" DATA MIGRATION "

PROJECT WORK DONE AT

COVANSYS INDIA LIMITED, CHENNAI.

PR0JECT REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE AWARD OF THE DEGREE OF
MASTER OF COMPUTER APPLICATIONS
OF BHARATHIAR UNIVERSITY, COIMBATORE.

SUBMITTED BY

Shibu Joseph
Reg.No. 9838M0521

GUIDED BY

EXTERNAL GUIDE

Mr. Ravi Rangarajan,
General Manager,
Covansys India Limited,
Chennai.

INTERNAL GUIDE

Mr. M.Saravanamuthu,
MCA, PGDBA, BLIS, MISTE,
Lecturer, Dept. of Comp. Science
& Engineering,
Kumaraguru College of Tech,
Coimbatore.

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641 006
MAY 2001

Certificates

CERTIFICATE

This is to certify that the project work entitled

DATA MIGRATION

Submitted to the

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

In partial fulfillment of requirements for the award of degree of Master of Computer Applications is a record of original work done **by Mr.Shibu Joseph, Reg. No 9838M0521** during his period of study in the Department of Computer Science engineering, Kumaraguru College of Technology, Coimbatore under my supervision and this project work has not formed the basis of award of any Degree/Diploma Associateship/Fellowship or similar title to any candidate of any University

Professor and Head 23/4

23/4
Guide

Submitted to University Examination held on ---------------------------

11/5
Internal Examiner

External Examiner 11/5/01

Covansys

(Formerly CBSI)

Unit 13, Block 2, SDF Buildings
Madras Export Processing Zone
Chennai - 600 045. India
Phone : 91-44-2628080, 2623880
Fax : 91-44-2628171

April 5, 2001

To Whomsoever It May Concern

This is to certify that Mr. Shibu Joseph, (Univ. Reg No: 9838M0521) final year MCA (Master of Computer Application) student of Kumaraguru College of Technology, has successfully completed his project titled "Data Migration" in Covansys (India) Ltd.

He did this project under the guidance of Mr. Ravi Rangarajan, and during his project period (from January 10, 2001 to April 5, 2001), his conduct and character was found to be good.

Yours Sincerely,
For Covansys (India) Ltd.

H Venkatesan
Vice President – HR

Declaration

DECLARATION

I hereby declare that this project work entitled

DATA MIGRATION

Submitted to Kumaraguru College of Technology, Coimbatore (Affiliated to Bharathiar University) is a record of original work done by me under the supervision and guidance of **Mr.M.Saravanamuthu, MCA, PGDBA, BLIS, MISTE** Lecturer, Department of Computer Science and Engineering, Kumaraguru College of Technology, Coimbatore under my supervision and guidance and his project work has not formed the basis of award of any Degree/ Diploma/ Associateship/ Fellowship or similar title to any candidate of any University.

Place: Coimbatore

Date: 23-04-01

Signature of the candidate

(Shibu Joseph)

Counter signed by

23/4

Guide

Mr. M. Saravanamuthu
Lecturer,
Department of Computer Science and
Engineering, Kumaraguru College
of Technology ,Coimbatore.

Acknowledgment

ACKNOWLEDGEMENT

I take this opportunity to express my gratitude to all those good people for their contribution to the successful completion of this project.

My sincere thanks to **Dr. S. Thangasamy, BE (Hon's), Ph.D, Head, Department of Computer Science and Engineering for his valuable guidance.**

I am highly indebted by to my guide **Mr. M. Saravanamuthu Lecturer, Department of Computer Science and Engineering** for providing me constant support and encouragement for successfully completing the project.

I express my heartfelt thanks to **Mr. Swamynathan, Associate Development Manager** for his valuable guidance and Help to the successful completion of this project.

I would like to express my sincere gratitude to **Mr. Ravi Rangarajan, General Manager** for providing opportunity to do this project in their esteemed organization. Let me admit that I really enjoyed carrying my project work in the pleasant and amiable atmosphere of this company

Synopsis

SYNOPSIS

Covansys supports the packages that run on the mainframes of a large shipping company in U.S. The support can be an enhancement for the existing packages or a bug fixing. In the clients mainframe system there are different regions like production, test etc. The actual packages runs in the production region. But the testing and the development take place in the test region. The project work entitled **Data Migration ~~Tool To Port Data From Production To Test~~** will down load database, as requested by the user, from the production region to the test region so that it will be easy to fix the bug and also to test the enhancements.

This is a tool written in MVS/ESA, VS-COBOL, IDMS, and proprietary architecture environment. The output of the tool will cater to the need of testing data in some critical areas of the Application. Test data is also very vital in arriving at quick solutions whenever the production problems occur and affect the business process. It is also useful in testing Programs that are written in a short notice to fix the production problems.

The scope of the tool and hence of the project will be in the purview to the specific client it is catering to. The tool can only be used to download data specific to the client. But it is generic as far as the project is concerned.

Contents

CONTENTS

Page No.

Introduction

1.INTRODUCTION

1.1 PROJECT OVERVIEW

The project named **Data Migration** is developed for the Covansys project team that supports a large shipping company's mainframe environment. The objective of the tool is to download the database from the production region to the test region.

To have the user needs an interview was conducted with the programmers. It was easy to conclude on the needs as all were from the IT industry and could understand the project.

The design of the process as well as database was done with the help of project guide. Database design was a replicate of the production region and hence had to have a thorough understanding of the production database.

The development was done in VS-COBOL, MVS/ESA, Integrated Data Base Management System (IDMS) and the proprietary architecture environment. It was challenging as it was not with the usual Graphic User Interface (G.U.I.).

The project was of two stages. In the first stage the production database was downloaded to a flat file by running the program in the production region. The second program was to upload the test database from the flat file by

1.2 COMPANY PROFILE

Covansys (formerly known as CBSI), founded more than fifteen years ago by an N.R.I. Mr. Raj Vattikuti has successfully delivered fast, focused, high-quality yet cost-effective business and technical solutions. The name was derived from **co**operative, **ad**van**ced**, **sys**tems. Its professional heritage bridges the largest 'Old World' legacy systems and the New Economy's leading edge Web applications.

Its range of experience, the scope of expertise, and scaleable delivery model make Covansys a special breed in the technology services field. Covansys put those attributes to work in order to guarantee clients short-cycle project completion and ingenious resource deployment with real-world budgets. The result: innovative, problem-solving technology services that improve efficiency, reduce costs, and increase revenues.

Covansys specializes in Web-to enterprise Integration (WEI), indutry-specific solutions, and strategic outsourcing. It helps industry leaders in the public and private sectors resolve a broad range of business and technology challenges. Satisfied Fortune 1000 customers provide the clearest testimony of success. Over 80 percent of our annual business derive from our existing client base of more than 500 companies.

This level of customer trust and confidence is a tribute to the commitment of more than 5,000 employees and consultants -- long recognized as one of our key strengths. Acting as local partners, the Covansys team becomes an integral part of the client's technology team, tapping into our collective experience and global resources to complete projects with skill, speed, and flexibility.

Covasys's unique delivery model means clients have the right resources at the right time and at the right cost -- a fact underscored by two recent industry honors. In December, 1999, Covansys became one of the few companies worldwide to achieve the highest possible maturity rating for its development centers in India -- the Software Engineering Institute's CMM Level 5 quality rating. In addition, Fortune Magazine named Covansys as one of the 100 fastest-growing U.S.-based companies in its September, 2000 issue.

Public- and private-sector enterprises looking for a technology services strategy for business transformation, specialized industry solutions, or outsourcing turn to Covansys with confidence -- and with good reason.

1.2.1 Company Highlights

- Fortune Magazine selected Covansys as one of 100 fastest-growing U.S.-based companies (September 2000).

- Deloitte & Touche Technology Fast 500 Program recognized Covansys, based on its 568% 5-year growth rate (December 1999).

- For two consecutive years, Covansys was recognized as one of the top three performing publicly traded companies in Michigan.

- Covansys are one of the few companies in the world to achieve an outstanding CMM Level 5 quality rating for our development centers in India from the Software Engineering Institute.

- It has one of the largest U.S. owned offshore capabilities in the world.

