td>4.1 Future Work</td>
<td>4-9</td>
</tr>
<tr>
<td>Bibliography</td>
<td>5-1</td>
</tr>
<tr>
<td>Appendix A: ADR Programs and Output</td>
<td>6-1</td>
</tr>
<tr>
<td>ADR - Batch Program Output</td>
<td>6-2</td>
</tr>
<tr>
<td>ADR - Batch Program</td>
<td>6-15</td>
</tr>
<tr>
<td>ADR - Online Program Output</td>
<td>6-33</td>
</tr>
<tr>
<td>ADR - Online Program</td>
<td>6-35</td>
</tr>
<tr>
<td>Appendix B: DB2 Programs and Output</td>
<td>7-1</td>
</tr>
<tr>
<td>DB2 - Batch Program Output</td>
<td>7-2</td>
</tr>
</tbody>
</table>

iii
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 - Batch Program</td>
<td>7-17</td>
</tr>
<tr>
<td>DB2 - Online Program Output</td>
<td>7-29</td>
</tr>
<tr>
<td>DB2 - Online Programs</td>
<td>7-31</td>
</tr>
<tr>
<td>Appendix C: Formal/DB2/ADR Entity Comparison</td>
<td>8-1</td>
</tr>
<tr>
<td>Abstract</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>1.0</td>
<td>1-4</td>
</tr>
<tr>
<td>1.1</td>
<td>1-6</td>
</tr>
<tr>
<td>2.0</td>
<td>2-3</td>
</tr>
<tr>
<td>2.1</td>
<td>2-17</td>
</tr>
<tr>
<td>2.2</td>
<td>2-18</td>
</tr>
<tr>
<td>2.3</td>
<td>2-19</td>
</tr>
<tr>
<td>2.4</td>
<td>2-20</td>
</tr>
<tr>
<td>2.5</td>
<td>2-21</td>
</tr>
<tr>
<td>3.0</td>
<td>3-3</td>
</tr>
<tr>
<td>3.1</td>
<td>3-5</td>
</tr>
<tr>
<td>3.2</td>
<td>3-7</td>
</tr>
<tr>
<td>3.3</td>
<td>3-9</td>
</tr>
<tr>
<td>3.4</td>
<td>3-10</td>
</tr>
<tr>
<td>3.5</td>
<td>3-11</td>
</tr>
<tr>
<td>3.6</td>
<td>3-12</td>
</tr>
<tr>
<td>3.7</td>
<td>3-14</td>
</tr>
<tr>
<td>3.8</td>
<td>3-16</td>
</tr>
<tr>
<td>3.9</td>
<td>3-17</td>
</tr>
<tr>
<td>3.10</td>
<td>3-18</td>
</tr>
<tr>
<td>3.11</td>
<td>3-19</td>
</tr>
</tbody>
</table>
Table of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.12 ADR Cardin Dataview</td>
<td>3-20</td>
</tr>
<tr>
<td>3.13 Side-by-side Comparison</td>
<td>3-22</td>
</tr>
<tr>
<td>3.14 SQL Phone View Definition</td>
<td>3-26</td>
</tr>
<tr>
<td>3.15 ADR Ideal Nested FOR Statements</td>
<td>3-32</td>
</tr>
<tr>
<td>3.16 Program Timings</td>
<td>3-35</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Data Base Management Systems (DBMS) are playing an increasingly important role in the development of computer systems. There are a great many commercial data base management systems available to chose from, many more in the last 10 years than ever before. Of the 3 major types of data bases (relational, hierarchical and network), this paper will concentrate on two commercially available relational data base management system products for mainframe computers, IBM's DB2 (Data Base 2) and ADR's Datacom/DB.

But first, a little information on relational database management. "Ever since the landmark work of E.F. Codd and C.J. Date in the early 1970s, the concept of relational database technology has engendered considerable excitement, as well as confusion, in the computer industry." [60]
The excitement revolves around the promise of relational technology: "the ability to efficiently access and manipulate data, regardless of the manner in which the data is physically stored." [60] No programmer navigation is required to move through a relational database.

"The manipulative part of the relational model consist of a set of operators known collectively as the relational algebra, together with a relational assignment operator which assigns the value of some arbitrary impression of the algebra to another relation. Each operator of the relational algebra takes either one or two relations as its operand(s) and produces a new relation as its result. Codd originally defined eight such operators, ... union, intersection, difference, Cartesian product, ... select, project, join and divide...". [18]

"According to C. J. Date, the term 'fully relational' refers to two principal components of the relational data base model: the relational data structure, and the relational algebra.... A data base system may be called fully relational if its supports: Relational data
bases (including the concepts of domain and key and the two integrity rules, data integrity and referential integrity); and a language that is at least as powerful as the relational algebra (and that would remain so, even if all facilities for loops and recursion were to be deleted)." [60]

The most fundamental property of a relational data base system is that data is presented to the user as tables — the mathematical name for a table with unique rows is a relation — and that the system provides suitable operators for the manipulation of these tables.

An example of a relational data structure is provided in Figure 1.0.
<table>
<thead>
<tr>
<th>LASTNAME</th>
<th>WORKDEPT</th>
<th>PHONENO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith</td>
<td>E11</td>
<td>2095</td>
</tr>
<tr>
<td>Spenser</td>
<td>E21</td>
<td>0972</td>
</tr>
<tr>
<td>Geyer</td>
<td>E01</td>
<td>6789</td>
</tr>
<tr>
<td>Perez</td>
<td>D21</td>
<td>9001</td>
</tr>
<tr>
<td>Haas</td>
<td>A00</td>
<td>3978</td>
</tr>
<tr>
<td>Johnson</td>
<td>D21</td>
<td>8953</td>
</tr>
<tr>
<td>Nicholls</td>
<td>C01</td>
<td>1793</td>
</tr>
<tr>
<td>Thompson</td>
<td>B01</td>
<td>3476</td>
</tr>
<tr>
<td>Lutz</td>
<td>D11</td>
<td>0672</td>
</tr>
<tr>
<td>Pulaski</td>
<td>D21</td>
<td>7831</td>
</tr>
<tr>
<td>Setright</td>
<td>E11</td>
<td>3332</td>
</tr>
<tr>
<td>Stern</td>
<td>D11</td>
<td>6423</td>
</tr>
</tbody>
</table>

Figure 1.0 Telephone Directory
Figure 1.0 shows only a single table. The table is a telephone directory containing name, work department and telephone number. There is one row for each employee. Each row has three values - one for each column in the table.

A relational data base is usually composed of many different tables, as shown in Figure 1.1 (the terms relation and table are synonymous). Thus, a relational data base appears as a collection of tables, each consisting of columns and rows. The rows in a table correspond to records in a file and the columns correspond to the fields within such records. "In order for the analogy with a file to be meaningful such a file should have records of one single type only, all occurrences having the same record structure and layout." [60]

The alternatives to presenting data as tables are to present data in the form of hierarchies (as DL/1) or in the form of networks. "A main difference between these alternatives is that in the case of network or hierarchical data structures (see Figure 1.1), the relationships between data are to some extent represented by the struc-
Hierarchical DBMS Structure

Network DBMS Structure

Relational DBMS Structure

<table>
<thead>
<tr>
<th>Course Relation</th>
<th>Course Number</th>
<th>Course Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>10003</td>
<td>History</td>
<td></td>
</tr>
<tr>
<td>10150</td>
<td>Spanish</td>
<td></td>
</tr>
<tr>
<td>10152</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>10288</td>
<td>Algebra</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student Relation</th>
<th>Student Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1016</td>
<td>Ruth Davis</td>
</tr>
<tr>
<td>1099</td>
<td>Sam Smith</td>
</tr>
<tr>
<td>1911</td>
<td>Jan Wilson</td>
</tr>
<tr>
<td>1902</td>
<td>Bill Jones</td>
</tr>
<tr>
<td>2008</td>
<td>Geri Thomas</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Courses-Taken Relation</th>
<th>Student Name</th>
<th>Course Name</th>
<th>Grade</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ruth Davis</td>
<td>Spanish</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Bill Jones</td>
<td>Algebra</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Bill Jones</td>
<td>English</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Bill Jones</td>
<td>Spanish</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Sam Smith</td>
<td>Algebra</td>
<td>B</td>
<td></td>
</tr>
<tr>
<td>Sam Smith</td>
<td>English</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Ceri Thomas</td>
<td>English</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Ceri Thomas</td>
<td>Algebra</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Jan Wilson</td>
<td>History</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Jan Wilson</td>
<td>Algebra</td>
<td>B</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1.1 Examples of the Three Models
ture itself - in a network through Owner-Member links, in a hierarchy through Parent-Child links. In relational data structures all relationships between tables are represented purely by the values in the tables.

"In a system logically modeled as a hierarchical or network structure, all potential access paths are explicitly defined by the data base administrator. Any application program that accessed the data base must work through a predefined and preauthorized access path. These paths are a major strength of the hierarchical and network data base." [60]

A relational data base management system is the software package that will provide facilities to access and update tables. As such it includes a language to define, access and update the databases, storage methods to maintain data on disk, utility functions, concurrency control facilities and various service functions.

The relational database is being held up by many as a panacea for the ills currently afflicting modern DP operations: the application backlog, the need to increase
increase development productivity, the complexity and changeability of business requirements and the demands of end users for access to their data.

Database entity terminology can vary widely from vendor to vendor. Appendix C shows the accepted formal relational term and counterparts for each used by the two DBMS vendors under study (ADR Datacom/DB and IBM DB2). For example, the formal term relation is referred to as a table by DB2 and as a file by ADR. [19]

Now, on to Chapter 2 and some detail about the two relational DBMS product families under study. Chapter 3 makes further comparisons through a miniature application system implemented in each environment. Chapter 4 concludes the paper with impressions of the two product families and future work which could be done to supplement the effort of this paper.
Chapter 2

Orientation to the DB2 and Datacom/DB product families

The purpose of this Chapter is to provide the reader with more detailed information about both of the relational DBMS product families under study - IBM's DB2 and ADR's Datacom/DB.

"In the late 1960s and early 1970s, Codd introduced the relational data model as an alternative way of structuring and managing data. Here, data is structured in two-dimensional tables and related by their value only, not by the logical structure of the schema. The term nonnavigational data structure was created because no programmer navigation is required to move through the data structure. In conjunction with the data structure, the relational model suggests data manipulation via a series of set-theoretic operators that help achieve significant economies in programming and end user access to data bases." [44]
IBM's DB2 is the product of the research into the relational model that began at IBM in 1969. DB2 (Data Base 2) is IBM's relational DBMS for large scale (MVS) mainframe data base systems. "Announced in June 1983, generally available in April 1985 after an intensive "field test" program, and recently enhanced (release 2, Feb. 1986) with subtle but important performance improvements. DB2 is a new product, very new. As of fourth quarter 1986 it had approximately 450 users." [16]

"ADR's Datacom/DB was first introduced in 1974. It had approximately 1200 customers as of fourth quarter 1986." [15]

Figure 2.0 provides a quick comparison of the two DBMS product families:
<table>
<thead>
<tr>
<th></th>
<th>DB2</th>
<th>Datacom/DB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Vendor</strong></td>
<td>IBM</td>
<td>ADR</td>
</tr>
<tr>
<td><strong>Structure</strong></td>
<td>Relational</td>
<td>Quasi-relational*</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td>IBM 370 family</td>
<td>IBM 370 family,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PC-AT, 3270 PC</td>
</tr>
<tr>
<td><strong>Operating System</strong></td>
<td>MVS, MVS/XA</td>
<td>DOS/VSE, VS1, VM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MVS, MVS/XA</td>
</tr>
<tr>
<td><strong>High-level non-procedural Langs.</strong></td>
<td>SQL</td>
<td>Ideal</td>
</tr>
<tr>
<td><strong>Min. Memory Req.</strong></td>
<td>2.5 - 5 MB</td>
<td>1 - 2.5 MB</td>
</tr>
<tr>
<td><strong>Data Dictionary</strong></td>
<td>IMS DB/DC</td>
<td>Data Dictionary</td>
</tr>
<tr>
<td><strong>Logical View</strong></td>
<td>Extensive</td>
<td>NO</td>
</tr>
<tr>
<td><strong>Data Independ.</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Query</strong></td>
<td>QMF, SQL, QBE</td>
<td>Data Query</td>
</tr>
<tr>
<td><strong>Appl. Pgmg Aid</strong></td>
<td>Cross System</td>
<td>Ideal</td>
</tr>
<tr>
<td><strong>Product</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Graphics support</strong></td>
<td>No</td>
<td>Data Query Graphics</td>
</tr>
<tr>
<td><strong>Record/file lock.</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Automatic recovery</strong></td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Authorization/security</strong></td>
<td>extensive</td>
<td>extensive</td>
</tr>
</tbody>
</table>

Figure 2.0 Quick Comparison of Features

2-3
<table>
<thead>
<tr>
<th>Networking Facility</th>
<th>SNA</th>
<th>SNA, DNET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transaction Proc.</td>
<td>CICS, TSO</td>
<td>CICS, Datacom-DC,</td>
</tr>
<tr>
<td>Facility</td>
<td></td>
<td>Roscoe</td>
</tr>
</tbody>
</table>

* Quasi-relational means that many relational algebra operations are provided, but not enough for the relational algebra to be complete.

**Figure 2.0** Quick Comparison of Features
Below is a brief description of several products from each DBMS family. The description will be organized in the following general categories:

ADR product descriptions:

ADR Data Base/Data Communications

1. Datacom/DB

ADR/Datacom/DB is a relational database management system designed for on-line and batch applications first installed in 1974. It services concurrent record-at-a-time and set-at-a-time data manipulation requests with comprehensive security and integrity protection, utilizing a hi-speed directory of active definitions, a unique, compound relational index that directs the location of data, and an intelligent access plan optimizer. It is available for both OS and DOS environments.

2. Data Designer

Data Designer is an interactive data base design tool that automates and simplifies data base design.
3. Datacom/DC

Datacom/DC is a reentrant, multi-tasking, multi-threaded TP monitor which controls programs and terminals for on-line data base applications.

4. D-Net and /D-DDB

D-Net and D-DDB support distributed database processing and permit data to be shared transparently across multiple computer systems in a network. D-DDB manages data updating across nodes, and supports replicated and partitioned databases. D-Net provides the connection and transportation services.

ADR Resource Control:

5. Librarian

Librarian is a source program management system for programming, systems, and operations personnel and managers.

6. Datadictionary

Datadictionary is the central control and resource manager for the ADR/Datacom product line. It is integrated and active, managing an organization's entire information resource, ensuring consistency of all definitions and re-
relationships.

ADR On-line Program Development

7. Roscoe

Roscoe is an on-line program development and maintenance system for OS environments.

8. Vollie

Vollie is an on-line program development and maintenance system for DOS/VS(E) environments.

ADR Application Development

9. Ideal

Ideal is a sophisticated, fourth-generation application development system suitable for a broad breadth of on-line and batch applications. It combines a very high-level, structured language with a screen painter, a report writer and change management, prototyping, and testing facilities into a workstation environment.

10. DE-II

An on-line data collection and validation facility with
generalized and user-defined formatted screens.

ADR Advanced Programs and Languages

11. MetaCOBOL

MetaCOBOL is a Cobol language pre-compiler that supports Cobol standards, structured programming, and higher-level language extensions for DB/DC applications.

12. ADR/DL

ADR/DL supports the development and enhancement of Cobol applications through interactive programming facilities, a high-level, structured language consistent with Ideal, and a batch Cobol generator. ADR/DL operates with ADR/ROSCOE and ADR/Vollie, with plans to support IBM's TSO/ISPF. ADR/DL is a high-level extension to the Cobol language designed especially for data base and data communications applications.

ADR Query and Reporting

13. Datareporter

Datareporter is an information retrieval and reporting system for generating management information reports from centralized files.
14. Dataquery

Dataquery supports ad hoc on-line and batch inquiry and reporting. Field value security and user-level processing limits allow a wide audience of users to safely and directly reference the same set of production data for decision making. Other features include updating, graphics, multilingual support, expert and novice modes, and an on-line tutorial. Dataquery is an English-language, relational query system for providing immediate access to information.

ADR Office Automation, Decision Support, Documentation

15. Empire

Empire is a decision support system with integrated color graphics and statistics form business analysis and planning.

16. ETC

ETC is an interactive word processing system for automated preparation, composition, and maintenance of textual material.
17. eMAIL

eMAIL is an interactive electronic mail system for sending, receiving, storing, and managing correspondence quickly and easily.

18. ASC

ASC is a system to automatically produce documentation for system and application-level information.

19. Autoflow II

Autoflow II is an advanced system development tool which provides automatic program documentation and analysis.

ADR Operational Performance

20. LOOK

LOOK is a real-time performance measurement system for improving performance and throughput. Look/Datacom permits the comprehensive performance management in database-oriented shops. In addition to dynamically analyzing activity levels, resource utilization, and response times of Datacom/DB, Look also determines the load and utilization of the entire processor, including CICS, batch, ROSCOE, the operating system, and the hardware.
ADR Transparency

21. ADR's migration software automates the process of porting data from older technologies (VSAM, DL/1, Total) to a relational environment, without the need to convert, recompile, or relink existing applications. Numerous options are available in the new relational environment to productively harness the information, manage the operation, and introduce change, particularly when the applications run as fast or faster than before.
DB2 product descriptions:

DB2 Data Base/Data Communications

1. Database 2 (DB2)

A large-system MVS relational DBMS, first released in 1983. It can be installed with the IMS/VS/DB hierarchical system or configured as a stand-alone DBMS. The DB2 system employes the SQL (Structured Query Language) as its host data base language, and is compatible, to some degrees, with the SQL/DS relational system designed for use the DOS/VS environment.

DB2 Query and reporting

2. Query Management Facility (QMF)

QMF provides interactive data base facilities to users with little of no technical background. It operates against SQL/DS data under VM as well as against DB2 under MVS. Data definition functions are provided through SQL. Features include ad hoc query in SQL or QBE (query by example) languages. Report preparation consists of relation of data for graphic presentation, defining and executing a procedure consisting of a series of query/report functions.

2-12
3. Cross System Product/Application Development (CSP/AD) and Cross System Product/Application Execution (CSP/AE)

With CSP/AD and CSP/AE, programming staff can completely define, test, generate, and execute application programs. These productivity tools support multiple development and production environments.

4. Data Base Edit Facility (DBEDIT)

DBEDIT helps you perform data base operations without requiring you to know SQL or data base structure. You can insert, delete, update, or select information from a data base. You can, in short, perform all the data manipulation tasks for which you would otherwise need to know SQL.

DB2 Data Migration, Product Interfacing

5. CSP-DB2 Interface

DB2 offers the powerful data manipulation capabilities of SQL. The Cross System Product offers the ease-of-use features of high-level languages. The CSP-DB2 Interface gives you both. The CSP-DB2 Interface is a PL/1 program
that allows Cross System Product applications to issue SQL statements against DB2 tables.

6. Data Base Migration Aid Utility
Assists the DB2 system of DBA in migrating data and data descriptions between DB2 systems. Data descriptors can also be redefined on an existing DB2 subsystem. The program is a set of nine functions and requires authorization to select data from the DB2 system catalog tables. The Utility is menu driven.

7. Data Extract (DXT)
Extracts data on a periodic or one time basis. The operational data can be in a DL/1 hierarchical database, a VSAM file, a physical sequential file, or a DB2 or SQL/DS-VM relational database. This extracted data can be put into a relational database for easy access and reporting by such programs as Query Management Facility (QMF) and can be moved from different subsystems on the same processor, moved between subsystems on different processors, or stored elsewhere, as defined by the installation's needs. End user dialogs are similar to
those of QMF, allowing users who have no data processing experience to create extract requests.

8. Data Dictionary DB2 Interface

Expands the Dictionary's (DB/DC) base of general functions to support DB2. It is designed to accelerate DB/DC system and application development by supporting these activities by using Dictionary information as the source for creating, modifying, and/or deleting DB2 objects and by using the DB2 catalog as the source for creating and/or modifying Dictionary subjects. A front end dialog under ISPF is optional.

DB2 Resource Control

9. DB/DC Dictionary

The DB/DC dictionary is an IMS productivity aid that consists of five DL/1 physical and five DL/1 logical databases containing information about the structure of data in five distinct levels. The system performs the following functions: translates dictionary user-input language requests; executes update commands to modify, delete, and add information to the dictionary databases; prepares output reports in either batch or on-line environment;
provides data definitions for Assembler, Cobol, or PL/1 COPY or INCLUDE libraries.

DB2 Operational Performance

10. DB2 Performance Monitor (DB2PM)

Designed to provide DP managers, system administrators, database administrators, and system programmers with essential information addressing the performance of DB2 database systems. With DB2PM you can obtain reports providing both system-wide and application-related information showing in varying levels of detail, DB2 performance characteristics during a given interval.
Here is a brief list of products in each DBMS family and prices as of July, 1986. Figure 2.1 shows pricing information for a typical ADR package with the following configuration:

Datacom/DB Kernel
Datadictionary
Datareporter
DataQuery
Ideal

Package price: Initial charge: $288,300
               Annual charge: $42,546

Figure 2.1 ADR Package Price, Typical Configuration
Figure 2.2 shows prices for selected ADR products.

<table>
<thead>
<tr>
<th>Product</th>
<th>License Purchase ($)</th>
<th>Annual Maint. ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datacom/DB Kernel</td>
<td>116,000</td>
<td>16,240</td>
</tr>
<tr>
<td>Datadictionary</td>
<td>39,600</td>
<td>5,545</td>
</tr>
<tr>
<td>Datareporter</td>
<td>5,545</td>
<td>2,184</td>
</tr>
<tr>
<td>Options:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dataquery</td>
<td>34,800</td>
<td>4,872</td>
</tr>
<tr>
<td>Ideal (4GL)</td>
<td>97,900</td>
<td>13,705</td>
</tr>
<tr>
<td>VSAM Transparency</td>
<td>25,300</td>
<td>3,542</td>
</tr>
<tr>
<td>DL/1 Transparency</td>
<td>40,000</td>
<td>5,600</td>
</tr>
<tr>
<td>Datasecure</td>
<td>14,100</td>
<td>1,974</td>
</tr>
<tr>
<td>Datadesigner</td>
<td>50,600</td>
<td>7,084</td>
</tr>
<tr>
<td>D-Net</td>
<td>41,300</td>
<td>5,780</td>
</tr>
<tr>
<td>DE-II (data entry)</td>
<td>24,500</td>
<td>3,430</td>
</tr>
</tbody>
</table>

Figure 2.2  ADR Price List, Selected Products
Figure 2.3 shows prices for a typical DB2 package with the following configuration:

- SQL (Data Manipulation/definition lang.)
- Query Management Facility (QMF)
- DB2 Performance Reporting Tool
- Data Base Edit
- Data Dictionary DB2 Interface

Package price: Initial charge: $16,050
Monthly charge: $2,675
(Annual charge: $32,100)

Figure 2.3 DB2 Package Price, Typical Configuration
Figure 2.4 shows prices for selected DB2 products.

<table>
<thead>
<tr>
<th>Product</th>
<th>1 Time Charge ($)</th>
<th>Initial Charge ($)</th>
<th>Monthly Charge ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. DB2</td>
<td>16,050.00</td>
<td>975.00</td>
<td></td>
</tr>
<tr>
<td>2. QMF</td>
<td>1500.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. DXT</td>
<td>300.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. CSP</td>
<td>1550.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5. CSP-DB2 Intf</td>
<td>4000.00</td>
<td>1150.00</td>
<td></td>
</tr>
<tr>
<td>6. Edit Facility</td>
<td></td>
<td></td>
<td>350.00</td>
</tr>
<tr>
<td>7. Mig Aid Util</td>
<td>4000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8. DD DB2 Intf</td>
<td>4000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Perf Monitor</td>
<td></td>
<td></td>
<td>975.00</td>
</tr>
</tbody>
</table>

Figure 2.4 DB2 Price List, Selected Products
Figure 2.5 shows abbreviations for DB2 products which were used in Figure 2.4.

DB2 - Database 2
QMF - Query Management
DXT - Data Extract
CSP - Cross System Product
CSP-DB2 Intf - Cross System Product DB2 Interface
Edit Facility - DB2 Edit Facility
Mig Aid Util - Migration Aid Utility
DD DB2 Intf - Data Dictionary DB2 Interface
Perf Monitor - Performance Monitor

Figure 2.5 DB2 Product Abbreviations
There are many features about the databases and other related products which are different. Below is a more detailed discussion about each DBMS and selected comparisons. But first, a brief overview of each DBMS system.

DB2 product system overview:

DB2 is a large-system MVS relational DBMS. It was first introduced in 1983 as part of IBM's dual database strategy. DB2 was to be used as an ad hoc query and data analysis tool, while IMS was to serve as the production DBMS. This plan has been changing as DB2's performance has continually improved.

DB2 can be installed with the IMS/VS/DB hierarchical system or configured as a stand-alone DBMS. The DB2 system employs SQL (Structured Query Language) as its host database language, and is compatible, to some degrees, with the SQL/DS relational system designed for use with the DOS/VS environment. DB2 provides the following integral database facilities: relational file
structure, views, table space, SQL, data space management, user interface, monitoring and accounting, security and authorization, and data set protection. All data in a DB2 data base is stored in VSAM entry sequenced data sets (EDSD), which can be defined and maintained by the user or automatically maintained by DB2. DB2 supports a relational data model. DB2's data base can be described as a collection of tables. Data is defined in terms of tables and accessed through operations on tables. Data definition, retrieval, manipulation, and control operations are supported by SQL. SQL is a high-level data language available to users through an interactive terminal and through applications written in Cobol, Fortran, PL/1, Basic, or Assembler language. DB2 can be accessed concurrently by the IMS/V5 Data Communication Feature, by CICS/OS/VS, by TSO users, and by batch jobs. It operates as an MVS subsystem and is designed to utilize the System/370 Extended Architecture (XA), including 31-bit virtual addressing and large real storage. DB2's architecture provides for very large data bases (up to 64 billion bytes per table). It is supported by a comprehensive set of data base utilities that operate online, including DXT and DB2 Performance Monitor. Its security
and authorization mechanism offers field content security and allows various levels of authority to be delegated to users as appropriate.
ADR Datacom/DB product system overview:

"ADR/Datacom/DB was originally marketed by a small firm that did not have the marketing resources of a major system software house. ADR acquired the company and the DBMS in late 1978. Soon thereafter, ADR spent considerable effort in redesigning and restructuring the architecture of the system, and the growth in the user base since then has been dramatic. In 1985 alone, the product doubled its user base, and should continue to make significant inroads in the market with its flexibility, ease of use, integration with other popular ADR products, and its recently introduced DL/1 Transparency option." [15]

Datacom/DB is a relational database management system designed for on-line and batch applications. Data is stored in tables (also known as logical relations) and accessed as rows and columns. Each database has a single, combined index providing logical relational operations such as select, project, and join. The system provides an interactive end-user language to support direct user access, updating, analysis and reporting.
Datacom/DB programs use dataviews or logical views of data when accessing information. It provides program development, maintenance and information center support for existing and future applications through ADR/Ideal, a fourth generation application development system for new applications and ADR/DL, a comprehensive Cobol application generator for database programs and a complete data manipulation language. Data access is supported by random, sequential or index-only processing. Data integrity is maintained through such features as concurrent update protection, exclusive control of data, automatic transaction backout, complete automatic restart/recovery, program data access security and encryption facilities. Datacom/DB supports a multi-user facility that provides the user with extended ability to access databases concurrently from multiple partitions. Accounting facilities for management control are definable by each installation and its preferred detail level of summarization point. The information is stored in table form, in a system controlled Datacom/DB database. Statistics can be accessed on-line or in batch with the ADR tool set. The Datacom/DB databases are defined through Datadictionary, a central resource manager for the infor-
The dictionary provides facilities for definition, design, reporting, auditing and control. ADR's migration software tools, the VSAM or DL/1 Transparencies, migrate current data written for VSAM or DL/1 to Datacom/DB with no modifications to existing application programs.

Below is more detailed information about selected areas for each DBMS:

1. Operating environment

DB2: Minimum memory 2.5 - 5 MB
ADR: Minimum memory 1 - 2.5 MB

Computer/Operating Systems supported:
ADR: IBM system/370, 3000, 4000 and compatible computers. SSX/VSE, DOS/VS, DOS/VSE, OS/VS1, OS/VS2(SVS), OS/VS2(MVS), MVS/XA, AND VM/CMS.

DB2: Any IBM or compatible processor supported by MVS/SP, MVS/XA.
ADR: can be accessed by Datacom/DC (ADR's reentrant, multi-tasking, multi-threaded TP monitor, by CICS and by other telecommunications monitors.

DB2: can be accessed concurrently by IMS/VS Data Communication Feature, by CICS/OS/VS, by TSO users, and by batch jobs.

DB2 was designed to exploit the 31-bit addressing architecture of MVS/XA, but it can also run on MVS/370, which uses 24-bit addressing.

2. File structure

All data in a DB2 data base is stored in VSAM entry sequenced data sets (EDSD), which can be defined and maintained by the user or automatically maintained by DB2.

3. Views of data

The general concept of a DB2 and ADR view of data are the same, a view allows you to present a smaller, simpler
version of a table upon which it is based. It permits you to authorize the use of a view to only specified users, and not every field in the entire table. A view only needs to contain fields to meet the needs of a user/program and not contain any unnecessary fields to complicate or confuse things.

However, a DB2 view can be created dynamically (does not need to previously exist) by a program if the program is authorized to access all fields making up the new view. An ADR dataview cannot be dynamically produced. A DB2 view also very importantly can contain fields from more than one table (i.e. join 2 or more tables), from other views or from a combination of views and tables. Views are used just like tables in SQL data manipulation statements. An ADR dataview currently can only contain fields from one table, not fields from multiple tables. A dataview processor has been promised for a future release which will remedy this situation.

4. Table space/Data space management

DB2 Table Space is user-defined to hold the data base
Tables. Each table space is divided into equal-sized units, called pages, which contain table data. A table space consists of one to 64 VSAM entry sequenced data sets (ESDS) and contain up to 64G bytes of information.

Table space is of two types, simple and partitioned. A partitioned table space holds exactly one table. The table space is divided into partitions, and each partition is stored in one VSAM ESDS. The partitions are defined as ranges of an index based on one or multiple columns. Through partitioned table spaces very large tables can be split into manageable units; partitions are independent of each other and can be reorganized and recovered individually.

In addition, partitioned table spaces can be assigned to different storage groups, each of which may be assigned to a different device type. This assignment scheme allows installations to store active data on faster devices than those on which history data is stored. Table spaces which are not partitioned are referred to as "simple" table spaces. A simple table space can hold one or more tables.
Table spaces are physically divided into storage units called pages. Each page holds one or more rows of a table (or multiple tables for simple table spaces). Two page sizes are available, 4K and 32K bytes. The larger page size must be chosen if rows with a length of more than 4K bytes exist in a table.

DB2 uses VSAM for DASD space management and data set cataloging. However, after the data sets are created, they are formatted and used by DB2 and cannot be processed by VSAM services.

At the top of ADR entity model are two related entities, Database and Area. These entities describe the database and area into which files (tables) are organized. The Area, hierarchically, is the ADR counterpart to the DB2 Table Space. An Area can have one or more Datacom/DB files defined to it. Typically, only one file is related to one area. If more than one file was related to an Area, the data would be cc-located, or co-mingled. If two files are nearly always accessed together, the number of I/Os could be cut down by physically locating the data together. These two files should probably be physically
together in the same Area. The records in the Area would resemble:

Record A/Record B/Record A/Record B/Record A/...

ADR Space management option information for an Area's files is also defined at the Area level within Datadictionary when the Area is defined. Options 0-3 can be selected to aid in the management of space within the files. Option selection controls things such as the reclamation of space after a record has been marked for deletion, whether it is important to maintain the native sequence in which records were added to the file or not, whether it is permissible or not to wrap-around to the beginning of the file when an end of file is detected.

ADR's CXX (control) File contains data base control information. It consists of the base segment describing data base characteristics, followed by area and file segments that describe the characteristics of each file, including keys and data elements. The CXX is constructed automatically from definitions in Datadictionary.
5. Data manipulation language

Users of both DB2 and ADR products do not have to navigate their way to data. When the requested data is located, DB2 and ADR returns, updates or deletes the entire collection (termed set-at-a-time processing) of data that met the conditions specified by the user.

SQL, Structured Query Language is used in DB2 for data manipulation. The language includes statements for retrieval, replacement, insertion, and deletion of data.

SQL statements may be issued interactively from a terminal and the results may be browsed. The interactive SQL facility is available to authorized TSO SPF/ISPF users.

SQL statements may be embedded in application programs written in Cobol, PL/1, Fortran, Basic, or Assembler.

SQL also provides Data Definition facilities for creating, changing, and deleting all DB2 objects. Objects are storage groups, table spaces, tables,
indexes, and views. A unique advantage of SQL is that these Data Definition statements may be used in a normal user session and mixed together with other types of statements, such as data manipulation statements. It is not necessary to stop DB2 or to invoke special utility programs to create a table for storing and manipulating some temporary result and drop the table when it is no longer needed.

ADR does not have a direct counterpart to DB2's SQL. ADR does not have one set method of manipulating data, but several products which provide for manipulation of data. First available was data base access by writing a traditional third generation language and embedding it with a low-level CALL interface (ie. CALL DBENTRY parameters). This is similar to DB2's SQL-embedded third language support. ADR's Ideal (standing for Interactive Development Environment for an Application's Life-cycle) is an alternative data manipulation language (4GL) plus an interactive tool for the design, development, and execution of applications in an online environment. Ideal lets the user access and update the data base with
an easy-to-use and very high-level set of statements.

Another ADR data manipulation product is ADR/DL. DL is an interactive, high-level application development and maintenance Cobol workstation. With DL, programmers can access and manipulate Datacom/DB data directly from Cobol applications without coding physical interfaces to the data base. DL has both Cobol-like commands and 4GL-like functions.

ADR announced in the first quarter of 1987 that it will support a subset of SQL data manipulation commands. No further information is available from ADR at this time on SQL support. ADR has recognized SQL as the recognized industry standard data manipulation language and the value of supporting that standard. ADR will introduce SQL support with a selected subset and is expected to expand over time the set of SQL commands it will support.

6. Indexes

DB2 and ADR indexes are very much alike. Both have a B-tree index structure. Each level of the index contains
a pointer to a block of pointers at the next lower level, except the last level, which contains pointers to actual data records. Programs or users accessing data never explicitly specify keys or indexes, indexes are only used by the DBMS. Programs specify field names. The DBMS maps the field or fields back to keys.

DB2 has one index for each key. ADR has 1 index for all keys of each database. Both have cluster keys or indexes. A cluster index or key determines the physical order in which rows are stored. A Cluster key or index is used to store data the way it would be most frequently accessed, for example, employee number, account number, or alphabetically by name. Both ADR and DB2 provide for unique and nonunique indexes/keys.

7. Program preparation

A DB2 application program issuing SQL calls must be precompiled using the DB2 precompiler to create a Data Base Request Module (DBRM). The DBRM is the input to the DB2 BIND process which produces a DB2 application plan that contains an optimized access path for each SQL statement. Besides the access path, the plan also
contains the tables to be accessed and the appropriate locking information. The application program must be compiled and then link-edited.

After programs are written, four steps must be performed before they can be run.

Precompilation: to check SQL syntax, produce a modified source program, and produce a data base request module (DBRM), an intermediate form of an SQL statement.

Compilation: to translate the modified source program using either a Cobol, PL/1, or Fortran compiler or an assembler.

Bind: to process the DBRM to produce an application plan, the control structure representing one or more SQL statements.

Link-edit: to produce the final object module.

The compiling and link-editing steps are the same in DB2 as they are in any other programming process. The
precompile and bind, however, are unique to DB2.

ADR programs can be written in several different languages, ADR/DL (an extension to Cobol), Ideal (ADR's 4GL) or Cobol, PL/1 or Assembler with embedded DBENTRY calls. When the program is successfully compiled, the program preparation process is finished. Access paths to data are determined dynamically by CBS (Compound Boolean Selection), which is discussed below.

8. Path Selection to data.

When all SQL statements are found to be correct in a program, the binder is authorized to access the data, DB2 builds and stores an application plan that contains information about both the program and the data the program uses.

The major advantage of the precompilation and bind process is that it removes operations that can be done once from the program's normal processing. When the program runs, SQL statements are not translated or semantically checked. Access paths are not selected. All that processing happens only once, before the program
If the data definitions or indexes that an application plan uses should change, rebinding occurs automatically. For example, suppose an index on a table used in an application plan were dropped. If the program using the plan were run, DB2 would automatically invoke the bind process to rebuild an application plan that did not use the index that had been dropped.

ADR's CBS (Compound Boolean Selection) facility provides full relational selection and ordering capabilities. CBS is totally key insensitive from the programmer viewpoint; selection and ordering can be specified for any fields in the for any fields in the table with full data typing (including floating point). CBS dynamically chooses the most efficient index path to satisfy the access request. If a key is modified, typically no program changes need to be made, CBS will continue to dynamically select the optimal path to data.
Chapter 3

Mini-application system - DB2 vs Datacom/DB

3.0 Introduction

In order to further study how DB2 and ADR products compare, a miniature application system was constructed in both environments. This section details the work involved in taking an existing small application system running in a DB2 environment and duplicating it as closely as possible in an ADR environment.

The mini-system consisted of programs at both extremes: an interactive on-line program and a batch report program and the necessary database, files, data views and other needed entities. This mini-system is part of a sample application system provided by IBM as part of the DB2 software (see Appendix B). It was decided to use this system provided by IBM instead of a system written in-house at the DB2 site because the DB2 site's personnel's' programming skills would not be a fac
tor and more importantly, the sensitivity of any of the site's data would be compromised in any way.

The purpose of the mini-application system was to aid the author in demonstrating future comparisons between DB2 and ADR DBMS software.
The batch program produces a report which is several pages of telephone directory listings. Each telephone list is based on card image input (see Figure 3.0).

A line number bar has been provided above the input for the reader's convenience.

```
0 1 2 3 4 5
1234567890123456789012345678901234567890123456789012345678901234567
L* LJO% L%SON LSMITH
LBROWN   ALAN
LBROWN   DAVID
U 0002304265
```

Figure 3.0 Batch Program Card Image Input
'LBROWN ALAN', is the fifth card image in Figure 3.0. The 'L' in the first byte indicates the action - 'L' = list, a 'U' in the last card image means update. The above example input indicates for the program to produce a telephone directory listing of all records (using the EMPLOYEE and DEPARTMENT Tables) with a last name of BROWN and first name of ALAN. If no match was found, an appropriate message was produced on the report.

The mini-system's on-line program is used to maintain the database (ie. add, change, delete, and display an index). Figure 3.1 shows the Employee Index screen.
### SELECTING AN EMPLOYEE TO DISPLAY

<table>
<thead>
<tr>
<th>NO</th>
<th>D/ID</th>
<th>DEPARTMENT NAME</th>
<th>E/ID</th>
<th>EMPLOYEE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>0000010</td>
<td>CI HASS</td>
</tr>
<tr>
<td>02</td>
<td>B01</td>
<td>PLANNING</td>
<td>0000020</td>
<td>ML THOMPSON</td>
</tr>
<tr>
<td>03</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>0000030</td>
<td>SA KWAN</td>
</tr>
<tr>
<td>04</td>
<td>E01</td>
<td>SUPPORT SERVICES</td>
<td>0000040</td>
<td>JB GEYER</td>
</tr>
<tr>
<td>05</td>
<td>D11</td>
<td>MANUFACTURING</td>
<td>0000050</td>
<td>IF STERN</td>
</tr>
<tr>
<td>06</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
<td>0000060</td>
<td>ED PULASKI</td>
</tr>
<tr>
<td>07</td>
<td>E11</td>
<td>OPERATIONS</td>
<td>0000070</td>
<td>EW HENDERSON</td>
</tr>
<tr>
<td>08</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td>0000080</td>
<td>TG SPENSER</td>
</tr>
<tr>
<td>09</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>0000090</td>
<td>VB LUCCHESI</td>
</tr>
<tr>
<td>10</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>0000100</td>
<td>E O'CONNELL</td>
</tr>
<tr>
<td>11</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>0000110</td>
<td>DM QUINTANA</td>
</tr>
</tbody>
</table>

PFK1 02=RESEND  03=END  08=NEXT

---

**Figure 3.1 On-line Program Employee Index Screen**
The best way to get a comparison of DB2 and ADR DBMS products would have been to have both DB2 and Datacom/DB running at the same site, physically on the same equipment. Since it was not possible to find one data processing shop with both ADR's Datacom/DB and IBM's DB2, the product comparison was conducted at the two sites. Care was given to ensure that the hardware and operating systems at the two sites were as similar as possible. Figure 3.2 compares the pertinent hardware and software at the two sites. Note that both sites have the same CPU and operating system.

The ADR Datacom/DB site selected was Johnson County, Kansas in Olathe, Kansas. The IBM DB2 site selected was Forrest T. Jones, a Kansas City, Missouri insurance company.
Johnson County (ADR) Forrest T. Jones (DB2)

Hardware:

CPU: IBM 4381 IBM 4381
Memory size: 16 MEG 16 MEG

Software:

Operating system: OS MVS/XA OS MVS/XA
TP Monitor: VTAM SNA/VTAM
Pertinent IBM CICS CICS
Products:

ADR Products: Datacom/DB
Data Dictionary
Ideal
Data Query
Data Reporter
VSAM Transparency
Inter Products Components

File storage DASD: IBM 3380's

Figure 3.2 Profiles of Test Sites

3-7
Now, more about the mini-system. First, the database configuration documentation supplied by the DB2 site is given in Figure 3.3, which shows the relationship between storage group, database, table space, and table. Figure 3.4 shows the Department and Employee table layouts (each is enclosed in a box). Characteristics of fields in each table are shown. Figure 3.5 shows Indexes (keys) for the Department and Employee tables. Fields making up each key are shown. Figure 3.6 shows the Phone View (enclosed in a box), which was used in the DB2 programs. Later in this chapter, entities shown in these figures are discussed in detail.
Storage group: DSN8G2DD

Data bases:
- DSN8D2AP application data
- DSN8D2PG programming tables

Table Spaces:
- DSN8S2DP Department Table
- DSN8S2EM Employee Table
- Separated spaces for other application tables
- DSN8S2CM Common for Programming Tables

Figure 3.3 DB2 Test Data Base Configuration
<table>
<thead>
<tr>
<th>Creator</th>
<th>TNAM1E</th>
<th>CNAME</th>
<th>COLNO</th>
<th>COLTYPE</th>
<th>LENGTH</th>
<th>SCALE</th>
<th>NULLS</th>
</tr>
</thead>
<tbody>
<tr>
<td>05N82</td>
<td>TAC11PE</td>
<td>ACTDESC</td>
<td>3</td>
<td>VARCHAR</td>
<td>20</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TAC11PE</td>
<td>AC11NO</td>
<td>2</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TAC11PE</td>
<td>AC11NO</td>
<td>1</td>
<td>SMALL1NT</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>ICONA</td>
<td>COMVIO</td>
<td>1</td>
<td>CHAR</td>
<td>16</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>ICONA</td>
<td>LASTMIC</td>
<td>5</td>
<td>LONGVAR</td>
<td>1574</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>ICONA</td>
<td>LASTPOS</td>
<td>3</td>
<td>CHAR</td>
<td>254</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>ICONA</td>
<td>LASTPOS</td>
<td>4</td>
<td>CHAR</td>
<td>254</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOLPT</td>
<td>ADMDEPT</td>
<td>4</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOLPT</td>
<td>DEPTNAM</td>
<td>2</td>
<td>VARCHAR</td>
<td>16</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOLPT</td>
<td>DEPTNO</td>
<td>1</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOLPT</td>
<td>DEPTNO</td>
<td>3</td>
<td>CHAR</td>
<td>8</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOSPTX1</td>
<td>OSP11N0</td>
<td>1</td>
<td>CHAR</td>
<td>79</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOSPTX1</td>
<td>SP11N0</td>
<td>2</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>BIRTHDATE</td>
<td>11</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>DECVL</td>
<td>9</td>
<td>SMALL1NT</td>
<td>2</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>EMPNO</td>
<td>1</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>FIRSTNAME</td>
<td>2</td>
<td>VARCHAR</td>
<td>12</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>HIREDATE</td>
<td>7</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>JOCODEC</td>
<td>8</td>
<td>DECIMAL</td>
<td>3</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>LASTNAME</td>
<td>4</td>
<td>VARCHAR</td>
<td>15</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>MIDNIT</td>
<td>3</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>PHONENO</td>
<td>6</td>
<td>CHAR</td>
<td>4</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>SALARY</td>
<td>12</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>SEX</td>
<td>16</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPL</td>
<td>WORKR11PT</td>
<td>3</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPRAC</td>
<td>AC11NO</td>
<td>3</td>
<td>SMALL1NT</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPRAC</td>
<td>EMPNO</td>
<td>1</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPRAC</td>
<td>EMPTIME</td>
<td>4</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPRAC</td>
<td>END111ATE</td>
<td>5</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TEMPRAC</td>
<td>PROJNO</td>
<td>6</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>ACTION</td>
<td>2</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>OSP11N0</td>
<td>11</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>HEADTXT</td>
<td>6</td>
<td>CHAR</td>
<td>50</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>HELP TXT</td>
<td>9</td>
<td>CHAR</td>
<td>79</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>INF11OTX</td>
<td>8</td>
<td>CHAR</td>
<td>79</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>MAJSYS</td>
<td>1</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>OBJECT</td>
<td>3</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>PRJTXT</td>
<td>10</td>
<td>CHAR</td>
<td>79</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>SCR111YPE</td>
<td>5</td>
<td>CHAR</td>
<td>1</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>SELTXT</td>
<td>7</td>
<td>CHAR</td>
<td>50</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TOPTVAL</td>
<td>SRCH1111RTT</td>
<td>4</td>
<td>CHAR</td>
<td>2</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>DEPTNO</td>
<td>1</td>
<td>CHAR</td>
<td>3</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>MAJPRJ</td>
<td>6</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>PROJNAME</td>
<td>2</td>
<td>VARCHAR</td>
<td>24</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>PROJNO</td>
<td>1</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>PRSTAFF</td>
<td>3</td>
<td>DECIMAL</td>
<td>5</td>
<td>2</td>
<td>Y</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>PRSTDATE</td>
<td>6</td>
<td>DECIMAL</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
<tr>
<td>05N82</td>
<td>TPROJ</td>
<td>RES111EMP</td>
<td>4</td>
<td>CHAR</td>
<td>6</td>
<td>0</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 3.4 DB2 Department and Employee Table Layouts.
<table>
<thead>
<tr>
<th>IXNAME</th>
<th>IXCREATOR</th>
<th>COLNAME</th>
<th>COLNO</th>
<th>CDLSEQ</th>
<th>ORDERING</th>
<th>IBMREQD</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDEPT2</td>
<td>OSN2</td>
<td>MGRNO</td>
<td>3</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XDEPT1</td>
<td>OSN2</td>
<td>ADMDEPT</td>
<td>4</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XDEPT1</td>
<td>OSN2</td>
<td>DEPTNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPL1</td>
<td>OSN2</td>
<td>EMPNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPL2</td>
<td>OSN2</td>
<td>WORKDEPT</td>
<td>5</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XONAT1</td>
<td>OSN2</td>
<td>CONT</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOPTVAL1</td>
<td>OSN2</td>
<td>MAJSYS</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOPTVAL1</td>
<td>OSN2</td>
<td>SGTYP</td>
<td>5</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOPTVAL1</td>
<td>OSN2</td>
<td>SGTYP</td>
<td>4</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOPTVAL1</td>
<td>OSN2</td>
<td>OBJECT</td>
<td>3</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOPTVAL1</td>
<td>OSN2</td>
<td>ACTION</td>
<td>2</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XPROJ2</td>
<td>OSN2</td>
<td>RESPMAP</td>
<td>4</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XPROJ1</td>
<td>OSN2</td>
<td>PROJNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XACTYPE1</td>
<td>OSN2</td>
<td>ACTNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XACTYPE2</td>
<td>OSN2</td>
<td>ACTNO</td>
<td>2</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XPROJAC1</td>
<td>OSN2</td>
<td>PROJNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XPROJAC1</td>
<td>OSN2</td>
<td>PROJNO</td>
<td>2</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPLAC1</td>
<td>OSN2</td>
<td>EMPNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPLAC1</td>
<td>OSN2</td>
<td>EMSTDATE</td>
<td>5</td>
<td>3</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPLAC2</td>
<td>OSN2</td>
<td>ACTNO</td>
<td>3</td>
<td>2</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XEMPLAC2</td>
<td>OSN2</td>
<td>EMPNO</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOSPTXT1</td>
<td>OSN2</td>
<td>GSPINDEX</td>
<td>1</td>
<td>1</td>
<td>A</td>
<td>N</td>
</tr>
<tr>
<td>XOSPTXT1</td>
<td>OSN2</td>
<td>LINO</td>
<td>2</td>
<td>2</td>
<td>A</td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 3.5 DB2 Department and Employee Indexes (Keys)

3-11
<table>
<thead>
<tr>
<th>Creator</th>
<th>Name</th>
<th>Name</th>
<th>Colno</th>
<th>Col Type</th>
<th>Length</th>
<th>Scale</th>
<th>Nulls</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMPLOYEENUMBER</td>
<td>1</td>
<td>CHAR</td>
<td>6</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>PHONENUMBER</td>
<td>2</td>
<td>CHAR</td>
<td>20</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMENDATE</td>
<td>3</td>
<td>SMALLINT</td>
<td>2</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMPNO</td>
<td>4</td>
<td>DECIMAL</td>
<td>6</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>ENFMNO</td>
<td>5</td>
<td>DECIMAL</td>
<td>6</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>ENFNAME</td>
<td>6</td>
<td>DECIMAL</td>
<td>2</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMENDATE</td>
<td>7</td>
<td>DECIMAL</td>
<td>6</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMPNO</td>
<td>8</td>
<td>DECIMAL</td>
<td>6</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VEMPLP</td>
<td>EMPDATE</td>
<td>9</td>
<td>DECIMAL</td>
<td>2</td>
<td></td>
<td>Y</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>ACTION</td>
<td>10</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>SPINDEX</td>
<td>11</td>
<td>CHAR</td>
<td>2</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>HELPTEXT</td>
<td>12</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>INFOX</td>
<td>13</td>
<td>CHAR</td>
<td>79</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>OBJECT</td>
<td>14</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>PERMIT</td>
<td>15</td>
<td>CHAR</td>
<td>2</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>SCRITUDE</td>
<td>16</td>
<td>CHAR</td>
<td>79</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>SETX</td>
<td>17</td>
<td>CHAR</td>
<td>79</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VTPTVAL</td>
<td>SHIELDCRT</td>
<td>18</td>
<td>CHAR</td>
<td>50</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>DEFPNAME</td>
<td>19</td>
<td>VARCHAR</td>
<td>20</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>OEPNUMBER</td>
<td>20</td>
<td>CHAR</td>
<td>3</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>EMPLOYEENUMBER</td>
<td>21</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>FIRSTNAME</td>
<td>22</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>MIJNAME</td>
<td>23</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
<tr>
<td>DSN82</td>
<td>VPHONE</td>
<td>MMEGINIT</td>
<td>24</td>
<td>CHAR</td>
<td>1</td>
<td></td>
<td>N</td>
</tr>
</tbody>
</table>

Figure 3.6 DB2 Phone View
The database configuration from the DB2 site has been presented. The database, tables, indexes, views of data, etc. at the DB2 site were duplicated as closely as possible at the ADR site. Below is database configuration documentation from the ADR site. Figure 3.7 shows, in an indented format, the relationship between database, area, file, record, key, element, and field. Figure 3.8 shows the Employee table layout and keys. Fields making up each table and key are shown. Figure 3.9 shows the Department table layout and keys. Fields making up each table and key are shown. Figures 3.10, 3.11, and 3.12 show the Employee, Department, and Cardin dataviews used in the ADR programs. Entities shown in these figures are discussed in detail later in this chapter.
```
| DATABASE | MS-CARDIN-OI  | 013 | P MINI-SYSTEM DATABASE  | 010 |
| AREA | MS-CARDIN | 001 | P MINI-SYSTEM ACTION AREA  | CARO O10 |
| FILE | MS-CARDIN | 001 | P MINI-SYSTEM MAJOR SYST TABLE  | CARO O10 |
| RECORD | MS-CARDIN | 001 | P CARDIN RECORD  | 001 |
| KEY | MS-CARDIN,MS-CARDIN-KEY-1 | 001 | P CARDIN KEY 1  | 001 |
| FIELD | MS-CARDIN,MS-CARDIN-KEY | 001 | P  | 001 |
| FIELD | MS-CARDIN-IN-RECO  | 001 | P  | 001 |
| FIELD | MS-CARDIN-IN-FRMNO | 001 | P  | 001 |
| FIELD | MS-CARDIN-IN-NAME | 001 | P  | 001 |
| FIELD | MS-CARDIN-IN-NEWNO | 001 | P  | 001 |

| AREA | MS-DEPARTMENT | 001 | P MINI-SYSTEM DEPARTMENT AREA  | DEP O10 |
| FILE | MS-DEPARTMENT | 001 | P MINI-SYSTEM DEPARTMENT TABLE  | DEP 001 |
| RECORD | MS-DEPARTMENT | 001 | P DEPARTMENT RECORD  | 001 |
| KEY | NS-DEPARTMENT,MS-DEPART-K-1 | 001 | P MINI-SYSTEM DEPARTMENT KEY 1  | 002 |
| KEY | MS-DEPARTMENT,MS-DEPART-K-2 | 001 | P MINI-SYSTEM DEPARTMENT ADOPT 004 |
| KEY | MS-DEPARTMENT,MS-DEPART-K-3 | 001 | P MINI-SYSTEM DEPARTMENT ORTN 005 |
| ELEMENT | NS-DEPARTMENT,MS-DEPARTMENT | 001 | P ENTIRE MINI-SYSTEM DEPT DEPT2 |
| FIELD | MS-DEPARTMENT,DEPTNO | 001 | P  | 001 |
| FIELD | MS-DEPARTMENT,DEPTNAME | 001 | P  | 001 |
| FIELD | MS-DEPARTMENT,DEPTNO | 001 | P  | 001 |
| FIELD | MS-DEPARTMENT,NAME | 001 | P  | 001 |

| AREA | MS-EMPLOYEE | 001 | P MINI-SYSTEM EMPLOYEE AREA  | EMP O10 |
| FILE | MS-EMPLOYEE | 001 | P MINI-SYSTEM EMPLOYEE TABLE  | EMP 002 |
| RECORD | MS-EMPLOYEE | 001 | P EMPLOYEE RECORD  | 001 |
| KEY | MS-EMPLOYEE,MS-EMPLOYEE-K-1 | 001 | P MINI-SYSTEM EMPLOYEE KEY 1  | 001 |
| KEY | MS-EMPLOYEE,MS-EMPLOYEE-K-2 | 001 | P MINI-SYSTEM EMPLOYEE KE ADOPT 003 |
| KEY | MS-EMPLOYEE,MS-EMPLOYEE-K-3 | 001 | P MINI-SYSTEM EMPLOYEE ORTN 004 |
| ELEMENT | MS-EMPLOYEE,MS-EMPLOYEE | 001 | P ENTIRE MINI-SYSTEM EMPLOYEE EMPLOYEE | 001 |
| ELEMENT | MS-EMPLOYEE,MS-EMPLOYEE-2 | 001 | P MINI-SYSTEM EMPLOYEE EMPLOYEE | 001 |
| FIELD | MS-EMPLOYEE,EMPHRNAME | 001 | P  | 001 |
| FIELD | MS-EMPLOYEE,EMPHRNAM | 001 | P  | 001 |
| FIELD | MS-EMPLOYEE,EMPHRNAM | 001 | P  | 001 |
| FIELD | MS-EMPLOYEE,FIRSTNAME | 001 | P  | 001 |

Diagram 3.7 ADR Test Data Base Configuration
```

3-14
<table>
<thead>
<tr>
<th>ENTITY-TYPE</th>
<th>OCCURRENCE</th>
<th>VER &amp; DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.FULLNAME</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.INICATE</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.JOBCODE</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.LASTNAME</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.LNAME</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.MIDNAME</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.MFONENO</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.POST</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.SALARY</td>
<td>008 P</td>
</tr>
<tr>
<td>FIELD</td>
<td>MS-EMPLOYEE.SEX</td>
<td>008 P</td>
</tr>
</tbody>
</table>

Figure 3.7 ADR Test Data Base Configuration
Figure 3.8 ADR Employee Table Layout and Keys
### VIEW ALL DISPLAY

**BASE NAME:** MINI-SYSTEM-DB  **BASE-ID:** 10  **RECORD NAME:** MS-DEPARTMENT

#### KEY NAME

- **MS-DEPART-K-1**
- **MS-DEPART-K-2**
- **MS-DEPART-K-3**

#### LV FIELD NAME

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>TYPE</th>
<th>LEN</th>
<th>DEC</th>
<th>SIGN</th>
<th>OCC</th>
<th>CL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADMRDEPT</td>
<td>C</td>
<td>3</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>DEPTNAME</td>
<td>C</td>
<td>36</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>DEPTNO</td>
<td>C</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>MGRNO</td>
<td>C</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>C</td>
<td>6</td>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### VIEW KEYS DISPLAY

**BASE NAME:** MINI-SYSTEM-DB  **BASE-ID:** 10  **RECORD NAME:** MS-DEPARTMENT

- **MS-DEPART-K-1** key consists of:
  - FIELD NAME | TYPE | LEN | DEC | SIGN | OCC | CL | DESCRIPTION
  - MGRNO       | C    | 6   |     |      | 1   | 2  |             |

- **MS-DEPART-K-2** key consists of:
  - FIELD NAME | TYPE | LEN | DEC | SIGN | OCC | CL | DESCRIPTION
  - ADMRDEPT   | C    | 3   |     |      | 1   | 2  |             |

- **MS-DEPART-K-3** key consists of:
  - FIELD NAME | TYPE | LEN | DEC | SIGN | OCC | CL | DESCRIPTION
  - DEPTNO     | C    | 3   |     |      | 1   | 2  |             |

---

Figure 3.9 ADR Department Table Layout and Keys
<table>
<thead>
<tr>
<th>SEQ</th>
<th>LEVEL</th>
<th>FIELD NAME</th>
<th>TYPE</th>
<th>CATALOGED</th>
<th>DATAVIEW</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>MS-EMPLOYEE-2-J</td>
<td>1</td>
<td>02/29/67</td>
<td>MS-EMPLOYEE-2-U</td>
<td>PRD</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>WORKCENT</td>
<td>X</td>
<td>3</td>
<td>UPD=NO</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>EMPNO</td>
<td>X</td>
<td>6</td>
<td>DEF=NO</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>FULLNAME</td>
<td>X</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>LASTNAME</td>
<td>X</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>FIRSTNAME</td>
<td>X</td>
<td>14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>MIDINIT</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>PHONEEND</td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **SEQ:** SEQUENCE NUMBER
- **LEVEL:** FIELD TYPE: X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, V=VARIABLE, G=DATE
- **TYPE:** I=INTERNAL NUMERIC TYPE: P=PECKED, Z=ZONED, B=BINARY
- **CATALOGED:** CH/OG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT. DEC
- **DATAVIEW:** K=KEY USAGE: K=COMPLETE KEY, P=PARTIAL KEY (HIGH ORDER POSITION)
- **BOUNDARY:** REDEF=REDEFINITION, DEF=DEPENDS ON

Figure 3.10 ADR Employee Dataview
**Figure 3.11 ADR Department Dataview**

<table>
<thead>
<tr>
<th>SEQ</th>
<th>FIELD NAME</th>
<th>FIELD TYPE</th>
<th>VALUE</th>
<th>DEPTNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MS-DEPARTMENT-2</td>
<td>X</td>
<td>3b</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>DEPTNAME</td>
<td>X</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- SEQ = SEQUENCE NUMBER
- FIELD TYPE:
  - X = ALPHANUMERIC
  - N = NUMERIC
  - U = UNSIGNED
  - V = VARIABLE
  - D = DATE
- I = INTERNAL NUMERIC TYPE:
  - P = PACKED
  - Z = ZONED
  - R = BINARY
- CHOG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT-DEC
- K (KEY USAGE):
  - W = WHOLE KEY
  - P = PARTIAL KEY
  - H = HIGH ORDER POSITION
- REDEF = REDEFINITION
- DEP = DEPENDING ON
### Figure 3.12 ADR Cardin Dataview

<table>
<thead>
<tr>
<th>FIELD NAME</th>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>SEQ LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-CARDIN-U</td>
<td>1</td>
<td>MS-CARDIN-U</td>
<td>1</td>
<td>MS-CARDIN-U</td>
<td>1</td>
<td>MS-CARDIN-U</td>
<td>1</td>
<td>MS-CARDIN-U</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>CARDIN-CYE</td>
<td>2</td>
<td>CARDIN-CYE</td>
<td>2</td>
<td>CARDIN-CYE</td>
<td>2</td>
<td>CARDIN-CYE</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>IN-ACTION</td>
<td>3</td>
<td>IN-ACTION</td>
<td>3</td>
<td>IN-ACTION</td>
<td>3</td>
<td>IN-ACTION</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>IN-LNAME</td>
<td>4</td>
<td>IN-LNAME</td>
<td>4</td>
<td>IN-LNAME</td>
<td>4</td>
<td>IN-LNAME</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>IN-ENAME</td>
<td>5</td>
<td>IN-ENAME</td>
<td>5</td>
<td>IN-ENAME</td>
<td>5</td>
<td>IN-ENAME</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>IN-EMPNO</td>
<td>6</td>
<td>IN-EMPNO</td>
<td>6</td>
<td>IN-EMPNO</td>
<td>6</td>
<td>IN-EMPNO</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>IN-NEWLO</td>
<td>7</td>
<td>IN-NEWLO</td>
<td>7</td>
<td>IN-NEWLO</td>
<td>7</td>
<td>IN-NEWLO</td>
<td>7</td>
</tr>
</tbody>
</table>

**Legend:**
- T (FIELD TYPE): X=HANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME
- V=VARIABLE, D=DATE
- I (INTERNAL NUMERIC TYPE): P=PACKED, Z=ZEROSHED, B=BINARY
- C/H/D (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT.DEC
- K (KEY USAGE): WHOLE KEY, PARTIAL KEY (HIGH ORDER POSITION)
- REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDEF=REDE
In Figure 3.13, the same DB2 and ADR entities (objects) are shown side-by-side in an indented report manner to allow the reader to judge the similarity of the two. This list of entities was the part implemented for the mini-system.
DB2

Storage group: DSN8G200

Data base: DSN8D2AP

Table space: DSN8S2DP

Table: DEPARTMENT

Index: DSN82.XDEPT1

Table space: DSN8S2EM

Area: MS-DEPARTMENT

Table: Employee

Index: DSN82.XEMPL1

Index: DSN82.XEMPL2

Area: MS-CARDIN

File: MS-CARDIN

Record: MS-CARDIN

Key: misc. keys

Element: misc. elms

Field: Misc. flds

Figure 3.13 Side-by-side Comparison

ADR

Database: Mini-system

Area: MS-DEPARTMENT

File: MS-DEPARTMENT

Record: MS-DEPARTMENT

Key: misc. keys

Element: misc. elms

Field: misc. flds
There were a few database duplication problems encountered in the implementing the DB2 database configuration at the ADR site that needed to be handled. The two databases used for the two mini-systems were essentially the same, see Figure 3.13.

DB2 has a Storage Group as its highest level Object. ADR has a Database as its highest level Entity. A DB2 Storage Group is a named set of DASD volumes on which DB2 can be stored.

In the hierarchy of a DB2 database are one or more Table Spaces logically comprising the database. Each Table Space has one or more table. A table is a relation of rows and columns. Data from one or more table can be co-located in the same Table Space (ie., file A record, file B record, file A record... co-located within the same Table Space). ADR's counterpart to the Table Space is the Area. An Area can contain one or more Files. Data can also be co-located within the same Area. Co-location of data can improve performance if the co-located records are frequently accessed together per
formance may suffer if they are not often accessed together.

3. In the hierarchy of an ADR File (table) are Records, which occur below the ADR file. DB2 has no counterpart the Record. ADR development has talked about eliminating the Record entity type entirely since there is always 1 to 1 relationship between record and file.

4. Below an ADR record is one or more keys, elements, and fields. Each ADR Database has one index which contains all keys for that database. A file can have one or more keys. An element is one or more contiguous fields from the file. There can be one or more elements for each record. An ADR dataview is related to fields by-way-of elements. There is a relationship between an ADR Dataview and one or more elements and there is a relationship between each element and one or more contiguous fields.

DB2's entities are a bit different. An ADR Key is called an Index. A Table can have one or more DB2 Indexes. An ADR database may have but one Index. There is

3-24
no counter part for an ADR Element.

DB2 calls fields, columns. A Table will have 1 or more Columns and 1 or more rows (records). DB2 Views of data are "mapped" to fields through SQL definition statements. See the example below:

Figure 3.14 provides a SQL definition of a View called PHONE, which is a view of telephone information. This view is used by the DB2 mini-system batch program. It was created externally to the batch program.
CREATE VIEW PHONE AS

SELECT DEPARTMENT.DEPTNAME, DEPARTMENT.DEPTNUMBER,
    EMPLOYEE.EMPLOYEENUMBER, EMPLOYEE.FIRSTNAME,
    EMPLOYEE.LASTNAME, EMPLOYEE.MIDDLEINITIAL,
    EMPLOYEE.PHONENUMBER
FROM DEPARTMENT,
    EMPLOYEE
WHERE DEPARTMENT.DEPTNO =
    EMPLOYEE.WORKDEPT

Figure 3.14 SQL Phone View Definition
At the ADR site, with the current software releases, a dataview can contain fields from only a single file. "Logical joins" of files are accomplished by nesting accesses to files (i.e., nesting Ideal "FOR ___ dataview" statements, see Figure 3.15). In future software releases, a "dataview processor" has been promised which would allow a single dataview to access fields from multiple files, or views simplifying the programming effort and providing a join of one or more files or views.
3.1 Program conversion problems

There were a few problems that came up in converting the two DB2 programs to Datacom/DB at the ADR site. This section contains a discussion of the problems that were encountered.

Both the DB2 batch report and the on-line program are written in Cobol embedded with SQL statements. Their counterparts were written in ADR/Ideal (not Cobol). Ideal was chosen instead of Cobol for a variety of reasons. First, since ADR products were acquired, Ideal has become the standard language used by the ADR site. Prior to Ideal, PL/1 was the shop standard. A second reason for programming in Ideal was that management at the ADR site preferred that IDEAL be used in this product comparison. The author of this paper works at that site and wanted the full cooperation of management on the project, so the author complied with the request.
Program listings and program output from the DB2 and ADR sites are provided in the Appendices as listed in the Table of Contents.

The DB2 on-line program was streamlined a bit at the ADR site out of practicality to remove irrelevant complication. The part that was streamlined was a series of DB2 programs involved in validation of user input. In timing program response times on the Employee Index, only valid input was used. Timings did not begin until after input was validated, hence the validation was not a factor in the timings and the DB2 validation programs were irrelevant complication.

The user is prompted for 5 things, 4 of which were validated (Major system, Action, Object and Search criteria). Valid input in the DB2 and ADR versions of the program was: "O" for Major System ("O" short for Organization), "D" for action ("D" short for Display), "EM" for Object ("EM" short for EMployee) and "EI" for Search Criteria ("EI" for Employee Id). A series of programs is involved in the DB2 system for this validation. Once valid input was received for these 4 prompts, the
Display Index portion of the on-line program, for example, then read a screen-full of records based on a read-key greater than or equal the user input provided on the fifth line of the screen (i.e., An employee id of 000001). Various records were read, data was gathered for each line of the index and the screen was then painted with an the first page of the index. The user could then press one of 3 PF keys to continue. (PF2 to resend the initial screen with all values removed, PF3 to end the program or PF8 to page forward on the index of records).

Data was gathered from two files, the EMPLOYEE file and DEPARTMENT file, in painting the screen.

DB2 SQL has CURSORS to aid in selection of EMPLOYEE and DEPARTMENT records. A cursor is a named control structure used by an application program to point to a row in a table. The position of the row is within some ordered set of rows, and the cursor is used to retrieve rows from the set. ADR has no such aid. Rows had to be selected from the MS-EMPLOYEE and MS-DEPARTMENT files by
nesting FOR statements in Ideal, see Figure 3.15.
FOR EACH MS-EMPLOYEE-2 NO UPDATE
    WHERE MS-EMPLOYEE-2.LASTNAME = WK-IN-LNAME
    ORDERED BY WORKDEPT EMPNO
    various statements
FOR EACH MS-DEPARTMENT-2 NO UPDATE
    WHERE MS-DEPARTMENT-2.DEPTNO =
        MS-DEPARTMENT-2.WORKDEPT
    various statements
WHEN NONE
    various statements
ENDFOR :FOR EACH MS-DEPARTMENT-2
WHEN NONE
    various statements
ENDFOR :FOR EACH MS-EMPLOYEE-2

Figure 3.15  ADR Ideal Nested FOR Statements
Another problem was access of JCL SYSIN card images with ADR's Ideal. One can do so only by calling a non-Ideal (i.e., Cobol or Assembler subprogram which accesses the SYSIN file). PL/1 subprogram interfaces were considered not a priority item with the Ideal development staff, and may be added at a later date.

Ideal handles I/O to and from ADR/Datacom/DB files and ADR/Ideal-defined Panels and Reports very well, non-Datacom file I/O is handled only through non-Ideal subprograms. In the mini-system, as is typically done at the ADR site, SYSIN card images were loaded in a Datacom/DB file using an online data entry utility.

ADR Panel definition and Report definition was very easy and flexible. ADR's Report Definition Facility made setting up and modifying a screen very easy. DB2 used CICS BMS (basic mapping support) which makes screen "painting" a very tedious task. DB2 provides no tools to aid in working with BMS, however, screen-painting software can be purchased from various vendors.
3.2 Program Timings

Program timing statistics were kept for each program. The batch and on-line programs were specifically run at times of the day when they were the only programs running and had no competition from any other program for computer resources. The DEPARTMENT file had 6 records in it. The EMPLOYEE file originally had just 32 records, but 1000 records were added at each site bringing the total number of EMPLOYEE records to 1032. Both sites had exactly the same data in their files. Those timings are given in Figure 3.16.
DB2
Batch report program
CPU time: 14.78 sec.
Actual time: .31 min.

On-line index display
Index 1st page: 3 sec.
PF8, Index 2nd pg: 3 sec.

ADR
Batch report program
CPU time: 13.05 sec.
Actual time: .25 min.

On-line index display
Index 1st page: 2 sec.
PF8, Index 2nd pg: 2 sec.

Figure 3.16 Program Timings
Chapter 4

Conclusions and Future Work

4.0 Conclusions

One would think that being a current user of ADR software, the author would be strongly prefer the more familiar ADR software over DB2 DBMS software. Prior to the study, this was the case. After the study of these two commercial DBMS software systems, that was not the case. An appreciation has developed for the potential flexibility and impressive features possessed by DB2 software.

Announced in June 1983, and not generally available until April 1985, DB2 has proved to be a very impressive set of products.

"ADR's DBMS is quasi-relational, utilizing a relational front end that provides end-user friendliness. This is not to say that it does not possess relational
capabilities. DB2 is slow compared to its competitors. However, it is a young product and because it is supported and enhanced by IBM, DB2's prospects for success are excellent...". [16] The logical data structure underlying the front end is hierarchical, which does not meet Date's criteria as being truly relational.

ADR/Datacom/DB is, nevertheless, an extremely popular DBMS. Although not a true representation of a relational system in the mold of IBM's SQL/DS or DB2, ADR/Datacom/DB does come with an impressive list of options and "add-ons" that allow the user to build a DBMS environment conducive to the operations of the installation.

In the second and third chapters of this paper, some differences between DB2 and ADR products were described. Out of all those differences, the most prominent is the view of data. DB2 provides much more flexibility in the construction of its views. DB2 view can be constructed dynamically by a program, if authorized. DB2 views can consist of fields from more than
one table, of fields from other views or from a combination of views and tables. The programmer does not need to be concerned with "logical joins" of 2 or more tables as he does with ADR software. A programmer using ADR software accomplishes "logical joins" of tables by nesting the file accesses. The programmer must be aware of the data that needs to be gathered and processed and whether it comes from one or more tables. A DB2 programmer may be provided with one preconstructed view of data which consists of fields from any combination of tables and/or views. The DB2 programmer's work is reduced in this respect. A dataview processor is promised in a future release of ADR products which should solve this for the ADR programmer.

ADR's fourth generation language, Ideal, provides many conveniences for the programmer which IBM's Cobol-embedded SQL does not have. The task of screen production in on-line programming is simplified greatly. ADR has PDF, a Panel (screen) Definition Facility, which provides separate fill-in screen-design aids which allows the programmer to "paint" sophisticated screen very simply and quickly. DB2 at this point uses CICS BMS (basic
mapping support). Screens can be painted using non-DB2 CICS screen design tools. None is provided with DB2 products.

Another Ideal time-saver is report production capabilities. Ideal provides RDF, a Report Definition Facility, to aid in the production of routine batch report production. Things such as column headings, page numbers, summary totals, level breaks, lines per page, etc. are handled very by RDF. The programmer defines the Report separately from the program and lists the report in the Resource Section of his Ideal program (along with any Dataviews, Sub programs or Panels used). DB2 provides a report writer, but the DB2 site in section 3's mini-system had not yet purchased it, so depended on an application programmer's report producing skills using Cobol.

Nothing is defined dynamically within an ADR program. All dataviews, indexes, etc. must be defined outside of the program. With the proper authorization, DB2 provides the flexibility of dynamic definition of entities within application programs. However, there are
some limitations in updating data when using views containing fields from more than 1 table. For non-update processing of data, however, this does remove some burden from the application programmer.

"4GLs (fourth generation languages) are easier for a programmer to use than Cobol. Novice programmers with very little commercial experience can learn Ideal 'and run with it' and become very productive very quickly. Independent software vendors will have to respond to this by modifying their tools to work with DB2. It's not difficult to speculate that soon you will be able to buy DB2 from IBM and acquire...Ideal from ADR to work with it." [54]

ADR in early 1987 announced support of DB2 and initially, a selected subset of SQL. ANSI (American National Standards Institute) recently recognized SQL as an industry data manipulation language standard.

All data in a DB2 data base is stored in VSAM entry sequenced data sets (ESDS), ADR uses Datacom /DA (direct access format) files.
DB2 product documentation, which is extensive, presents DB2 in a well-organized and straightforward manner and as such should be considered an additional product strength. ADR could use a some improvement in this area. ADR should pattern their documentation after DB2's.