1.2.2 Mission and Value Proposition

***The Covansys Mission*:**

Mission of Covansys is to bring value to our employees, stakeholders, and the global community by delivering advanced technology solutions that enable its clients and trading partners to operate, interoperate, and compete more effectively.

Our Value Proposition:

Covansys is the global technology Services Company that brings competitive advantage to its clients by delivering innovative, results-oriented information management solutions

1.2.3 Global Offices

Covansys has its head quarters in Michigan, U.S.A and has offices across the world. In U.S. it has offices in different states like New Jersey, New York, California to name a few. Across the globe it has offices in India, UK, France, Netherlands, Belgium, Philippine, Singapore, Germany and Australia.

1.2.4 Core Competency

Web-to-Enterprise-Integration

Covansys has been integrating Web applications with core business systems since the infancy of the Web. We call it Web-to-Enterprise Integration, and it's what we do best.

Web-to-Enterprise Integration requires an understanding of business and legacy systems, deep traditional application development, a breadth of knowledge across disparate software and platforms, and outstanding technical and architectural skills. We look at the big picture and the potential for leveraging **CRM, ERP,** and **SCM** in order to create an integrated Web-based business and technology strategy. Goal of Covansys is to help our clients improve efficiency, build stronger business relationships, expedite key decisions, and enrich customer relationships so as to develop

an information strategy that reduces costs and creates improved revenue generation opportunities for the clients.

Industry specific solutions

Covansys believe that knowledge of a client's industry is every bit as important as technology expertise. There are no cookie-cutter miracles. There is no one-size-fits-all answer. It's 15 years of collective knowledge and experience means a short learning curve on the way to a customized implementation for our clients.

Strategic outsourcing

Covansys has years of experience with a number of industries, such as health care, public sector, telecommunications, retail, manufacturing, and financial services.

It's specialist teams focus on the changes generated by such issues as e-business, globalization, and deregulation -- and provide high-value, client-driven, timely business solutions.

Strategic Outsourcing

Speed, flexibility, and round-the-clock productivity are the hallmarks of Covansys outsourcing services. It's fast and flexible on-site, off-site, offshore delivery model lends itself to a full spectrum of outsourcing services -- from applications maintenance, development, and modernization to full technology outsourcing, including data center operations.

In order to meet aggressive schedules, Covansys offer clients one of the largest U.S.-owned offshore capabilities in the world. This capability sets Covansys apart from many 'land-locked' companies in the technology services field.

1.2.5 Customers

Covansys has over 500 satisfied customers in industries such as health care, retail, manufacturing, telecommunications, public sector, and financial services

Banking & Financial Services

Bank One Services Corporation

Chase Manhattan Corporation

Foremost Corporation of America

Putnam Investments

UNUM Corporation

Public Sector

State of California

State of Hawaii

State of Indiana

State of Michigan

State of Ohio

State of Rhode Island

Health Care

Anthem Blue Cross Blue Shield

Retail and Distribution

Fred Meyer, Inc.

of Connecticut

Baystate Health System

Blue Cross and Blue Shield of Michigan

The Gap, Inc.

Lands′ End, Inc.

Spartan Stores, Inc.

Manufacturing

DaimlerChrysler AG

Ford Motor Company

General Motors Corporation

Johnson Controls, Inc.

Weyerhaeuser Company

Utilities and Telecommunications

Ameritech Corporation

Lucent Technologies, Inc.

South Carolina Electric and Gas Company

Southern California Edison

System Study And Analysis

2. SYSTEM STUDY AND ANALYSIS.

System analysis and design has been one of the important milestones of the project life cycle. Since the project named **Data Migration ~~Tool To Port Data From Production To Test~~** was of the first of its kind in the project for the large shipping company it had to start from the scratch. It was really a challenging job and since it was starting fresh, there was no legacy carrying to the new system.

2.1 User Characteristics

One of the reasons, the project went without any hurdles were because the users were real time programmers, coding and testing the projects in Covansys. It has helped in a big way to find out the system requirements. I had to have a question-answer session with the employees to narrow down to which database has to be down loaded so that the tool will be of utmost help. The knowledge and experience they had in the project has helped in a big way to make the tool highly useful. Because of the in-depth knowledge of the users about the system it was also easy to make the input design.

To use the tool the user has to input the calc key of the record to the input file. The tool will down load all the fields of the record and also the fields in the owner and member records. The user has to know the Multiple Virtual Storage (M.V.S.) environment to handle the tool. Also he should have access to the production region to use it.

2.2 User Specifications of the System.

One of the general specifications is have a quick response as most of the time the data may be needed to the user in short span because he may be attending to the real time production bugs. Also it was important to take care of the security threat since it is the real time business data. Another specification is to develop the system in the proprietary environment. A latent need is to make it as user-friendly as possible, with the knowledge that it is out of question to have a graphic user interface (GUI) in mainframe environment.

2.3 Functional Requirements.

The system is conceived by a need to have a sample of the actual data in the test region, as it is present in the production region. So the primary requirement is to migrate the database from the production to the test region. Also, it is not necessary and also impossible to have all the data of the production region. Hence the tool had to get input from the user so that it will download only the related data.

2.4 Proposed System.

Keeping in mind all the specifications, the new system was proposed. The system will take the input from a file and the user only has to input only the calc key of the required field. It will make it as easy as possible to use the system. The tool is of two parts - first part will download the data to intermediate files and the second part will upload the test region database from the intermediate files. The security issues are handled by having the first part of the tool in the production region and are given access only with the correct password. To assure speed the tool will fetch only the records, which are directly related to the record of the inputted calc key. This will also help to curtail the excess data available in the production region and that is not necessary in the test region. The above said is made sure by having calc key navigation of the database rather than the area sweeping which will fetch all the data. The system is proposed to develop in MVS with VS-COBOL and IDMS as database and the hardware configuration of the client so that it will match the proprietary architecture.

Programming Environment

3 PROGRAMMING ENVIRONMENT.

3.1 HARDWARE CONFIGURATION.

- IBM 390
- 3270 TERMINAL
- TCP/IP

SOFTWARE SPECIFICATION

• PLATFORM	: MAINFRAME
• OS	: MULTIPLE VIRTUAL STORAGE (MVS)
• LANGUAGE	: COBOL
• DATABASE	: INTEGRATED DATABASE MANAGEMENT SYSTEM (IDMS)

3.2 DESCRIPTION OF SOFTWARE CHARACTERISTICS OF MAINFRAME OPERATING SYSTEMS

The following are the general characteristics of Mainframe Operating Systems

VIRTUAL STORAGE

Virtual storage is a technique that lets a large amount of main storage be stimulated by the operating system that actually has a smaller amount of real storage.

MULTIPROGRAMMING

Multiprogramming means the simultaneous execution of more than one program in a single CPU. A typical business program, which performs a significant number of I/O operations, makes processor idle for a large percentage of the time, as they wait for I/O operations to complete. Multiprogramming simply reclaims the CPU during these idle periods to let other programs execute. Multiprogramming make better use of system resources namely CPU. Storage space, and CPU time.

SPOOLING

To provide shared access to printer devices, spooling is used. Spooling manages printer output for applications by interpreting it and directing it to a device instead. Then, when the program is finished, the operating system collects its spooled print output and directs it to the printer. In a multiprogramming environment, each program's spooled output is stored separately on disk so that it can be printed separately. Another benefit of spooling is that disk devices are much faster than printers.

BATCH PROCESSING

Batch Processing is the normal way of using the mainframe computer system. In batch processing, work is processed in units called jobs. A job may cause one or more programs to be executed in sequence.

TIME SHARING

In a time sharing system, each user has access to the system through a terminal device. Time sharing is sometimes called interactive processing, because it lets users interact directly with the computer. For technical reasons, time sharing is sometimes called foreground processing, while batch job processing is called background processing.

MULTIPROCESSING

In the advanced models of the System/370 family, more than one CPU is included in the processor. In those multiprocessor systems, two or more CPUs share access to main memory.