The fact that DB2 can coexist with and complement IMS/VS/DB and allows concurrent access to data is a definite advantage. However, at this point DB2 is still being used primarily as an adjunct to the IMS/VS/DB, which causes DB2 to function less as a full relational DBMS, and more as a data language that gives IMS/VS/DB users relational capability. "DB2 coexists with IMS and can share data via an extraction module. Users can move from IMS to DB2 and still protect their investment in IMS applications". 16

"Except for SQL/DS, DB2 and a few others, most of the products offer an interpretive data sublanguage facility. To improve efficiency and obtain a higher performance, relational systems should support data sublanguage compilers instead of interpreters. In
addition to this, they should provide a precompiler for support of the callable procedural languages such as COBOL. This reduces the run-time system overhead and thus improves performance." 16

DB2 uses IMS DB/DC as its data dictionary. IBM has obviously not "re-invented the wheel" in a few areas of its software; VSAM files are used for data storage and IMS's DB/DC data dictionary is used for DB2. DB/DC would need to be purchased.

"Relational systems utilize a higher-level language for separating the translation from source code from the execution of target object code. Consequently, they require more computing resources for the translation to be effective, and human productivity is increased at the expense of the additional computing resources necessary." 16 With labor costs steadily rising and hardware costs dropping, this hardly appears to be a problem.

Figure 3.16 shows timings were kept for programs in the miniature application system programming effort of chapter 3. The purpose of the timings was to make
further comparison of the two DBMS product families through on-line and batch programs written in each environment. In this test there was very little difference in the timings for either set of programs, on-line or batch. The ADR programs did execute a bit faster than their counter parts.

It was interesting to have the timings as close as they were. The two test sites were selected because of their similarities in operating environments (see Figure 3.2, profiles of test sites). Conditions at the DB2 site were duplicated at the ADR site as closely as possible. Yet differences, as discussed in detail in chapter 3, did exist (i.e., Differences in the data base configurations, programming languages used, dynamic verses static access paths to data, etc.). On the small scale in which this test product comparison was conducted, approximately 1000 Employee and 10 Department file records were processed, timings were almost the same. On a much larger scale the differences in the timings may have been much greater. However, the mini-application programming effort of chapter 3 was an excellent vehicle in discussing DBMS product family differences in detail.
4.1 Future Work

It was mentioned before that ADR has announced a "support commitment for such industry standards ... as SQL, DB2, SQL/DS,... to equip users with a foundation for investment protection while being positioned for technology refresh". [16] ADR plans to start with a "subset of SQL". These plans were just announced in early 1987. No further details are available. The SQL subset will very probably contain no data definition statements. It will very likely contain only selected data manipulation statements. Providing SQL support will allow one to write a program access either a DB2 database or ADR database or both within the same program.

Future work could include a study of the subset of SQL that ADR chooses to support. Details of the interfacing products should be documented as well. A performance comparison similar to the effort in this paper could be done. Programs could be written demonstrating performance comparisons between ADR's SQL subset and DB2's data manipulation language.
ADR and DB2 DBMS products continue to evolve to meet their customer's needs. The versions of software for each family of products will need to provide downward support of existing customer database configurations, but likely will contain many new and exciting enhancements for the customer.
Bibliography

1. ADR Manual No. DB46-DB-00, Data Base Administration, 1986.
5. Anonymous, IBM upgrades, repositions DB2; Product's role poses threat to independent DBMS vendors, Computerworld, July 9, 1986, pg 15 focus.


19. Date, C.J., Relational Data Base Selected Writings, 1986.


26. Gallant, John, IBM pushes relational, updates DB2/IBM to support IMS full function for the 'Indefinite Future', Computerworld, February 10, 1986, pg 1, 4-5.


32. Hessinger, Paul, DB2 and IMS, Computerworld, December 4, 1985, pg 57-68.

33. IBM Alert, Query Management Facility (QMF) Version 2 Release 1, November 1986, pg 1-10.


43. IBM Software Catalog, Data Systems, Database 2 (DB2), 1986, pg 1-5.


49. McCusker, Tom, IBM bounces back, Datamation, July 1, 1985, pg 32, 34, 39, 42.

50. McEnaney, Maura, Codd: Relational approach to DBMS embraced by users, Computerworld, May 6, 1985, pg 41, 50.


Appendix A

ADR Programs and Output
<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>PHONE</th>
<th>EMPLOYEE NUMBER</th>
<th>DEPT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAAS</td>
<td>CHRISTINE</td>
<td>I</td>
<td>3978</td>
<td>000110</td>
<td>A01</td>
</tr>
<tr>
<td>LUCCHESI</td>
<td>VINCENTO</td>
<td>G</td>
<td>3490</td>
<td>000110</td>
<td>A01</td>
</tr>
<tr>
<td>OECDONELL</td>
<td>SEAN</td>
<td>M</td>
<td>2167</td>
<td>000120</td>
<td>A01</td>
</tr>
<tr>
<td>THOMPSON</td>
<td>MICHAEL</td>
<td>L</td>
<td>3476</td>
<td>000220</td>
<td>801</td>
</tr>
<tr>
<td>KHAN</td>
<td>SALLY</td>
<td>A</td>
<td>4328</td>
<td>000030</td>
<td>C01</td>
</tr>
<tr>
<td>QUINTANA</td>
<td>DELORES</td>
<td>M</td>
<td>4578</td>
<td>000130</td>
<td>C01</td>
</tr>
<tr>
<td>NICHOLLS</td>
<td>HEATHER</td>
<td>A</td>
<td>1193</td>
<td>000140</td>
<td>C01</td>
</tr>
<tr>
<td>STERN</td>
<td>DIVINE</td>
<td>F</td>
<td>5423</td>
<td>000050</td>
<td>A11</td>
</tr>
<tr>
<td>ADAMSON</td>
<td>BRUCE</td>
<td>T</td>
<td>4510</td>
<td>000150</td>
<td>A11</td>
</tr>
<tr>
<td>PIANKA</td>
<td>ELIZABETH</td>
<td>R</td>
<td>3782</td>
<td>000160</td>
<td>A11</td>
</tr>
<tr>
<td>YOSHIMURA</td>
<td>MASAOSHI</td>
<td>J</td>
<td>2890</td>
<td>000170</td>
<td>A11</td>
</tr>
<tr>
<td>SCONTEN</td>
<td>MARYLYN</td>
<td>S</td>
<td>1682</td>
<td>000180</td>
<td>A11</td>
</tr>
<tr>
<td>WALKER</td>
<td>JAMES</td>
<td>M</td>
<td>2986</td>
<td>000190</td>
<td>A11</td>
</tr>
<tr>
<td>HUNN</td>
<td>DAVID</td>
<td>T</td>
<td>4501</td>
<td>000200</td>
<td>A11</td>
</tr>
<tr>
<td>JONES</td>
<td>WILLIAM</td>
<td>M</td>
<td>0942</td>
<td>000210</td>
<td>A11</td>
</tr>
<tr>
<td>LUTZ</td>
<td>JENNIFER</td>
<td>K</td>
<td>0672</td>
<td>000220</td>
<td>A11</td>
</tr>
<tr>
<td>PULASKI</td>
<td>EVA</td>
<td>O</td>
<td>7831</td>
<td>000070</td>
<td>A21</td>
</tr>
<tr>
<td>JEFFERSON</td>
<td>JAMES</td>
<td>J</td>
<td>4265</td>
<td>000230</td>
<td>A21</td>
</tr>
<tr>
<td>MARINO</td>
<td>SALVATORE</td>
<td>M</td>
<td>3760</td>
<td>000240</td>
<td>A21</td>
</tr>
<tr>
<td>SMITH</td>
<td>DANIEL</td>
<td>S</td>
<td>0961</td>
<td>000250</td>
<td>A21</td>
</tr>
<tr>
<td>JOHNSON</td>
<td>SYBIL</td>
<td>V</td>
<td>0953</td>
<td>000260</td>
<td>A21</td>
</tr>
<tr>
<td>PEREZ</td>
<td>MARIA</td>
<td>L</td>
<td>9091</td>
<td>000270</td>
<td>A21</td>
</tr>
<tr>
<td>GEVER</td>
<td>JOHN</td>
<td>B</td>
<td>6789</td>
<td>000350</td>
<td>E01</td>
</tr>
<tr>
<td>MENDERSON</td>
<td>ELEEN</td>
<td>M</td>
<td>5648</td>
<td>000090</td>
<td>E11</td>
</tr>
<tr>
<td>SCHNEIDER</td>
<td>ETHEL</td>
<td>R</td>
<td>8991</td>
<td>000280</td>
<td>E11</td>
</tr>
<tr>
<td>PARKER</td>
<td>JOHN</td>
<td>R</td>
<td>4502</td>
<td>000290</td>
<td>E11</td>
</tr>
<tr>
<td>SMITH</td>
<td>PHILIP</td>
<td>X</td>
<td>2095</td>
<td>000300</td>
<td>E11</td>
</tr>
</tbody>
</table>

ADR - Batch program output
<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>PHONE</th>
<th>EMPLOYEE NUMBER</th>
<th>WORK NUMBER</th>
<th>DEPT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SETRIGHT</td>
<td>MAUDE</td>
<td>F</td>
<td>3332</td>
<td>000310</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>SPENSER</td>
<td>THEODORE</td>
<td>Q</td>
<td>0912</td>
<td>000100</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>LEE</td>
<td>RAMAL</td>
<td></td>
<td>2103</td>
<td>000320</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>GOUNTOT</td>
<td>JASON</td>
<td>R</td>
<td>5698</td>
<td>000330</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>GOUNTOT</td>
<td>JASON</td>
<td>R</td>
<td>5698</td>
<td>000340</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300002</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300003</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300004</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300005</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300006</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300007</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300008</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300009</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300010</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300011</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300012</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300013</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300014</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300015</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300016</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300017</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300018</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300019</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300020</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300021</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK DEPT NAME</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>--------</td>
<td>-----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300022</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300023</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300024</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300025</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300026</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300027</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300028</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300029</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300030</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300031</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300032</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300033</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300034</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300035</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300036</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300037</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300038</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300039</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300040</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300041</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300042</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300043</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300044</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300045</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300046</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300047</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>300048</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE NUMBER</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK DEPT NAME</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>--------------</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300049</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300050</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300051</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300052</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300053</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300054</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300055</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300056</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300057</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300058</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300059</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300060</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300061</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300062</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300063</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300064</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300065</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300066</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300067</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300068</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300069</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300070</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300071</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300072</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300073</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300074</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300075</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL PHONE</td>
<td>EMPLOYEE WORK WORK NUMBER</td>
<td>DEPT NAME</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>----------------</td>
<td>---------------------------</td>
<td>-----------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300076</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300077</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300078</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300079</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300080</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300081</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300082</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300083</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300084</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300085</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300086</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300087</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300088</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300089</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300091</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300092</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300093</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300094</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300095</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300096</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300097</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300098</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300099</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300100</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300001</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300002</td>
<td>E21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE NUMBER</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK DEPT NAME</td>
<td>WORK NUMBER</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>--------------</td>
<td>----------------</td>
<td>----------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300103</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300104</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300105</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300106</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300107</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300108</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300109</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300110</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300111</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300112</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300113</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300114</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300115</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300116</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300117</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300118</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300119</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300120</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300121</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300122</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300123</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300124</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300125</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300126</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300127</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300128</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000</td>
<td>300129</td>
<td>E21 SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE NUMBER</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK NUMBER</td>
<td>DEPT NAME</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
<td>-----------------</td>
<td>-------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300994</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300995</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300996</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300997</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300998</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>DOE</td>
<td>JOHN</td>
<td>E</td>
<td>0000 300999</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE NUMBER</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK DEPT NAME</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>--------------</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>THOMPSON</td>
<td>MICHAEL</td>
<td>L</td>
<td>3475</td>
<td>000020</td>
<td>BOE PLANNING</td>
<td></td>
</tr>
<tr>
<td>ADAMSON</td>
<td>BRUCE</td>
<td>J</td>
<td>4510</td>
<td>000130</td>
<td>011 MANUFACTURING SYSTEMS</td>
<td></td>
</tr>
<tr>
<td>JEFFERSON</td>
<td>JAMES</td>
<td>J</td>
<td>4265</td>
<td>000230</td>
<td>021 ADMINISTRATION SYSTEMS</td>
<td></td>
</tr>
<tr>
<td>JOHNSON</td>
<td>SYBIL</td>
<td>V</td>
<td>8953</td>
<td>000240</td>
<td>021 ADMINISTRATION SYSTEMS</td>
<td></td>
</tr>
<tr>
<td>HENDERSON</td>
<td>EILEEN</td>
<td>W</td>
<td>5478</td>
<td>000090</td>
<td>E11 OPERATIONS</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK NUMBER</td>
<td>DEPT NAME</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td>-----------------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
</tbody>
</table>

DSN90081 - NO EMPLOYEE FOUND IN TABLE
<table>
<thead>
<tr>
<th>LAST NAME</th>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>PHONE</th>
<th>EMPLOYEE NUMBER</th>
<th>WORK NUMBER</th>
<th>DEPT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brown</td>
<td>David</td>
<td>4501</td>
<td>00200</td>
<td>Oil Manufacturing Systems</td>
<td>6-11</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK NUMBER</td>
<td>DEPT NAME</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td>-----------------</td>
<td>-------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>JONES</td>
<td>WILLIAM</td>
<td>T</td>
<td>0942</td>
<td>000210</td>
<td>011</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>JOHNSON</td>
<td>SYBIL</td>
<td>V</td>
<td>8953</td>
<td>000260</td>
<td>021</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE</td>
<td>EMPLOYEE NUMBER</td>
<td>WORK DEPT NAME</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td>----------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>SMITH</td>
<td>DANIEL</td>
<td>S</td>
<td>0961</td>
<td>000250</td>
<td>AD ADMINISTRATION SYSTEMS</td>
<td></td>
</tr>
<tr>
<td>SMITH</td>
<td>PHILIP</td>
<td>X</td>
<td>2095</td>
<td>000300</td>
<td>E11 OPERATIONS</td>
<td></td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE</td>
<td>EMPLOYEE</td>
<td>WORK NUMBER</td>
<td>DEPT NAME</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
<td>-------</td>
<td>----------</td>
<td>-------------</td>
<td>-----------</td>
</tr>
<tr>
<td>DSM80041</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

DSM80041 - EMPLOYEE SUCCESSFULLY UPDATED
<table>
<thead>
<tr>
<th>IDENTIFICATION:</th>
<th>CCFA9581</th>
<th>VERSION:</th>
<th>001</th>
<th>STATUS:</th>
<th>TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM:</td>
<td>CCFA9581</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CREATED:</td>
<td>02/23/67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST MODIFIED:</td>
<td>02/25/67 at 16:50</td>
<td>BY GARTEN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST COMPILED:</td>
<td>02/25/67 at 16:51</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RUN STATUS:</td>
<td>PRIVATE</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SHORT DESCRIPTION:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LANGUAGE:</td>
<td>IDEAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DESCRIPTION:</td>
<td>ADR - Batch Program</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACTUAL DATE:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RESOURCES:</td>
<td>CCFA95BI</td>
<td>VERSION: 001</td>
<td>STATUS: TEST</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
<td>--------------</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATAVIEW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PANEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>REPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROGRAM VERS.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SYS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS-CARDIN-U</td>
<td>0001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS-DEPARTMENT-2</td>
<td>0002</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS-EMPLOYEE-2</td>
<td>0006</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCFA95R1 0001</td>
<td>CCFA95V1 0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CCFA99V1 0005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ LEVEL</td>
<td>FIELD NAME</td>
<td>T</td>
<td>CH/DG</td>
<td>OCCUR</td>
<td>K</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---</td>
<td>-------</td>
<td>-------</td>
<td>---</td>
</tr>
<tr>
<td>CATALOGED</td>
<td>02/24/87 15:22</td>
<td>DATAADV/DB UPD=YES DBID=010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>NS-CARDIN-U</td>
<td>X</td>
<td>1</td>
<td>P</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IN-ACTION</td>
<td>X</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IN-LNAME</td>
<td>X</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>IN-FNAME</td>
<td>X</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>IN-NEWMC</td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- SEQ = SEQUENCE NUMBER
- T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COMMON, NAME, V=VARIABLE, D=DATE
- I (INTERNAL NUMERIC TYPE): P=PACKED, Z=ZONED, B=BINARY
- CH/DG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT. DEC
- K (KEY USAGE): K=WHOLE KEY, P=PARTIAL KEY (HIGH ORDER POSITION)
- REDEF = REDEFINITION, DEP ON = DEPENDING ON
<table>
<thead>
<tr>
<th>SEQ</th>
<th>LEVEL</th>
<th>FIELD NAME</th>
<th>TYPE</th>
<th>OCCUR</th>
<th>VALUE/REDEF/DEP ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>HS-DEPARTMENT-2</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>DEPTNO</td>
<td>X</td>
<td>3</td>
<td>K</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>DEPTNAME</td>
<td>X</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- SEQ: SEQUENCE NUMBER
- T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND. NAME, V=VARIABLE, D=DATE
- I (INTERNAL NUMERIC TYPE): P=PACKED, Z=ZONED, B=BINARY
- CN/DS (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT. DEC
- K (KEY USAGE): W=WHOLE KEY, P=PARTIAL KEY (HIGH ORDER POSITION)
- REDEF=REDEFINITION, DEP ON=DEPENDING ON

**Cataloged:** 02/16/87 14:02

**Data On DB:** UPD=YES DBID=010

**Data View:** MS-DEPARTMENT-2

**Version:** 002

**Status:** PROD

**Date View:** HS-DEPARTMENT-2

**Version:** 002

**Status:** PROD
## DATAVIEW: MS-EMPLOYEE-2

<table>
<thead>
<tr>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>T</th>
<th>I</th>
<th>CH/DS</th>
<th>OCCUR</th>
<th>K</th>
<th>VALUE/REDEF/DEP ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MS-EMPLOYEE-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>WORKDEPT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>3</td>
<td>K</td>
</tr>
<tr>
<td>3</td>
<td>EMPNO</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>6</td>
<td>K</td>
</tr>
<tr>
<td>4</td>
<td>FIRSTNAME</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LASTNAME</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>15</td>
<td>P</td>
</tr>
<tr>
<td>6</td>
<td>MIDINIT</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PHONENO</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **SEQ**: SEQUENCE NUMBER
- **T**: FIELD TYPE: X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=CONST. NAME, V=VARIABLE, D=DATE
- **I**: (INTERNAL NUMERIC TYPE): P=PACKED, Z=ZONED, Z=BINARY
- **CH/DS**: (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INT.DEC
- **K**: KEY USAGE: W=WHOLE KEY, P=PARTIAL KEY (HIGH ORDER POSITION)
- **REDEF**: REDEFINITION, DEP ON: DEPENDING ON

**Cataloged Date/Time:** 02/16/87 15:45
**Datacom/DB UPO=** YES
**OBID=DID**
<table>
<thead>
<tr>
<th>SEQ</th>
<th>LEVEL</th>
<th>FIELD NAME</th>
<th>T CH/OG</th>
<th>OCCUR</th>
<th>VAL/COMMENT/REDEF/OEP ON/COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>000300</td>
<td>1</td>
<td>WK-IN-LNAME-LEN</td>
<td>N</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>000301</td>
<td>1</td>
<td>WK-IN-FNAME-LEN</td>
<td>N</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>000303</td>
<td>1</td>
<td>WK-LENGTH</td>
<td>N</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>000304</td>
<td>1</td>
<td>WK-BLANK-POS</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>000305</td>
<td>1</td>
<td>WK-BLANK-POS-1</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>000307</td>
<td>1</td>
<td>WK-START-POS</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>000308</td>
<td>1</td>
<td>WK-INPUT-LEN</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>000309</td>
<td>1</td>
<td>WK-LASTNAME</td>
<td>X</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>000310</td>
<td>1</td>
<td>WK-IN-LNAME</td>
<td>V</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>000311</td>
<td>1</td>
<td>WK-IN-FNAME</td>
<td>V</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>000312</td>
<td>1</td>
<td>WK-PERCENT-POS</td>
<td>N</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>000500</td>
<td>1</td>
<td>ERROR-MSG-LINE</td>
<td>X</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>000502</td>
<td>2</td>
<td>EM-OSNB</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>004900</td>
<td>2</td>
<td>EM-MSGCODE</td>
<td>X</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>004901</td>
<td>2</td>
<td>EM-TEXT</td>
<td>X</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>004902</td>
<td>1</td>
<td>RPT-99-BLANKS</td>
<td>X</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>005001</td>
<td>1</td>
<td>RPT-Detail-Line</td>
<td>X</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>005100</td>
<td>2</td>
<td>RPT-LNAME</td>
<td>X</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>005101</td>
<td>2</td>
<td>RPT-FILLER-1</td>
<td>X</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>005200</td>
<td>2</td>
<td>RPT-FNAME</td>
<td>X</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>005201</td>
<td>2</td>
<td>RPT-FILLER-2</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>005300</td>
<td>2</td>
<td>RPT-MIDLEINIT</td>
<td>X</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>005301</td>
<td>2</td>
<td>RPT-FILLER-3</td>
<td>X</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>005400</td>
<td>2</td>
<td>RPT-PHONENO</td>
<td>X</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>005401</td>
<td>2</td>
<td>RPT-FILLER-4</td>
<td>X</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>005500</td>
<td>2</td>
<td>RPT-EMPLNO</td>
<td>X</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>005501</td>
<td>2</td>
<td>RPT-FILLER-5</td>
<td>X</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>005600</td>
<td>2</td>
<td>RPT-WORKDEPT</td>
<td>X</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>005601</td>
<td>2</td>
<td>RPT-FILLER-6</td>
<td>X</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>005700</td>
<td>2</td>
<td>RPT-DEPTNAME</td>
<td>X</td>
<td>36</td>
<td></td>
</tr>
</tbody>
</table>

LEGEND:

T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNSIGNED, C=COND NAME, F=FLAG, V=VARIABLE, D=DATE

I (INTERNAL NUMERIC TYPE): BLANK OR P=PACKED, Z=ZONED, B=BINARY

CN/CG (CHARACTERS/DIGITS): NUMBER OF CHARACTERS, INTEGERS, OR INTEGERS.DECIMALS

REDEF= REDEFINITION, OEP ON= DEPENDING ON
REPORT: CCFA90R1
VERSION: ODI
STATUS: TEST

IDENTIFICATION:

REPORT NAME: CCFA90R1
CREATED: 02/12/87
LAST MODIFIED: 02/21/87 AT 11:47

DESCRIPTION:

REPORT PARAMETERS:

LINES PER PAGE ON PRINTOUT: 60 (1 THRU 250)
REPORT WIDTH: 132 (40 THRU 230)
SPACING BETWEEN LINES: 2 (1 THRU 3)
SPACING BETWEEN COLUMNS: 2 (0 THRU 66 OR A=AUTOMATIC)
SUMMARY ONLY: N (Y=YES, N=NO)
COLUMN HEADINGS DESIRED: Y (Y=YES, N=NO)
COLUMN HEADINGS INDICATION: N (U=UNDERSCORE, N=NONE, D=DASHES)
CONTROL BREAK INDICATION: N (Y=YES, N=NO)
CONTROL BREAK FOOTING: Y (Y=YES, N=NO)
COLUMN CONTINUATION AT TOP OF PAGE: N (Y=YES, N=NO)
ANNOTATED COUNT IN CONTROL FOOTING: N (Y=YES, N=NO)
REPORT FINAL SUMMARY TITLE: N (Y=YES, N=NO)
SPACING BEFORE SUMMARY TITLE: I (1 THRU 9 = LINES, P=NEW PAGE)

DATE POSITION: NO (N=NONE, BR=BOT. RIGHT, BL=BOT. LEFT, BC=BOT. CTR., TR=TOP RIGHT, TL=TOP LEFT, IC=TOP CENTER)

PAGE NUMBERS POSITION: NO (D=DIGITS ONLY, N=WITH HYPHENS, P=PAGE NNN)

PAGE HEADING POSITION: C (C=CENTE, L=LEFT JUSTIFY, R=RIGHT JUSTIFY)
## REPORT: CCFA90R1

**VERSION:** 001  **STATUS:** TEST

### PAGE HEADING:

<table>
<thead>
<tr>
<th>SEQ</th>
<th>FIELD NAME, LITERAL, FUNCTION, OR ARITHMETIC EXPRESSION</th>
<th>COLUMN</th>
<th>WIDTH</th>
<th>EDIT PATTERN</th>
</tr>
</thead>
<tbody>
<tr>
<td>00300</td>
<td>TELEPHONE DIRECTORY*</td>
<td>LO1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00301</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00400</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>00500</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SEQ</th>
<th>FIELD NAME, LITERAL, FUNCTION, OR ARITHMETIC EXPRESSION</th>
<th>COLUMN</th>
<th>WIDTH</th>
<th>EDIT PATTERN</th>
</tr>
</thead>
<tbody>
<tr>
<td>00600</td>
<td>LAST NAME*</td>
<td>LO2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00700</td>
<td>FIRST NAME*</td>
<td>018</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00800</td>
<td>INITIAL*</td>
<td>034</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00900</td>
<td>PHONE*</td>
<td>039</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01000</td>
<td>EMPLOYEE*</td>
<td>046</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01100</td>
<td>WORK*</td>
<td>055</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01200</td>
<td>DEPT*</td>
<td>046</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01300</td>
<td>NAME*</td>
<td>055</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### LEGEND:

- SEQ=SEQUENCE NUMBER
- WIDTH: 0-99=USER DEFINED
- TAB: +NN=RELATIVE SPACING, NNN=ABSOLUTE SPACING, LNN=VERTICAL SPACING
- P=TOP OF NEW PAGE

---

6-22
### REPORT: CCFA9901

**VERSION:** CCFA9901  
**STATUS:** TEST

#### DETAIL:

<table>
<thead>
<tr>
<th>SEQUENCE</th>
<th>FUNCTION</th>
<th>ARITHMETIC EXPRESSION</th>
<th>SORT</th>
<th>BREAK</th>
<th>HEADING</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>LITERAL</td>
<td>FUNCTION</td>
<td>COLUMN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field</td>
<td>FIELD NAME</td>
<td>LITERAL</td>
<td>FUNCTION</td>
<td>ARITHMETIC EXPRESSION</td>
<td>SORT</td>
</tr>
<tr>
<td>NAME</td>
<td>FIELD NAME</td>
<td>LITERAL</td>
<td>FUNCTION</td>
<td>ARITHMETIC EXPRESSION</td>
<td>SORT</td>
</tr>
<tr>
<td>NAME</td>
<td>FIELD NAME</td>
<td>LITERAL</td>
<td>FUNCTION</td>
<td>ARITHMETIC EXPRESSION</td>
<td>SORT</td>
</tr>
</tbody>
</table>

#### LEGEND:

- **LEVEL:** 1 = MAJOR, 2-9 = MINOR
- **A/D:** ASCENDING/DESCENDING: A = ASCENDING, D = DESCENDING
- **SORT:** T = TOP OF NEW PAGE, 0 = NUMBER OF LINES
- **BREAK:** P = TOP OF NEW PAGE, 0 = NUMBER OF LINES
- **FUNCTION:** O = OMITTED, G = NEW GROUP, N = NONE, R = REPEATED
- **EDIT:** T = TOTAL, S = SINGLE SUMMARY LINE
- **ARGUMENTS:** MAX = MAXIMUM, N = NONE, U = USER DEFINED

- **REPORT INTERNAL RECORD SIZE:** 95 BYTES

---

6-23
PROCEDURE: CCFA95BI

STATEMENT

100 IDEAL BATCH PGM CCFA95BI - FOR NNNI-SYSTEM DMO
200 <MAIN> PROCEDURE
300 FOR EACH MS-CARDIN-U NO UPDATE
400 SET RPT-DETAIL-LINE = RPT-99-BLANKS
500 GO PROCESS-INPUT
600 WHEN NONE
700 PRODUCE CCFA90R1
800 RELEASE REPORT CCFA90R1
900 ENDFOR

2000 :
2100 :
2200 : PROCESS INPUT CARD IMAGES
2300 :
2400 :
2500 <PROCESS-INPUT> PROCEDURE
2600 :
2700 SELECT FIRST ACTION
2800 WHEN IN-ACTION = "F"
2900 GO LIST-FUNCTION
3000 WHEN IN-ACTION = "U"
3100 WHEN PHONE-UPDATE
3200 WHEN OTHER
3300 MOVE "0681" TO EN-MSGCODE
3400 CALL CCFA95VI USING EN-MSGCODE EN-TEXT
3500 PRODUCE CCFA90R1
3600 ENDPROC
3800 ENDFOR :PROCESS-INPUT PROC
4000 :
4100 := LIST EMPLOYEES
4300 :
4400 :
4500 <LIST-FUNCTION> PROCEDURE
4600 WHEN IN-LNAME = "F"
4700 IF IN-LNAME = "F"
4800 MOVE "0681" TO WK-IN-LNAME
4900 ELSE
5000 :NULL
5100 ENDFOR
5300
PROCEDURE:  CCA9581  

VERSION:  001  STATUS:  TEST

SEQ  STATEMENT

0360  "NO FIRST NAME GIVEN"
0370  0380  IF IN-FNAME = **SPACES
0390  0400  MOVE 'Z' TO WK-IN-FNAME
0410  0420  ELSE
0430  0440  :NULL
0450  0460  ENOIF
0470  0480  0490  :LIST ALL EMPLOYEES
0500  0510  IF IN-LNAME = **
0520  0530  :NO LIST-ALL
0540  0550  ELSE
0560  0570  :TRIM TRAILING BLANKS FROM LAST NAME
0580  0590  SET WK-PERCENT-POS = $LENGTH(IN-LNAME)
0600  0610  SET WK-IN-LNAME = STRIM(IN-LNAME,RIGHT='')
0620  0630  SET WK-IN-LNAME = STRIM(IN-LNAME,RIGHT='')
0640  0650  SET WK-IN-LNAME = STRIM(IN-LNAME,RIGHT='')
0660  0670  SET WK-IN-LNAME = STRIM(IN-LNAME,RIGHT='')
0680  0690  IF WK-PERCENT-POS > 0
0700  0710  ELSE LIST SPECIFIC NAMES
0720  0730  ELSE LIST GENERIC NAMES
0740  0750  ENDIF
0760  0770  :ENDPROC  "LIST-FUNCTION"

0780  0790  :ENDPROC  "LIST-FUNCTION"

0800  0810  :ENDPROC  "LIST-FUNCTION"

0820  0830  :ENDPROC  "LIST-FUNCTION"

0840  0850  :ENDPROC  "LIST-FUNCTION"

0860  0870  :ENDPROC  "LIST-FUNCTION"

0880  0890  :ENDPROC  "LIST-FUNCTION"

0900  0910  :ENDPROC  "LIST-FUNCTION"

0920  0930  :ENDPROC  "LIST-FUNCTION"

0940  :ENDPROC  "LIST-FUNCTION"

9500  FOR EACH MS-EMPLOYEE-2 NO UPDATE
9600  ORDERED BY WORKDEPT EMPNO
9700  SET RPT-LNAME = LASTNAME
9800  SET RPT-FNAME = FIRSTNAME
9900  SET RPT-MIDINIT = MIDDLEINIT
1000  SET RPT-EMPNO = EMPNO
1010  SET RPT-PHONENO = PHONENO
1020  SET RPT-WORKDEPT = WORKDEPT
1030  LIST ALL EMPLOYEES

END
PROCEDURE: CCFA9ROI  VERSION: OD  STATUS: TEST

SEQ STATEMENT
10500 FOR EACH MS-DEPARTMENT-2 NO UPDATE
10600 WHERE MS-DEPARTMENT-2.OEPTNO =
10700 MS-EMPLOYEE-2.WORKDEPT
10800 SET RPT-OEPTNAME = OEPTNAME
10900 PRODUCE CCFA901
11000 WHEN NONE
11100 SET RPT-DETAIL-LINE =
11200 NO DEPARTMENT RECORDS FOUND - CONTACT PROGRAMMER;
11300 PRODUCE CCFA901
11400 ENFOR

11500
11600 WHEN NONE
11700 MOVE *0091* TO EM-MSGCODE
11800 CALL CCFA9501 USING EM-MSGCODE EM-TEXT
11900 MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
12000 PRODUCE CCFA901
12100 ENFOR
12200
12300 RELEASE REPORT CCFA901
12400 ENPROC
12500
12600 
12700 : LIST EMPLOYEES GIVEN GENERIC KEY
12800 :
12900 :
13000 <LIST-GENERIC> PROCEDURE
13100 : FIND OUT IF HAVE LEADING OR TRAILING 
13200 
13300 IF WK-PERCENT-POS > 1
13400 00 NAME-BEGIN-WITH
13500 ELSE
13600 00 NAME-END-WITH
13700 ENOPF
13800
13900 ENPROC
14000
14100
14200 : LIST EMPLOYEES WHOSE LASTNAME BEGINS WITH...
14300 :
14400 :
14500 <NAME-BEGIN-WITH> PROCEDURE
14600 : WHERE JO3
14700 SET WK-IN-LNAME-LEN = LENGTH(WK-IN-LNAME) - 1
14800 SET WK-IN-LNAME = $SUBSTR(WK-IN-LNAME,START=1*,
14900 LENGTH=WK-IN-LNAME-LEN)
15000
15300

6-26
PROCEDURE: CCA9501

SEQ STATEMENT

15400 FOR EACH MS-EMPLOYEE-2 NO UPDATE
15500 WHERE MS-EMPLOYEE-2.LASTNAME >= WK-IN-LNAME ORDERED BY WORKDEPT EMPNO
15700
15800 IF SUBSTR(LASTNAME,START=1,LENGTH=WK-IN-LNAME-LEN) = WK-IN-LNAME
15900 SET RPT-LNAME = LASTNAME
16000 SET RPT-FNAME = FIRSTNAME
16100 SET RPT-MIDDLEINIT = MIDDLEINIT
16200 SET RPT-PHONENO = PHONENO
16300 SET RPT-EMPNO = EMPNO
16400 SET RPT-WORKDEPT = WORKDEPT
16500
16600 FOR FIRST MS-DEPARTMENT-2 NO UPDATE
16700 WHERE MS-DEPARTMENT-2.DEPTNO = MS-EMPLOYEE-2.WORKDEPT
16900
17000 SET RPT-DEPTNAME = DEPTNAME;
17200 WHEN NONE
17300 SET RPT-DETAIL-LINE =
17400 "NO DEPARTMENT RECORDS FOUND - CONTACT PROGRAMMER;"
17500 PRODUCE CCAF901
17600 ENDFOR ;FOR EACH MS-DEPARTMENT-2
17700
17800 ENDIS
17900 WHEN NONE
18000 MOVE '0091' TO EM-MSGCODE
18100 CALL CCAF9501 USING EM-MSGCODE EM-TEXT
18200 MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
18300 PRODUCE CCAF901
18400 ENDFOR ;FOR EACH MS-EMPLOYEE-2
18500
18600 RELEASE REPORT CCAF901
18700 ENDPROC 'NAME-BEGIN-WITH'
18800
18900 "LIST EMPLOYEES WHOSE LASTNAME ENDS WITH ***
19000 "
19100 "
19200 "
19300 "NAME-BEGIN-WITH" PROCEDURE
19400 "
19500 "
19600 "
19700 SET WK-IN-LNAME-LEN = SLENGTH(WK-IN-LNAME) - 1
19900 SET WK-IN-LNAME = SUBSTR(WK-IN-LNAME,START=2)
20000 "LENGTH(WK-IN-LNAME-LEN)
20100 "
20200 FOR EACH MS-EMPLOYEE-2 NO UPDATE
20300 WHERE MS-EMPLOYEE-2.LASTNAME >= WK-IN-LNAME
6-27
**PGM CCFA950I**  **COL: TEST**  **SYS: &IO**  **FEBRUARY 25, 1987 18:29:01**

**PROCEDURE: CCFA950I**  **VERSION: 001**  **STATUS: TEST**

```
0 STATEMENT

20400 ORDERED BY WORKDEPT. EMPNO
20600 SET WK-BLANK-POS-1 = 3 + INDEX(LASTNAME, SEARCH=' ')
21300 IF KSUBSTR(MS-EMPLOYEE-2.LASTNAME,START=WK-START-POS, LENGTH=3) = WK-IN-LNAME
21400 IF SUBSTR(MS-EMPLOYEE-2.LASTNAME,START=WK-START-POS,
21500   LENGTH=3) = WK-IN-LNAME
21300   SET RPT-LNAME = LASTNAME
21700   SET RPT-FNAME = FNAME
21800   SET RPT-MIDDLEINIT = MIDDLE
21900   SET RPT-NAME = NAME
22000   SET RPT-EMPNO = EMPNO
22100   SET RPT-WORKDEPT = WORKDEPT
22200   FOR EACH MS-DEPARTMENT-2 NO UPDATE
22300     WHERE MS-DEPARTMENT-2.DEPTNO = WK-DEPTNO
22400     MS-EMPLOYEE-2.EMPNO = WK-EMPNO
22500     USE RPT-DEPTNAME = DEPTNAME
22600     PRODUCE CCFA90R1
22700     WHEN NONE
22800       SET RPT-DETAIL-LINE = "NO DEPARTMENT RECORDS FOUND - CONTACT PROGRAMMER"
23000       PRODUCE CCFA90R1
23100 ENDIF
23200 ENDFOR FOR EACH MS-DEPARTMENT-2
23300 ENDOF
23400 WHEN NONE
23500   MOVE "0801" TO EM-MSGCODE
23600   CALL CCFA95VI USING EM-MSGCODE SM-TEXT
23700   MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
23800   PRODUCE CCFA90R1
23900 ENDOF
24000 WHEN NONE
24100   ENDFOR FOR EACH MS-EMPLOYEE-2
24200 ENDOF
24300 RELEASE REPORT CCFA90R1
24500 ENDOF NAME=END-WITH
24600 ENDOF
24800 LIST EMPLOYEES MATCHING SPECIFIC NAME PROVIDED
24900 LIST-SPECIFIC PROCEDURE
25000 ORDERED BY WORKDEPT. EMPNO
25000 IF SUBSTR(WK-IN-FNAME, START=1, LENGTH=1) = " " OR $SPACE$;
25000   USE JUST THE LAST NAME, NOT THE FIRST
25000 FOR EACH MS-EMPLOYEE-2 NO UPDATE
25000 ORDERED BY WORKDEPT. EMPNO
25000 WHERE MS-EMPLOYEE-2.LASTNAME = WK-IN-LNAME
```

6-28
**PGM CCFAF850I  JO1  TEST  SYSTEM  FEBRUARY 25, 1987  18:29:01**

**PROCEDURE: CCFAF850I**  
**VERSION: 001**  
**STATUS: TEST**

```
25010  SET RPT-LNAME = LASTNAME
25011  SET RPT-FNAME = FIRSTNAME
25013  SET RPT-MIDDLEINIT = MIDDLEINIT
25014  SET RPT-PHONENO = PHONENO
25015  SET RPT-EMPNO = EMPNO
25016  SET RPT-WORKDEPT = WORKDEPT

25017  FOR EACH MS-DEPARTMENT-2 NO UPDATE
25019    WHERE MS-DEPARTMENT-2-DPTNO =
25020    MS-EMPLOYEE-2-WORKDEPT
25021    SET RPT-DPTNAME = DPTNAME
25022    PRODUCE CCFAF90R1
25023    WHEN NONE
25024      SET RPT-DETAIL-LINE =
25025      "NO DEPARTMENT RECORDS FOUND - CONTACT PROGRAMMER"
25026      PRODUCE CCFAF90R1
25027    ENDFOR
25028  ENDFOR :FOR EACH MS-DEPARTMENT-2

25029  WHEN NONE
25031    MOVE "00001" TO EN-MSGCODE
25032    CALL CCFAF95VT USING EN-MSGCODE EM-TEXT
25033    MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
25034    PRODUCE CCFAF90R1
25035    ENDFOR :FOR EACH MS-EMPLOYEE-2
25036
25100  ELSE
25200  FOR EACH MS-EMPLOYEE-2 NO UPDATE
25300    ORDERED BY WORKDEPT EMPNO
25400    WHERE MS-EMPLOYEE-2-LASTNAME = IN-LNAME AND
25500      MS-EMPLOYEE-2-FIRSTNAME = IN-FNAME
25600
25700    SET RPT-LNAME = LASTNAME
25800    SET RPT-FNAME = FIRSTNAME
25900    SET RPT-MIDDLEINIT = MIDDLEINIT
26000    SET RPT-PHONENO = PHONENO
26100    SET RPT-EMPNO = EMPNO
26200    SET RPT-WORKDEPT = WORKDEPT

26400  FOR EACH MS-DEPARTMENT-2 NO UPDATE
26500    WHERE MS-DEPARTMENT-2-DPTNO =
26600    MS-EMPLOYEE-2-WORKDEPT
26700    SET RPT-DPTNAME = DPTNAME
26800    PRODUCE CCFAF90R1
26900    WHEN NONE
```

6-29
SEQ STATEMENT

27000 SET RPT-DETAIL-LINE =
27100 "NO DEPARTMENT RECORDS FOUND - CONTACT PROGRAMMER"
27200 PRODUCE CCFA99R1
27300
27400 ENDFOR :FOR EACH MS-DEPARTMENT-2
27500
27600 WHEN NONE
27700 MOVE "0081" TO EM-MSGCODE
27800 CALL CCFA99VI USING EM-MSGCODE EM-TEXT
27900 MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
28000 PRODUCE CCFA99R1
28100 ENDFOR :FOR EACH MS-EMPLOYEE-2
28101 EN0IF
28200
28300 RELEASE REPORT CCFA99R1
28400 EN0PROC :LIST-SPECIFIC
28500
28600:
28700 : UPDATE EMPLOYEE PHONE NUMBERS
28800 :
28900 :
29000 "<PHONE-UPDATE>" PROCEDURE
29100
29200 FOR EACH MS-EMPLOYEE-2
29300 ORDERED BY WORKDEPT EMPNO
29400 WHERE MS-EMPLOYEE-2.EMPNO = IN-EMPNO
29500 SET RPT-PHNOENO = "*4269"
29600 MOVE "0041" TO EM-MSGCODE
29700 CALL CCFA99VI USING EM-MSGCODE EM-TEXT
29800 MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
29900 PRODUCE CCFA99R1
30000 WHEN NONE
30100 MOVE "0081" TO EM-MSGCODE
30200 CALL CCFA99VI USING EM-MSGCODE EM-TEXT
30300 MOVE ERROR-MSG-LINE TO RPT-DETAIL-LINE
30400 PRODUCE CCFA99R1
30500 ENDFOR
30600
30700 RELEASE REPORT CCFA99R1
30800 EN0PROC :PHONE-UPDATE
30900
31000 "<ERROR>" PROCEDURE
31100 ERROR PROCEDURE TRAPS AND PRINTS ERRORS NOT TRAPPED ABOVE
31300 :
31400 "<ERROR>" PROCEDURE
31500 CALL CCFA99VI USING INPUT "10"

6-30
<table>
<thead>
<tr>
<th>SEQ</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>31400</td>
<td>QUIT RUN</td>
</tr>
<tr>
<td>31700</td>
<td>ENDPROC ERROR</td>
</tr>
<tr>
<td>31800</td>
<td></td>
</tr>
</tbody>
</table>

6-31
PROGRAM HAS BEEN SUCCESSFULLY COMPILED

NO ERROR MESSAGE(S) FLAGGED IN THIS COMPILATION
NO WARNING MESSAGE(S) FLAGGED IN THIS COMPILATION
NO ADVISORY MESSAGE(S) FLAGGED IN THIS COMPILATION

COMPILED PROGRAM STATISTICS:
- NUMBER OF SHAREABLE MEMBERS OF THE PROGRAM = 001
- LENGTH OF SHAREABLE PORTIONS OF THE PROGRAM = 7,012 BYTES
- NUMBER OF NON-SHAREABLE MEMBERS OF THE PROGRAM = 001
- LENGTH OF NON-SHAREABLE PORTIONS OF THE PROGRAM = 5,812 BYTES
SELECTING AN EMPLOYEE TO DISPLAY

MAJOR SYSTEM ......: O
ACTION .............: D
OBJECT .............: EM
SEARCH CRITERIA ...: EN
DATA ..............: %

<table>
<thead>
<tr>
<th>NO</th>
<th>D/ID</th>
<th>DEPARTMENT NAME</th>
<th>E/ID</th>
<th>EMPLOYEE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000010</td>
<td>CH HASS</td>
</tr>
<tr>
<td>02</td>
<td>B01</td>
<td>PLANNING</td>
<td>000020</td>
<td>ML THOMPSON</td>
</tr>
<tr>
<td>03</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>000030</td>
<td>SAI Kawan</td>
</tr>
<tr>
<td>04</td>
<td>E01</td>
<td>SUPPORT SERVICES</td>
<td>000050</td>
<td>JB BEYER</td>
</tr>
<tr>
<td>05</td>
<td>D01</td>
<td>MANUFACTURING</td>
<td>000060</td>
<td>JF STERN</td>
</tr>
<tr>
<td>06</td>
<td>D02</td>
<td>ADMINISTRATION SYSTEMS</td>
<td>000070</td>
<td>ED PULASKI</td>
</tr>
<tr>
<td>07</td>
<td>E02</td>
<td>OPERATIONS</td>
<td>000090</td>
<td>EH HENDERSON</td>
</tr>
<tr>
<td>08</td>
<td>E03</td>
<td>SOFTWARE SUPPORT</td>
<td>000100</td>
<td>JQ SPENSER</td>
</tr>
<tr>
<td>09</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000110</td>
<td>VLUCCHESI</td>
</tr>
<tr>
<td>10</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000120</td>
<td>S O'CONNELL</td>
</tr>
<tr>
<td>11</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>000130</td>
<td>DM QUINTANA</td>
</tr>
</tbody>
</table>

PFK: 02=RESEND 03=END 06=NEXT

EMPLOYEE ADD

MAJOR SYSTEM ......: O
ACTION .............: A
OBJECT .............: EM
SEARCH CRITERIA ...: EN
DATA ..............: 000099

<table>
<thead>
<tr>
<th>EMPLOYEE ID</th>
<th>FIRST NAME</th>
<th>MIDDLE INITIAL</th>
<th>LAST NAME</th>
<th>WORK DEPT ID</th>
<th>PHONE NUMBER</th>
</tr>
</thead>
<tbody>
<tr>
<td>000099</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PFK: 02=RESEND 03=END 06=NEXT

ADR - Online program screens

6-33
MAJOR SYSTEM ......: O
ACTION ..............: E
OBJECT ..............: EM
SEARCH CRITERIA ....: EN
DATA ...............: 000030

EMPLOYEE ID: 000030
FIRST NAME: JANE
MIDDLE INITIAL: E
LAST NAME: DOE
WORK DEPT ID: E21
PHONE NUMBER: 0000

PFK: 02=RESEND 03=END

MAJOR SYSTEM ......: O
ACTION ..............: U
OBJECT ..............: EM
SEARCH CRITERIA ....: EN
DATA ...............: 000030

EMPLOYEE ID: 000030
FIRST NAME: JANE
MIDDLE INITIAL: E
LAST NAME: DOE
WORK DEPT ID: E21
PHONE NUMBER: 0000

ADR - Online program screens
<table>
<thead>
<tr>
<th>SEQ</th>
<th>LEVEL</th>
<th>FIELD NAME</th>
<th>T</th>
<th>CH/DG</th>
<th>OCCUR</th>
<th>K</th>
<th>VALUE/REDEF/DEP ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>MS-DEPARTMENT-U</td>
<td>X</td>
<td>3</td>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>ADMDEPT</td>
<td>X</td>
<td>36</td>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>DEPTNO</td>
<td>X</td>
<td>3</td>
<td>K</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>MGRNO</td>
<td>X</td>
<td>6</td>
<td>K</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Legend:**
- **SEQ**: Sequence Number
- **T**: Field Type: X=Alphanumeric, N=Numeric, U=Unsigned, C=Co-Named, V=Variable, D=Date
- **I**: Internal Numeric Type: P=Packed, Z=Zoned, B=Binary
- **CH/DG**: Characters/Digits: Number of Characters, Integers, or Int-Dec
- **K**: Key Usage: WH=Whole Key, PR=Partial Key (High Order Position)
- **REDEF**: Redefinition, DEP ON=Dependent
### DATA VIEW: MS-EMPLOYEE-2-U

<table>
<thead>
<tr>
<th>SEQ LEVEL</th>
<th>FIELD NAME</th>
<th>T</th>
<th>CH/DG</th>
<th>OCCUR</th>
<th>VALUE/REDEF/DEP ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MS-EMPLOYEE-2-U</td>
<td>I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>WORKDEPT</td>
<td>X</td>
<td>3</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>EMPNO</td>
<td>X</td>
<td>6</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>FULLNAME</td>
<td>X</td>
<td>3</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>LASTNAME</td>
<td>X</td>
<td>12</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>FIRSTNAME</td>
<td>X</td>
<td>13</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MIDINIT</td>
<td>X</td>
<td>1</td>
<td>K</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>PHONENO</td>
<td>X</td>
<td>4</td>
<td>K</td>
<td></td>
</tr>
</tbody>
</table>

#### LEGEND:
- **SEQ**: SEQUENCE NUMBER
- **T**: FIELD TYPE: **X**=ALPHANUMERIC, **M**=NUMERIC, **U**=UNSIGNED, **C**=COMMON, **N**=NONE
- **V**: VARIABLE **D**=DATE
- **I**: INTERNAL NUMERIC TYPE: **P**=PACKED, **Z**=ZONED, **B**=BINARY
- **CH/DG**: CHARACTERS/DIGITS: NUMBER OF CHARACTERS, INTERGERS, OR INTEGER
- **K**: KEY USAGE: **K**=WHOLE KEY, **P**=PARTIAL KEY (HIGH ORDER POSITION)
- **REDEF**: REDEFINITION, **DEP ON**: DEPENDING ON

---

6-38
SELECTING AN EMPLOYEE TO DISPLAY

MAJOR SYSTEM

ACTION

OBJECT

SEARCH CRITERIA

DATA

PFK: 02=RESEND 03=END
<table>
<thead>
<tr>
<th>PANEL7</th>
<th>FA9201I</th>
<th>VERSION: 001 STATUS: TEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>LAYOUT:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>MAJOR SYSTEM</td>
<td>=1</td>
</tr>
<tr>
<td>2</td>
<td>ACTION</td>
<td>=1</td>
</tr>
<tr>
<td>3</td>
<td>OBJECT</td>
<td>=1</td>
</tr>
<tr>
<td>4</td>
<td>SEARCH CRITERIA</td>
<td>=1</td>
</tr>
<tr>
<td>5</td>
<td>DATA</td>
<td>=1</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SELECTING AN EMPLOYEE TO DISPLAY:**

1. MAJOR SYSTEM
2. ACTION
3. OBJECT
4. SEARCH CRITERIA
5. DATA

**PFKZ: OZ=RESEND O3=END**

**6-41**
### FIELD SUMMARY TABLE

<table>
<thead>
<tr>
<th>SEQ</th>
<th>LV</th>
<th>FIELD NAME</th>
<th>ATTR</th>
<th>T LEN</th>
<th>IN,OP</th>
<th>OCC</th>
<th>COMMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>MAJSYS</td>
<td>PSL</td>
<td>X</td>
<td>12</td>
<td>1</td>
<td>SELECTING AN EMPLOYEE</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>MAJSYS-MSG</td>
<td>PSL</td>
<td>X</td>
<td>19</td>
<td>1</td>
<td>MAJOR SYSTEM ........</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>ACTION</td>
<td>VALE</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>ACTION .............</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>ACTION-MSG</td>
<td>PSL</td>
<td>X</td>
<td>52</td>
<td>1</td>
<td>OBJECT ..............</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>OBJECT</td>
<td>PSL</td>
<td>X</td>
<td>52</td>
<td>1</td>
<td>SEARCH CRITERIA....</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>SEARCH</td>
<td>PSL</td>
<td>X</td>
<td>52</td>
<td>1</td>
<td>DATA ...............</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>MGLINE</td>
<td>PSL</td>
<td>X</td>
<td>78</td>
<td>1</td>
<td>SEARCH-CRITERIA....</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>COLUMN-HEADS</td>
<td>PSH</td>
<td>X</td>
<td>73</td>
<td>1</td>
<td>USER-INPUT ..........</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>POL-HEADER</td>
<td>PSL</td>
<td>X</td>
<td>55</td>
<td>1</td>
<td>USER-INPUT ..........</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>POL-INPUT</td>
<td>PSH</td>
<td>X</td>
<td>39</td>
<td>1</td>
<td>USER-INPUT ..........</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>PFK-MSG</td>
<td>PSL</td>
<td>X</td>
<td>21</td>
<td>1</td>
<td>PFK: 02=RESEND 03=END</td>
</tr>
</tbody>
</table>

### LEGEND:

- SEQ=SEQUENCE NUMBER
- ATTR=SCREEN ATTRIBUTES: U=UNPROT O=OUTER L=LOW-LIGHT
- T=FIELD TYPE: A=ALPHANUMERIC N=NUMERIC
- IN,OP=INTEGER-PLACES,DECIMAL-PLACES
- OCC=NUMBER OF OCCURRENCES

### PANEL PARAMETERS:

- START FIELD SYMBOL
- END FIELD SYMBOL
- REPEATER GROUP SYMBOL
- INPUT FILL CHARACTER
- OUTPUT FILL CHARACTER
- NON-DISPLAY CHARACTER
- ERROR FILL CHARACTER
- CASE TRANSLATION
- REQUIRED
  - Y=YES, N=NO
PANEL: FA9201I  VERSION: D01  STATUS: TEST

PANEL PARAMETERS:

ERROR HANDLING B (N=None, F=Fill w/Error Fill, H=High Intensity)

PF1=HELP, PF3=CLARIFY Y (Y=Yes, N=No)
PFT=SCR-, PF9=SCR+ N (Y=Yes, N=No)
PFT0=SCR Top, PFT1=SCR Bot

EDIT-RULE ERROR PROC C (C=CLARIFY COMMAND, A=APPLICATION)

HELP PANEL NAME VERSION
PREFIX PANEL NAME VERSION
SUFFIX PANEL NAME VERSION

6-43
## PANEL: PFA9201I
### VERSION: 001
### STATUS: TEST
#### INPUT RULES:

| SEQ FIELD NAME | E | R MINIMUM VALUE | MAXIMUM VALUE | J | I | C | M | N | D | N | A | C | M | S | F | S | D | S | O | S | C | F |
| 1              | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 2              | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 3              | NAJSYS | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 4              | NAJSYS-NSG | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 5              | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 6              | ACTION | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 7              | ACTION-NSG | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 8              | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 9              | OBJECT | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 10             | OBJECT-NSG | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 11             | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 12             | SEARCH | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 13             | SEARCH-NSG | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 14             | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 15             | USER-INPUT | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 16             | NSGLINE | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 17             | COLUMN-NEADS | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 18             | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 19             | PDL-HEADER | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 20             | PDL-INPUT | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 21             | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
| 22             | PFK-NSG | B | N               |               |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |

### LEGEND:
- **SEQ**: SEQUENCE NUMBER
- **E/N**: ERROR HANDLING: **N**=NONE, **@**=FILL WITH @, **H**=HIGH INTENSITY
- **U/B**: BOTH (U IF ILLEGAL VALUE, @ IF REQUIRED FIELD MISSING)
- **R/Q**: REQUIRED FIELD: **Y**=YES, **N**=NO, **C**=CONDITIONAL
- **J/S**: JUSTIFICATION: **N**=NONE, **L**=LEFT, **R**=RIGHT, **A**=ALIGN BY DECIMAL POINT
- **F**: INPUT FILL CHARACTER: **S**=SPACES, **L**=LOW-VALUES, **Z**=ZEROS, **U**=UNDERSCORE, **O**=OTHER=ITSELF
- **C/S**: CASES: **U**=UPPER CASE, **N**=NAVY CASE
- **M/D**: MINIMUM REQUIRED DECIMALS (FOR JUSTIFICATION=H)
- **O/S**: ALLOW DIGIT SEPARATOR: **T**=YES, **N**=NO
- **M/H**: MUST MINUS SIGN: **Y**=YES, **N**=NO
- **A/C**: ALLOW CURRENCY SYMBOL: **Y**=YES, **N**=NO
- **M/C**: CHECK DIGIT: **N**=NONE, **T**=MODULO 10, **E**=MODULO 11
- **M/F**: MUST FILL: **Y**=YES, **N**=NO
<table>
<thead>
<tr>
<th>SEQ FIELD NAME</th>
<th>EDIT PATTERN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U</td>
</tr>
<tr>
<td>2</td>
<td>N</td>
</tr>
<tr>
<td>3 MAJSYS</td>
<td>S</td>
</tr>
<tr>
<td>4 MAJSYS-MSG</td>
<td>N</td>
</tr>
<tr>
<td>5</td>
<td>U</td>
</tr>
<tr>
<td>6 ACTION</td>
<td>S</td>
</tr>
<tr>
<td>7 ACTION-MSG</td>
<td>N</td>
</tr>
<tr>
<td>8</td>
<td>U</td>
</tr>
<tr>
<td>9 OBJECT</td>
<td>S</td>
</tr>
<tr>
<td>10 OBJECT-MSG</td>
<td>N</td>
</tr>
<tr>
<td>11 SEARCH</td>
<td>S</td>
</tr>
<tr>
<td>12 SEARCH-MSG</td>
<td>N</td>
</tr>
<tr>
<td>13 USER-INPUT</td>
<td>S</td>
</tr>
<tr>
<td>14 MSGLINE</td>
<td>N</td>
</tr>
<tr>
<td>15 COLUMN-HEADS</td>
<td>S</td>
</tr>
<tr>
<td>16 PDL-HEADER</td>
<td>S</td>
</tr>
<tr>
<td>17 PDL-INPUT</td>
<td>U</td>
</tr>
<tr>
<td>18 PFK-MSG</td>
<td>S</td>
</tr>
</tbody>
</table>

**Legend:**
- SEQ=SEQUENCE NUMBER
- D/F=OUTPUT FILL CHARACTER: S=SPACES, L=LOW-VALUES, Z=ZEROES, U=UNDERSCORE, OTHER=ITSELF
- C=COLORS: N=NEUTRAL, B=BLUE, R=RED, P=PINK, G=GREEN
- T=TURQUOISE, Y=YELLOW, W=WHITE/BLACK
- E/H=EXTENDED HIGHLIGHTING: N=NONE, BL=BLINK, R=REVERSE VIDEO, U=UNDERSCORE
### Working Data: CCF92CI

<table>
<thead>
<tr>
<th>SEQ</th>
<th>LEVEL</th>
<th>FIELD NAME</th>
<th>T</th>
<th>CH/OG</th>
<th>OCCUR</th>
<th>VAL/COMMENT</th>
<th>REDEF</th>
<th>DEP ON/COPY</th>
</tr>
</thead>
<tbody>
<tr>
<td>000101</td>
<td>1</td>
<td>WK-FNAME-BYTE-I</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000200</td>
<td>1</td>
<td>WK-MOLO-INPUT</td>
<td>X</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000201</td>
<td>1</td>
<td>WK-MOLO-NAME</td>
<td>X</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000202</td>
<td>1</td>
<td>WK-MOLDC-KEY</td>
<td>X</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000203</td>
<td>1</td>
<td>WK-READ-KEY</td>
<td>X</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000204</td>
<td>1</td>
<td>WK-CTR</td>
<td>N</td>
<td>4</td>
<td></td>
<td>VALUE 1-12 COUNTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000205</td>
<td>1</td>
<td>WK-PNL-CTR</td>
<td>N</td>
<td>4</td>
<td></td>
<td>NUMERIC PNL INDEX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000206</td>
<td>1</td>
<td>WK-RESEND-CTR</td>
<td>N</td>
<td>2</td>
<td></td>
<td>RESEND COUNTER</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000207</td>
<td>1</td>
<td>WK-VALIO-INPUT</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000208</td>
<td>1</td>
<td>WK-REC-DISP</td>
<td>X</td>
<td>1</td>
<td></td>
<td>F</td>
<td></td>
<td></td>
</tr>
<tr>
<td>000209</td>
<td>1</td>
<td>WK-GOOD-EDIT</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000210</td>
<td>1</td>
<td>WK-GOOD-COMPARE</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000211</td>
<td>1</td>
<td>WK-SAVE-EMPNO</td>
<td>X</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000212</td>
<td>1</td>
<td>WK-SAVE-LNAME</td>
<td>X</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000213</td>
<td>1</td>
<td>WK-SAVE-FNAME</td>
<td>X</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000214</td>
<td>1</td>
<td>WK-SAVE-MIDINIT</td>
<td>X</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000215</td>
<td>1</td>
<td>WK-SAVE-PMONENO</td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000216</td>
<td>1</td>
<td>WK-SAVE-WKDEPT</td>
<td>X</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000400</td>
<td>1</td>
<td>WK-DETAIL-LINE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000500</td>
<td>2</td>
<td>WK-OL-NDX-ALPN</td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000600</td>
<td>2</td>
<td>WK-FILLER-1</td>
<td>X</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000700</td>
<td>2</td>
<td>WK-OL-WORKOEP</td>
<td>X</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000800</td>
<td>2</td>
<td>WK-FILLER-2</td>
<td>X</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>000900</td>
<td>2</td>
<td>WK-OL-DEPTNAME</td>
<td>X</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001000</td>
<td>2</td>
<td>WK-FILLER-3</td>
<td>X</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001100</td>
<td>2</td>
<td>WK-FILLER-4</td>
<td>X</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001200</td>
<td>2</td>
<td>WK-FILLER-5</td>
<td>X</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>001300</td>
<td>2</td>
<td>WK-OL-FULL-NAME</td>
<td>X</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Legend:
- T (FIELD TYPE): X=ALPHANUMERIC, N=NUMERIC, U=UNIGNED, C=CON. NAME, F=FLAG, V=VARIABLE, D=DATE
- I (INTERNAL NUMERIC TYPE): B=UNSIGNED, P=PACKEO, Z-ZONE, B=BINARY
- CH/OG (CHARACTERS/DIGITS): NUMER OF CHARACTERS, INTEGRERS, OR NUMERICs/DECIMALS
- REDEF= REDEFINITION, DEP ON= DEPENDING ON

---

6-46
PROCEDURE   CCFA92CI

SEQ STATEMENT
100 :  
200 <<MAIN>> PROCEDURE
220  
300  
400  
500  
600  
700  
800  
900  
1000 LOOP
1100 
1200 DO EDIT-4-LINES-IN
1300 IF WK-VALID-INPUT = +T+
1400 SET ATTRIBUTE *P* ON FA9201I.ACTION
1500 SET ATTRIBUTE *P* ON FA9201I.OBJECT
1600 
1700 
1800 MOVE *SPF=NEXT* TO PFK-MSG
2000 MOVE *STRING("SELECT AN EMPLOYEE FROM FOLLOWING LIST BY *,
2100   *SPECIFYING THE LINE NUMBER") TO FA9201I.MSGLINE
2200 SET ATTRIBUTE *HP TEMP ON FA9201I.MSGLINE
2300 
2400 SELECT FIRST ACTION
2500 WHEN SPF2
2600 OD RESEND-PNL
2700 WHEN SPF8
2800 OD DISPLAY-INDEX
2900 WHEN OTHER
3000 SET WK-READ-KEY = FA9201I.USER-INPUT
3100 OD DISPLAY-INDEX
3200 ENDSEL
3300 ELSE
3400 OD RESEND-PNL
3500 ENDIF
3600 
3700 UNTIL SPF3
3800 
3900 ENDOLOOP
4000 
4100 ENPROC <<MAIN
4200 
4300 
4400 <<EDIT-4-LINES-IN>> PROCEDURE
4500 MOVE +T+ TO WK-VALID-INPUT
4600 
4700 "VALIDATE USER *SEARCH CRITERIA* INPUT

6-47
PROCEDURE CCFA92CI VERSION: 001 STATUS: TEST

4800 IF FA9201I.SEARCH = 'EI'
4900 SET FA9201I.SEARCH-HS& = EMPLOYEE ID'
5000 ELSE
5100 SET FA9201I.SEARCH-HS& = 'INVALID SEARCH CRITERIA, USE 'EI''
5200 SET ATTRIBUTE 'CH' TEMP ON FA9201I.SEARCH
5300 MOVE 'F' TO WK-VALID-INPUT
5400 ENOIF

5500 IFFA9201I.OBJECT = 'EM'
5600 SET FA9201I.OBJECT-HS& = 'EMPLOYEE'
5700 ELSE
5800 SET FA9201I.OBJECT-HS& = 'INVALID OBJECT, USE 'EM''
5900 SET ATTRIBUTE 'CH' TEMP ON FA9201I.OBJECT
6000 MOVE 'F' TO WK-VALID-INPUT
6100 ENOIF

6200 WHEN FA9201I.ACTION = 'A'
6300 DO ADD-REC
6400 WHEN FA9201I.ACTION = 'D'
6500 DO DISPLAY-INDEX
6600 WHEN FA9201I.ACTION = 'E'
6700 DO ERASE-REC
6800 WHEN FA9201I.ACTION = 'U'
6900 DO UPDATE-REC
7000 WHEN OTHER
7100 SET FA9201I.MSGLINE = 'SELECT AN ACTION FROM FOLLOWING LIST''
7200 SET FA9201I.POL-HEADER(1) = ' A ADD (INSERT)'
7300 SET FA9201I.POL-HEADER(2) = ' O DISPLAY (SHOW)'
7400 SET FA9201I.POL-HEADER(3) = ' E ERASE (REMOVE)'
7500 SET FA9201I.POL-HEADER(4) = ' U UPDATE (CHANGE)'
7600 SET ATTRIBUTE 'CH' TEMP ON FA9201I.ACTION
7700 SET WK-VALID-INPUT = 'F'
7800 ENOSEL

7900 WHEN MAJORSYS = '0'
8000 SET FA9201I.MAJSYS-HS& = 'ORGANIZATION'
8100 ELSE
8200 SET FA9201I.MAJSYS-HS& = 'INVALID MAJOR SYSTEM, USE '0''
8300 SET ATTRIBUTE 'CH' TEMP ON FA9201I.MAJSYS
8400 MOVE 'F' TO WK-VALID-INPUT
8500 ENOIF
8600 ENOPROC
PROCEDURE:  CCFA92CI  

PROEDURE: CCFA92CI VERSION: 001 STATUS: TEST

360 STATEMENT

14300 SET MSGLINE = "CORRECT HIGHLIGHTED FIELDS"
14400 ENDIF
14500 WHEN DUPLICATE
14600 SET MSGLINE = "EMPLOYEE ID ALREADY EXISTS ON FILE"
14700 ENDFOR
14800 ENOPROC IADD-REC

15000 :--
15100 : UPDATE RECORD
15200 <--UPDATE-REC--> PROCEDURE
15400 IF WK-REC-DISP = 'F'
15500 FOR THE FIRST NS-EMPLOYEE-2-U NO UPDATE
15600 WHERE MS-EMPLOYEE-2-U.EMPNO = WK-READ-KEY
15700 15800 SET FA9201L.PDL-HEADER(1) = *
15900 STRING('EMPLOYEE ID') : *NS-EMPLOYEE-2-U.EMPNO)
16000 SET FA9201L.PDL-HEADER(2) = *
16100 STRING('FIRST NAME') : *NS-EMPLOYEE-2-U.FIRSTNAME)
16200 SET FA9201L.PDL-HEADER(3) = *
16300 STRING('MIDDLE INIT') : *NS-EMPLOYEE-2-U.MIDINIT)
16400 SET FA9201L.PDL-HEADER(4) = *
16500 STRING('LAST NAME') : *NS-EMPLOYEE-2-U.LASTNAME)
16600 SET FA9201L.PDL-HEADER(5) = *
16700 STRING('PHONE NUMBER') : *NS-EMPLOYEE-2-U.PHONENO)
16800 SET FA9201L.PDL-HEADER(6) = *
16900 STRING('WORK DEPT') : *NS-EMPLOYEE-2-U.WORKDEPT)
17000 SET WK-SAVE-EMPNO = NS-EMPLOYEE-2-U.EMPNO
17100 SET WK-SAVE-FNAME = NS-EMPLOYEE-2-U.FIRSTNAME
17200 SET WK-SAVE-MIDINIT = NS-EMPLOYEE-2-U.MIDINIT
17300 SET WK-SAVE-LNAME = NS-EMPLOYEE-2-U.LASTNAME
17400 SET WK-SAVE-PHONENO = NS-EMPLOYEE-2-U.PHONENO
17500 SET WK-SAVE-WKDEPT = NS-EMPLOYEE-2-U.WORKDEPT
17600 WHEN NONE
17700 SET FA9201L.MSGLINE = "EMPLOYEE ID NOT FOUND FOR UPDATE"
17800 SET WK-REC-DISP = 'F'
17900 ENDFOR
18000 ELSE
18200 : RECORD FOUND AND DISPLAYED FOR USER
18300 SET WK-GOOD-EDIT = 'F'
18400 DO EDIT-INPUT
18500 IF WK-GOOD-EDIT = 'T'
18600 SET WK-GOOD-COMPARE = 'T'
18700 DO COMPARE-RECS
18800 ELSE
18900 SET FA9201L.MSGLINE = "$STRING('CORRECT HIGHLIGHTED FIELDS')"
PROCEDURE: CCFA92CI

VERSION: 001

STATEMENT

16000 IF FA9201I.POL-INPUT(1) > 5SPACES
16100 SET MSGLINE = 5SPACES
16150 ELSE
16200 SET MSGLINE = *LAST NAME NOT SUPPLIED*
16250 SET WK-GOOD-EDIT = *F*
16300 ENOIF
16350 IF FA9201I.POL-INPUT(5) > 5SPACES
16400 IF NUMERIC(FA9201I.POL-INPUT(5))
16450 SET MSGLINE = 5SPACES
16500 ELSE
16550 SET MSGLINE = *PHONE NUMBER NOT NUMERIC*
16600 SET WK-GOOD-COMPARE = *F*
16650 ELSE
16700 SET WK-GOOD-COMPARE = *F*
16750 ENOIF
16800 ENOPROC :EDIT-INPUT
16850 :
16900 /*COMPARE-RECS*/ PROCEDURE
16950 FOR THE FIRST NS-EMPLOYEE-2-U WHERE NS-EMPLOYEE-2-U.EMPNO = WK-READ-KEY
17000 IF WK-SAVE-EMPNO = FA9201I.POL-INPUT(1)
17050 := NULL
17100 ELSE
17150 SET WK-GOOD-COMPARE = *F*
17200 IF WK-SAVE-FNAME = FA9201I.POL-INPUT(2)
17250 := NULL
17300 ELSE
17350 SET WK-GOOD-COMPARE = *F*
17400 ENOIF

6-51
PROCEDURE: CCFA92CI

VERSION: 001

STATEMENT

22700 IF WK-SAVE-MIOINIT = FA92011,POL-INPUT(3)
22750 THEN
22800 NULL
22850 ELSE
22900 SET WK-GOOD-CMPARE = 'F'
22950 ENDIF
23000 IF WK-SAVE-LNAME = FA92011,POL-INPUT(4)
23050 THEN
23100 NULL
23150 ELSE
23200 SET WK-GOOD-CMPARE = 'F'
23250 ENDIF
23300 IF WK-SAVE-PHONENO = FA92011,POL-INPUT(5)
23350 THEN
23400 NULL
23450 ELSE
23500 SET WK-GOOD-CMPARE = 'F'
23550 ENDIF
23600 IF WK-SAVE-WKDEPT = FA92011,POL-INPUT(6)
23650 THEN
23700 NULL
23750 ELSE
23800 SET WK-GOOD-CMPARE = 'F'
23850 ENDIF
23900 IF WK-GOOD-CMPARE = 'T'
23950 THEN
24000 ENDIF
24050 ELSE
24100 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(1)
24150 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(2)
24200 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(3)
24250 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(4)
24300 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(5)
24350 SET ATTRIBUTE 'P' TEMP ON FA92011,POL-INPUT(6)
24400 SET FA92011,MSGLINE = "NO UPDATE: DATA CHANGED SINCE UPDATE BEGAN; CONTACT DP"
24450 QUIT COMPARE-LOOP
24500 ENDIF
24550 WHEN NONE
24600 SET MSGLINE = *EMPLOYEE IO NOT FOUND FOR UPDATE*
24650 ENDIF
24700 QUIT COMPARE-RECS
24750 "CHECK-RECS"
24800 IF WK-SAVE-WKDEPT = FA92011,COLUMN-HEADS
24850 THEN
24900 SET ATTRIBUTE 'IP' TEMP ON FA92011,MSGLINE
24950 ENDIF
### PROCEDURE: CCFA92CI

<table>
<thead>
<tr>
<th>SEQ</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>2400</td>
<td>SET ATTRIBUTE 'TP' TEMP ON FA92011.PFK-MSG</td>
</tr>
<tr>
<td>2430</td>
<td>SET ATTRIBUTE 'UAL' ON FA92011.MAJSYS</td>
</tr>
<tr>
<td>2460</td>
<td>SET ATTRIBUTE 'UAL' ON FA92011.ACTION</td>
</tr>
<tr>
<td>2470</td>
<td>SET ATTRIBUTE 'UAL' ON FA92011.OBJECT</td>
</tr>
<tr>
<td>2480</td>
<td>SET ATTRIBUTE 'UAL' ON FA92011.SEARCH</td>
</tr>
<tr>
<td>28800</td>
<td>SET ATTRIBUTE 'UAL' ON FA92011.OBJECT</td>
</tr>
<tr>
<td>29000</td>
<td>MOVE $SPACES TO FA92011.ACTION</td>
</tr>
<tr>
<td>29500</td>
<td>LOOP VARYING WK-RESEND-CTR FROM 1 BY 1 UP THRU 12</td>
</tr>
<tr>
<td>29600</td>
<td>MOVE $SPACES TO FA92011.OBJECT</td>
</tr>
<tr>
<td>29700</td>
<td>ENDLOOP</td>
</tr>
<tr>
<td>29900</td>
<td>ENDPROC</td>
</tr>
</tbody>
</table>