HISTORY OF MVS

IBM announced the System/360 family of processors in 1964. OS/360 is the original ancestor for MVS. OS/MVT (Multiprogramming with variable number of Tasks), which allow upto 15 jobs of varying sizes to run simultaneously. In the 1972 OS/VS2 has been released, was later called SVS (Single Virtual Storage), with one 16 MB view of virtual storage. SVS was modified to newer version, that allowed multiple views of part of its 16 MB virtual storage area, it was named MVS(Multiple Virtual Storage).

The following are the versions of MVS

1) MVS
2) MVS/XA (Extended Architecture)
3) MVS/ESA (Extended System Architecture)

FEATURES

It supports Uniprocessor, Attached processor, Coupled multiprocessor configuration.

It provides integrated Time Sharing Option (TSO) facility.

System Resource Manager (SRM) for dynamic workload management.

Job entry functions supported standard Job Entry Subsystems JES2 and JES3.

Functional recovery routines for system components.

Support for 31-bit virtual storage address (2 Giga bytes) (MVS/XA)

BENEFITS

Provides unrestricted multiprogramming.

Multiprocessor configurations offers increased internal performance and greater configuration possibilities with real storage, channels and I/O devices.

The Job Entry Subsystems support system restart, automatic job step restart and check point/restart which reduce restart and rerun costs.

Virtual Input/Output facility (VIO) for temporary data sets uses a combination of buffer space within an address space and the paging mechanism thereby saving storage space on DSAD's.

System software errors are handled by means of Functional Recovery Routines (FRR).

HIGH -LEVEL LANGUAGES SUPPORTED BY MVS

The following are the languages supported by MVS operating system

FORTRAN

VS COBOL II

PL/I

CLIST

IBM 'C'

PROCESSORS SUPPORTED BY MVS

MVS Version	Processor No.
MVS	4341 through 3090
MVS/XA	4341 and 3090
MVS/ESA	3090, 9000

SUBSYSTEMS THAT RUNS ON MVS

- TSO
- ISPF
- PDF

- CICS
- DB2
- IMS
- JES2
- JES3

Software

- VSAM (Virtual Storage Access Method)
- ACF/VTAM (Access Control Facility/Virtual Telecommunication Access Method)
- ACF/TCAM (Access Control Facility/Telecommunication Access Method)
- RACF (Resources Access Control Facility)

MEMORY MANAGEMENT UNDER MVS

Pageable Supervisor and Link-pack Area
Master Scheduler Region
Dynamic Region
Non-Pageable Dynamic Area
MVS Nucleus

- Nucleus resides in the lowest area of fixed storage.
- The Kernel consists of the most heavily used parts of the Operating system and control.
- One of the main function of the nucleus is that the operation of the virtual storage.

Non-pageable Dynamic Area

- This area provided for additional programs that must operate in real storage.
- Time dependent programs such as telecommunications access methods are executed here.

Dynamic Region Area

- This area is allocated to job steps as they execute.

Master Scheduler Region

- This area contains the master scheduler program, which is responsible for communication between MVS and the system operator.

Pageable Supervisor Area

- This is part of the highest address portion of the virtual storage.
- The less frequently used parts of the MVS operating system are used in this area and swapped in and out as needed.

Link Pack Area

- This area is for heavily used I/O routine or other system subroutines.
-

- If the subroutines are placed in Link pack Area, they can be shared by many programs.

JOB SHEDULING

- A special program called the Job Scheduler selects jobs for processing.
- The job scheduler selects a job from the queue for each step in the job, then it starts an Initiator/Terminator task that allocates the devices and main storage needed to run the program.

JOB ENTRY SUBSYSTEM

- The JES combines the functions of scheduling and spooling to provide a comprehensive facility for processing jobs through a MVS system.
- It read and process the Job Control Language (JCL) statements.
- It also provides many options for the final disposition of spooled output data based on the class and device specified in the JCL.There are two versions of JES called

- JES2
- JES3

JES2

- It is designed for single-processor systems.

- Can handle only simple job scheduling functions.

JES3

- It is designed for multi-processor systems.
- Can handle complicated job scheduling functions.

DATA MANAGEMENT SERVICES

- MVS provides Data Management Services for handling Data Sets (Files).
- The following are the important services it provides.

- Data set storage Control
- Cataloging
- Space allocation
- Security

DATA SET LABELS

- Under MVS, a storage entity such as a tape or disk pack is called as Volume and may contain many data sets.
- MVS uses the data set labels to identify the data set on a volume.
- The Volume itself identified by a Volume Serial Number.
- For direct-access volumes, a special area called the Volume Table Of Contents (VTOC), contains the data set labels for each data set on the volume.

- Each label in the VTOC gives
- Data set Name
- Data Control Block Information
- Location of the Data Set on the disk

DATA SET ORGANIZATION

- TSO mainly uses two methods of data set organization.
- Physical Sequential Data Set (PS)
- Partitioned Data Set (PDS)

Physical Sequential Data Set

- The records of the data set are written in processing sequence and the records are always retrieved in the sane order.

Eg.

PROJECT.TRAINING.DATA

Partitioned Data Set

- Partitioned data sets are often called as PDS or library, consists of a directory and one or more members.
- A PDS directory is a list of the members in the library.
- Each member is functionally as same as the sequential as data set.
- The advantage of PDS is that the entire library can be processed as a single file.

- In a PDS, each member has one to eight character member name.
- To refer a member, the user has to specify the member name in parentheses.

Eg.

PROJECT.TRAINING.COURSES(ONE)

DATA CONTROL BLOCK (DCB)

- When a program opens a file, MVS creates a Data Control Block (DCB) for the file based on the information taken from the program statements and the data set label for the file.

SPACE ALLOCATION

- The allocation of space on direct-access device is an important function of data management services.
- It lets the user to specify
  - How much space the file will need
  - Which volume the file will go on
- If the volume specified does not have enough space for the data set MVS will choose a volume that has enough space.

SECURITY

- MVS provides for data security by allowing passwords to be set for data sets.

- Usually, special security system called RACF (Resources Access Control Facility) will be used in installations that need enhanced security features.

TIME SHARING OPTION

- TSO is a feature of MVS operating system, which has
- Time-sharing Capabilities
- Interact-through a terminal
- Easy to use commands

- Time-sharing
- Computer time is divided into time slices
- Each terminal user is allocated a time slice
- Can supports hundreds or even thousands of users

ISPF

Interactive System Productivity Facility (ISPF) is designed to increase programmer productivity in an MVS/TSO environment. ISPF/PDF(for Program Development Facility), lets you use most of the features of MVS in an interactive, menu-driven fashion, bypassing the cumbersome commands typical of native TSO.

COBOL

COBOL (Common Business-Oriented Language) was developed in 1959 by the CODASYL (Conference on Data Systems Languages) Committee.

Although the language is nearly 40 years old, it is still one of the most widespread commercial applications languages currently in use. In the business field, COBOL is used for commercial applications such as payroll and inventory; usually operating on a large amount of data.

COBOL is also a standard language. COBOL compilers are available in a standard form for most computers, and a single COBOL program can be compiled and run on a variety of computers from mainframes to personal computers, with only minor variations. COBOL has the same syntax rules regardless of what type of computer the compiler is on.

COBOL is also an English-like language. Any COBOL instruction can be coded using English words instead of symbols or complex codes. For example, the word ADD is used to add two numbers together. The rules for programming in COBOL are similar to the rules for writing in English, so learning COBOL and writing COBOL programs are rather simple.

By being an English-like language, COBOL is considered to be a user-friendly language. By using English words instead of overly technical and sometimes cryptic code, any user with little programming experience can begin to understand the logic and order of instructions of a COBOL program.

Writing COBOL Programs

Every program written in COBOL consists of four separate divisions, with each one having a specific function:

Identification Division - This division identifies the program to the computer. This division may also contain documentation about the program.

Environment Division - This division defines all of the file names and describes the specific computer equipment that the program will use.

Data Division - This division contains descriptions of the input and output formats that will be used by the program. Within this division, there may be definitions of constants and work areas necessary for processing data.

Procedure Division - This division contains all instructions necessary for reading the input, processing it, and creating the desired output.

Every COBOL program must contain all four of the above divisions in that order.

COBOL Coding Rules

All COBOL programs must be structured in this way:

Columns 1 - 6: Used for sequence numbers or page and line numbers. This is optional and was previously used for sequence checking when programs were punched into cards.

Column 7: Used for continuations, comments, or starting a new page. Placing a * in column 7 denotes that line as being a comment. Placing a / in column 7 causes the printer to skip to a new page when printing the source listing, and this column can also be used to continue nonnumeric literals.

Columns 8 - 11: Known as "Area A"; entries such as DIVISION, SECTION, and paragraph-names must begin in Area A.

Columns 12 - 72: Known as "Area B"; most COBOL entries including PROCEDUREDIVISION statements and sentences are coded in Area B.

Columns 73 - 80: Used for program identification (this is optional.)

Identification Division:

As the name implies, the identification division supplies any identifying information about the program. While it has no effect on the execution of the program, it is required for program compilation. The following is an example of the divisions format:

```
IDENTIFICATION DIVISION.
PROGRAM-ID. program-name.
[AUTHOR. [comment-entry] ]
[INSTALLATION. [comment-entry] ]
[DATE-WRITTEN. [comment-entry] ]
[DATE-COMPILED. [comment-entry] ]
[SECURITY. [comment-entry] ]
```

The division names, and all paragraph names, are coded in Area A (columns 8-11). Each must end with a period. Brackets [] mean that the paragraph is optional. Therefore, PROGRAM- ID is the only necessary paragraph within this division. Each entry within the paragraph must be coded in Area B (columns 12-72).

If used, AUTHOR includes the programmers name. INSTALLATION includes the name of the company or organization. DATE-WRITTEN and DATE-COMPILED are self- explanatories. SECURITY indicates whether the program is classified or confidential. Any additional comments may be included using a * in column 7.

Environment Division:

This is the only machine-dependent part of a COBOL program. There are two section of this division the configuration section and the input-output section. The configuration section indicates the source computer the computer used for compiling the program and the object computer the computer that will be used for executing or running the program. This is not required if you running COBOL 85. Also these would be coded in the Area A.

```
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. IBM-ES900.
OBJECT-COMPUTER. VAX-8800.
```

This code will cause the program to be compiled on an IBM computer and executed on a VAX.

The other section of this division is called the input-output section. This section supplies information about specific devices used in the program. This is where the printers and disk drives are typically referred to in. The file control paragraph consists of Select statements, each of which is coded in Area B followed by a period. A select statement defines a file-name and assigns a device name to that file. The file-name must conform to the rules for forming user-defined words. If the file that contain the data is created with each record appearing on a separate line the clause organization is line sequential is needed. This will ensure that each record is on a separate line. Here is an example of the input-output section of the program.

```
INPUT-OUTPUT SECTION.
FILE-CONTROL.
SELECT EMPLOYEE-FILE ASSIGN TO C:\EMPFILE .
SELECT TRANS-FILE ASSIGN TO PRINTER.
ORGANIZATION IS LINE SEQUENTIAL.
```

Data Division:

This division is the part that defines and describes fields, records, and files in storage. Any area of storage that is required for the processing of data must be established in the DATA DIVISION. The two main sections are the file section, which defines all input and output files , and the working storage section which reserves storage for fields not part of input or output but required for processing. These include constants, end of file indicators, and work areas.

Each FD (file description) will describe a file defined in the select statement in the environment division. The record contain clause indicates the size of each record. This is mostly used with a print file and would be coded RECORD CONTAINS 80 CHARACTERS because most printers are 80 characters. Each FD must be followed by record description entries for the file. Records are defined on the 01 level. Each field within this record is coded on a level subordinate. Any level between 02 and 49 may be used. A field that is not further subdivided is called and elementary item. All elementary items must be described with a PICTURE clause that indicates the size and type of the field.

The three types of data fields are Alphabetic -only contains letters, Alphanumeric - which can contain letters and numbers, and Numeric- only contains numbers. The characters used in picture clauses are A for Alphabetic, X for Alphanumeric, and 9 for numeric. The code 05 CODE-IN PICTURE XXXX defines a four position storage area that will contain alphanumeric data. A more common way of coding the picture clause is with parentheses so instead of XXXX the code would be X(4). When coding a program you have to allot space for unused area in record description entries. The picture clause should account for the total number of positions of the file. The symbol V denotes an implied decimal point. PICTURE 999V99 will be interpreted as 100.00 not 10000.

The WORKING-STORAGE SECTION is used for storing intermediate results, counters, end of file indicators, and interactive data. The VALUE clause may be used to initialize fields.

Procedure Division:

This division corresponds to the heart of most other programming languages (especially those in the imperative/algorithmic paradigm). It is here that calculations are made, and conditional statements, loops, and jumps from one part of the program to another take place. The procedure division consists of, at least, a main module (usually, labeled 100-MAIN-MODULE). Any other routines are generally numbered using increments of 100. The following statements are those typically coded in COBOL.

The OPEN statement accesses the input and output files for a program. A file cannot be read or written to until it is accessed with the OPEN command. This statement indicates the input files and the output files, makes files available for processing, and performs header label routines if label records are STANDARD. A CLOSE statement is coded at the end of record processing. This command releases the files and frees up and deactivates any devices.

The READ statement transmits data from the input device, assigned in the ENVIRONMENT DIVISION, to the input storage area, defined in the FILE

SECTION of the DATA DIVISION. An AT END clause is used to tell the program what to do when there is no more data to be read. A NOT AT END clause may also be used with the READ statement.

The PERFORM UNTIL statement works similarly to that of a do while statement in C. This statement transfers control to the procedure of the paragraph named. This paragraph is executed until the specified condition is met. Using PERFORM without the UNTIL will execute the paragraph only once.

The MOVE statement is used to transfer fields in primary storage to other fields. A MOVE operation copies input data to the output area.

The WRITE statement takes data in the output area (defined in the DATA DIVISION) and transmits it to the specified device in the ENVIRONMENT DIVISION. While we READ files, we WRITE records.

The four basic arithmetic operations in COBOL include ADD, SUBTRACT, MULTIPLY, and DIVIDE. These have the following instruction format:

ADD {identifier/literal} TO {identifier}

SUBTRACT {identifier/literal} FROM {identifier}

MULTIPLY {identifier/literal} BY {identifier}

DIVIDE {identifier/literal} INTO {identifier}

A GIVING clause may also be applied to store the calculated value in an identifier. An IF THEN ELSE statement is also available in COBOL. Condition tests include: IS EQUAL TO, IS NOT EQUAL TO, IS LESS THAN, IS GREATER THAN, etc. Finally, the STOP-RUN statement indicates program termination. All programs should include this statement at the end of the 100-MAIN-MODULE.

JOB CONTROL LANGUAGE (J C L)

- It is the command language for MVS operating system
- The users use JCL to specify:
    - ! What program the user want to run , and where it is ?
    - ! What are the files that the program will use ?
    - ! What the user want to after the program is over ?
- JCL is the way that the user can talk with MVS.
- JCL is a means of communication between a program written in COBOL. Assembler or PL/1 and the MVS operating system.

It has set of control statement that provide the specifications necessary To process a job.

- The tasks of a job are informed to OS through JCL statements.

- For different versions of MVS such as MVS/XA, MVS/ESA, etc.

JCL remains same

IDMS/R CAPABILITIES

IDMS/R combines the advantage of both RELATIONAL AND NETWORK database architecture. Users, benefit from the ease of the Relational model for special requests and smaller applications.The power of the Network model is available for large, high-volume production applications.

Features

Automatic System Facility (ASF)

When a user defines a relational record, ASF generates all the necessary supporting structures such as data definitions, screen, formats, and application processing logic and documentation. No system developments are required by the user in order to develop an application around the defined relational record.

An Active Data Dictionary

Active data dictionary is used to maintain data definitions and control information.

Integrated Indexing

Indexes are defined in the database to provide the access to data through symbolic keys.

Automatic Database Backup and Recovery

Backup and recovery features, including automatic journaling, protect the integrity of the database against program or system failure.

Multiple Database Support.

Multiple database is multiple copies of the same database definitions but with different data.

SETS

The logical relationship between two or more record type is defined by a SET. In a set, one record type is the OWNER and other record types are MEMBERS.