---

### DISPLAY THE EMPLOYEE/OEDEPARTMENT INDEX

<table>
<thead>
<tr>
<th>SEQ</th>
<th>STATEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>30700</td>
<td>SET WK-PNL-CTR = 0</td>
</tr>
<tr>
<td>30800</td>
<td>FOR THE FIRST 12 MS-EMPLOYEE-2-U NO UPDATE</td>
</tr>
<tr>
<td>31000</td>
<td>WHERE MS-EMPLOYEE-2-U.EMPNO &gt;= WK-READ-KEY</td>
</tr>
<tr>
<td>31100</td>
<td>ORDERED BY MS-EMPLOYEE-2-U.EMPNO</td>
</tr>
<tr>
<td>31200</td>
<td>SET WK-CTR = WK-CTR + 1 VALUES 1 TO 12</td>
</tr>
<tr>
<td>31300</td>
<td>SET WK-PNL-CTR = WK-PNL-CTR + 1 VALUES 1 TO #EMP</td>
</tr>
<tr>
<td>31400</td>
<td>SET WK-DL-NOX-ALPM = WK-PNL-CTR</td>
</tr>
<tr>
<td>31500</td>
<td>SET WK-DL-NOX-ALPM = TRANSLATE(WK-DL-NOX-ALPM, FROM= ' ', TO= '0')</td>
</tr>
<tr>
<td>31700</td>
<td>SET WK-DL-WORKDEPT = MS-EMPLOYEE-2-U.WORKDEPT</td>
</tr>
<tr>
<td>31800</td>
<td>SET WK-DL-EMPNO = MS-EMPLOYEE-2-U.EMPNO</td>
</tr>
<tr>
<td>31900</td>
<td>SET WK-FNAME-BYTE-1 = SUBSTR(MS-EMPLOYEE-2-U.LASTNAME, START=1, LENGTH=1)</td>
</tr>
<tr>
<td>32100</td>
<td>SET WK-DL-FULL-NAME = $STRINGS(WK-FNAME-BYTE-1, MIDINIT=' ', FIRSTNAME)</td>
</tr>
<tr>
<td>32200</td>
<td>FOR FIRST MS-DEPARTMENT-U NO UPDATE</td>
</tr>
<tr>
<td>32400</td>
<td>WHERE MS-DEPARTMENT-U.WORKDEPT = MS-EMPLOYEE-2-U.WORKDEPT</td>
</tr>
<tr>
<td>32600</td>
<td>SET WK-DL-OEPTNAME = MS-DEPARTMENT-U.OEPTNAME</td>
</tr>
<tr>
<td>32700</td>
<td>WHEN NONE</td>
</tr>
<tr>
<td>32800</td>
<td>WHEN NULL</td>
</tr>
<tr>
<td>32900</td>
<td>ENDFOR</td>
</tr>
<tr>
<td>33000</td>
<td>6-53</td>
</tr>
</tbody>
</table>
PROCEDURE: CCFA92CI

SEQ STATEMENT

33100 MOVE MS-EMPLOYEE-2-U-EMPNO TO WK-READ-KEY
33200 MOVE WK-DETAIL-LINE TO FA92D11-PDL-HEADER(WK-CTR)
33300
33400 SET WK-HOLD-NAME = WK-OL-FULL-NAME
33500 WHEN NONE
33600 SET FA92D11.MSGLINE = 'END OF FILE'
33700 ENDFOR
33800 ENDP:DISPLAY-INDEX
33900
34000 :MOVE THE COLUMN HEADINGS TO THE REPORT
34100 :<LDAD-COL-HEADS> PROCEDURE
34200 :MOVE STRING('ID' "ID/ID", 'DEPARTMENT NAME' "E/ID", 'EMPLOYEE NAME')
34300 ID FA92D11.COLUMN-HEADS
34400 ENDP:DISPLAY-INDEX
34500
34600 :<ERROR> PROCEDURE
34700 CALL CCFA999 USING INPUT "ID"
34800 QUIT RUN
34900 ENDP:DISPLAY-INDEX :ERROR
31500 CMSNP24-1 4K-PNL-CTR, A NUMERIC FIELD OR LITERAL, MAY BE TRUNCATED IN MOVE TO AN ALPHANUMERIC FIELD
PROGRAM HAS BEEN SUCCESSFULLY COMPILED

NO ERROR MESSAGE(S) FLAGGED IN THIS COMPILATION
NO WARNING MESSAGE(S) FLAGGED IN THIS COMPILATION
1 ADVISORY MESSAGE(S) FLAGGED IN THIS COMPILATION

COMPILED PROGRAM STATISTICS:

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of shareable members of the program</td>
<td>001</td>
</tr>
<tr>
<td>Length of shareable portions of the program</td>
<td>8,996 bytes</td>
</tr>
<tr>
<td>Number of non-shareable members of the program</td>
<td>001</td>
</tr>
<tr>
<td>Length of non-shareable portions of the program</td>
<td>2,540 bytes</td>
</tr>
</tbody>
</table>
Appendix B

DB2 Programs and Output
<table>
<thead>
<tr>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>LAST NAME</th>
<th>PHONE NUMBER</th>
<th>EMPLOYEE NUMBER</th>
<th>DEPT</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHRISTINE</td>
<td>C</td>
<td>AAS</td>
<td>3978</td>
<td>000010</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
</tr>
<tr>
<td>VINCENZO</td>
<td>C</td>
<td>UCCHESI</td>
<td>3490</td>
<td>000110</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE OLY.</td>
</tr>
<tr>
<td>SEAN</td>
<td>I</td>
<td>CONNELL</td>
<td>2167</td>
<td>000120</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
</tr>
<tr>
<td>MICHAEL</td>
<td>L</td>
<td>HEPSON</td>
<td>3476</td>
<td>000020</td>
<td>E01</td>
<td>PLANNING</td>
</tr>
<tr>
<td>SALLY</td>
<td>A</td>
<td>WAN</td>
<td>6718</td>
<td>000030</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
</tr>
<tr>
<td>DOLORES</td>
<td>A</td>
<td>VINTANA</td>
<td>4578</td>
<td>000130</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
</tr>
<tr>
<td>HEATHER</td>
<td>A</td>
<td>ICHOLLS</td>
<td>1793</td>
<td>000140</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
</tr>
<tr>
<td>IRYING</td>
<td>F</td>
<td>TERN</td>
<td>6423</td>
<td>000060</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>BRUCE</td>
<td></td>
<td>OAKSON</td>
<td>4510</td>
<td>000150</td>
<td>D12</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>ELIZABETH</td>
<td>R</td>
<td>SANKA</td>
<td>3782</td>
<td>000160</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>MASATOMI</td>
<td>J</td>
<td>OSHIMURA</td>
<td>2890</td>
<td>000170</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>MARILYN</td>
<td>S</td>
<td>COURON</td>
<td>1682</td>
<td>000180</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>JAMES</td>
<td>N</td>
<td>ALKER</td>
<td>2966</td>
<td>000190</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>DAVID</td>
<td>T</td>
<td>ROWN</td>
<td>4501</td>
<td>000200</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>WILLIAM</td>
<td>T</td>
<td>OINES</td>
<td>0942</td>
<td>000210</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>JENNIFER</td>
<td>K</td>
<td>UTLER</td>
<td>0672</td>
<td>000220</td>
<td>D11</td>
<td>MANUFACTURING SYSTEMS</td>
</tr>
<tr>
<td>EYA</td>
<td>D</td>
<td>ULASKI</td>
<td>7831</td>
<td>000070</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>JAMES</td>
<td>J</td>
<td>EFFERON</td>
<td>4265</td>
<td>000230</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>SALVATORE</td>
<td>M</td>
<td>ARINO</td>
<td>3780</td>
<td>000240</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>DANIEL</td>
<td>S</td>
<td>MITH</td>
<td>0961</td>
<td>000250</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>SYBIL</td>
<td>V</td>
<td>DUNSON</td>
<td>8953</td>
<td>000260</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>MARIA</td>
<td>L</td>
<td>EREZ</td>
<td>9001</td>
<td>000270</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
</tr>
<tr>
<td>JOHN</td>
<td>B</td>
<td>EYER</td>
<td>6789</td>
<td>000050</td>
<td>E01</td>
<td>SUPPORT SERVICES</td>
</tr>
<tr>
<td>EILEEN</td>
<td>W</td>
<td>ENDERSON</td>
<td>2498</td>
<td>000090</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>ETHEL</td>
<td>R</td>
<td>CHNEIDER</td>
<td>8997</td>
<td>000280</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>JOHN</td>
<td>R</td>
<td>ARK</td>
<td>4502</td>
<td>000290</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>PHILIP</td>
<td>X</td>
<td>MITH</td>
<td>2095</td>
<td>000300</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>MAUDE</td>
<td>F</td>
<td>ETRIGHT</td>
<td>3332</td>
<td>000310</td>
<td>E11</td>
<td>OPERATIONS</td>
</tr>
<tr>
<td>THEODORE</td>
<td>G</td>
<td>PENNER</td>
<td>0972</td>
<td>000100</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>RAMRL</td>
<td>W</td>
<td>EMT</td>
<td>9990</td>
<td>000320</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
</tr>
</tbody>
</table>

DB2 - Batch Program Output
<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Code</th>
<th>Name</th>
<th>Code</th>
<th>Name</th>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>OE</td>
<td>JASON</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
<td>OE</td>
<td>TEST</td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>500031 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----</td>
<td>------</td>
<td>--------</td>
<td>----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300032 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>500033 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300034 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>500035 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300036 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300037 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300038 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300039 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300040 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300041 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300042 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300043 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300044 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300045 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300046 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300047 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300048 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300049 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300050 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300051 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300052 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300053 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300054 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300055 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300056 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300057 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300058 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300059 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300060 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300061 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300062 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OE</td>
<td>TEST</td>
<td>E 0000</td>
<td>300063 E21 SOFTWARE SUPPORT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7-4
<p>| OE TEST | E 0000 300064 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300065 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300066 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300067 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300068 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300069 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300070 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300071 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300072 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300073 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300074 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300075 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300076 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300077 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300078 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300079 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300080 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300081 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300082 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300083 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300084 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300085 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300086 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300087 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300088 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300089 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300090 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300091 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300092 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300093 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300094 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300095 | E21 SOFTWARE SUPPORT |
| OE TEST | E 0000 300096 | E21 SOFTWARE SUPPORT |</p>
<table>
<thead>
<tr>
<th>OE TEST</th>
<th>E 0000 300064</th>
<th>E21 SOFTWARE SUPPORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>OE TEST</td>
<td>E 0000 300065</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300066</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300067</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300068</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300069</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300070</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300071</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300072</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300073</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300074</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300075</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300076</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300077</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300078</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300079</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300080</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300081</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300082</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300083</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300084</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300085</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300086</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300087</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300088</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300089</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300090</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300091</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300092</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300093</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300094</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300095</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>OE TEST</td>
<td>E 0000 300096</td>
<td>E21 SOFTWARE SUPPORT</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>DE</td>
<td>TEST</td>
<td>E</td>
</tr>
<tr>
<td>FIRST NAME</td>
<td>INITIAL</td>
<td>PHONE NUMBER</td>
</tr>
<tr>
<td>-----------</td>
<td>---------</td>
<td>--------------</td>
</tr>
<tr>
<td>WILLIAM</td>
<td>T</td>
<td>0942</td>
</tr>
<tr>
<td>SYBIL</td>
<td>V</td>
<td>3953</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>HOPKINSON</td>
<td>MICHAEL</td>
<td>L</td>
</tr>
<tr>
<td>DAMSON</td>
<td>BRUCE</td>
<td></td>
</tr>
<tr>
<td>EFFERSON</td>
<td>JAMES</td>
<td>J</td>
</tr>
<tr>
<td>ONNASON</td>
<td>SYBIL</td>
<td>V</td>
</tr>
<tr>
<td>ENDESON</td>
<td>EILEEN</td>
<td>W</td>
</tr>
<tr>
<td>AST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>MITH</td>
<td>DANIEL</td>
<td>S</td>
</tr>
<tr>
<td>MITH</td>
<td>PHILIP</td>
<td>X</td>
</tr>
<tr>
<td>ASI NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
</tr>
<tr>
<td>----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>SNH0801</td>
<td>DSNH83C7</td>
<td>NO</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>FIRST NAME</td>
<td>INITIAL</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>BROWN</td>
<td>DAVID</td>
<td></td>
</tr>
</tbody>
</table>

7-15
<table>
<thead>
<tr>
<th>ASI NAME</th>
<th>FIRST NAME</th>
<th>INITIAL</th>
<th>PHONE</th>
<th>EMPLOYEE WORK NUMBER</th>
<th>DEPT WORK DEPT NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNB0041</td>
<td>OSNABC3-EMPLOYEE</td>
<td>SUCCESSFULLY UPDATED</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7-16
**DATASET:** DSN120.DSNSSAMP

**MEMBER:** DSN8BC3

**START**

1 000100** DSN8BC3 - DB2 SAMPLE PHONE APPLICATION - COBOL - BATCH *** 00010000
1 000200* MODULE NAME = DSN8BC3
1 000300* DESCRIPTIVE NAME = DB2 SAMPLE APPLICATION
1 000400* PHONE APPLICATION
1 000500* BATCH
1 000600* COBOL
1 000700* COPYRIGHT = 5740-XYA ICI COPYRIGHT IBM CORP 1982, 1985
1 000800* REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083
1 000900* STATUS = RELEASE 2, LEVEL 0
1 001000* FUNCTION = THIS MODULE LISTS EMPLOYEE PHONE NUMBERS AND
1 001100* UPDATES THEM IF DESIRED.
1 001200* NOTES = NONE
1 001300* MODULE TYPE = COBOL PROGRAM
1 001400* PROCESSOR = DB2 PRECOMPILER, VS COBOL
1 001500* MODULE SIZE = SEE LINK EDIT
1 001600* ATTRIBUTES = NOT REENTRANT OR REUSABLE
1 001700* ENTRY POINT = DSN8BC3
1 001800* PURPOSE = SEE FUNCTION
1 001900* LINKAGE = INVOKED FROM DSM RUN
1 002000* INPUT =
1 002100* SYMBOLIC LABEL/NAME = CARDIN
1 002200* DESCRIPTION = INPUT REQUEST FILE
1 002300* SYMBOLIC LABEL/NAME = VPHONE
1 002400* DESCRIPTION = VIEW OF TELEPHONE INFORMATION
1 002500* OUTPUT =
1 002600* SYMBOLIC LABEL/NAME = REPORT
1 002700* DESCRIPTION = REPORT OF EMPLOYEE PHONE NUMBERS
1 002800* SYMBOLIC LABEL/NAME = VEMPLP
1 002900* DESCRIPTION = VIEW OF EMPLOYEE INFORMATION
1 003000* EXIT-NORMAL = RETURN CODE 0 NORMAL COMPLETION
1 003100* EXIT-ERROR =
1 003200* RETURN CODE = NONE

**DB2 - Batch Program**

7-17
START

1 005500*  ABEND CODES = NONE
1 005600*  ERROR-MESSAGES =
1 005700*  DSN8004I - EMPLOYEE SUCCESSFULLY UPDATED
1 005800*  DSN8007E - EMPLOYEE DOES NOT EXIST, UPDATE NOT DONE!
1 005900*  DSN8008I - NO EMPLOYEE FOUND IN TABLE
1 006000*  DSN8053I - ROLLBACK SUCCESSFUL, ALL UPDATES REMOVED
1 006200*  DSN8060E - SQL ERROR, RETURN CODE IS:
1 006300*  DSN8061E - ROLLBACK FAILED, RETURN CODE IS:
1 006400*  DSN8065E - INVALID REQUEST, SHOULD BE 'L' OR 'U'
1 006500*  DSN8075E - MESSAGE FORMAT ROUTINE ERROR
1 006600*  RETURN CODE IS:
1 006700*  
1 006800*  EXTERNAL REFERENCES =
1 006900*  ROUTINES/SERVICES =
1 007000*  DSN01AR - TRANSLATE SQLCA INTO MESSAGES
1 007100*  DSN01RCG - ERROR MESSAGE ROUTINE
1 007200*  DATA-AREAS = NONE
1 007300*  CONTROL-BLOCKS =
1 007400*  SQLCA = SQL COMMUNICATION AREA
1 007500*  
1 007600*  TABLES = NONE
1 007700*  
1 007800*  
1 007900*  
1 008000*  
1 008100*  CHANGE-ACTIVITY = NONE
1 008200*  
1 008300*  PSEUDO-CODE =
1 008400*  
1 008500*  PROCEDURE
1 008600*  007000*  GET FIRST INPUT
1 008700*  008800*  DO WHILE MORE INPUT
1 008900*  009000*  CREATE REPORT HEADING
1 009100*  009200*  CASE (ACTION)
1 009300*  009400*  SUBCASE (L*):
1 009500*  009600*  IF LASTNAME IS "" THEN
1 009700*  009800*  LIST ALL EMPLOYEES
1 009900*  009900*  ELSE
1 010000*  010100*  IF LASTNAME CONTAINS '%' THEN
1 010200*  010300*  LIST EMPLOYEES GENERIC
1 010400*  010500*  ELSE
1 010600*  010700*  LIST EMPLOYEES SPECIFIC
1 010800*  ENDSUB
1 010900*  ENDPROC

Posting Address

SQL Communication Area

Error Message Routine

SQL Communication Area

Error Message Routine

Data Areas

Control Blocks

Tables

Change Activity

Pseudo Code

Procedure

Get First Input

Do While More Input

Create Report Heading

Case (Action)

Subcase (L*)

If Lastname Is ""

List All Employees

Else

If Lastname Contains "%"

List Employees Generic

Else

List Employees Specific

Endsub

Endprocedure

7-18
IDENTIFICATION DIVISION.

PROGRAM-ID. DSN8BC3.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SPECIAL-NAMES.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

SELECT CARDIN ASSIGN TO DA-S-CARDIN.

SELECT REPOUT ASSIGN TO UT-S-REPORT.

DATA DIVISION.

FILE SECTION.

FD CARDIN

FD

FD

FD

FD

FD

FD

FD

WORKING-STORAGE SECTION.

STRUCTURE FOR INPUT

STRUCTURE FOR OUTPUT

ACTION PIC X(11).

LNAME PIC X(15).

FNAME PIC X(12).
<table>
<thead>
<tr>
<th>COL</th>
<th>016300</th>
<th>02 ENO</th>
<th>PIC X(06).</th>
<th>016400</th>
<th>02 NEWD</th>
<th>PIC X(04).</th>
<th>016500</th>
<th>02 FILLER</th>
<th>PIC X(42).</th>
</tr>
</thead>
<tbody>
<tr>
<td>016600</td>
<td></td>
<td></td>
<td></td>
<td>016700</td>
<td></td>
<td></td>
<td>016800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>016900</td>
<td></td>
<td></td>
<td></td>
<td>017000</td>
<td></td>
<td></td>
<td>017100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>017200</td>
<td></td>
<td></td>
<td></td>
<td>017300</td>
<td></td>
<td></td>
<td>017400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>017500</td>
<td></td>
<td></td>
<td></td>
<td>017600</td>
<td></td>
<td></td>
<td>017700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>017800</td>
<td></td>
<td></td>
<td></td>
<td>017900</td>
<td></td>
<td></td>
<td>018000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>018100</td>
<td></td>
<td></td>
<td></td>
<td>018200</td>
<td></td>
<td></td>
<td>018300</td>
<td></td>
<td></td>
</tr>
<tr>
<td>018400</td>
<td></td>
<td></td>
<td></td>
<td>018500</td>
<td></td>
<td></td>
<td>018600</td>
<td></td>
<td></td>
</tr>
<tr>
<td>018700</td>
<td></td>
<td></td>
<td></td>
<td>018800</td>
<td></td>
<td></td>
<td>018900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>019000</td>
<td></td>
<td></td>
<td></td>
<td>019100</td>
<td></td>
<td></td>
<td>019200</td>
<td></td>
<td></td>
</tr>
<tr>
<td>019300</td>
<td></td>
<td></td>
<td></td>
<td>019400</td>
<td></td>
<td></td>
<td>019500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>019600</td>
<td></td>
<td></td>
<td></td>
<td>019700</td>
<td></td>
<td></td>
<td>019800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>019900</td>
<td></td>
<td></td>
<td></td>
<td>020000</td>
<td></td>
<td></td>
<td>020100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020200</td>
<td></td>
<td></td>
<td></td>
<td>020300</td>
<td></td>
<td></td>
<td>020400</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020500</td>
<td></td>
<td></td>
<td></td>
<td>020600</td>
<td></td>
<td></td>
<td>020700</td>
<td></td>
<td></td>
</tr>
<tr>
<td>020800</td>
<td></td>
<td></td>
<td></td>
<td>020900</td>
<td></td>
<td></td>
<td>021000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>021100</td>
<td></td>
<td></td>
<td></td>
<td>021200</td>
<td></td>
<td></td>
<td>021300</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
021700 49 LNAME-WORKL PIC S9(1) COMP.
021800 49 LNAME-WORKC PIC X(15).
021900 01 FNAME-WORK PIC S9(3) COMP.
022100 49 FNAME-WORK PIC S9(12) COMP.
022200 77 INPUT-SWITCH PIC X VALUE 'YY'
022400 77 NOT-FOUND PIC S9(1) COMP VALUE +100.
022500
022600 01 ERROR-MESSAGE.
023000 02 ERROR-LEN PIC S9(1) COMP VALUE +100.
023300 02 ERROR-TEXT PIC X(120) OCCURS 8 TIMES
023500 INDEXED BY ERROR-INDEX.
023700 77 ERROR-TEXT-LEN PIC S9(1) COMP VALUE +120.
023800
023900
024000 EXEC SQL INCLUDE SQLCA
024100 EXEC SQL DECLARE VIEW VPHONE
024200 EXEC SQL DECLARE VPHONE TABLE
024400 (LASTNAME VARCHAR(15) NOT NULL,
024500 FIRSTNAME VARCHAR(12) NOT NULL,
024600 MIDDLEINITIAL CHAR(10) NOT NULL,
024700 PHONENUMBER CHAR(10) NOT NULL,
024800 EMPLOYEEID NUMBER CHAR(10) NOT NULL,
024900 DEPTNUMBER CHAR(10) NOT NULL,
025000 DEPTNAME VARCHAR(36) NOT NULL)
025100 END-EXEC.
025200
025300
025400 01 STRUCTURE FOR PPHONE RECORD
025500
025600 01 PPHONE.
025700
025800 49 LASTNAMEL PIC S9(1) COMP.
025900 49 LASTNAME PIC X(15) VALUE SPACES.
026000 49 FNAME-WORK PIC S9(3) COMP.
026100 49 FNAME-WORK PIC X(120) OCCURS 8 TIMES
026200 INDEXED BY ERROR-INDEX.
026300 77 NOT-FOUND PIC S9(1) COMP VALUE +100.
026400
026500
026600
026700
026800
026900
027000
SQL DECLARATION FOR VIEW VEMPLP

EXEC SQL DECLARE TELE1 CURSOR FOR
SELECT *
FROM VPHONE
END-EXEC.

EXEC SQL DECLARE TELE2 CURSOR FOR
SELECT *
FROM VPHONE
WHERE LASTNAME LIKE :LNAME-WORK
AND FIRSTNAME LIKE :FNAME-WORK
END-EXEC.

EXEC SQL DECLARE TELES CURSOR FOR
SELECT *
FROM VPHONE
WHERE LASTNAME = :LNAME
AND FIRSTNAME LIKE :FNAME-WORK
END-EXEC.

EXEC SQL DECLARE VEMPLP TABLE
EMPLOYEE NUMBER, CNAM(30), NOT NULL,
PHONE NUMBER, CNAM(30)
END-EXEC.

EXEC SQL DECLARE TELE1 CURSOR FOR
SELECT *
FROM VPHONE
END-EXEC.

EXEC SQL DECLARE TELE2 CURSOR FOR
SELECT *
FROM VPHONE
WHERE LASTNAME LIKE :LNAME-WORK
AND FIRSTNAME LIKE :FNAME-WORK
END-EXEC.

EXEC SQL DECLARE TELES CURSOR FOR
SELECT *
FROM VPHONE
WHERE LASTNAME = :LNAME
AND FIRSTNAME LIKE :FNAME-WORK
END-EXEC.
DATASET: DSN120.DSNsamp  DATE: 07/02/04
MEMBER: DSNBC3   TIME: 08:33
PAGE: 7

```
032500 02 OUTMSG2 PIC X(69).
032600
032700
032500 PROCEDURE DIVISION.
033000
033100
033200 SQL RETURN CODE HANDLING
033300 EXEC SQL WHENEVER SQLERROR GOTO OBERROR END-EXEC.
033400 EXEC SQL WHENEVER SQLWARNING GOTO OBERROR END-EXEC.
033500 EXEC SQL WHENEVER NOT FOUND CONTINUE END-EXEC.
033600
033700
033800
033900 MAIN PROGRAM ROUTINE
034000
034100 PROG-START.
034200 OPEN INPUT CARDIN.
034300 OPEN OUTPUT REPOUT.
034400
034500
034600 READ CARDIN RECORD INTO IQAREA.
034700 AT END MOVE IQAREA TO INPUT-SWITCH.
034800
034900
035000 PERFORM PROCESS-INPUT UNTIL NONORE-INPUT.
035100
035200
035300 PROG-END.
035400 **CLOSE FILES
035500 CLOSE CARDIN.
035600 REPOUT.
035700 GOBACK.
035800
035900
036000 CREATE REPORT READING.
036100 SELECT ACTION
036200
036300 PROCESS-INPUT.
036400 **PRINT READING
036500 WRITE REPREC FROM REPDR1.
036600 AFTER ADVANCING TO-TOP-OF-PAGE.
036700 WRITE REPREC FROM REPDR2.
036800 AFTER ADVANCING 2 LINES.
036900 WRITE REPREC FROM REPDR3.
037000
037100 **SELECT ACTION
037200 IF ACTION = 'L'
037300 PERFORM LIST-FUNCTION.
037400 ELSE
037500 IF ACTION = 'U'
037600 PERFORM TELEPHONE-UPDATE.
037700 ELSE
037800
```

7-23
**INVALID REQUEST**

**PRINT ERROR MESSAGE**

MOVE '068E' TO MSGCODE

CALL 'DSNMGCC' USING MAJOR MSGCODE OUTMSG

MOVE OUTMSG TO OUTMSG2

WRITE REPREF FROM MSG-REC2

AFTER ADVANCING 2 LINES.

READ CARDIN RECORD INTO IOAREA

AT END MOVE 'N' TO INPUT-SWITCH.

MOVE '068E' TO MSGCODE

CALL 'DSN8NCGC' USING

NO OUTMSG TO 0UTHSG2

WRITE REPREC FROM MSG-REC2

AFTER ADVANCING 2 LINES.

READ CARDIN RECORD INTO IOAREA

AT END MOVE 'N' TO INPUT-SWITCH.

**INVALID REQUEST**

**PRINT ERROR MESSAGE**

MOVE '068E' TO MSGCODE

CALL 'DSNMGCC' USING MAJOR MSGCODE OUTMSG

MOVE OUTMSG TO OUTMSG2

WRITE REPREF FROM MSG-REC2

AFTER ADVANCING 2 LINES.

READ CARDIN RECORD INTO IOAREA

AT END MOVE 'N' TO INPUT-SWITCH.

MOVE '068E' TO MSGCODE

CALL 'DSN8NCGC' USING

NO OUTMSG TO 0UTHSG2

WRITE REPREC FROM MSG-REC2

AFTER ADVANCING 2 LINES.

READ CARDIN RECORD INTO IOAREA

AT END MOVE 'N' TO INPUT-SWITCH.

**INVALID REQUEST**

**PRINT ERROR MESSAGE**

MOVE '068E' TO MSGCODE

CALL 'DSNMGCC' USING MAJOR MSGCODE OUTMSG

MOVE OUTMSG TO OUTMSG2

WRITE REPREF FROM MSG-REC2

AFTER ADVANCING 2 LINES.

READ CARDIN RECORD INTO IOAREA

AT END MOVE 'N' TO INPUT-SWITCH.
**MEMBER:**

```
EXEC SQL FETCH TELE1 INTO :PHONE END-EXEC.
IF SQLCODE = NOT-FOUND
  **NO EMPLOYEE FOUND**
  **PRINT ERROR MESSAGE**
  EXEC SQL OPEN TELE1 INTO :PHONE END-EXEC.
  EXEC SQL FETCH TELE1 INTO :PHONE END-EXEC.
  IF SQLCODE = NOT-FOUND
    **NO EMPLOYEE FOUND**
    **PRINT ERROR MESSAGE**
    EXEC SQL CLOSE TELE1.
  ELSE
    **LIST ALL EMPLOYEES**
    PERFORM PRINT-AND-GET1 UNTIL SflLCODE IS NOT EQUAL TO ZERO.
  END-EXEC.
EXEC SQL CLOSE TELE1 END-EXEC.
```
LIST SPECIFIC EMPLOYEES

EXEC SQL OPEN TELE3 END-EXEC.

EXEC SQL FETCH TELE3 INTO JPPHONE END-EXEC.

IF SQLCODE = NOT-FOUND
**NO EMPLOYEE FOUND**
**PRINT ERROR MESSAGE** 04980000

MOVE '0000' TO MSGCODE
CALL F0SMHNG USING MAJOR MSGCODE OUTMSG
MOVE OUTMSG TO OUTMSG2
WRITE REPREC FROM MSG-REC2
AFTER ADVANCING 2 LINES
ELSE
**LIST SPECIFIC EMPLOYEE(S)**
**PRINT ERROR MESSAGE** 05040000
PERFORM PRINT-AND-GET3 UNTIL SQLCODE IS NOT EQUAL TO ZERO.

EXEC SQL CLOSE TELE3 END-EXEC.

UPDATE PHONE NUMBERS FOR EMPLOYEES
**PRINT INFORMATION** 05350000
EXEC SQL UPDATE VEMPLP
SET PHONENUMBER = :NEWNO
WHERE EMPLOYEEID = :ENO END-EXEC.

IF SQLCODE = ZERO
**EMPLOYEE FOUND
**UPDATE SUCCESSFUL
**PRINT CONFIRMATION
**MESSAGE
ELSE
**NO EMPLOYEE FOUND
**UPDATE FAILED
**PRINT ERROR MESSAGE

CALL 'DSNM8CG' USING major MSGCODE OUTMSG.
MOVE OUTMSG TO OUTMSG1.
WRITE REPRE FROM MSG-REC2 AFTER ADVANCING 2 LINES.

EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.
EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
EXEC SQL WHENEVER NOT FOUND CONTINUE END-EXEC.

**PERFORM ROLLBACK
EXEC SQL ROLLBACK END-EXEC.

IF SQLCODE = ZERO
**ROLLBACK SUCCESSFUL**
**PRINT CONFIRMATION**
**MESSAGE**

**ROLLBACK FAILED**
**PRINT ERROR MESSAGE**

*061000*
*061100*
*061200*
*061300*
*061400*
*061500*
*061600*
*061700*
*061800*
*061900*
*062000*
### SELECTING AN EMPLOYEE TO DISPLAY

**MAJOR SYSTEM:** O
**ACTION:** D
**OBJECT:** EM
**SEARCH CRITERIA:** EN

<table>
<thead>
<tr>
<th>NO</th>
<th>D/ID</th>
<th>DEPARTMENT NAME</th>
<th>E/ID</th>
<th>EMPLOYEE NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000010</td>
<td>CI BASS</td>
</tr>
<tr>
<td>02</td>
<td>B01</td>
<td>PLANNING</td>
<td>000020</td>
<td>ML THOMPSON</td>
</tr>
<tr>
<td>03</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>000030</td>
<td>SA KWAN</td>
</tr>
<tr>
<td>04</td>
<td>E01</td>
<td>SUPPORT SERVICES</td>
<td>000050</td>
<td>JB GEYER</td>
</tr>
<tr>
<td>05</td>
<td>D01</td>
<td>MANUFACTURING</td>
<td>000060</td>
<td>IF STERN</td>
</tr>
<tr>
<td>06</td>
<td>D21</td>
<td>ADMINISTRATION SYSTEMS</td>
<td>000070</td>
<td>ED PULASKI</td>
</tr>
<tr>
<td>07</td>
<td>E11</td>
<td>OPERATIONS</td>
<td>000090</td>
<td>EW HENDERSON</td>
</tr>
<tr>
<td>08</td>
<td>E21</td>
<td>SOFTWARE SUPPORT</td>
<td>000100</td>
<td>TQ SPENSER</td>
</tr>
<tr>
<td>09</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000110</td>
<td>VG LUCCHESI</td>
</tr>
<tr>
<td>10</td>
<td>A00</td>
<td>SPIFFY COMPUTER SERVICE DIV.</td>
<td>000120</td>
<td>S O'CONNELL</td>
</tr>
<tr>
<td>11</td>
<td>C01</td>
<td>INFORMATION CENTER</td>
<td>000130</td>
<td>DM QUINTANA</td>
</tr>
</tbody>
</table>

**PFK:** 02-RESEND 03-END 06-NEXT

### EMPLOYEE ADD

**MAJOR SYSTEM:** O
**ACTION:** A
**OBJECT:** EM
**SEARCH CRITERIA:** EN
**DATA:** 000030

<table>
<thead>
<tr>
<th>EMPLOYEE ID</th>
<th>FIRST NAME</th>
<th>MIDDLE INITIAL</th>
<th>LAST NAME</th>
<th>WORK DEPT ID</th>
<th>PHONE NUMBER</th>
</tr>
</thead>
</table>

**PFK:** 02-RESEND 03-END

---

**DB2 - Online program screens**

---

7-29
### EMPLOYEE ERASE

<table>
<thead>
<tr>
<th>MAJOR SYSTEM</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>E</td>
</tr>
<tr>
<td>OBJECT</td>
<td>EN</td>
</tr>
<tr>
<td>SEARCH CRITERIA</td>
<td>EN</td>
</tr>
<tr>
<td>DATA</td>
<td>000030</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMPLOYEE ID</th>
<th>000030</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST NAME</td>
<td>JANE</td>
</tr>
<tr>
<td>MIDDLE INITIAL</td>
<td>E</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>DOE</td>
</tr>
<tr>
<td>WORK DEPT ID</td>
<td>E21</td>
</tr>
<tr>
<td>PHONE NUMBER</td>
<td>0000</td>
</tr>
</tbody>
</table>

**PFK: 02-RESEND 03-END**

### EMPLOYEE UPDATE

<table>
<thead>
<tr>
<th>MAJOR SYSTEM</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTION</td>
<td>U</td>
</tr>
<tr>
<td>OBJECT</td>
<td>EN</td>
</tr>
<tr>
<td>SEARCH CRITERIA</td>
<td>EN</td>
</tr>
<tr>
<td>DATA</td>
<td>000030</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EMPLOYEE ID</th>
<th>000030</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIRST NAME</td>
<td>JANE</td>
</tr>
<tr>
<td>MIDDLE INITIAL</td>
<td>E</td>
</tr>
<tr>
<td>LAST NAME</td>
<td>DOE</td>
</tr>
<tr>
<td>WORK DEPT ID</td>
<td>E21</td>
</tr>
<tr>
<td>PHONE NUMBER</td>
<td>0000</td>
</tr>
</tbody>
</table>

**PFK: 02-RESEND 03-END**

DB2 - Online program screens
**COPYRIGHT (C) COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2003**

**NAME** = DSN8MCG

**MESSAGE ROUTINE**

**MESSAGE CODE** = BS2

**DESCRIPTION** = A MESSAGE CODE

**EXTERNAL REFERENCES**

**SYMBOLIC LABEL/NAME** = MSGCODE

**RETURN CODE** = NONE

**EXIT-ERROR** = NONE

**EXIT-NORMAL** = NONE

**EXIT-ERROR** = NONE

**EXIT-NORMAL** = NONE

**EXIT-ERROR** = NONE
IDENTIFICATION DIVISION.

PROGRAM-ID. DSN3MCG.

ENVIRONMENT DIVISION.

DATA DIVISION.

WORKING-STORAGE SECTION.

*OUTPUT MESSAGE

01 WORK-MSG.
  01 WORK-HEAD PIC X(04) VALUE 'DSMIX'.
  02 CODE PIC X(04).
  02 MESSAGE PIC X(50).
  02 MSGCODE PIC X(04).
  02 INPUT MESSAGE-CODE PIC X(04).

*INPUT MESSAGE CODE

01 050000
  0054000 Routines/Services = None
  0055000 Data-areas = None
  0057000 Control-blocks = None
  0058000 Tables = None
  0059000 Change-activity = None
  0056000 Pseudocode
  0060000 Procedure

GET INPUT FROM CALLING ROUTINE
SEARCH CODE ARRAY

IF CODES MATCH
GET APPROPRIATE MESSAGE
ELSE
USE 'MESSAGE TEXT NOT FOUND' AS MESSAGE

SEND MESSAGE TO CALLING ROUTINE

END.
DATE: 57/02/12
TIME: 17:24

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.