SET RELATIONSHIP

Set relationships are defined according to the following rules.

* Any record type can participate as a MEMBER in one or more sets.
* Any record type can be the OWNER of one or more sets.
* Any record type can participate as a MEMBER in one set and the OWNER in another.

CHAINED SETS

A chained set establishes a logical relationship between two or more user-defined record types, and consists of an owner type and one or more member record types.

Set Membership Options

Set membership options indicate the manner in which a member record is Connected and Disconnected from a set occurrences.

Set membership is defined in two parts:

Mandatory or Optional.

Indicates the manner in which the record is Disconnected from a set.

Automatic or Manual.

Indicates the manner in which the record is Connected to a set.

Disconnect Options

Mandatory: -

Record occurrences cannot be Disconnected from the set without being erased from the database.

Optional: -

Record occurrence can be Disconnected from the set without being erased from the database.

The record occurrence can be Connected to another set.

Connect Options.

Automatic:-

The membership of the record in a set is established automatically whenever a member record is stored in a database.

Manual:-

A record occurrence is not automatically connected to a set when it is stored in the database. Membership established by using the CONNECT statement.

System Design
And Development

4. SYSTEM DESIGN AND DEVELOPMENT

4.1Input Design.

Data Migration ~~Tool To Port Data From Production To Test~~ is a tool for downloading records from production database, namely IISA-AREA. This tool does not download all the records from IISA-AREA, since that consists of huge number of records. So this tool is designed to download records according to user input. In this design have one input file for writing calc key of record, which, user wants to download from production.

The second part of design is to download records from production. First, the tool will obtain the record with the user-specified calc key. After that, it will download all records on the basis of relationships and writes to separate intermediate files.

4.2 Output Design.

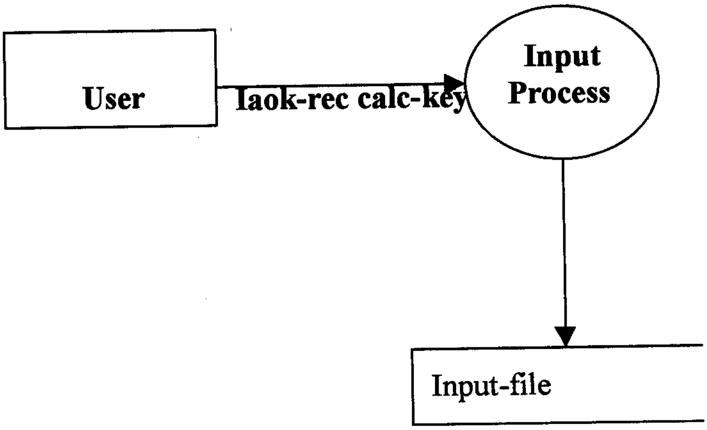
The third part of this tool is designed to upload records from the intermediate files to Test Region.This is a testing region of records from Production Database.In the input design have intermediate files. So this tool will read from file and store records in to Test Database.The tool will care Mandatory Automatic (MA) and Optional Manual (OM) connections will satisfies through the uploading of all the records. And it will also care about the Owner and Member relationship.

After uploading to Test Database, user can obtain all records from Test Region.

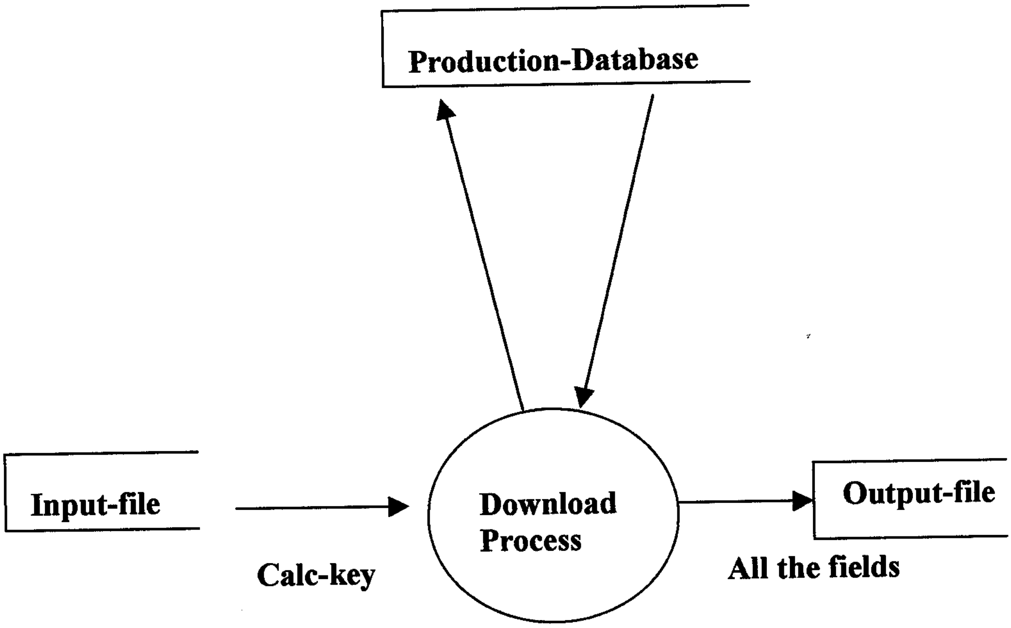
4.3 DATA FLOW DIAGRAM

The data flow diagram (DFD) is one of the most important modeling tools used by system analysts. DFD should be the first tool used by the system analysts to model system components. These components are the system processes, the data used by these processes, any external entities that interact with the system and the information flows in the system.