010000

010100

PROCEDURE DIVISION USING MAJOR MSG CODE OUT MSG.
1 016000
1 016500
1 020600
  + DEPARTMENT +
1 016000
1 016500+ IF MSGCODE EQUAL '0111' THEN
1 016700 MOVE 'DEPARTMENT NOT FOUND' + TO 016700
1 016000 0-MESSAGE OF WORK-MSG.* 016000
1 016900+ IF MSGCODE EQUAL '0121' THEN
1 017000 MOVE 'DEPARTMENT SUCCESSFULLY ADDED' + TO 017100
1 017200 0-MESSAGE OF WORK-MSG.* 017200
1 017300+ IF MSGCODE EQUAL '0131' THEN
1 017400 MOVE 'DEPARTMENT SUCCESSFULLY ERASED' + TO 017500
1 017600 0-MESSAGE OF WORK-MSG.* 017600
1 017700+ IF MSGCODE EQUAL '0141' THEN
1 017800 MOVE 'DEPARTMENT SUCCESSFULLY UPDATED' + TO 017900
1 018000 0-MESSAGE OF WORK-MSG.* 018000
1 018100+ IF MSGCODE EQUAL '0151' THEN
1 018200 MOVE 'DEPARTMENT EXISTS ALREADY, ADD NOT DONE' + TO 018300
1 018400 0-MESSAGE OF WORK-MSG.* 018400
1 018500+ IF MSGCODE EQUAL '0161' THEN
1 018600 MOVE 'DEPARTMENT DOES NOT EXIST, ERASE NOT DONE' + TO 018700
1 018800 0-MESSAGE OF WORK-MSG.* 018800
1 018900+ IF MSGCODE EQUAL '0171' THEN
1 019000 MOVE 'DEPARTMENT EXISTS ALREADY, UPDATE NOT DONE' + TO 019100
1 019200 0-MESSAGE OF WORK-MSG.* 019200
1 019300+ IF MSGCODE EQUAL '0181' THEN
1 019400 MOVE 'CURRENT' DEPARTMENT NOT FOUND + TO 019500
1 019600 0-MESSAGE OF WORK-MSG.* 019600
1 019700+ IF MSGCODE EQUAL '0191' THEN
1 019800 MOVE 'NO 'HIGHER' DEPARTMENT EXISTS + TO 019900
1 020000 0-MESSAGE OF WORK-MSG.* 020000
1 020100
1 020200
1 020300+ GENERAL INFO MESSAGES +
1 020400+ IF MSGCODE EQUAL '0501' THEN
1 020500 MOVE 'PROGRAM STARTED' + TO 020600
1 020700 0-MESSAGE OF WORK-MSG.* 020700
1 020800+ IF MSGCODE EQUAL '0511' THEN
1 020900 MOVE 'PROGRAM ENDED' + TO 021000
1 021100 0-MESSAGE OF WORK-MSG.* 021100
1 021200+ IF MSGCODE EQUAL '0521' THEN
1 021300 MOVE 'NO HIGHER WARNING, RETURN CODE IS:' + TO 021400
1 021500 0-MESSAGE OF WORK-MSG.* 021500

DATASET: DSN1234,SMYSAP
MEMBER: DSN1346

DATE: 07/02/12
TIME: 17:24
PAGE: 4
02160000
02170000
02180000
02190000
02200000
02210000
02220000
02230000
02240000
02250000
02260000
02270000
02280000
02290000
02300000
02310000
02320000
02330000
02340000
02350000
02360000
02370000
02380000
02390000
02400000
02410000
02420000
02430000
02440000
02450000
02460000
02470000
02480000
02490000
02500000
02510000
02520000
02530000
02540000
02550000
02560000
02570000
02580000
02590000
02600000
02610000
02620000
02630000
02640000
02650000
02660000
02670000
02680000
02690000

I-O-MESSAGE

O-MESSAGE

MOVE

026300*

024800

*051I

024000

MSGCODE

MOVE

TO

025400

TO

O-MESSAGE

O-MESSAGE

HSGCDE

024400

U2&7UUD0

023000

M5GC0DC

024200

025000

*068E*

IF

022900

023500

024600

024700

*068E*

O-MESSAGE

TQ

O-MESSAGE

WDV

TABLE

O-MESSAGE

WORK-HSG.

HOVE

TO,

*069E*

*069E*

*069E*

*069E*

平等

*069E*

IF

023200

MSGCODE

*069E*

then

equal

*069E*

WORK-MSG.

DL/I

SECONDARY

DETAIL

ERROR,

*069E*

ERROR

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*

RETURN

*069E*

EQUAL

*069E*
```
| START | COL | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- | --------------------------------- |
|-------|-----|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
| 1     | 027000 | O-MESSAGE OF WORK-MSG. | "072E" | 0270000000 | 1     | 0271000 | IF MSGCODE EQUAL "072E" THEN | 0271000000 | 1     | 0272000 | MOVE "INVALID SELECTION ON SECONDARY SCREEN" TO 0273000000 | 1     | 0274000 | O-MESSAGE OF WORK-MSG. | 0274000000 | 1     | 0275000 | IF MSGCODE EQUAL "073E" THEN | 0275000000 | 1     | 0276000 | MOVE "SPECIFIED LINE-NUMBER NOT FOUND IN PREVIOUS SCREEN" TO 0277000000 | 1     | 0278000 | O-MESSAGE OF WORK-MSG. | 0278000000 | 1     | 0279000 | IF MSGCODE EQUAL "074E" THEN | 0279000000 | 1     | 0280000 | MOVE "DATA IS TOO LONG FOR SEARCH CRITERIA" TO 0281000000 | 1     | 0282000 | O-MESSAGE OF WORK-MSG. | 0282000000 | 1     | 0283000 | IF MSGCODE EQUAL "075E" THEN | 0283000000 | 1     | 0284000 | MOVE "MESSAGE FORMAT ROUTINE ERROR, RETURN CODE 152" TO 0285000000 | 1     | 0286000 | O-MESSAGE OF WORK-MSG. | 0286000000 | 1     | 0287000 | O-MESSAGE OF WORK-MSG. | 0287000000 | 1     | 0288000 | O-MESSAGE OF WORK-MSG. | 0288000000 | 1     | 0289000 | PROG-END. | **RETURN TO CALLER** 0289000000 | 1     | 0290000 | **END OF PROGRAM** 0290000000 | 1     | 0291000 | MOVE OMSG TO OUTMSG. | 0291000000 | 1     | 0292000 | MOVE OMSG TO OUTMSG. | 0292000000 | 1     | 0293000 | GOBACK. | 0293000000 |
|-------|-----|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|-----------------------------------|
```
DATASET: OSM127.OSNAMP
MEMBER: OSM1NICE

START
COL

7  * COMMAREA PART 2
  1  WORK.
  2  VMK PIC X OCCURS 40.
  2  LINE-SELECT PIC X(2).
  2  LINE-SELECT-P REDEFINES LINE-SELECT-C PIC 99.
  3  77 1 PIC 99 COMP.
  8  77 77 TOPLINE PIC 99 COMP.
  8  77 77 TOPL-1 PIC 99 COMP.
  8  77 77 CURNLINE PIC 99 COMP.
  8  77 77 CURR-L PIC 99 COMP.
  8  77 77 BOTTLINE PIC 99 COMP.
  8  77 77 CURSOR-VALUE PIC S9999 COMP.
  8  77 77 RNC-INDEX PIC X(4).
  8  77 77 SAVC-COMVU PIC X(16).
  8  77 77 HELPBIT PIC X.
  8  77 77 SENDBIT PIC X.
  8  77 77 ENDBIT PIC X.
  8  77 77 NEXTBIT PIC X.
  8  77 77 ON-1 PIC X.
  8  77 77 OFF-1 PIC X.
  8  77 77 PEACENT PIC X VALUE 'X'.
  8  77 77 OPTRE PIC X(70).
  17 77 77 VALUE ENTRY MISSING IN TABLE TOPIVAL.
  8  77 77 OSPNF - PIC X(70).
  17 77 77 VALUE ENTRY MISSING IN TABLE TDSPTXT.
  8  77 77 77 ONA-LEN PIC S9199 COMP VALUE +79.
  8  1  ERAGA-MESSAGE.
  7  2  KLEN PIC S9(61) COMP VALUE +632.
  9  2  ERROR-DATA.
  10 3  ERR-TEXT1 PIC X(79).
  10 3  ERR-TEXT2 PIC X(79).
  10 3  ERR-TEXT3 PIC X(79).
  10 3  ERR-TEXT4 PIC X(79).
  10 3  ERR-TEXT5 PIC X(79).
  10 3  ERR-TEXT6 PIC X(79).
  10 3  ERR-TEXT7 PIC X(79).
  10 3  ERR-TEXT8 PIC X(79).
  8  1  OSNMCAS.
  9  3  BLANK-DEPTH PIC X(1) VALUE SPAC.
  9  3  BLANK-DEPTHNAME.
  10 49 49 BLANK-DEPTHNAME PIC S9(4) COMP-4 VALUE +36.
  10 49 49 BLANK-DEPTHNAME PIC X(136) VALUE SPAC.
  9  3  BLANK-LASTNAME.
  10 49 49 BLANK-LASTNAME PIC S9(4) COMP-4 VALUE +12.
  10 49 49 BLANK-LASTNAME PIC X(12) VALUE SPAC.
  9  3  BLANK-LASTNAME.
  10 49 49 BLANK-LASTNAME PIC S9(4) COMP-4 VALUE +15.
  10 49 49 BLANK-LASTNAME PIC X(15) VALUE SPAC.

DATE: 87/02/12
TIME: 17:35
PAGE: 1

7-37.
**MODULE FOR IMS AND CICS**

**COBOL**

**COPYRIGHT © 5740-XYR (C) COPYRIGHT IBM CORP 1932, 1985**

**REFER TO COPYRIGHT INSTRUCTIONS FOR NEWCOPYR**

**STATUS = RELEASE 2, LEVEL 0**

**FUNCTION = RETRIEVE LAST CONVERSATION.**

**HANDLES 'RESEND' AND 'PEND' CASES.**

**CALLS VALIDATION ROUTINES DSN8NC3 THRU DSN8NC5.**

**CALLS SQL ROOT (DSN8CC2 OR DSN8IC2),**

**NOTES = NONE.**

**MODULE TYPE = DB2 PRECOMPILER, VS COBOL**

**MODULE SIZE = SEE LINKEDIT**

**ATTRIBUTES = REUSABLE**

**ENTRY POINT = DSN8MC1**

**PURPOSE = SEE FUNCTION**

**LINKAGE = INCLUDED BY MODULE DSN8IC1 OR DSN8IC2**

**INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:**

**DESCRIPTION = NONE**

**OUTPUT = PARAMETERS EXPLICITLY RETURNED:**

**DESCRIPTION = NONE**

**EXIT-NORMAL = DROP THRU TO NEXT LINE OF CODE IN DSN8CP1/IP1**

**EXIT-ERROR = IF SQLERROR OR SQLWARNING, SQL WHENEVER**

**CONDITION SPECIFIED IN DSN8CC1/IC1 WILL BE RAISED**

**AND PROGRAM WILL GO TO THE LABEL DB-ERROR.**

**RETURN CODE = NONE**

**ABEND CODES = NONE**

**ERROR MESSAGES = DSN8051 - PROGRAM ENDED**

**EXTERNAL REFERENCES = MOST VARIABLES ARE GLOBAL AND DEFINED**

**IN DSN8CC1/IC1.**

---

**ENTRY POINT = DSN8MC1**
PROCEDURE

ELIMINATE LEADING BLANKS ON DATA LINE IF NOT ALL OF DATA

LINE IS BLANK.

SET UP CONTROL FLAGS FOR 'RESEND' 'END' 'NEXT'

FIRST BY EXAMINING THE DATA LINE AND THEN COMPARING THE

PF KEYS (COMPARN,PFKEY)

RETRIEVE LAST CONVERSATION IF FROM VCONA.

IF LAST CONVERSATION IS NOT FOUND THEN DO.

COMPARN.NEWCONV = 'Y'.

PCONVSTA = ***.

END.

ELSE DO.

PCONVSTA = LAST CONVERSATION RETRIEVED.

IF RESEND, BYPASS VALIDATION AND SAVE, JUST RESEND.

IF END, DELETE CONVERSATION, SEND MESSAGE & GOTO CCEXIT.

IF NO SYSTEMS FIELD WAS CHANGED, BYPASS VALIDATION.

END.

WHILE RETURN CODE IS 0 DO

CALL VALIDATION MODULES DSN8MC3 THRU DSN8HC5

OTHERWISE

GO TO MC1SAVE.

GO TO CCICALL IN DSNRCC1/I1 TO CALL DSNRCC2/I2.

MC1SAVE:

INSERT/UPDATE CURRENT CONVERSATION INTO VCONA.

END.

INITIAL EDITING FOR DATA INPUT
1. The data line is shifted left until all leading blanks have been eliminated.

2. The appropriate bits for 'RESEND', 'END' etc. are then set according to input on data line.

3. If PF keys 1, 2, or 8 have been used, the appropriate bit is set for 'RESEND', 'END' etc. This takes precedence over the setting of the same bits in step 2.

* I.e. if someone types in 'RESEND' on the data line and uses the PF1 key at the same time, the PF1 'END' function is assumed to be the actual request.

---

**INITIALIZE CONTROL FLAGS**

**SKIP LEADING BLANKS**

**IF ALL OF LINE**

**IS OF BLANKS**

**SEE IF A CONTROL**

**FLAG IS SET**

**MC1-LOOP**

**MC1-10 LOOP**

**IF FIRST CHARACTER IS**

**NON-BLANK, SEE IF A**

**CONTROL FLAG IS SET**

IF I = 1 THEN

GO TO MC1-18.

MOVE 1 TO J.
```
COL  1  2  3  4  5  6  7  8
3  MCl-17.
7  **GET NON-BLANK CHARACTERS
12  MOVE DATAIN(I) TO QDAIN1(I).
12  ADD 1 TO I.
12  ADD 1 TO J.
7  **MCl-12 LOOP
12  PUTF MD MCl-12.
15  UNTIL I > 60.
8  MCl-14.
7  **PUT BLANKS AT END
12  MOVE SPACE TO DATAIN(I).
12  ADD 1 TO J.
7  **MCl-14 LOOP
12  PUTF MD MCl-16.
15  UNTIL J > 60.
7  ****************************
7  **SET UP CONTROL FLAGS FOR *RESEND* *END* *NEXT*
7  ****************************
8  MCl-18.
15  IF DATAIN = *RESEND* OR
7  *  **RESEND COMMAND OR
7  **PF KEY O2
13  PFKIN IN INAREA = 'O2' THEN
13  MOVE 04+1 TO SENDBIT
15  ELSE
15  IF DATAIN = *END* OR
7  *  **END COMMAND OR
7  **PF KEY O3
21  PFKIN IN INAREA = 'O1' THEN
21  MOVE 04+1 TO ENDBIT
21  ELSE
21  IF DATAIN = *NEXT* OR
7  *  **NEXT COMMAND OR
7  **PF KEY O8
24  PFKIN IN INAREA = 'O8' THEN
24  MOVE 04+1 TO NEXTBIT.
4  MCl-20.
7  ****************************
7  **RESTORE LAST MESSAGE AND DETERMINE IF VALIDATION IS NECESSARY
7  **
7  ***1. ATTEMPT TO RETRIEVE LAST MESSAGE STORED IN VCNA. IF
7  *** NOT SUCCESSFUL, THEN CONVERSATION IS NEW.
7  ***2. IF RETRIEVAL IS SUCCESSFUL, THEN TRANSFER THE DATA
7  *** INTO PCONVTA.
```

<table>
<thead>
<tr>
<th>START COL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>* 3. IF RESCN REQUEST, DON'T VALIDATE &amp; DON'T SAVE, JUST RSEND.</td>
<td>02170000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>* 4. IF END REQUEST=DELETE CONVERSATION=SEND END MESSAGE, EXIT.</td>
<td>02180000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>* 5. IF ALL SYSTEM FIELDS HAVE NOT CHANGED SINCE THEY WERE</td>
<td>02190000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>LAST SAVED, BYPASS VALIDATION ALSO.</td>
<td>02200000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>* 6. OTHERWISE VALIDATE EACH OF THE SYSTEM FIELDS.</td>
<td>02210000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*******************************************</td>
<td>02220000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE &quot;*&quot; TO NEWREV IN COMPARN.</td>
<td>02230000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE CONVID IN PCONVSTA TO SAVE-CONVID.</td>
<td>02250000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>EXEC SQL SELECT * INTO :PCONA FROM PCONVSTA</td>
<td>02260000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>WHERE PCONVSTA = :SAVE-CONVID END-EXEC.</td>
<td>02270000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02280000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IF SOLCODE = +100 THEN</td>
<td>02290000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE ** TO NEWREV IN COMPARN</td>
<td>02300000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE SPACE TO PCONVSTA</td>
<td>02310000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE SAVE-CONVID TO CONVID IN PCONVSTA</td>
<td>02320000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE SAVE-CONVID TO CONVID IN PCONA</td>
<td>02330000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE &quot;DSNREG1&quot; TO LASTSCR IN PCONVSTA</td>
<td>02340000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>GO TO MC1-VAL.</td>
<td>02350000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02360000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE LASTSCR IN PCONA TO LASTSCR IN PCONVSTA</td>
<td>02370000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE LASTPOS IN PCONA TO LASTPOS IN PCONVSTA</td>
<td>02380000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE LASTPOS IN PCONA TO LASTPOS IN PCONVSTA</td>
<td>02390000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02400000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE LASTMSG-TEXT IN PCONA TO OUTAREA.</td>
<td>02410000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02420000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IF PFHIN IN INAREA = &quot;00&quot; OR SENDB=ON-1 THEN GO TO CC1-EXIT.</td>
<td>02430000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02440000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>EXEC SQL UPDATE, FROM VCONA</td>
<td>02450000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>WHERE PCONVSTA = :SAVE-CONVID END-EXEC.</td>
<td>02460000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02470000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IF END3IT NOT ON-1 THEN GO TO CC1-EXIT.</td>
<td>02480000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02490000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>IF ENDSIT NOT ON-1 THEN GO TO CC1-EXIT.</td>
<td>02500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE &quot;DELETE&quot; TO MINOR IN DSNM-DISNAME</td>
<td>02510000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>MOVE SPACE TO OUTAREA.</td>
<td>02520000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>EXEC SQL DELETE FROM PCONA</td>
<td>02530000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>WHERE PCONVSTA = :PCONA, CONVID END-EXEC</td>
<td>02540000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*******************************************</td>
<td>02550000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
END

**PROGRAM ENDED**

02710000

CALL "DSN60500" USING DSNRCODE DUTMSG

02720000

**IF OLD CONVERSATION AND SYSTEM**

**FILEDS HAVE NOT CHANGED THEN**

02730000

**BYPASS VALIDATION**

02740000

**NEW CONVERSATION**

02750000

**IF ACTION IN INAREA NOT = ACTION IN OUTAREA OR**

02760000

**OBJECT IN INAREA NOT = OBJECT IN OUTAREA OR**

02770000

**SUCH IN INAREA NOT = SUCH IN OUTAREA THEN**

02780000

GO TO MCI-EXIT.

02790000

**OLD CONVERSATION**

**SAME IN INAREA NOT**

**DATAOUT IN OUTAREA**

02800000

GO TO MCI-VAL.

02810000

**VALIDATES FIELDS**

02820000

**VALIDATE FIELDS**

02830000

**VALIDATE OBJECT**

02840000

**VALIDATE SEARCH**

02850000

**IF ALL SYSTEM FIELDS ARE OK, CONTINUE**

02860000

**MCI-both.**

02870000

02880000

02890000

02900000

02910000

02920000

02930000

02940000

02950000

02960000

02970000

02980000

02990000

03000000

03010000

03020000

03030000

03040000

03050000

03060000

03070000

03080000

03090000

03100000

03110000

03120000

03130000

03140000

03150000

03160000

03170000

03180000

03190000

03200000

03210000

03220000

03230000

03240000
15 IF RV request space THEN
16 IF CVST1 = *S* THEN
15 IF CVST1 IN INAREA NOT = SPACE1 AND
16 IF NEXTCV = OFF-1 IF THEN
16 MOVE **V** TO NEWVRS IN COMPARM.

7 * **GO TO CCI-CALL WHERE A CALL **0520000
7 * **TO EITHER DSNBCCZ OR DSNBIC2 **0330000
7 * **IS PERFORMED. **03340000

7 * **RETURN CCI-CALL **03350000

13 **GO TO CCI-CALL. **03370000
13 **DSN5CC1 OR DSN5IC1 WILL **03380000
13 **BRANCH BACK TO MSAVE AFTER **03390000
13 **CALLING SOL2 AT MSAVE; **03400000
13 **THE DATA RETURNED BY SOL2 OR THE **03410000
13 **VALIDATION ROUTINES WILL BE **03420000
13 **MSAVE IN VCNA **03430000
13 **END-EXEC. **03440000

13 MOVE CVST1 IN INAREA TO DATAOUT IN OUTAREA. **03450000
13 MOVE +1602 TO LASTMSG-LEN. **03460000
13 MOVE OUTREAD TO LASTMSG-TEXT. **03470000
13 MOVE CVST1 IN PCONVSTA TO CONVST IN PCONA. **03480000
13 MOVE LASTSCR IN PCONVSTA TO LASTSCR IN PCONA. **03490000
13 MOVE LASTPOS IN PCONVSTA TO LASTPOS IN PCONA. **03500000
13 MOVE LASTPOSC IN PCONVSTA TO LASTPOSC IN PCONA. **03510000
13 MOVE *MSAVE* TO VENDS IN DSNM-MODULE-NAME. **03520000
13 MOVE *MSAVE* TO VENDS IN DSNM-MODULE-NAME. **03530000
13 MOVE *MSAVE* TO VENDS IN DSNM-MODULE-NAME. **03540000

7 * **IF NEWCVV = **V** THEN **03550000
16 EXEC SQL INSERT **03560000
21 INTO VCNA **03570000
21 VALUES (:PCONA) END-EXEC. **03580000

7 * **UPDATE OLD VALUES **03590000
15 **IF NEWCVV NOT = **V** THEN **03600000
16 EXEC SQL UPDATE VCNA **03610000
21 SET LASTSCR = :PCONA.LASTSCR **03620000
27 LASTPOS = :PCONA.LASTPOS **03630000
27 LASTPOSC = :PCONA.LASTPOSC **03640000
27 LASTMSG = :PCONA.LASTMSG **03650000
27 WHERE CONVID = :SAVE-CONDEND END-EXEC. **03660000

7 *
### DESCRIBING DSN3CC1

**Module Name:** DSN3CC1

**Descriptive Name:** DB2 Sample Application

**Copyright:** 5/74-XR (C) Copyright IBM Corp 1974, 1975

**Status:** Release 2, Level 0

**Function:** This module performs the INCLUDES to bring in the SQL Table DCLS and DCLS Structures as well as the Parameter Area.

**Dependencies:** Called by DSN3CC0; calls DSN3CC2 (CICS LINKS).

**Restrictions:** None

**Module Type:** Processor = DB2 Precompiler, CICS Translator, COBOL Compiler

**Attributes:** Reusable

**Entry Point:** DSN3CC1

**Purpose:** SEE FUNCTION

**Linkage:** Included by Module DSN3CC1

**Input:** Parameters Explicitly Passed to This Function

**Symbolic Label/Name:** None

**Description:** None

**Output:** Parameters Explicitly Returned

**Symbolic Label/Name:** None

**Description:** None

**Exit-Normal:** DSN3CC0

**Exit-Error:** DSN3CC0

**Return Code:** None

**Abend Code:** None

**Error-Messages:** None

**External References:**

**Routine/Services:** DSN3CC2
DATA DIVISION.

WORKING-STORAGE SECTION.

IDENTIFICATION DIVISION.

PROGRAM-ID. DSNMCC1.

ENVIRONMENT DIVISION.

DATA DIVISION.
DECLARE COMMON A4EA HhO C31 liSf!
AREA PARI
HSGCOPE
01 DUMSG
EXEC SQL INCLUDE DSN$MCCS END-EXEC.

LINKAGE SECTION.
01 DFHCOMMAREA.
EXEC SQL INCLUDE DSN$MCC4 END-EXEC.

PROCEDURE DIVISION.
EXEC SQL INCLUDE DSN$MCC5 END-EXEC.

EXEC SQL INCLUDE DSN$MCC2 END-EXEC.

EXEC CICS LINK PROGRAM 'DSN$CC2' COMMAREA (DFHCOMMAREA) LENGTH (3000) END-EXEC.
GO TO +C1-SAVE.
EXEC SQL INCLUDE DSN$MCC3 END-EXEC.
EXEC SQL INCLUDE DSN$MCC4 END-EXEC.
EXEC SQL INCLUDE DSN$MCC5 END-EXEC.
EXEC SQL INCLUDE DSN$MCCX END-EXEC.
GOBACK.
*DSN8MC4 - VALIDATION MODULE FOR OBJECT - COBOL ********

** *DSN8MC4 - MODULE NAME = DSN8MC4

** DESCRIPTION NAME = ORZ SAMPLE APPLICATION Validation Module For Object

** COBOL

** COPYRIGHT = 5740-XVR ICJ COPYRIGHT IBM CORP 1982, 1985

** REFER TO COPYRIGHT INSTRUCTIONS FOR NUMBER 6120-2083

** STATUS = RELEASE 2, LEVEL 0

** FUNCTION = THIS NODULE VALIDATES SPECIFIC INPUT AND MOVES IT TO THE OUTPUT MESSAGE TOGETHER WITH A TEXT FIELD.

** NOTES = NONE

** MODULE TYPE =

** PROCESSOR = ORZ PRECOMPILER, COBOL COMPILER

** MODULE SIZE = SEE LINKEDIT

** ATTRIBUTES = NONE

** ENTRY POINT = DSN8MC

** PURPOSE = SEE FUNCTION

** LINKAGE = INCLUDED BY NODULE DSN8CC1

** INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:

** SYMBO L/NAME = NONE

** DESCRIPTION = NONE

** OUTPUT = PARAMETERS EXPLICITLY RETURNED:

** SYMBO L/NAME = NONE

** DESCRIPTION = NONE

** EXIT-NORMAL = THIS CODE IS "PERFORMED", SO IT EXITS TO THE CODE FOLLOWING THE "PERFORM" STATEMENT

** EXIT-ERROR = IF SOLERROR OR SALWARING, SOL UNNEVER

** CONDITION SPECIFIED IN DSN8CC1/ICJ WILL BE RAISED AND PROGRAM WILL GO TO THE LABEL DB-ERROR.

** RETURN CODE = NONE

** ABEND CODES = NONE

** ERROR MESSAGES =

** DSN1070E VITAL DATA MISSING IN TABLE TOPTVAL

** EXTERNAL REFERENCES = MOST VARIABLES ARE GLOBAL AND
**START**

1. **ROUTES/SERVICES** = DEFINED IN DSN4CC1/IC1.
2. DSNMCQ - ERROR MESSAGE ROUTINE
3. **DATA AREAS** = NONE
4. **CONTROL BLOCKS** = SQLCA - SQL COMMUNICATION AREA
5. **TABLES** = NONE
6. **CHANGE ACTIVITY** = NONE
7. **PSEUOCODE**
8. **PROCEDURE**
9. **INITIALIZE** RETURN CODE TO '0'.
10. **FILL IN THE DISPLAY AREA** FROM VOPTVAL OBJEXTXT, DEPENDING ON OBJECT REQUIRED RETURN.
11. IF OBJECT NOT FOUND RETRIEVE A LIST OF OBJECTS WHICH EXISTS, HEADXT, INFOXT AND PKXT.
12. FROM VOPTVAL
13. DEPENDING ON MAJSYS = MAJSYS + ACTION = ACTION AND OBJECT = BLANK SET RETURN CODE TO '*'.
14. **END**.

3. **DSNMCQ**.

---

**DSN120, DSN5AMP**

**DATE:** 07/02/12

**TIME:** 17:37

**PAGE:** 2

---

**EXEC SQL SELECT SELTXT INTO :VOPTVAL, SELTXT FROM VOPTVAL**
START

COL

17   WHERE MAJSYS = 'INAREA.MAJSYS' AND ACTION = 'INAREA.ACTION' AND OBJECT = 'INAREA.OBJECT'

17   END-EXEC.

7   **OBJECT EXISTS

17   **FILL IN DISPLAY AREA

17   MOVE OBJECT IN INAREA TO OBJECT IN OUTAREA.

17   MOVE SELTXT IN POPTVAL TO DESC3 IN OUTAREA.

12   IF SOLCODE = +0 THEN

12   **RETURN

12   GO TO END-DSNAMC4.

12   OPEN CURSOR

12   EXEC SQL OPEN VO3 END-EXEC.

12   MOVE +1 TO I.

8   MC4-10.

8   **RETRIEVE LIST

12   IF I NOT > 15 THEN EXEC SQL FETCH VO3 INTO :POPTVAL.OBJECT.

25   :POPTVAL.SELTXT END-EXEC

20   **OBJECT NOT FOUND

12   IF SOLCODE NOT EQUAL TO +10 THEN

12   **OPEN CURSOR

12   **CLOSE CURSOR

12   **PUT BLANKS AT END OF LINE

12   IF J NOT > 15 THEN MOVE SPACE TO LINE0(J)

12   ADD I TO J.

12   GO TO MC4-30.

7   **CHECK FOR CONDITION WHERE THERE ARE NO VALID ENTRIES
**START**

**DATE:** 07/02/12  
**TIME:** 17:37  
**PAGE:** 1

---

```plaintext
7 ****************************************************************************** **
7 * **IF NO VALID ENTRY IN **
7 * **OPTION VALIDATION **
7 * **BASE TABLE (OPTVAL) **
7 * **TRY TO GET ERROR TEXT **
12 * **IF 1 = 1 THEN MOVE '1' TO RETCODE **
12 * **ERROR TEXT FOUND **
17 * MOVE '0700' TO MSGCODE **
17 * CALL 'DSNMC4' USING MAJOR MSGCODE OUTMSG **
17 * MOVE OUTMSG TO MSGTEXT IN MSG **
7 * **RETURN **
17 * GO TO END-DSNMC4. **
7 * **IF ONLY ONE OBJECT EXISTS THEN USE IT AS DEFAULT **
7 * **SET UP OBJECT AND DESCRIPTION IN OUTPUT **
12 * **IF 2 AND OBJECT IN INAREA = * * THEN **
15 * MOVE '0' TO RETCODE **
15 * MOVE FIELD-2111 TO OBJECT IN INAREA **
15 * MOVE FIELD-3111 TO DESC3 IN OUTAREA **
15 * MOVE SPACE * TO LINE011 **
7 * **RETURN **
15 * GO TO END-DSNMC4. **
7 * **OBJECT WAS NOT FOUND **
7 * **RETURN **
12 * MOVE '1' TO RETCODE. **
12 * EXEC SQL SELECT * **
17 * INTO 2POPTVAL **
17 * FROM OPTVAL **
17 * WHERE MAJSYS = :INAREA,MAJSYS **
17 * AND ACTION = :INAREA,ACTION **
17 * AND OBJECT = * * **
17 * END-EXEC. **
7 * **FILL IN DISPLAY AREA **
7 * **WITH HEADING, PKKEY **
7 * **AND MESSAGE INFO. **
12 * MOVE HEARTX IN POPTVAL TO HTITLE IN OUTAREA. **
12 * MOVE INFOTX IN POPTVAL TO MSG IN OUTAREA. **
12 * MOVE PKTXT IN POPTVAL TO PKTEXT IN OUTAREA. **
7 * **RETURN TO **
```

---

**END**
*******DSNMC3 - VALIDATION MODULE FOR ACTION - COBOL *******

MODULE NAME = DSNMC3

DESCRIPTIVE NAME = DBZ SAMPLE APPLICATION

VALIDATION MODULE FOR ACTION - COBOL

COPYRIGHT = 5740-XVR 1C1 COPYRIGHT IBM CORP 1982, 1985

REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G1Z0-2033

STATUS = RELEASE 2, LEVEL 0

FUNCTION = THIS MODULE VALIDATES SPECIFIC INPUT AND MOVES IT TO THE OUTPUT MESSAGE TOGETHER WITH A TEXT FIELD.

NOTES = NONE

MODULE TYPE = BLOCK OF COBOL CODE

PROCESSOR = DBZ PRECOMPILER, COBOL COMPILER

MODULE SIZE = SEE LINKEDIT

ATTRIBUTES = REUSABLE

ENTRY POINT = DSNMC3

PURPOSE = SEE FUNCTION

LINKAGE = INCLUDED BY MODULE DSNBCC1

INPUT = PARAMETERS EXPICITLY PASSED TO THIS FUNCTION:

SYMBOLIC LABEL/NAMES = NONE

DESCRIPTION = NONE

OUTPUT = PARAMETERS EXPICITLY RETURNED:

SYMBOLIC LABEL/NAMES = NONE

DESCRIPTION = NONE

EXIT-NORMAL = THIS CODE IS "PERFORMED", SO IT EXITS TO CODE FOLLOWING THE "PERFORM" STATEMENT

EXIT-ERROR = IF SOLERR OR SOLWARNING, SQL WHENEVER CONDITION SPECIFIED IN DSNACCI/1C1 WILL BE RAISED, AND PROGRAM WILL GO TO THE LABEL DB-ERROR.

RETURN CODE = NONE

ABEND CODES = NONE

ERROR MESSAGES = 

DSN070 - VITAL DATA IS MISSING IN TABLE TOPTVAL

EXTERNAL REFERENCES = MOST VARIABLES ARE GLOBAL AND DEFINED IN DSNACCI/1C1.
procedure initialize returncode to 0.

fill in the display area from voptval (action, seltxt)

depending on required major and action

return.

if action not found retrieve a list of action which exists.

headtxt, infotxt and pkftxt fill in display area

depending on major = major and action = blank

set returncode to *!.

end.

dsnmc3.

************************************************************

** initialize return code

lemma.

move *0* to retcode.

move *dsnmc3* to major in dsna-module-name.

************************************************************

** let's see if the action specified on input exists

** by trying to retrieve action and text.

************************************************************

exec sql select seltxt

into :voptval:seltxt

from voptval

where major = :inarea:major

and action = :inarea:action

and action = **
DATE: 07/02/12
TIME: 17:36

START

MOVE OBJET = '•' TO END-EXEC.

**ACTION EXISTS**
**FILL IN DISPLAY AREA**

MOVE ACTION IN ENAREA TO ACTION IN OUTAREA.

MOVE SELTXT IN POPTVAL TO DES2 IN OUTAREA.

**RETURN**

IF SOLCODE = +0 THEN
GO TO END-DSN8MC3.

**ACTION NOT FOUND**

**PROVIDE A LIST OF ACTIONS WHICH EXIST**

MOVE SPACE TO ACTION IN OUTAREA.

MOVE SPACE TO DES2 IN OUTAREA.

EXEC SQL OPEN V02 END-EXEC. **OPEN CURSOR**

MOVE +1 TO I. **RETRIEVE LIST OF ACTIONS**

EXEC SQL OPEN V02 END-EXEC.

IF SOLCODE = +100 THEN
GO TO END-DSN8MC3.

MOVE SPACE TO ACTION IN OUTAREA.

EXEC SQL CLOSE V02 END-EXEC. **CLOSE CURSOR**

MOVE I TO J. **PUT BLANKS AT END OF LINE**

**NO VALID ENTRY**
**OPTION VALIDATION**
**BASE TABLE (OPTVAL)**

END-EXEC.
**TRY TO GET ERROR TEXT**

```
7 12 IF I = 1 THEN MOVE '1' TO RETCODE
22   MOVE '070E' TO MSGCODE
22   CALL 'DSNEMCG' USING MAJOR MSGCODE OUTMSG
22   MOVE OUTMSG TO MSGTEXT IN MSG
   **RETURN**
```

```
7 22 GO TO END-DSNEMCS.
7  **RETURN**
7 ** IF ONLY THE ACTION EXISTS THEN USE AS DEFAULT**
7 ** SET UP ACTION AND DESCRIPTION IN OUTPUT**
```

```
12 IF I = 2 AND ACTION IN INAREA = ' ' THEN
15   MOVE '0' TO RETCODE
15   MOVE FIELD-2111 TO ACTION IN INAREA
15   MOVE FIELD-5111 TO DESC2 IN OUTAREA
15   MOVE SPACE TO LINENO (1)
15 **RETURN**
15 GO TO END-DSNEMCS.
```

```
7 ** ACTION WAS NOT FOUND**
7 ** ACTION WAS NOT FOUND**
```

```
12 MOVE '1' TO RETCODE.
12 EXEC SQL SELECT *
17   FROM VOPTVAL
17   WHERE MAJSYS = :INAREA,MAJSYS
17   AND ACTION = ' ';
17   END-EXEC.
```

```
7 ** FILL DISPLAY AREA**
7 ** FILL DISPLAY AREA**
7 ** FILL DISPLAY AREA**
```

```
15 MOVE HEADTXT IN POPVAL TO HTITLE IN OUTAREA.
15 MOVE INFOTXT IN POPVAL TO MSG IN OUTAREA.
15 MOVE PFKTXT IN POPVAL TO PFTEXT IN OUTAREA.
```

```
7 **RETURN TO**
7 **RETURN TO**
7 **RETURN TO**
```

```
**END-DSNEMCS.**
```
### Module: DSN8CCO

**Subsystem Interface Module for CICS/VS - COBOL**

<table>
<thead>
<tr>
<th>COL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Module Name</strong> = DSN8CCO</td>
</tr>
<tr>
<td>2</td>
<td><strong>Descriptive Name</strong> = DB2 Sample Application</td>
</tr>
<tr>
<td>3</td>
<td><strong>Subsystem Interface Module</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>CICS</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>COBOL</strong></td>
</tr>
<tr>
<td>6</td>
<td><strong>Copyright</strong> = 5740-YVR (C) Copyright IBM Corp. 1982, 1985</td>
</tr>
<tr>
<td>7</td>
<td><strong>Refer to Copyright Instructions Form Number 6120-2083</strong></td>
</tr>
<tr>
<td>8</td>
<td><strong>Status</strong> = Release 2, Level 0</td>
</tr>
<tr>
<td>9</td>
<td><strong>Function</strong> = This module issues CICS receive map to retrieve input, calls OSMCCS, and issue CICS send map after returning.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Notes</strong> = 1. This is a CICS pseudo conversation program which initializes itself when a terminal operator enters input after viewing the screen sent by previous iterations of the program.</td>
</tr>
<tr>
<td>11</td>
<td><strong>Dependencies</strong> = Two CICS maps (OSMCECS) are required: OSMRMC6 and OSMRMC0</td>
</tr>
<tr>
<td>12</td>
<td><strong>Module OSMCCS is required.</strong></td>
</tr>
<tr>
<td>13</td>
<td><strong>Included COBOL structure OSMRMC6 is required.</strong></td>
</tr>
<tr>
<td>14</td>
<td><strong>Restrictions</strong> = None</td>
</tr>
<tr>
<td>15</td>
<td><strong>Module Type</strong> = COBOL</td>
</tr>
<tr>
<td>16</td>
<td><strong>Processors</strong> = DB2 Precompiler, CICS Translator, COBOL Compiler</td>
</tr>
<tr>
<td>17</td>
<td><strong>Module Size</strong> = See Link-Edit</td>
</tr>
<tr>
<td>18</td>
<td><strong>Attributes</strong> = Reusable</td>
</tr>
<tr>
<td>19</td>
<td><strong>Entry Point</strong> = DSN8CC0</td>
</tr>
<tr>
<td>20</td>
<td><strong>Purpose</strong> = See Function</td>
</tr>
<tr>
<td>21</td>
<td><strong>Linkage</strong> = CICS/OS/VS ENTRY</td>
</tr>
<tr>
<td>22</td>
<td><strong>Input</strong> = Parameters explicitly passed to this function:</td>
</tr>
<tr>
<td>23</td>
<td><strong>Symbolic Label/Name</strong> = OSM8CCGI</td>
</tr>
<tr>
<td>24</td>
<td><strong>Description</strong> = CICS/OS/VS BMS Map for General Input</td>
</tr>
<tr>
<td>25</td>
<td><strong>Symbolic Label/Name</strong> = OSM8CCCO</td>
</tr>
<tr>
<td>26</td>
<td><strong>Description</strong> = CICS/OS/VS BMS Map for Detail Input</td>
</tr>
<tr>
<td>27</td>
<td><strong>Output</strong> = Parameters explicitly returned:</td>
</tr>
</tbody>
</table>

---

**Notes:**

- This is a CICS pseudo conversation program which initializes itself when a terminal operator enters input after viewing the screen sent by previous iterations of the program.
- Included COBOL structure OSMRMC6 is required.
- Restrictions = None

**Input:** Parameters explicitly passed to this function:

- Symbolic Label/Name = OSM8CCGI
- Description = CICS/OS/VS BMS Map for General Input

**Output:** Parameters explicitly returned:

- Symbolic Label/Name = OSM8CCCO
- Description = CICS/OS/VS BMS Map for Detail Input
SYMBOLIC LABEL/NAME = DSN8CC0
DESCRIPTION = CICS/OS/VS RRS MAP FOR GENERAL OUTPUT 00550000

EXIT-NORMAL = CICS RETURN TRANSID 00610000
EXIT-ERROR = SQL ERROR FOR SQL ERRORS 00640000
CICS ABEND FOR CICS PROBLEMS 00650000
RETURN CODE = NONE 00660000
ABEND CODES = LSCR - LOGICAL SCREEN SET INCORRECTLY 00660000
ERROR-MESSAGES = NONE 00700000
EXTERNAL REFERENCES = COMMON CICS REQUIREMENTS 00730000
ROUTINES/SERVICES = 00740000
CICS/VS SERVICES = SQL I MAINLINE CODE 00760000
DSN8CC1 - DSN8CC1 - COMMON AREA 00770000
DATA AREAS = DSNRMCCA - PARAMETER TO BE PASSED TO DSN8CC1 00800000
DSNRMCS - DECLARE CONVERSATION STATUS 00820000
DSNRMCC2 - COMMON AREA PART 2 00830000
DSNRMCMO - CICS/OS/VS COBOL MAP, ORGANIZATION 00840000
DSNRMCMG - CICS/OS/VS COBOL MAP, ORGANIZATION 00850000
CONTROL BLOCKS = SQLCA - SQL COMMUNICATION AREA 00880000
TABLES - NONE 00900000
CHANGE-ACTIVITY - NONE 00910000
PSEUDOCODE = PROCEDURE
DECLAREATIONS,
ALLOCATE COBOL WORK AREA FOR CMMAREA.
PUT MODULE NAME "DSN8CC0" IN AREA USED BY ERROR-HANDLER.
PUT CICS EIRTRANID IN PCONSTA.CONVID TO BE PASSED TO DSN8CC1.
RETRIEVE LASTCR FROM VCONA USING THE CONVID TO DETERMINE WHICH OF THE TWO RRS MAPS SHOULD BE USED TO MAP DATA.
IF RETRIEVAL OF MAPS IS SUCCESSFUL, THEN DO
EXEC CICS RECEIVE MAP ACCORDING TO SPECIFIED LASTCR. 01050000
3ATE: 37/02/12
riW: 17:17
PAGE: 3

PFAIL
CONDITION IS RAISED.