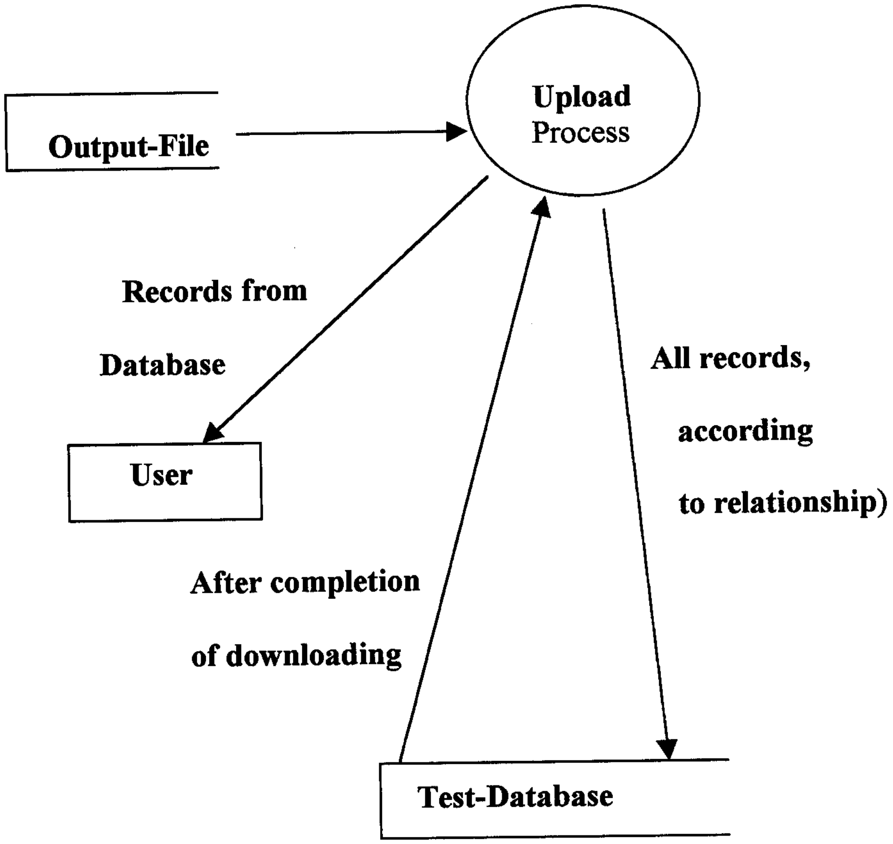
DFD for Input Process



DFD FOR DOWNLOADING DATA FROM PRODUCTION



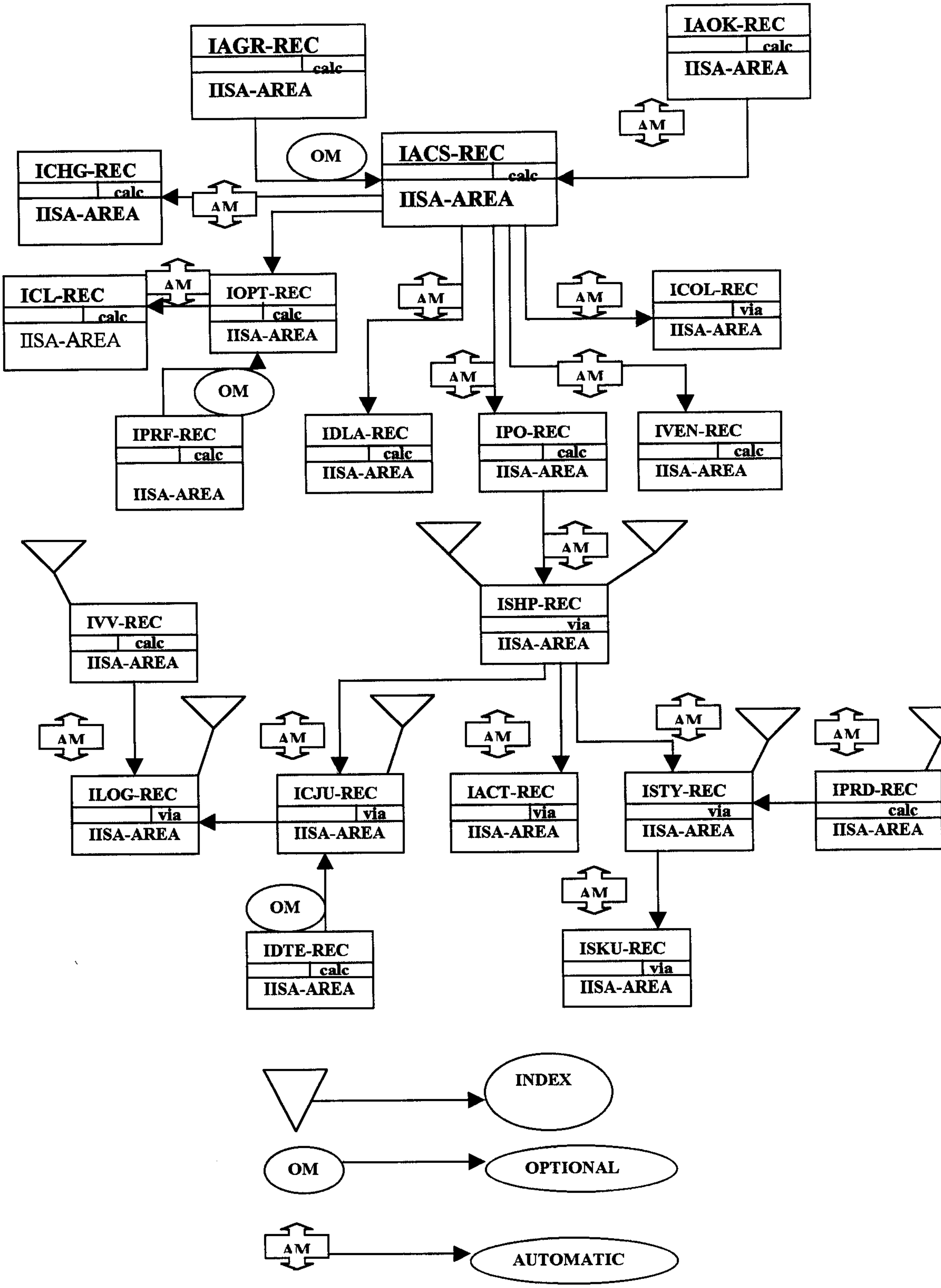
DFD FOR UPLOADING DATA TO TEST REGION



Output-File
Upload Process
Records from Database
User
All records, according to relationship)
After completion of downloading
Test-Database

BACHMAN DIAGRAM

IMPORT INFORMATION SYSTEM



2.4 DATABASE DESIGN

The objective of the database design is to treat data as an organizational resource and as an integrated whole. Database management systems allows the data to be protected and organized separately from other resources.

IAGR-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IAGR-LAST-UPDATE-GROUP	Y	GROUP	
03	IAGR-LAST-USER-ID	Y	DISPLAY	X(008)
03	IAGR-LAST-UPDATE-DATE	Y	GROUP	
04	IAGR-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IAGR-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IAGR-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IAGR-LAST-RACN	Y	COMP	9(004)
02	**IAGR-CALC-KEY**	N	GROUP	
03	**IAGR-ACS-ACCT-NUMBER**	N	DISPLAY	9(006)
02	FILLER	Y	DISPLAY	X(026)

IAOK-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IAOK-LAST-UPDATE-GROUP	Y	GROUP	
03	IAOK-LAST-USER-ID	Y	DISPLAY	X(008)
03	IAOK-LAST-UPDATE-DATE	Y	GROUP	
04	IAOK-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IAOK-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IAOK-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IAOK-LAST-RACN	Y	COMP	9(004)
02	**IAOK-CALC-KEY**	N	GROUP	
03	**IAOK-IAOK-VALUE**	N	DISPLAY	X(004)
02	FILLER	Y	DISPLAY	X(004)

IACS-REC

LEVEL	FIELD NAME	NULL	#OCCURS	USAGE	PICTURE
02	IACS-LAST-UPDATE-GROUP	Y		GROUP	
03	IACS-LAST-USER-ID	Y		DISPLAY	X(008)
03	IACS-LAST-UPDATE-DATE	Y		GROUP	

04	IACS-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	IACS-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	IACS-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	IACS-LAST-RACN	Y		COMP	9(004)
02	**IACS-CALC-KEY**	N		GROUP	
03	**IACS-ACS-ACCT-NUMBER**	N		DISPLAY	9(006)
02	IACS-ACS-ACCT-NAME	Y		DISPLAY	X(020)
02	IACS-ACS-ACCT-ADDRESS	Y	3	DISPLAY	X(020)
02	IACS-ACS-ACCT-PHONE	Y		DISPLAY	X(020)
02	IACS-ACS-DELETE-FLAG	Y		DISPLAY	X(001)

02	IACS-EST-DAYS-TO-ARRIVE	NULL		DISPLAY	9(003)
02	IACS-PROD-STATUS-FLAG	Y		DISPLAY	X(001)
02 03	IACS-USER-FINANCIAL-LITERALS	Y		GROUP	
03	IACS-PLAN-FINANCIAL-LIT	Y	6	DISPLAY	X(010)
03	IACS-ACT-FINANCIAL-LIT	Y	10	DISPLAY	X(010)
02	IACS-ACSA-ACCT-CODE	Y		DISPLAY	X(006)
02 03	IACS-SHIP-ADVICE-DEFAULTS	Y		GROUP	
04 05	IACS-PROC-ALL-ACCTS-FLAG	Y		DISPLAY	X(001)
04 05	IACS-DOMESTIC-CARRIER-NAME	Y		DISPLAY	X(15)
03	IACS-IMP-PRINTER	Y		GROUP	
04	IACS-IMP-CITY-CODE	Y		DISPLAY	X(003)
04	IACS-IMP-PRINTER-ID	Y		DISPLAY	X(003)
03 04	IACS-IMP-REMOTE-TERM-NUMBER	Y		DISPLAY	X(003)
03	IACS-IMP-MAILBOX	Y		DISPLAY	X(006)
03	IACS-DC-PRINTER	Y		GROUP	
02	IACS-DEST-LIT	Y		DISPLAY	X(008)

ICHG-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ICHG-LAST-UPDATE-GROUP	Y	GROUP	
03	ICHG-LAST-USER-ID	Y	DISPLAY	X(008)
03	ICHG-LAST-UPDATE-DATE	Y	GROUP	
04	ICHG-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ICHG-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ICHG-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ICHG-LAST-RACN	Y	COMP	9(004)
02	**ICHG-SORT-KEY**	N	GROUP	
03	**ICHG-ACS-OFFICE**	N	DISPLAY	X(003)