CONPAR.H.Pf
«

SO
to COSENO 010000
01100000
01110000
01120000
01150000
01140000

PUT DATA FROM MAP INTO CONPAR

ELSE
IT IS A NEW CONVERSATION
AND NO EXEC CICS RECEIVE MAP IS ISSUED.

ELSE
EXEC CICS SEND MAP ACCORDING TO THE TYPE SPECIFIED IN MCONVSTA.LASTSCR.
EXEC CICS RETURN TRANSID(ORCS).

END.

I.E. LAST CONVERSATION EXISTS, DUE OPERATOR HAD ENTERED DATA FROM A CLEARED SCREEN OR HAD ERASED ALL DATA ON A FORMATTED SCREEN AND PRESSED ENTER.

** CONPAR.PFKIN = PF KEY ACTUALLY USED I.E. '01' FOR PF1 **

IDENTIFICATION DIVISION.
PROGRAM-ID. DSN8CCO.
ENVIRONMENT DIVISION.
WORKING-STORAGE SECTION.
77 FOUND PIC S99.
EXEC SQL INCLUDE SOLCA END-EXEC.
EXEC SQL INCLUDE DSN8CC2 END-EXEC.
EXEC SQL INCLUDE DSNMCC2 END-EXEC.
EXEC SQL INCLUDE DSNMCC END-EXEC.
EXEC SQL INCLUDE DSNMCCG END-EXEC.
EXEC SQL INCLUDE DSNMCCG END-EXEC.

MAPD REDEFINES THE COBOL STRUCTURE ASSOCIATED WITH THE

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
02 FILLER PIC X(387).

01 MAPD REDEFINES DSN8CCO,
PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREVER SQLWARNING GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES

    EXEC SQL SELECT LASTSCR INTO I-PCONA.
    LAST SCREEN SPECIFIED TO RECEIVE INPUT FROM TERMINAL.

    EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLWARNING GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.

**********

EXEC SQL WHENEVER SQLERROR GO TO DS-ERROR END-EXEC.

PROCEDURE DIVISION.

* ************************************************************
* ** EXEC SQL WHEREEVER SQLERROR GO TO DS-ERROR END-EXEC. **
* ************************************************************

    IF LAST CONVERSATION DOES NOT EXIST, THEN DO NOT ATTEMPT TO

* RECEIVE INPUT MAP. GO DIRECTLY TO VALIDATION MODULES.
IF SOLCODE = *100 THEN
   GO TO CCOSEND.

EXEC CICS HANDLE CONDITION MAPFAIL (CCOSEND) END-EXEC.

EXEC CICS RECEIVE MAP ('DSN8BBCG') MAPSET ('DSN8BCCG')
   END-EXEC.

GO TO CCO-LABEL1.

EXEC CICS RECEIVE MAP ('DSN8BBCG') MAPSET ('DSN8BCCG')
   END-EXEC.

EXEC CICS RECEIVE MAP ('DSN8BBCG') MAPSET ('DSN8BCCG')
   END-EXEC.

GO TO CCO-LABELX.

ERROR ON LAST SCREEN

GO BACK.
**CICE** Send Spaces to INAREA.

MOVE '00' TO PFKIN OF INAREA.

EXEC CICS LINK PROGRAM (*DSN8MCXX*)

**CCO-NORMAL.**

IF LASTSCR OF PCONSTA = '03002' THEN GO TO **CCO-LABEL9.**

MOVE HTITLE OF OUTAREA TO ATITLEO.

MOVE NMSYS OF OUTAREA TO ANMSYSO.

MOVE ACTION OF OUTAREA TO ACTIUNO.

MOVE OBJECT OF OUTAREA TO AOBJCTO.

MOVE SRCH OF OUTAREA TO ASARCHO.

MOVE DATOUT TO AOUTAO.

MOVE MSG OF OUTAREA TO AASCO.

MOVE DESC2 OF OUTAREA TO ADESC2O.

MOVE DESC3 OF OUTAREA TO ADESC3O.

MOVE DESC4 OF OUTAREA TO ADESC4O.

MOVE PFTEXT OF OUTAREA TO APFKEYO.

MOVE 1 TO I.

**CCO-LABEL7.**

**CCO-LABEL7 LOOP**

**PREVIOUS SCREEN**

**SEND MAP ACCORDING TO**

**OUTPUT DATA INTO**

**C IC-Lo0GFi.**

**CCO-LABEL7 LOOP**

PERFORM **CCO-LABEL7 UNTIL**

**SET CURSOR POSITION**
**START**

**COLE**:

12  "MOVE ZER0ES TO CURSOR-VALUE".
12  IF AXTR54 = SPACES THEN MOVE +177 TO CURSOR-VALUE.
12  ELSE IF AcR5JSCTO = SPACES THEN MOVE +219 TO CURSOR-VALUE.
12  ELSE IF Ac3EARCH9 = SPACES THEN MOVE +339 TO CURSOR-VALUE.
12  ELSE IF 2ADATA = SPACES OR AxACTION = "0" OR "E" THEN
12  MOVE +419 TO CURSOR-VALUE.

7  **END**

12  "**SEND OUTPUT MAP**

12  **FINISHED?**
12  IF EXITCODE = **'1' THEN GO TO **COO-LABEL2.

**EXEC** CICS SEND MAP(*S3NRCG*) MAPSET(*S3NRCG*) END-EXEC.

**GENERAL**

12  "**MOVES DATA FROM OUTPUT MAP AREA TO **RECEIVE** CAP ACCORDING TO MAP SPECIFIED IN LAST**C60** OF PCOO**V**T**

**COPY**

12  **RECEIVE** CAP ACCORDING TO MAP SPECIFIED IN LAST**C60** OF PCOO**V**T**

**EXEC** CICS RETURN TRANSID('DESC5') END-EXEC.

**COO-LABEL2**

12  "MOVE TITLE OF OUTAREA TO BTITLECO.
12  MOVE MAJSYS OF OUTAREA TO BMAJSYS.

12  "MOVE ACTION OF OUTAREA TO BACTIONIO.
12  MOVE OBJECT OF OUTAREA TO BOBJECTJ.
12  "MOVE SEARCH OF OUTAREA TO 9SEARCH9.

12  "MOVE Dateout TO 9DATAA.
12  "MOVE HSG OF OUTAREA TO BHSO.
12  MOVE DESC2 OF OUTAREA TO 2DE5CL0.
12  MOVE DESC3 OF OUTAREA TO 3DE5CL0.
12  "MOVE P**FTEXT** OF OUTAREA TO 3PP**K**T**
12  "MOVE 1 TO 1.

**RECEIVE** CAP ACCORDING TO MAP SPECIFIED IN LAST**C60** OF PCOO**V**T**

**EXEC** CICS RETURN TRANSID('DESC5') END-EXEC.

**END**

7  **RECEIVE** CAP ACCORDING TO MAP SPECIFIED IN LAST**C60** OF PCOO**V**T**

7  "**TO PREVIOUS** SCREEN.

**END**

7  **COO-LABEL8**

12  MOVE FIELD111 TO COLDATA111.

7  "**CHECK** FOR ATTRIBUTE OF XFC6G1.
12  IF ATTR111 = -16191 THEN MOVE -1 TO COLDLEN111.

7  "**COO-L4P**
12  **COO-L4P**.

12  PERFORM COO-LABEL10 UNTIL
SET CURSOR POSITION

**SET CURSOR POSITION**

MOVE ZEROS TO CURSOR-VALUE.

ELSE IF BACTION = SPACES THEN MOVE 179 TO CURSOR-VALUE

ELSE IF BACTION = SPACES THEN MOVE 259 TO CURSOR-VALUE

ELSE IF BACTION = 'D' OR 'E' THEN

MOVE 419 TO CURSOR-VALUE.

ELSEIF CURSOR-VALUE = ZEROS THEN

**SEND INPUT MAP**

EXEC CICS SEND MAP('DSNACCD') MAPSET('DSNACCD') END-EXEC

EXEC CICS SEND MAP('DSNACCD') MAPSET('DSNACCD') ERASE END-EXEC.

**FINISHED?**

EXEC CICS RETURN TRANSID('DBCS') END-EXEC.

**RETURN**

EXEC CICS RETURN END-EXEC.

GObACK.
EXEC SQL DECLARE VO2 CURSOR FOR
    SELECT ACTION, SELTXT
    FROM VOPTVAL
    WHERE MAJSYS = :INAREA.MAJSYS
    AND ACTION = :ACTION
    AND OBJECT = :OBJECT
    ORDER BY ACTION ASC
END-EXEC.

EXEC SQL DECLARE VO3 CURSOR FOR
    SELECT OBJECT, SELTXT
    FROM VOPTVAL
    WHERE MAJSYS = :INAREA.MAJSYS
    AND ACTION = :INAREA.ACTION
    AND OBJECT = :INAREA.OBJECT
    AND SRCHCRIT = :SRCHCRIT
    ORDER BY OBJECT ASC END-EXEC.

EXEC SQL DECLARE VO4 CURSOR FOR
    SELECT SRCHCRIT, SELTXT
    FROM VOPTVAL
    WHERE MAJSYS = :INAREA.MAJSYS
    AND ACTION = :INAREA.ACTION
    AND OBJECT = :INAREA.OBJECT
    AND SRCHCRIT = :SRCHCRIT
    AND (SCRTYPE = 'S' OR SCRTYPE = 'E')
    ORDER BY SRCHCRIT ASC END-EXEC.
THE CONTAINER STRUCTURE DECLARED BELOW IS USED TO PASS INPUT DATA BETWEEN THE SUBSYSTEM DEPENDENT MODULES (CICS, IMS, TSO).

PCONVSTA.

PCONVSTAO REDEFINES PCONVSTA PIC X.(32).

DATAOUT.

DATAOUT2 RED

OUTAREA.

ACTION PIC X.

OBJE

SRCH PIC X.

DATAOUT2 RED

DATOUT PIC X(12).

DATOUT PIC X(2).

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.

PCONVSTAO.
<table>
<thead>
<tr>
<th>START</th>
<th>COL</th>
<th>PIC X</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>6</td>
<td>FILLER</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>TITLE</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>DESC3</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>DESC4</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>MSG</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>SLAT</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>FILLER</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>MSGN0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>MSGN02</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>MSGTXT REDEFINES MSGTXT</td>
<td>PIC X(70).</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>FILLER</td>
<td>PIC X(9).</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>PICTXT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>OUTPUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>LINE1 OCCURS</td>
<td>15.</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>LINE0</td>
<td>PIC X(79).</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>FIELD-1</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>FIELD-2</td>
<td>PIC X(16).</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>FIELD-3</td>
<td>PIC X(170).</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>FIELD-2</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>FIELD1</td>
<td>PIC X(37).</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>ATTR</td>
<td>PIC S(64) COMP.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>ATTR1</td>
<td>REDEFINES ATTR.</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>OUTPUT</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>FIELD</td>
<td>PIC X.</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>FIELD2</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>FIELD2</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>FIELD2</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>FIELD2</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>FIELD2</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>FIELD2</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>FIELD</td>
<td>PIC X.</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>X11</td>
<td>PIC X(21).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X10A</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>FILLER</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>GNU</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X20</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>XK10</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X110</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X12</td>
<td>PIC X(34).</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>RIGHT-DPT</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>X12</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X120</td>
<td>PIC X(33).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X13</td>
<td>PIC X(33).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X130</td>
<td>PIC X(33).</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>RIGHT-MGR</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>X23</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X24</td>
<td>PIC X.</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X20</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X25</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X250</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>RIGHT-EMP</td>
<td>REDEFINES LINEO.</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X14</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X26</td>
<td>PIC X(40).</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>X15</td>
<td>PIC X(40).</td>
</tr>
</tbody>
</table>
**DATASET:** US#123
**MEMBER:** DB2CC

<table>
<thead>
<tr>
<th>COL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>DSN8CC2 - COMM MODULE FOR CICS - COBOL</td>
<td>00813000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MODULE NAME = DSN8CC2</td>
<td>00020000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DESCRIPTIVE NAME = DB2 SAMPLE APPLICATION</td>
<td>00040000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CICS</td>
<td>00050000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CALLS DETAIL MODULES</td>
<td>00060000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>COPYRIGHT = 3740-XYR (C) COPYRIGHT IBM CORP 1982, 1985</td>
<td>00100000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2029</td>
<td>00110000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>STATUS = RELEASE 2 PLUS LEVEL 0</td>
<td>00130000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>FUNCTION = ROUTER FOR SECONDARY SELECTION AND/OR DETAIL PROCESSING</td>
<td>00160000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CALLS SECONDARY SELECTION MODULES</td>
<td>00170000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8CM DSN8CM</td>
<td>00190000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CALLS DETAIL MODULES</td>
<td>00200000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8CD DSN8CM DSN8MCF</td>
<td>00210000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC DSN8MCM DSN8MCM DSN8MC</td>
<td>00220000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CALLED BY DSN8MC1 (SOL1)</td>
<td>00230000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MUTES = NONE</td>
<td>00240000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MODULE TYPE =</td>
<td>00250000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PROCESSOR = DB2 PRECOMPILER, CICS TRANSLATOR</td>
<td>00260000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>VS COBOL</td>
<td>00270000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>MODULE SIZE = SEE LINKEDIT</td>
<td>00280000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ATTRIBUTES = REUSABLE</td>
<td>00290000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ENTRY POINT = DSN8CC2</td>
<td>00310000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PURPOSE = SEE FUNCTION</td>
<td>00320000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>LINKAGE = NONE</td>
<td>00330000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>INPUT</td>
<td>00340000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SYMATIC LABEL/NAME = COBOL</td>
<td>00350000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DESCRIPTION = POINTER TO COMMAREA</td>
<td>00360000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(COMMUNICATION AREA)</td>
<td>00370000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SYMATIC LABEL/NAME = COBOL</td>
<td>00380000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DESCRIPTION = POINTER TO COMMAREA</td>
<td>00390000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(COMMUNICATION AREA)</td>
<td>00400000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SYMATIC LABEL/NAME = COBOL</td>
<td>00410000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DESCRIPTION = POINTER TO COMMAREA</td>
<td>00420000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(COMMUNICATION AREA)</td>
<td>00430000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SYMATIC LABEL/NAME = COBOL</td>
<td>00440000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DESCRIPTION = POINTER TO COMMAREA</td>
<td>00450000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(COMMUNICATION AREA)</td>
<td>00460000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>EXIT-NORMAL = RETURN CODE 0 NORMAL COMPLETION</td>
<td>00470000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>EXIT-ERROR</td>
<td>00480000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SPECIFIED IN DSN8CC2 WILL BE RAISED AND PROGRAM</td>
<td>00490000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>WILL GO TO THE LABEL DB-ERROR</td>
<td>00500000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7-71
PREVIOUS: | NEXT:

```
<table>
<thead>
<tr>
<th>START COL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>RETURN CODE = NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ABEND CODES = NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ERROR MESSAGES =</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8072E-INVALID SELECT ON SECONDARY SCREEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>EXTERNAL REFERENCES = NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ROUTINES/SERVICES = 10 MODULES LISTED ABOVE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8NCO - ERROR MESSAGE ROUTINE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DATA AREAS =</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8NCA - SECONDARY SELECTION FOR ORGANIZATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8NCAE - DECLARE ADMINISTRATION DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8NCAF - DECLARE DEPARTMENT MANAGER</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8NCC2 - COMMON AREA PART 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0 - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0A - DECLARE ADMINISTRATION DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0B - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0C - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0D - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0E - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0F - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0G - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0H - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0I - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0J - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0K - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0L - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0M - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0N - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0O - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0P - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0Q - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0R - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0S - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0T - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0U - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0V - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0W - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>DSN8MC0X - DECLARE DEPARTMENT DETAIL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CONTROL BLOCKS =</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>SQLCA - SQL COMMUNICATION AREA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TABLES = NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CHANGE ACTIVITY = NONE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><em>PSEUDOCODE</em></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>THIS MODULE DETERMINES WHICH SECONDARY SELECTION AND/OR DETAIL MODULE(S) ARE TO BE CALLED IN THE CICS/COBOL ENVIRONMENT.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>WHAT HAS HAPPENED SO FAR? THE SUBSYSTEM DEPENDENT MODULE (IMS,CICS,TSO) OR (SQL) HAS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

7-72
THE INPUT SCREEN FORMATTED THE INPUT AND PASSED CONTROL TO THIS MODULE. PASSED PARAMETERS CONSIST ONLY OF A POINTER WHICH POINTS TO A COMMUNICATION CONTROL AREA USED TO COMMUNICATE BETWEEN SOL 1 AND SOL 2 AND THE SECONDARY SELECTION MODULE. ALL SYSTEM FIELDS ARE VALID SOL 1 PASSED CONTROL TO THIS MODULE. ALL VARIABLES KNOWN IN THIS PROCEDURE ARE KNOWN IN THE SQL CURSOR DEFINITIONS AND SQL INCLUDES ARE DONE IN THIS PROCEDURE. THE CURSOR DEFINITION ALL CURSOR HOST VARIABLES ARE DECLARED BECAUSE OF THE restriction that CURSOR HOST VARIABLES MUST BE DECLARED BEFORE THE CURSOR DEFINITION ALL CURSOR HOST VARIABLES ARE DECLARED.

IF ANSWER TO DETAIL SCREEN AND DETAIL PROCESOR IS NOT WILLING TO ACCEPT AN ANSWER THEN NEW REQUEST.

ELSE IF ANSWER TO A SECONDARY SELECTION THEN DETERMINE IF NEW REQUEST.

CASE (NEW REQUEST)

SUBLACE ('ADD') DETAIL PROCESSOR RETURN TO SOL 1

SUBLACE ('ERASE', 'DISPLAY', 'UPDATE') CALL SECONDARY SELECTION IF # OF POSSIBLE CHOICES IS = 1 THEN RETURN TO SOL 1

ELSE CALL THE DETAIL PROCESSOR RETURN TO SOL 1

ENDSUBLACE

IF ANSWER TO SECONDARY SELECTION AND A SELECTION HAS ACTUALLY BEEN MADE THEN IF IS, IS A VALID SELECTION NUMBER THEN CALL DETAIL PROCESSOR

END RETURN TO SOL 1
**ELSE** PRINT ERROR MSG RETURN TO SOL 1

**IF ANSWER TO SECONDARY SELECTION THEN** CALL SECONDARY SELECTION RETURN TO SOL 1 END.

**IF ANSWER TO DETAIL THEN** CALL DETAIL PROCESSOR RETURN TO SOL 1 END.

RETURN TO SOL 1.

**EXAMPLE- A ROW IS SUCCESSFULLY ADDED, THE OPERATOR RECEIVES**

THE SUCCESSFULLY ADDED MESSAGE AND JUST HITS ENTER.

**IDENTIFICATION DIVISION.**

**ENVIRONMENT DIVISION.**

**DATA DIVISION.**

**WORKING-STORAGE SECTION.**

**F i e l d s \ s e n t t o \ m e s s a g e \ r u n t i n e .**

**O 1 \ m s g c o d e \ p i c \ x ( 6 9 ) .**

**O 1 \ o u t m s g \ p i c \ x ( 6 9 ) .**

**L i n k a g e \ s e c t i o n .**
EXEC SQL INCLUDE DSN8MSCA END-EXEC.
EXEC SQL INCLUDE DSNHCAE END-EXEC.
EXEC SQL INCLUDE DSN8MCC-H END-EXEC.
EXEC SQL INCLUDE DSN8HCDA END-EXEC.

DATE: 87/02/12
TIME: 17:18

PROCEDURE DIVISION.

EXEC SQL INCLUDE DSN8MCCA END-EXEC.
EXEC SQL INCLUDE DSN8MCAE END-EXEC.
EXEC SQL INCLUDE DSN8MCAL END-EXEC.
EXEC SQL INCLUDE DSN8MCC-H END-EXEC.
EXEC SQL INCLUDE DSN8HCDA END-EXEC.

INITIALIZATIONS

IF NEWREQ OF COMPARM = 'N' AND DATA01 NOT = 'Y' AND DATA01 NOT = 'NEXT' THEN MOVE 'V' TO NEWREQ OF COMPARM.
IF NEWREQ OF COMPARM NOT = 'V' THEN GO TO IC2000.
IF PREV OF PCONVSTA = ' ' THEN MOVE 'V' TO NEWREQ OF COMPARM.
IF NEWRQ OF COMPARM = 'N' AND PREV OF PCONVSTA = 'S' THEN MOVE 'V' TO NEWRQ OF COMPARM.
IF NEWRQ OF COMPARM NOT = 'V' THEN GO TO IC2000.

IF ACTION OF INAREA = 'A' THEN **DETAIL PROCESSOR
IF ACTION OF INAREA = 'B' THEN **SECONDARY SELECTION
IF ACTIONS = 1 **GO TO DETAIL PROCESSOR
PERFORM SECSSEL THRU END-SECSSEL.

GO TO DETAILO.
GO TO DETAILO.
GO TO DETAILO.
GO TO EXITO.

DETERMINES WHETHER NEW REQUEST OR NOT

DETERMINES WHETHER NEW REQUEST OR NOT
DETERMINES WHETHER NEW REQUEST OR NOT
DETERMINES WHETHER NEW REQUEST OR NOT

DETERMINES WHETHER NEW REQUEST OR NOT
DETERMINES WHETHER NEW REQUEST OR NOT
DETERMINES WHETHER NEW REQUEST OR NOT
DETERMINES WHETHER NEW REQUEST OR NOT
**DETERMINES IF VALID SELECTION NUMBER**

**IF PREV OF PCONVST = '5' OR**

**MAXSEL < 1 OR**

**DATA1 = 'NEXT' OR**

**DATA2 = DATA2 TYPN**

**GO TO IC201.**

**DETERMINES SECONDARY SELECTION OR DETAIL**

**IF PREV OF PCONVST = THE* AND DATA2 NOT > MAXSEL THEN**

**MOVE 'Y' TO NEWREQ OF CONPARM**

**GO TO DETAIL.**

**DETERMINES WHETHER SECONDARY SELECTION OR DETAIL**

**SECONDARY SELECTION**

**IF PREV OF PCONVST = '5' THEN**

**PERFORM SECSEL THRU END-SECSEL**

**GO TO EXIT.**

**ELSE**

**PRINT ERROR MESSAGE**

**CALL 'DSN8MCG' USING MAJOR MSGCODE OUTMSG.**

**MOVE OUTMSG TO MS6 OF OUTAREA.**

**GO TO EXIT.**

**DETAIL PROCESSOR**

**IF PREV OF PCONVST = '9' THEN GO TO DETAIL.**

**PRINT ERROR MESSAGE**

**CALL 'DSN8MCG' USING MAJOR MSGCODE OUTMSG.**

**MOVE OUTMSG TO MS6 OF OUTAREA.**

**GO TO EXIT.**

**EXEC SQL INCLUDE DSNAMXX END-EXEC.**

**HANDLES ERRORS**

**CALLS SECONDARY SELECTION AND RETURNS TO SQL 1**

**SECONDARY SELECTION**

**MOVE 'DSNR001' TO LASTSCR IN PCONVST.**
START

EXEC CICS RETURN END-EXEC.

EXEC SQL INCLUDE DSN8MCA END-EXEC.

EXEC SQL INCLUDE DSN8MCO END-EXEC.

EXEC SQL INCLUDE DSN8MCE END-EXEC.

EXEC SQL INCLUDE DSN8MCN END-EXEC.

GOBACK.

DATE: 87/02/12
TIME: 17:13

DATASET: DSN8MCO
MEMBER: DSN8MCO2

PAGE: 3

PAGE: 7-78
### DSN8MC5

**Module Name:** DSN8MC5

**Purpose:** This module validates specific input and moves it to the output message together with a text field.

**Input Parameters Explicitly Passed to This Function:**
- Parameters: None
- Symbolic Label/Name: None
- Description: None

**Output Parameters Explicitly Returned:**
- Parameters: None
- Symbolic Label/Name: None
- Description: None

**Exit-Error Statement:**
- Exit-Error: If SQLERROR or SQLWARNING, SQL whenever
- Condition Specified in DSN8CC1/II WILL BE RAISED AND PROGRAM WILL GO TO THE LABEL
- DB-ERROR:
- Return Code: None
- Abend Codes: None
- Error Messages:
  - DSN5070E - Vital data missing in table 'TOPTVAL'
  - DSN5070E - Vital data missing in table 'TOPTVAL'
  - DSN5070E - Vital data missing in table 'TOPTVAL'
  - DSN5070E - Vital data missing in table 'TOPTVAL'

**Module Attributes:**
- Attributes: None

**Entry Point:** DSN8MC5

**Linkage:** Included by module DSN8CC1

**Input Parameters:**
- Parameters: Explicitly passed to this function.
- Symbolic label/name: None
- Description: None

**Output Parameters:**
- Parameters: Explicitly returned.
- Symbolic label/name: None
- Description: None

**Exit-Normal:**
- This code is "PERFORMED", so it exits to the code following the "PERFORM" statement
- Return Code: None
- Abend Codes: None

**Error Messages:**
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'

**COPYRIGHT:**
- Copyright IBM Corp. 1982, 1985
- Refer to Copyright instructions form number G120-2083

**Status:**
- Status: Release 2.1, Level 0

**Language:**
- Language: COBOL

**Module Size:**
- Size: See Linkedit

**Attributes:**
- Attributes: None

**Linkage:**
- Linkage: Included by module DSN8CC1

**Input Parameters:**
- Parameters: Explicitly passed to this function.
- Symbolic label/name: None
- Description: None

**Output Parameters:**
- Parameters: Explicitly returned.
- Symbolic label/name: None
- Description: None

**Exit-Normal:**
- This code is "PERFORMED", so it exits to the code following the "PERFORM" statement
- Return Code: None
- Abend Codes: None

**Error Messages:**
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'

**Module Type:**
- Processor: D82 Precompiler, COBOL Compiler

**Module Size:**
- Size: See Linkedit

**Attributes:**
- Attributes: None

**ENTRY POINT:**
- Entry point: DSN8MC5

**Purpose:**
- Purpose: See function

**Linkage:**
- Linkage: Included by module DSN8CC1

**Input Parameters:**
- Parameters: Explicitly passed to this function.
- Symbolic label/name: None
- Description: None

**Output Parameters:**
- Parameters: Explicitly returned.
- Symbolic label/name: None
- Description: None

**Exit-Normal:**
- This code is "PERFORMED", so it exits to the code following the "PERFORM" statement
- Return Code: None
- Abend Codes: None

**Error Messages:**
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'

**COPYRIGHT:**
- Copyright IBM Corp. 1982, 1985
- Refer to Copyright instructions form number G120-2083

**Status:**
- Status: Release 2.1, Level 0

**Language:**
- Language: COBOL

**Module Size:**
- Size: See Linkedit

**Attributes:**
- Attributes: None

**ENTRY POINT:**
- Entry point: DSN8MC5

**Purpose:**
- Purpose: See function

**Linkage:**
- Linkage: Included by module DSN8CC1

**Input Parameters:**
- Parameters: Explicitly passed to this function.
- Symbolic label/name: None
- Description: None

**Output Parameters:**
- Parameters: Explicitly returned.
- Symbolic label/name: None
- Description: None

**Exit-Normal:**
- This code is "PERFORMED", so it exits to the code following the "PERFORM" statement
- Return Code: None
- Abend Codes: None

**Error Messages:**
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
- DSN5070E - Vital data missing in table 'TOPTVAL'
EXTERNAL REFERENCES = *ST Variables are global and defined in DSN4CC1/1C1.*

ROUTES/SERVICES = DSN3MC

DATA-AREAS = NONE

CONTROL-BLOCKS = SQLCA

SQLCA = SQL Communication area

TABLES = NONE

CHANGE-ACTIVITY = NONE

**PSEUDOCODE**

PROCEDURE

**INITIALIZE** RETURN CODE TO '0'.

**FILL IN THE DISPLAY AREA** FROM VOPTVAL (SEARCH-SELIXT)

**DEPENDING ON SEARCH REQUIRED**

RETURN.

IF SEARCH CRITERIA NOT FOUND

**RETRIEVE A LIST OF SEARCH CRITERIA WHICH EXISTS,**

HEADTXT, INFOTXT AND PFK.TXT

**DEPENDING ON MAJSYS = MAJSYS* ACTION = ACTION;**

OBJECT = OBJECT AND SEARCH = BLANK

**FILL IN DISPLAY AREA**

**SET RETURN CODE TO '1'.**

END.

**INPUT EXISTS BY TRYING TO RETRIEVE SEARCH CRITERIA**

**AND TEXT**

**RETRIEVAL**

DATASET: DSN120.DSN4CRP
HOST: DSN4CRP

START

12 EXEC SQL SELECT SELTXT
17 INTO :POPTVAL, SELTXT
18 FROM DSN4CRP
19 WHERE AJSYS = 'INAREA:aJSYS'
20 AND ACTION = 'INAREA: ACTION'
21 AND OBJECT = 'INAREA:OBJECT'
22 AND SRCHCRT = 'INAREA:SRCHCRT'
23 AND SCRTYPE = ' ' OR SCRTYPE = 'S':
24 AND SRCHCRIT =: INAREA..14JSYS
25 AND ACTION =: INAREA.ACTION
26 AND OBJECT =: INAREA.OBJECT
27 AND SRCHCRT =: INAREA.SRCHCRT
28 AND SCRTYPE = ' ' OR SCRTYPE = 'S':
29 WHERE SRCHCRIT = INAREA..14JSYS
30 AND ACTION = INAREA.ACTION
31 AND OBJECT = INAREA.OBJECT
32 AND SRCHCRT = INAREA.SRCHCRT
33 AND SCRTYPE = ' ' OR SCRTYPE = 'S':
34 END-EXEC.

7 * **SEARCH CRITERIA EXIST**
7 * **FILL IN DISPLAY AREA**

12 MOVE SELTXT IN POPTVAL TO DESCP IN OUTAREA.
12 MOVE SRCH IN SRCHCRT IN OUTAREA.
7 * IF SQLCODE = 0 THEN
7 * GO TO END-DSN4CRP.

12 **SEARCH CRITERIA NOT FOUND**
12 **PROVIDE A LIST OF SEARCH CRITERIA WHICH EXIST**
12 **SEARCH CRITERIA NOT FOUND**
12 **PROVIDE A LIST OF SEARCH CRITERIA WHICH EXIST**
12 MOVE SPACE TO SRCH IN OUTAREA.
12 MOVE SPACE TO DESCP IN OUTAREA.
7 * EXEC SQL OPEN VOL4 END-EXEC.
7 * MOVE +1 TO I.

7 * **RETRIEVE LIST OF**
7 * **SEARCH CRITERIA**

8 MDC-10.
8 IF I NOT > IS THEN
8 EXEC SQL FETCH VOL4 INTO :POPTVAL,SRCHCRT ,
9 INTO :POPTVAL,SELTXT END-EXEC
16 IF SQLCODE IS NOT EQUAL TO +100 THEN
23 MOVE SPACES TO FIELD-111
23 MOVE SCRCHRT IN POPTVAL TO FIELD-211
23 MOVE SELTXT IN POPTVAL TO FIELD-311
23 ADD 1 TO I
23 GO TO MDC-10.
8 MDC-20.
8 **CLOSE CURSOR**

8 MDC-30.
8 **PUT BLANKS AT**
8 **END OF LINE**

8 MDC-30.
8 IF J NOT > IS THEN
17 MOVE SPACE TO LINE011J

PAGE: 3
IF NC1 VALID ENTRY 
OPTION VALIDATION **BASE TABLE (TOPTVAL) **TRY TO SET ERROR TEXT

MOVE '070E' TO MSGCODE 
CALL *DSN8NC6 USING MAJOR MSGCODE OUTFSG

MOVE OUTTEXT TO MSGTEXT IN MSG

GO TO END-DSNC6.

IF 1 = 1 THEN MOVE '1' TO RETCODE

IF 1 = 2 THEN SEARCH CRITERIA WAS NOT FOUND

EXEC SQL SELECT * INTO :TOPTVAL 
FROM TOPTVAL 
WHERE MAJSYS = 'INAREA,MAJSYS' 
AND ACTION = 'INAREA,ACTION' 
AND OBJECT = 'INAREA,OBJECT' 
AND SRCRCRT = 'T'.

END-EXEC.

**FILL IN DISPLAY AREA 02120000
**WITH HEADING: FFKEY, 02130000
**MESSAGE: 02140000

MOVE HEADTXT IN TOPTVAL TO HTITLE IN OUTAREA.
<table>
<thead>
<tr>
<th>START</th>
<th>CEL</th>
<th>DATE: 17/02/12</th>
<th>TIME: 17:17</th>
<th>PAGE: 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>MOVE IFTPTEXT IN POUTRAL TO 456 IN OUTAREA.</td>
<td>02170000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>MOVE PFTEXT IN POUTVAL TO PFTEXT IN OUTAREA.</td>
<td>02180000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>RETURN TO</strong></td>
<td>02200000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td><strong>DSNMCSI MODULE</strong></td>
<td>02210000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>END-DSNMCSI</td>
<td>02230000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MODULE NAME = DSNTIAR

DESCRIPTION = SQAERROR HANDLING MODULE

COPYRIGHT = SDG-XMV (C) COPYRIGHT 1982, 1985

REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER 0120-2033

STATUS = RELEASE 2, LEVEL 0

FUNCTION = THIS MODULE IS ENTERED AS STANDARD ACTION

WHEN A 'SQAERROR' OR 'SWARNINIX' OCCURS

NOTES =

MODULE TYPE = BLOCK OF COBOL CODE

PROCESSOR = DB2 PRECOMPILED, COBOL COMPILER

MODULE SIZE = SEE LINKEDIT

ATTRIBUTES = REUSEABLE

ENTRY POINT =

PURPOSE = SEE FUNCTION

LINKAGE =

INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:

SYMBOLIC LABEL/NAME = /A

DESCRIPTION = .DNAAME

OUTPUT = PARAMETERS EXPLICITLY RETURNED:

PCONVSTA.OUTPUT.LINE(*)

SYMBOLIC LABEL/NAME = /A

DESCRIPTION = /A

EXIT-NORMAL =

EXIT-ERROR =

RETURN CODE = /A

REASON CODE = /A

MESSAGE ID = /A

ABEND CODES = N/A

ERROR-MESSAGES =

EXTERNAL REFERENCES =

ROUTINES/SERVICES = MODULE DSNTIAR
THIS CODE IS ENTERED AS STANDARD ACTION WHEN AN 'SQLERROR' OR 'SQLWARNING' OCCURS.

INFORMATION DESCRIBING THE ERROR WILL BE PLACED IN THE DISPLAY AREA.

AREA OF THE OUTPUT MESSAGE / PCNAVSTA.OUTPUT.LINE(*) /

IN THE FOLLOWING WAY:

LINE 4 WILL BE BLANK

LINE 5 CONTAINS A MESSAGE INCLUDING NAME (MAJOR AND MINOR) OF THE MODULE WHERE THE ERROR OCCURRED

LINE 6 WILL BE BLANK

LINES 7-14 CONTAIN THE CONTENTS OF 'SQL COMMUNICATION AREA'.
**DETAILED EMPLOYEE MODULE - COBOL**

**NAME = DSNMCE**

**DESCRIPTION NAME = DB2 SAMPLE APPLICATION**

**DESCRIPTION = DETAIL EMPLOYEE MODULE**

**CALL**

**ORGANIZATION**

**COPYRIGHT = 1964-1965 IBM CORPORATION**

**REMARKS TO COPYRIGHT INSTRUCTIONS FOR NUMBER C120-2083**

**STATUS = RELEASE 2, LEVEL 0**

**FUNCTION = THIS MODULE HANDLES THE DETAIL OPERATIONS FOR AN EMPLOYEE SUCH AS DISPLAY, ADD(INSERT), UPDATE, AND ERASE in the major system organization.**

**NOTES**

**DEPENDENCIES = NONE**

**RESTRICTIONS = THE VALID OPTIONS ARE:**

- D-DM-DE-EM-EIE-EIE-END
- D-DM-DE-EM-EIE-BI
- D-DM-DE-EM-EIE-END

**ENTRY POINT =**

**PURPOSE = SEE FUNCTION**

**LINKAGE = MODULE CALLED BY**

- DSNMCE FOR DISPLAY, AND FIRST STEP UPDATE OR ERASE
- DSNMCE FOR FIRST STEP ADD, AND ALL SECOND STEPS.

**INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:**

**COMMON AREA.**

**SYMBOLIC LABEL/NAME = PCONVSTA.PREV**

**DESCRIPTION = 'D' OR 'P' PREVIOUS REQUEST**

**SYMBOLIC LABEL/NAME = MAXSEL**

**DESCRIPTION = 1-13 NUMBER OF SELECTIONS**

**SYMBOLIC LABEL/NAME = OUTAREA.OUTPUT**

**DESCRIPTION = SECONDARY SELECTION OUTPUT**

**SYMBOLIC LABEL/NAME = COMPARM.NEWREQ**

**DESCRIPTION = 'Y' OR 'N' NEW REQUEST**

**SYMBOLIC LABEL/NAME = INAREA**

**COPYRIGHT = 1964-1965 IBM CORPORATION**

**REMARKS TO COPYRIGHT INSTRUCTIONS FOR NUMBER C120-2083**

**DATE: 6/1/72**

**TIME: 1/7/72**

**PAGE: 1**
| START CO | DESCRIPTION = USER INPUT | 0055000 |
|          |                          | 0056000 |
|          |                          | 0057000 |
|          | OUTPUT = PARAMETERS EXPLICITLY RETURNED: COMMON AREA. | 0058000 |
|          | 7 | SYMBO[ICAL LABEL/NAME = OUTAREA.OUTPUT DESCRIPTION = SCREEN DETAIL OUTPUT | 0061000 |
|          | 7 | SYMBO[ICAL LABEL/NAME = PCONVSTA.PREV DESCRIPTION = 'D' OR 'E' DEPENDING ON STEP NUMBER | 0065000 |
|          | 7 |                                                        | 0066000 |
|          | 7 | EXIT-NORMAL = | 0067000 |
|          | 7 | EXIT-ERROR = | 0069000 |
|          | 7 | RETURN CODE = NONE | 0071000 |
|          | 7 | ABEND CODES = NONE | 0072000 |
|          | 7 | ERROR-MESSAGES = | 0073000 |
|          | 7 | DSN3001I EMPLOYEE NOT FOUND | 0074000 |
|          | 7 | DSN3003I EMPLOYEE SUCCESSFULLY ADDED | 0075000 |
|          | 7 | DSN3003I EMPLOYEE SUCCESSFULLY ERASED | 0076000 |
|          | 7 | DSN3004I EMPLOYEE SUCCESSFULLY UPDATED | 0077000 |
|          | 7 | DSN4005E EMPLOYEE EXISTS ALREADY; ADD NOT DONE | 0078000 |
|          | 7 | DSN4006E EMPLOYEE DOES NOT EXIST; ERASE NOT DONE | 0079000 |
|          | 7 | DSN4007E EMPLOYEE DOES NOT EXIST; UPDATE NOT DONE | 0080000 |
|          | 7 | DSN4008E NO VALID SELECTIONS QUALIFY FOR THIS REQUEST | 0081000 |
|          | 7 | ERROR-MESSAGES = | 0082000 |
|          | 7 | DSN4001I EMPLOYEE NOT FOUND | 0083000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ADDED | 0084000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ERASED | 0085000 |
|          | 7 | DSN4004I EMPLOYEE SUCCESSFULLY UPDATED | 0086000 |
|          | 7 | DSN4005E EMPLOYEE EXISTS ALREADY; ADD NOT DONE | 0087000 |
|          | 7 | DSN4006E EMPLOYEE DOES NOT EXIST; ERASE NOT DONE | 0088000 |
|          | 7 | DSN4007E EMPLOYEE DOES NOT EXIST; UPDATE NOT DONE | 0089000 |
|          | 7 | DSN4008E NO VALID SELECTIONS QUALIFY FOR THIS REQUEST | 0090000 |
|          | 7 | ERROR-MESSAGES = | 0091000 |
|          | 7 | DSN4001I EMPLOYEE NOT FOUND | 0092000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ADDED | 0093000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ERASED | 0094000 |
|          | 7 | DSN4004I EMPLOYEE SUCCESSFULLY UPDATED | 0095000 |
|          | 7 | DSN4005E EMPLOYEE EXISTS ALREADY; ADD NOT DONE | 0096000 |
|          | 7 | DSN4006E EMPLOYEE DOES NOT EXIST; ERASE NOT DONE | 0097000 |
|          | 7 | DSN4007E EMPLOYEE DOES NOT EXIST; UPDATE NOT DONE | 0098000 |
|          | 7 | DSN4008E NO VALID SELECTIONS QUALIFY FOR THIS REQUEST | 0099000 |
|          | 7 | ERROR-MESSAGES = | 0100000 |
|          | 7 | DSN4001I EMPLOYEE NOT FOUND | 0101000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ADDED | 0102000 |
|          | 7 | DSN4003I EMPLOYEE SUCCESSFULLY ERASED | 0103000 |
|          | 7 | DSN4004I EMPLOYEE SUCCESSFULLY UPDATED | 0104000 |
|          | 7 | DSN4005E EMPLOYEE EXISTS ALREADY; ADD NOT DONE | 0105000 |
|          | 7 | DSN4006E EMPLOYEE DOES NOT EXIST; ERASE NOT DONE | 0106000 |
|          | 7 | DSN4007E EMPLOYEE DOES NOT EXIST; UPDATE NOT DONE | 0107000 |
|          | 7 | DSN4008E NO VALID SELECTIONS QUALIFY FOR THIS REQUEST | 0108000 |

PSEUDOCODE:
*INITIALIZE.*

*CHECK IF OPTION IS VALID FOR THIS MODULE* 01100000
*MAJOR SYSTEM = '0' AND OBJECT = 'EMP'* 01110000
*IF NOT* RETURN WITH ERROR MSG 069 INVALID REQUEST. 01160000
*STEP-1.*

*FILL IN TEXT LINES (HEADER, INFORMATION, AND PF) FROM VOPTVAL DEPENDING ON ACTION REQUIRED.* 01105000
*IF NOT ADD* SAVE EMPLOYEE ID DEPENDING ON MAXSEL. 01120000
*IF MAXSEL=1 EMP-ID IS ON THE FIRST DETAIL LINE:* 01125000
*IF MAXSEL=1 THE INPUT DATA CONTAINS THE DETAIL LINE NUMBER.* 01130000

*SET DEPARTMENT AND EMPLOYEE FIELD NAMES,* 01135000
*FROM VOPTTXT.* 01140000
*IF DISPLAY OR DELETE ACTION,* 01145000
*PROTECT EVERY DETAIL INPUT FIELD.* 01150000
*IF ADD OR UPDATE ACTION,* 01155000
*PROTECT EMPLOYEE-ID AND ALL DEPARTMENT FIELDS.* 01160000
*POSITION THE SCREEN CURSOR TO EMPLOYEE NAME FIELD.* 01165000
*IF ADD* UNPROTECT EMPLOYEE-ID FIELD.* 01170000
*MOVE USER INPUT TO CORRESPONDING OUTPUT DATA FIELD.* 01175000
*AND FOR DISPLAY, UPDATE AND ERASE,* 01180000
*FETCH EMPLOYEE AND DEPARTMENT CURRENT VALUES,* 01185000
*PREV='D' AND RETURN.* 01190000
*OR MSG *'EMPLOYEE NOT FOUND'* AND RETURN.* 01195000

*STEP-2.*

*IF ADD, DO IT AND MSG* 01200000
*EITHER EMPLOYEE ADDED SUCCESSFULLY* 01205000
*OR *'EMPLOYEE EXISTS ALREADY, ADD NOT DONE'* 01210000
*PREV=' ' AND RETURN.* 01215000
*IF UPDATE, DO IT AND MSG* 01220000
*EITHER EMPLOYEE UPDATED SUCCESSFULLY* 01225000
*OR *'EMPLOYEE DOES NOT EXIST, UPDATE NOT DONE'* 01230000
*RETURN.* 01235000
*IF ERASE, DO IT AND MSG* 01240000
*EITHER EMPLOYEE ERASED SUCCESSFULLY* 01245000
*OR *'EMPLOYEE DOES NOT EXIST, ERASE NOT DONE'* 01250000
*PREV=' ' AND RETURN.* 01255000
*OR MSG CPYC INVALID REQUEST.* 01260000
*RETURN.* 01265000

*END.* 01270000

**************************************
**********
**************************************

DSNAMECF.

******************************************************************************
* CHECKS IF OPTION IS VALID*
*******************************************************************************

7-88
**INICLAR LVALUES**

**STORE MAM SPACES TO MAJOR.**

**STORE MAJOR SYSTEM-O**

**STORE OBJECT-EM**

**IF MAJSYS OF INAREA NOT = 'O' OR**

**OBJECT OF INAREA NOT = 'EM' THEN**

**MOVE I TO I**

**IF ACTION OF INAREA = 'D' THEN**

**GO TO MCF1-STEP.**

**IF NEWREQ = 'N' THEN**

**GO TO MCF2-STEP.**

**IF NEWREQ NOT = 'Y' THEN**

**MOVE ? TO I**

**GO TO MCFNSUP.**

**FETCHES AND PROTECTS FIELDS FOR A CERTAIN COMMAND**

**EXEC SQL SELECT * INTO :POPTVAL FROM YOPTVAL**

**WHERE MAJSYS='O'**

**AND ACTION=INAREA.ACTION**

**AND OBJECT='EM'**

**AND SRCRTYPE='D'**

**AND SRCRT='E1'**

**END-EXEC.**

**EXEC SQL**

**ERR?**

**IF SOLCODE = +100 THEN**

**MOVE DPINF TO MSG OF OUTAREA**

**GO TO END-EXEC.**

**SET**

**FILL IN TEXT LINES**

**MOVE HEAOTXT OF POPTVAL TO HTITLE.**

**MOVE INFOTXT OF POPTVAL TO MSG OF OUTAREA.**

**MOVE PFKTXT OF POPTVAL TO PFTEXT OF OUTAREA.**

**SAVE EMPLOYEE ID**

**IF ACTION OF INAREA = 'A' THEN**

**GO TO MCF010.**

**IF MAXSEL = 1 THEN**

**MOVE MSGNUM(1) TO EMPNO OF PEMP.**

**GO TO MCF010.**

**IF MAXSL < 1 THEN**

**MOVE I TO I**

**GO TO MCFNSUP.**

**DATE: 07/02/12**

**TIME: 17:39**

**PAGE: 5**
IF DAT1 NOT NUMERIC THEN
GO TO MCFNSUP.

IF DAT2 NOT NUMERIC THEN
MOVE DAT1 TO DAT2
MOVE '0' TO DAT1.

**INPUT DATA CONTAINS NUMERIC
**THE DETAIL LINE NO.

IF OAT? NOT NUMERIC THEN
MOVE DAT1 TO DAT2
MOVE '0' TO DAT1.

**SAVE EMPLOYEE ID

MOVE HORNUM111 TO EMPNO OF PEMP1.

**CLEAR FIELD WITH BLANKS

MOVE O TO I.

**OPEN CURSOR

MOVE 0 TO I.

**OPEN CURSOR

EXEC SQL OPEN DH END-EXEC.

**OPEN CURSOR

EXEC SQL OPEN DH END-EXEC.

**CLOSE CURSOR

EXEC SQL CLOSE DH END-EXEC.

**CLOSE CURSOR

EXEC SQL CLOSE DH END-EXEC.
**SET** NAME,0,540,$5,<
**TERM** NAME,0,540,$5,

**START**

<table>
<thead>
<tr>
<th>COL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>*</td>
<td>MCF24.</td>
<td></td>
<td></td>
<td>03250000</td>
<td>03260000</td>
<td>03270000</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>MCF24.</td>
<td></td>
<td></td>
<td>03250000</td>
<td>03260000</td>
<td>03270000</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>MCF24.</td>
<td></td>
<td></td>
<td>03290000</td>
<td>03300000</td>
<td>03310000</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>ADD 1 TO I.</td>
<td><strong>MCF24 LOOP</strong></td>
<td></td>
<td>03320000</td>
<td>03330000</td>
<td>03340000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>MCF-LOOP24.</td>
<td></td>
<td></td>
<td>03340000</td>
<td>03350000</td>
<td>03360000</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>MCF-LOOP24.</td>
<td></td>
<td></td>
<td>03370000</td>
<td>03380000</td>
<td>03390000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>CURSOR POSITION</strong></td>
<td>03400000</td>
<td>03410000</td>
<td>03420000</td>
<td>03430000</td>
<td>03440000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>OPEN UPDATES, OR ERASES AND PRINTS A MESSAGE</strong></td>
<td>03450000</td>
<td>03460000</td>
<td>03470000</td>
<td>03480000</td>
<td>03490000</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>FETCH EMPLOYEE</strong></td>
<td>03500000</td>
<td>03510000</td>
<td>03520000</td>
<td>03530000</td>
<td>03540000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>EXEC SQL SELECT * INTO :PEMPL FROM VEMPL WHERE EMPNO=:PEMPL.EMPNO</td>
<td>03550000</td>
<td>03560000</td>
<td>03570000</td>
<td>03580000</td>
<td>03590000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>EMPLOYEE NOT FOUND</strong></td>
<td>03600000</td>
<td>03610000</td>
<td>03620000</td>
<td>03630000</td>
<td>03640000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>FETCH DEPARTMENT</strong></td>
<td>03650000</td>
<td>03660000</td>
<td>03670000</td>
<td>03680000</td>
<td>03690000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>DEPARTMENT FOUND</strong></td>
<td>03700000</td>
<td>03710000</td>
<td>03720000</td>
<td>03730000</td>
<td>03740000</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td><strong>DEPARTMENT NOT FOUND</strong></td>
<td>03750000</td>
<td>03760000</td>
<td>03770000</td>
<td>03780000</td>
<td>03790000</td>
</tr>
</tbody>
</table>

**END-EXEC.**

**START**

<table>
<thead>
<tr>
<th>COL</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td>EXEC SQL SELECT * INTO :DEPFT FROM VDEPT WHERE DEPTNO=:DEPFT.WKDEPT</td>
<td>03700000</td>
<td>03710000</td>
<td>03720000</td>
<td>03730000</td>
<td>03740000</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td><strong>DEPARTMENT NOT FOUND</strong></td>
<td>03750000</td>
<td>03760000</td>
<td>03770000</td>
<td>03780000</td>
<td>03790000</td>
</tr>
</tbody>
</table>

**END**

7-92
START

1. Move '0' to PREV.
2. Move '1' to MIND.
3. Move '0' to ;
4. Move ;?
5. Move TRANDATA(1) to FIELD2(I).
6. Move TRANDATA(2) to FIELD2(I).
7. Move TRANDATA(3) to FIELD2(I).
8. Move TRANDATA(4) to FIELD2(I).
9. Move TRANDATA(5) to FIELD2(I).
10. Move TRANDATA(6) to TRANDATA.
11. Move TRANDATA(7) to FIRSTNAME-TEXT OF PEMPL.
12. Move TRANDATA(8) to LASTNAME-TEXT OF PEMPL.
13. Move TRANDATA(9) to WORKDATE OF PEMPL.
14. Move TRANDATA(10) to WORKDATE OF PEMPL.
15. Move TRANDATA(11) to WORKDATE OF PEMPL.
16. Move TRANDATA(12) to WORKDATE OF PEMPL.
17. Move TRANDATA(13) to WORKDATE OF PEMPL.
18. Move TRANDATA(14) to WORKDATE OF PEMPL.
19. Move TRANDATA(15) to WORKDATE OF PEMPL.
20. Move TRANDATA(16) to WORKDATE OF PEMPL.
21. Move TRANDATA(17) to WORKDATE OF PEMPL.
22. Move TRANDATA(18) to WORKDATE OF PEMPL.
23. Move TRANDATA(19) to WORKDATE OF PEMPL.
24. Move TRANDATA(20) to WORKDATE OF PEMPL.
25. Move TRANDATA(21) to WORKDATE OF PEMPL.
26. Move TRANDATA(22) to WORKDATE OF PEMPL.
27. Move TRANDATA(23) to WORKDATE OF PEMPL.
28. Move TRANDATA(24) to WORKDATE OF PEMPL.
29. Move TRANDATA(25) to WORKDATE OF PEMPL.
30. Move TRANDATA(26) to WORKDATE OF PEMPL.
31. Move TRANDATA(27) to WORKDATE OF PEMPL.
32. Move TRANDATA(28) to WORKDATE OF PEMPL.
33. Move TRANDATA(29) to WORKDATE OF PEMPL.
34. Move TRANDATA(30) to WORKDATE OF PEMPL.
35. Move TRANDATA(31) to WORKDATE OF PEMPL.
36. Move TRANDATA(32) to WORKDATE OF PEMPL.
37. Move TRANDATA(33) to WORKDATE OF PEMPL.
38. Move TRANDATA(34) to WORKDATE OF PEMPL.
39. Move TRANDATA(35) to WORKDATE OF PEMPL.
40. Move TRANDATA(36) to WORKDATE OF PEMPL.
41. Move TRANDATA(37) to WORKDATE OF PEMPL.
42. Move TRANDATA(38) to WORKDATE OF PEMPL.
43. Move TRANDATA(39) to WORKDATE OF PPLEMENT.
44. Move TRANDATA(40) to WORKDATE OF PEMPL.
45. Move TRANDATA(41) to WORKDATE OF PEMPL.
46. Move TRANDATA(42) to WORKDATE OF PEMPL.
47. Move TRANDATA(43) to WORKDATE OF PEMPL.
48. Move TRANDATA(44) to WORKDATE OF PEMPL.
49. Move TRANDATA(45) to WORKDATE OF PEMPL.
50. Move TRANDATA(46) to WORKDATE OF PEMPL.
51. Move TRANDATA(47) to WORKDATE OF PEMPL.
52. Move TRANDATA(48) to WORKDATE OF PEMPL.
53. Move TRANDATA(49) to WORKDATE OF PEMPL.
54. Move TRANDATA(50) to WORKDATE OF PEMPL.
55. Move TRANDATA(51) to WORKDATE OF PEMPL.
56. Move TRANDATA(52) to WORKDATE OF PEMPL.
57. Move TRANDATA(53) to WORKDATE OF PEMPL.
58. Move TRANDATA(54) to WORKDATE OF PEMPL.
59. Move TRANDATA(55) to WORKDATE OF PEMPL.
60. Move TRANDATA(56) to WORKDATE OF PPLEMENT.
61. Move TRANDATA(57) to WORKDATE OF PEMPL.
62. Move TRANDATA(58) to WORKDATE OF PEMPL.
63. Move TRANDATA(59) to WORKDATE OF PEMPL.
64. Move TRANDATA(60) to WORKDATE OF PEMPL.
65. Move TRANDATA(61) to WORKDATE OF PEMPL.
66. Move TRANDATA(62) to WORKDATE OF PEMPL.
67. Move TRANDATA(63) to WORKDATE OF PEMPL.
68. Move TRANDATA(64) to WORKDATE OF PEMPL.
69. Move TRANDATA(65) to WORKDATE OF PEMPL.
70. Move TRANDATA(66) to WORKDATE OF PEMPL.
71. Move TRANDATA(67) to WORKDATE OF PEMPL.
72. Move TRANDATA(68) to WORKDATE OF PEMPL.
73. Move TRANDATA(69) to WORKDATE OF PEMPL.
17 EXEC SQL WHENEVER SOLError CONTINUE END-EXEC.
16 EXEC SQL INSERT INTO EMPLOYEES (EMPNO, FIRSTNAME, MIDINIT, LASTNAME, WORKDEPT) VALUES (:PEMPL.EMPNO, :PEMPL.FIRSTNAME, :PEMPL.MIDINIT, :PEMPL.LASTNAME, :PEMPL.WORKDEPT) **PERFORM INSERT
15 WHENEVER SQLERROR CONTINUE END-EXEC.
14 EXEC SQL UPDATE EMPLOYEES SET FIRSTNAME = :PEMPL.FIRSTNAME, MIDDLEINIT = :PEMPL.MIDINIT, LASTNAME = :PEMPL.LASTNAME, WORKDEPT = :PEMPL.WORKDEPT WHERE EMPNO = :PEMPL.EMPNO **PERFORM UPDATE
13 WHENEVER SQLERROR CONTINUE END-EXEC.
12 EXEC SQL UPDATE EMPLOYEES WHERE EMPNO = :PEMPL.EMPNO MOVE :PEMPL.EMPNO TO EmployeeID **EMPLOYEE SUCCESSFULLY UPDATED
11 MOVE :PEMPL.LASTNAME TO Field21 **EMPLOYEE SUCCESSFULLY ADDED
10 EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.
9 WHENEVER SQLERROR CONTINUE END-EXEC.
8 WHENEVER SQLERROR CONTINUE END-EXEC.
7 WHENEVER SQLERROR CONTINUE END-EXEC.
6 WHENEVER SQLERROR CONTINUE END-EXEC.
5 WHENEVER SQLERROR CONTINUE END-EXEC.
4 WHENEVER SQLERROR CONTINUE END-EXEC.
3 WHENEVER SQLERROR CONTINUE END-EXEC.
2 WHENEVER SQLERROR CONTINUE END-EXEC.
1 WHENEVER SQLERROR CONTINUE END-EXEC.
DATA:  DSNAME, input
MEMO:  DSNAME.

DATE:  07/22/12
TIME:  17:37
PAGE:  10

START

10  MOVE SPACES TO DEPTNAME-TEXT OF DEPT1.
11  EXEC SQL SELECT * FROM DEPT WHERE DEPTNO=:W0UDEPT
12  END-EXEC.
13  IF SQLCODE = 0 THEN
14      MOVE DEPTNAME-TEXT OF DEPT1 TO FIELD2(12).
15      MOVE MGRNO OF DEPT1 TO FIELD2(13).
16      MOVE ADMDPT OF DEPT1 TO FIELD2(4).
17      GO TO MCFSG.
18  MOVE I TO I.
19  **PUT SPACES AT END OF FIELD**
20  END-EXEC.
21  ADD 1 TO I.
22  MOVE SPACES TO FIELD2(12).
23  **MCF342 LOOP**
24  MCF-L00P42.
25  PERFORM MCF342 UNTIL I > 3.
26  GO TO MCFSG.
27  MCF050.
28  **PERFORM ERASE**
29  **ERASE**
30  EXEC SQL DELETE FROM VEMP1
31  WHERE EMPNO=PEMPL-EMPNO.
32  END-EXEC.
33  **EMPLOYEE SUCCESSFULLY ERASED**
34  IF SQLCODE = 0 THEN
35      MOVE 'Y' TO PREW.
36      MOVE '0066' TO MSGCODE.
37  **EMPLOYEE DOES NOT EXIST**
38  IF SQLCODE = +100 THEN
39      **ERASE NOT DONE**
40      MOVE '0066' TO MSGCODE.
41      GO TO MCFSG.
42  IF SQLCODE = 0 OR +100 THEN
43      GO TO MCFSG.
44  **ERROR - INVALID REQUEST**
45  MCFNUP.
47  **MOVE '066E' TO MSGCODE.**
48  **PRINT MESSAGE**
49  **PERFORM ERASE**
50  MCFMSG.
51  CALL 'DSN8MCG' USING MAJOR MSGCODE OUTMSG.
52  MOVE OUTMSG TO MSGTXT OF MSG.
53  **PRINT MESSAGE**
54  **ERROR - INVALID REQUEST**
START

* DSNN4CA - SQL 2 SECONDARY SELECTION FOR MAJOR SYSTEM 0 - COBOL 00000000
  * MODULE NAME = DSNN4CA
  * DESCRIPTIVE NAME = DB2 SAMPLE APPLICATION
  * SQL 2 SECONDARY SELECTION
  * CICS
  * COBOL
  * ORGANIZATION
  * COPYRIGHT = 5740-XYR ECI COPYRIGHT IBM CORP 1982, 1985
  * REFER TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083
  * STATUS = RELEASE 2, LEVEL 0
  * FUNCTION = THIS MODULE PRODUCES A SECONDARY SELECTION SCREEN
    FOR OBJECTS IN MAJOR SYSTEM '0' (ORGANIZATION)
    CALLED BY DSNN4C2 (SQL2 MAINLINE)
  * NOTES = NONE
  * MODULE TYPE = BLOCK OF COBOL CODE
  * PROCESSOR = DB2 PRECOMPILER, COBOL COMPILER
  * MODULE SIZE = SEE LINKEDIT
  * ATTRIBUTES = REUSABLE
  * ENTRY POINT = DSNN4CA
  * PURPOSE = SEE FUNCTION
  * LINKAGE = NONE
  * INPUT = SYMBOIC LABEL/NAME = COMPMTR
  * DESCRIPTION = POINTEO TO COMMAREA
  * (COMMUNICATION AREA)
  * OUTPUT = SYMBOIC LABEL/NAME = COMPMTR
  * DESCRIPTION = POINTER TO COMMAREA
  * (COMMUNICATION AREA)
  * EXIT-NORMAL = END OF COBO RLOCK
  * EXIT-ERROR = IF SQL ERROR OR SQL WARNING,
  * SQL WHENEVER CONDITION
  * SPECIFIED IN DSNN4C2 WILL BE RAISED AND PROGRAM
  * WILL GO TO THE LABEL DB_ERR0R.
  * RETURN CODE = NONE
  * ABEND CODES = NONE
  * ERROR MESSAGES = DSNN067E - UNSUPPORTED SEARCH CRITERIA FOR OBJECT

DATE: 07/02/22
TIME: 17:24
PAGE: 1

COPYRIGHT = 5740-XYR ECI COPYRIGHT IBM CORP 1982, 1985

REFoR€ TO COPYRIGHT INSTRUCTIONS FORM NUMBER G120-2083

STATUS = RELEASE 2, LEVEL 0

FUNCTION = THIS MODULE PRODUCES A SECONDARY SELECTION SCREEN
FOR OBJECTS IN MAJOR SYSTEM '0' (ORGANIZATION)
CALLED BY DSNN4C2 (SQL2 MAINLINE)

NOTES = NONE

MODULE TYPE = BLOCK OF COBOL CODE
PROCESSOR = DB2 PRECOMPILER, COBOL COMPILER
MODULE SIZE = SEE LINKEDIT
ATTRIBUTES = REUSABLE
ENTRY POINT = DSNN4CA
PURPOSE = SEE FUNCTION
LINKAGE = NONE
INPUT = SYMBOIC LABEL/NAME = COMPMTR
DESCRIPTION = POINTER TO COMMAREA
(COMMUNICATION AREA)
OUTPUT = SYMBOIC LABEL/NAME = COMPMTR
DESCRIPTION = POINTER TO COMMAREA
(COMMUNICATION AREA)
EXIT-NORMAL = END OF COBO RLOCK
EXIT-ERROR = IF SQL ERROR OR SQL WARNING,
SQL WHENEVER CONDITION
SPECIFIED IN DSNN4C2 WILL BE RAISED AND PROGRAM
WILL GO TO THE LABEL DB_ERR0R.
RETURN CODE = NONE
ABEND CODES = NONE
ERROR MESSAGES = DSNN067E - UNSUPPORTED SEARCH CRITERIA FOR OBJECT
Pseudocode

/* SECONDARY SELECTION FOR NAJSYS 'D' - OBJECTS
1. DS - ADMINISTRATIVE LISTING
2. DE - INDIVIDUAL DEPARTMENTS
3. EM - INDIVIDUAL EMPLOYEES
DS AND DE USE THE SAME CURSORS WHICH SELECTS DEPARTMENTS AND MANAGERS. FM USES ANOTHER CURSOR WHICH SELECTS DEPARTMENTS AND EMPLOYEES. THE FIELDS SELECTED BY THE TWO DIFFERENT CURSORS ARE THE SAME IN NUMBER AND MAY MATCHING THEREFORE IT IS POSSIBLE TO USE THE SAME CODE FOR BOTH SITUATIONS MOST OF THE TIME.

THERE ARE TWO SITUATIONS UNDER WHICH THIS MODULE CAN BE CALLED
1. THE SYSTEM FIELDS MAY HAVE CHANGED - NEW REQUEST
2. AN ANSWER TO A PREVIOUS REQUEST IF COMPARE NEWSREQUEST THIS SYSTEM FIELDS CHANGED AND THIS IS A NEW REQUEST
THIS MODULE SHOULD SET THE FOLLOWING TWO FIELDS BEFORE EXITING
1. PCONVST.SAME.IND = NEXT TIME AROUND
2. PCONVST.MAXSEL.IND. OF ENTRIES ON SEC SEL SCREEN BUILT

PROCEDURE
INITIALIZE TWO CONTROL FIELDS
CASE NEW REQUEST
INITIALIZE MINIMUM VALUES
ASSIGN FIELD VALUES FOR 'LIKE' IN SQL SELECT
RETRIEVE DESCRIPTIVES AND IMF MESSAGE
RETRIEVE DESCRIPTIVE LINES
ENDCASE
ASSIGN DATA VALUE FROM SCREEN FOR 'LIKE' PROCESSING
IF 'EM' SEARCH CRITERIA THEN
OPEN EMPLOYEE CURSOR ASCENDING

ELSE
OPEN ADMIN ST CURSOR ASCENDING
SET UP 'DU LOOP' VALUES
*FETCH* FROM THE APPROPRIATE CURSOR UP TO MAX OF 13 TIMES
IF NO VALU Entries THEN
SEND MESSAGE
SAVE MIN VALUE FOR POSSIBLE SCROLLING REQUEST
RETURN
END.

OSNAMCA.

MOVE 'OSNAMCA' TO MAJOR.
MOVE SPACES TO MINOR.

**INITIALIZE CONTROL FIELDS

MOVE 'S' TO PREV OF LASTPOS.
MOVE TO I.
MOVE SPACES TO LINE OF I.

PERFORM SQL-OP10.
PERFORM UNTIL I > 14.
**NEW REQUEST

IF NEWREQ OF COMPARE = 'Y' THEN
**NEW REQUEST

**INITIALIZE MINIMUM
**VALUES

MOVE LOW-VALUES TO DIM, EMIN
**ASSIGN FIELD VALUES
**FOR LIKE SQL SELECT

MOVE 1 TO 'LDEPTNO', 'LDEPTNAME', 'MGRTNO', 'MGRTNAME', 'LEMPNO', 'LEMPNAME'

MOVE PERCENT TO 'LDEPTNO', 'LDEPTNAME', 'MGRTNO', 'MGRTNAME', 'LEMPNO', 'LEMPNAME'.

7-99

EXEC SQL SELECT DSPLINE INTO :POPTVAL.DSPLINE FROM VPSPTXT WHERE DSPINDEX = :POPTVAL.DSPLINDEX

IF SOLCODE = '100' THEN
  MOVE * TO PREV OF LASTPOS
  STRING MAJSYS OF INARE REGION, ACTION OF INARE REGION, OBJECT OF INARE REGION, SRCH OF INARE REGION
  INTO MSGMOD2
  GO TO END-DSNVMA.
ENDIF.
**SPLINE**

**ASSIST DATA VALUE**

*FOR* 'LIKE' PROCESSING

**NOTE** THAT ALL THE FOLLOWING SEARCH CRITERIA MAY NOT BE SUPPORTED IN ALL SITUATIONS — HOWEVER SOL 1 WILL ONLY

**PEM** IT VALID ENTRIES TO BE PASSED.

**DATE:** 7/02/12

**TIME:** 17:24

**PAGE:** 5
**DATE:** 17/02/12  
**TIME:** 17:24  
**PAGE:** 1

**STAKT**

**CCL**

**-<**

**7**

**<**

**8**

**>l**

**lo**

*t:inc.-.j*  

**11**

**U**

**i**

**»U**

**P?ORT**

**£**

**3**

**_S**

**EARCH**

**•CRITERIA**

**FOR**

**OBJECT**

**•PKINT**

**ERROR**

**MESSAGE**

**WOVg**

**'0**

**676***

**TO**

**HS**

**GO.**

**CALL**

**V**

**3SN8HC6'**

**USING**

**.iAJQR**

**flS'SCODE**

**OUTffS**

**HOVf**

**OUTM56**

**TO**

**KS'lTEXT**

**60**

**TO**

**^WD-PS.HaHCfl.**

**_______**

****f>ATA**

**TOO**

**LUNG**

****P.UMT**

**ERROR**

**MESSAGE**

**NCA025.**

**move**

**CALL**

**MQ='**

**OJTFISu**

**TO**

**r,iG7EXT**

**OF**

***.**

**HOVE**

**•**

**TO**

**PREV**

**HF**

**LASTPOS.**

**60**

**TO**

**END-DSNSFICA.**

***»**

**OPEN**

**CURSORS**

**••OPEN**

**EflPLQVEE**

**02710000**

**02720000**

**02730000**

**02740000**

**02750000**

**02760000**

**02770000**

**02780000**

**02790000**

**02800000**

**02810000**

**02820000**

**02830000**

**02840000**

**02850000**

**02860000**

**02870000**

**02880000**

**02890000**

**02900000**

**02910000**

**02920000**

**02930000**

**02940000**

**02950000**

**02960000**

**02970000**

**02980000**

**02990000**

**03000000**

**03010000**

**03020000**

**03030000**

**03040000**

**03050000**

**03060000**

**03070000**

**03080000**

**03090000**

**03100000**

**03110000**

**03120000**

**03130000**

**03140000**

**03150000**

**03160000**

**03170000**

**03180000**

**03190000**

**03200000**

**03210000**

**03220000**

**03230000**

**07-102**
*CLOSE CURSORS AND RETURN

**CLOSE CURSOR

**CLOSE EMPLOYEE

**CLOSE DEPARTMENT

**CLOSE ALA

**RETURN
Appendix C

Formal/DB2/ADR Database Entity Comparison
<table>
<thead>
<tr>
<th>Formal Relational</th>
<th>DB2</th>
<th>ADR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database</td>
<td>Database</td>
<td>Database</td>
</tr>
<tr>
<td>Relation</td>
<td>Table</td>
<td>File</td>
</tr>
<tr>
<td>View</td>
<td>View</td>
<td>Dataview</td>
</tr>
<tr>
<td>Tuple</td>
<td>Row</td>
<td>Record</td>
</tr>
<tr>
<td>Attribute</td>
<td>Column</td>
<td>Field</td>
</tr>
<tr>
<td>Domain</td>
<td>**</td>
<td>**</td>
</tr>
</tbody>
</table>

** The system has no term for domain. Domains are implicitly defined when the attributes (columns or fields) are declared.
TWO RELATIONAL DBMS: A COMPARISON

by

GARY F. GARTEN

B.S., Kansas State University, 1978
B.A., Kansas State University, 1979
B.S., Kansas State University, 1979

AN ABSTRACT OF A MASTER'S DISSERTATION

submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE, COMPUTER SCIENCE

College of Arts and Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1987
Abstract

Data Base Management Systems (DBMS) are playing an increasingly important role in the development of computer systems. Few studies comparing the relational systems available for mainframe computers exist. This work compares two commercially available systems, IBM's DB2 (Data Base 2) and Applied Data Research's Datacom/DB.

After a brief introduction to the subject of relational data base management, DB2 and Datacom/DB product families are compared based on product descriptions, current product prices, and system overviews. Further comparisons of the two DBMS product families are made by comparing a miniature application system implemented. Difficulties encountered in reproducing the DB2 mini-system at the ADR site are described in detail.