02	ICHG-CONSOL-FEE-CBM	Y	DISPLAY	9(2)V9(2)
02	ICHG-CONSOL-FEE-BOX	Y	DISPLAY	9(4)V9(2)
02	ICHG-IIS-CHARGE-CBM	Y	DISPLAY	9(2)V9(2)
02	ICHG-CONSOL-FEE-CBM-CY	Y	DISPLAY	9(2)V9(2)
02	ICHG-CONSOL-FEE-BOX-CY	Y	DISPLAY	9(4)V9(2)
02	ICHG-CONSOL-FEE-CBM-MCC	Y	DISPLAY	9(2)V9(2)
02	ICHG-IIS-CHARGE-IND	Y	DISPLAY	X(001)
02	ICHG-IIS-CHARGE-BOX-CBM	Y	DISPLAY	X(001)
02	FILLER	Y	DISPLAY	X(0023)

ICL-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ICL-LAST-UPDATE-GROUP	Y	GROUP	
03	ICL-LAST-USER-ID	Y	DISPLAY	X(008)
03	ICL-LAST-UPDATE-DATE	Y	GROUP	
04	ICL-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ICL-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ICL-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ICL-LAST-RACN	Y	COMP	9(004)
02	**ICL-CALC-KEY**	N	GROUP	
03	**ICL-ACS-ACCT-NUMBER**	N	DISPLAY	9(006)
03	**ICL-ACS-DEPT-NUMBER**	N	DISPLAY	X(005)
03	**ICL-ACS-CLASS-NUMBER**	N	DISPLAY	9(003)
02	ICL-ACS-CLASS-NAME	Y	DISPLAY	X(015)
02	FILLER	Y	DISPLAY	X(0011)

IDPT-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IDPT-LAST-UPDATE-GROUP	Y	GROUP	
03	IDPT-LAST-USER-ID	Y	DISPLAY	X(008)
03	IDPT-LAST-UPDATE-DATE	Y	GROUP	
04	IDPT-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IDPT-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IDPT-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IDPT-LAST-RACN	Y	COMP	9(004)
02	**IDPT-CALC-KEY**	N	GROUP	
03	**IDPT-ACS-ACCT-NUMBER**	N	DISPLAY	9(006)
03	**IDPT-ACS-DEPT-NUMBER**	N	DISPLAY	X(005)
02	IDPT-ACS-DEPT-NAME	Y	DISPLAY	X(015)
02	IDPT-ACS-BUYER-NAME	Y	DISPLAY	X(015)
02	FILLER	Y	DISPLAY	X(0015)

ICOL-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ICOL-LAST-UPDATE-GROUP	Y	GROUP	
03	ICOL-LAST-USER-ID	Y	DISPLAY	X(008)
03	ICOL-LAST-UPDATE-DATE	Y	GROUP	
04	ICOL-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ICOL-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ICOL-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ICOL-LAST-RACN	Y	COMP	9(004)
02	**ICOL-SORT-KEY**	N	GROUP	
03	**ICOL-COLOR-CODE**	N	DISPLAY	X(003)
02	ICOL-COLOR-DESCRIPTION	Y	DISPLAY	X(010)
02	FILLER	Y	DISPLAY	X(0011)

IPRF-REC

LEVEL	FIELD NAME	NULL	OCCURS	USAGE	PICTURE
02	IPRF-LAST-UPDATE-GROUP	Y		GROUP	
03	IPRF-LAST-USER-ID	Y		DISPLAY	X(008)
03	IPRF-LAST-UPDATE-DATE	Y		GROUP	
04	IPRF-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	IPRF-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	IPRF-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	IPRF-LAST-RACN	Y		COMP	9(004)
02	**IPRF-CALC-KEY**	N		GROUP	
03	**IPRF-ACS-ACCT-NUMBER**	N		DISPLAY	9(006)
03	**IPRF-PROFILE-CODE**	N		DISPLAY	X(03)
02	IPRF-STAGE-PROFILE	Y	6	GROUP	
03	IPRF-STAGE-DESCRIPTION	Y		DISPLAY	X(20)
03	IPRF-EST-COMPLETION-DAYS	Y		DISPLAY	9(03)
03	IPRF-SUBSTAGES	Y		GROUP	
04	IPRF-SUB-DESCRIPTION-1Y	Y		DISPLAY	X(20)
04	IPRF-SUB-DESCRIPTION-4Y	Y		DISPLAY	X(20)
04	IPRF-SUB-DESCRIPTION-5Y	Y		DISPLAY	X(20)
04	IPRF-SUB-DESCRIPTION-6Y	Y		DISPLAY	X(20)
02	FILLER	Y		DISPLAY	X(021)

IDLA-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IDLA-LAST-UPDATE-GROUP	Y	GROUP	
03	IDLA-LAST-USER-ID	Y	DISPLAY	X(008)
03	IDLA-LAST-UPDATE-DATE	Y	GROUP	

04	IDLA-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IDLA-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IDLA-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IDLA-LAST-RACN	Y	COMP	9(004)
02	**IDLA-CALC-KEY**	N	GROUP	
03	**IDLA-ACS-ACCT-NUMBER**	N	DISPLAY	9(006)
03	**IDLA-RCPT-LOC-CODE**	N	DISPLAY	X(005)
03	**IDLA-FINAL-DEST-PORT-CODE**	N	DISPLAY	X(003)
03	**IDLA-ACS-COUNTRY-CODE**	N	DISPLAY	X(003)
02	IDLA-ORIG-CNTRYRCPT-DAYS	Y	DISPLAY	9(003)
02	IDLA-FDP-TO-RCPT-LOC-DAYS	Y	DISPLAY	9(003)
02	FILLER	Y	DISPLAY	X(017)

IPO-REC

LEVEL	FIELD NAME	NULL	OCC URS	USAGE	PICTURE
02	IPO-LAST-UPDATE-GROUP	Y		GROUP	
03	IPO-LAST-USER-ID	Y		DISPLAY	X(008)
03	IPO-LAST-UPDATE-DATE	Y		GROUP	
04	IPO-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	IPO-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	IPO-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	IPO-LAST-RACN	Y		COMP	9(004)
02	**IPO-CALC-KEY**	N		GROUP	
03	**IPO-ACS-ACCT-NUMBER**	N		DISPLAY	9(006)
03	**IPO-ACS-PO-NUMBER**	N		DISPLAY	X(016)
02	IPO-CREATE-DATE	Y		GROUP	
03	IPO-CREATE-DATE-YY	Y		DISPLAY	9(002)
03	IPO-CREATE-DATE-MM	Y		DISPLAY	9(002)
03	IPO-CREATE-DATE-DD	Y		DISPLAY	9(002)
02	IPO-ACS-BUYER-NAME	Y		DISPLAY	X(015)
02	IPO-EVENT-NUMBER	Y		DISPLAY	X(005)
02	IPO-ACS-OFFICE	Y		DISPLAY	X(003)
02	IPO-ACS-VENDOR	Y		GROUP	
03	IPO-VENDOR-TICKET-FLAG	Y		DISPLAY	X(001)
03	FILLER	Y		DISPLAY	X(007)
03	IPO-VENDOR-NAME	Y		DISPLAY	X(020)
03	IPO-VENDOR-ADDRESS	Y	3	DISPLAY	X(020)
03	IPO-VENDOR-PHONE	Y		DISPLAY	X(020)
02	IPO-AGENT-NAME	Y		DISPLAY	X(020)
02	IPO-ACS-IMP-DEPT-NUMBER	Y		DISPLAY	X(005)
02	IPO-ACS-IMP-DEPT-NAME	Y		DISPLAY	X(015)

IVEN-REC

LEVEL	FIELD NAME	NULL	OCCURS	USAGE	PICTURE
03	IVEN-LAST-USER-ID	Y		DISPLAY	X(008)
03	IVEN-LAST-UPDATE-DATE	Y		GROUP	
04	IVEN-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	IVEN-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	IVEN-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	IVEN-LAST-RACN	Y		COMP	9(004)
02	**IVEN-CALC-KEY**	N		GROUP	
03	**IVEN-ACS-ACCT-NUMBER**	N		DISPLAY	9(006)
03	**IVEN-VENDOR-BANK-REC-TYPE**	N		DISPLAY	X(001)
03	**IVEN-ACS-VENDOR-BANK-CODE**	N		DISPLAY	X(010)
02	IVEN-NAME-AND-ADDRESS	Y		GROUP	
03	IVEN-NAME-ADDRESS-DATA	Y	12	DISPLAY	X(20)
02	IVEN-VENDOR-DATA	Y		GROUP	
03	IVEN-VENDOR-NAME	Y		DISPLAY	X(020)
03	IVEN-VENDOR-ADDRESS	Y	3	DISPLAY	X(20)
03	IVEN-ACS-CONTACT-NAME	Y	2	DISPLAY	X(020)
03	IVEN-VENDOR-PHONE	Y	2	DISPLAY	X(020)
03	IVEN-AGENT-NAME	Y		DISPLAY	X(020)
03	IVEN-AGENT-ADDRESS	Y	3	DISPLAY	X(20)
02	IVEN-BANK-DATA	Y		GROUP	
03	IVEN-BANK-NAME	Y	2	DISPLAY	X(20)
03	IVEN-BANK-PHONE	Y	2	DISPLAY	X(020)
03	IVEN-BANK-ADDRESS-1	Y	3	DISPLAY	X(020)
03	IVEN-BANK-ADDRESS-2	Y	3	DISPLAY	X(020)
03	FILLER	Y		DISPLAY	X(40)
02	IVEN-ACTIVE-VENDOR-FLAG	Y		DISPLAY	X(001)
02	IVEN-NOTES-49	Y		DISPLAY	X(049)
02	IVEN-BANK-ACS-CTRY-CODE	Y	15	DISPLAY	X(003)
02	FILLER	Y		DISPLAY	X(024)

ISHP-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ISHP-LAST-UPDATE-GROUP	Y	GROUP	
03	ISHP-LAST-USER-ID	Y	DISPLAY	X(008)
03	ISHP-LAST-UPDATE-DATE	Y	GROUP	
04	ISHP-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ISHP-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ISHP-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ISHP-LAST-RACN	Y	COMP	9(004)
02	ISHP-ACS-FIRST-RCPT-DATE	Y	GROUP	
03	ISHP-ACS-FIRST-RCPT-YY	Y	DISPLAY	9(002)
03	ISHP-ACS-FIRST-RCPT-MM	Y	DISPLAY	9(002)
03	ISHP-ACS-FIRST-RCPT-DD	Y	DISPLAY	9(002)
02	ISHP-ACS-ACCT-NUMBER	Y	DISPLAY	9(006)
02	ISHP-IMP-FIRST-RCPT-DATE	Y	GROUP	
03	ISHP-IMP-FIRST-RCPT-YY	Y	DISPLAY	9(002)

03	ISHP-IMP-FIRST-RCPT-MM	Y	DISPLAY	9(002)
03	ISHP-IMP-FIRST-RCPT-DD	Y	DISPLAY	9(002)
02	ISHP-SORT-KEY	N	GROUP	
03	ISHP-ACS-SHIP-NUMBER-R	N	GROUP	
04	ISHP-ACS-SHIP-NUMBER-PLAN	N	DISPLAY	9(2)
04	ISHP-ACS-SHIP-NUMBER-ACT	N	DISPLAY	9(02)
02	ISHP-ACS-PO-NUMBER	Y	DISPLAY	X(016)
02	ISHP-SHP-ACCUM-ACT-QTY	Y	DISPLAY	S9(07)
02	ISHP-SHP-ACCUM-EST-QTY	Y	DISPLAY	S9(07)
02	ISHP-ACS-FIRST-RCPT-EA	Y	DISPLAY	X(001)
02	ISHP-IMP-FIRST-RCPT-EA	Y	DISPLAY	X(001)
02	ISHP-HM-TRANS-FLAG	Y	DISPLAY	X(001)
02	ISHP-ACCUM-ACT-CARTONS	Y	DISPLAY	S9(06)
02	ISHP-EST-DAYS-TO-ARRIVE	Y	DISPLAY	9(003)
02	ISHP-SHIP-COMPLETE-FLAG	Y	DISPLAY	X(001)

IVV-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IVV-LAST-UPDATE-GROUP	Y	GROUP	
03	IVV-LAST-USER-ID	Y	DISPLAY	X(008)
03	IVV-LAST-UPDATE-DATE	Y	GROUP	
04	IVV-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IVV-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IVV-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IVV-LAST-RACN	Y	COMP	9(004)
02	**IVV-CALC-KEY**	N	GROUP	
03	**IVV-ORIG-VV**	N	GROUP	
04	**IVV-ORIG-VESSEL-NAME**	N	DISPLAY	X(020)
04	**IVV-ORIG-VOYAGE**	N	DISPLAY	X(010)
03	**IVV-ORIG-PORT-CODE**	N	DISPLAY	X(003)
02	IVV-ACS-CARRIER-CODE	Y	DISPLAY	X(004)

ILOG-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ILOG-LAST-UPDATE-GROUP	Y	GROUP	
03	ILOG-LAST-USER-ID	Y	DISPLAY	X(008)
03	ILOG-LAST-UPDATE-DATE	Y	GROUP	
04	ILOG-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ILOG-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ILOG-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ILOG-LAST-RACN	Y	COMP	9(004)
02	**ILOG-IX-KEY**	N	GROUP	
03	ILOG-ACS-CTR-ID	Y	DISPLAY	X(013)
02	ILOG-OCEAN-DEP-DATE	Y	GROUP	

03	ILOG-OCEAN-DEPART-YY	Y	DISPLAY	9(002)
03	ILOG-OCEAN-DEPART-MM	Y	DISPLAY	9(002)
03	ILOG-OCEAN-DEPART-DD	Y	DISPLAY	9(002)
02	ILOG-ACS-GOH	Y	DISPLAY	X(001)
02	ILOG-CY-NON-ACS-CHARGES	Y	GROUP	
03	ILOG-CONSOL-FEE-RATE	Y	DISPLAY	9(4)V9(2)
03	ILOG-CONSOL-FEE-RATE-TYPE	Y	DISPLAY	X(001)
02	ILOG-MODE1	Y	DISPLAY	X(003)
02	ILOG-MODE2	Y	DISPLAY	X(004)
02	ILOG-DISCH-PORT-CODE	Y	DISPLAY	X(003)
02	ILOG-ACS-CHARGE-TYPE	Y	DISPLAY	X(003)

ICJU-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	ICJU-LAST-UPDATE-GROUP	Y	GROUP	
03	ICJU-LAST-USER-ID	Y	DISPLAY	X(008)
03	ICJU-LAST-UPDATE-DATE	Y	GROUP	
04	ICJU-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	ICJU-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	ICJU-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	ICJU-LAST-RACN	Y	COMP	9(004)
02	ICJU-OCEAN-BL-NUMBER	Y	GROUP	
03	ICJU-CARRIER-CODE	Y	DISPLAY	X(004)
02	ICJU-ACS-WEIGHT	Y	DISPLAY	9(005)
02	ICJU-ACS-CUBIC-MEASURE	Y	DISPLAY	9(2)V9(3)
02	ICJU-QTY-UNITS	Y	DISPLAY	9(007)
02	ICJU-SHIPMENT-CBM-TYPE	Y	DISPLAY	X(001)
02	ICJU-SHIP-NBR-CARTONS	Y	DISPLAY	S9(006)
02	ICJU-ACS-ACCT-NUMBER	N	DISPLAY	9(006)
02	ICJU-BL-FDP	Y	DISPLAY	X(003)
02	ICJU-PORT-OF-ENTRY	Y	DISPLAY	X(003)
02	ICJU-CARGO-RCPT-NUMBER	Y	DISPLAY	X(008)
02	ICJU-STYLE-CODE	N	DISPLAY	X(010)

IACT-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IACT-LAST-UPDATE-GROUP	Y	GROUP	
03	IACT-LAST-USER-ID	Y	DISPLAY	X(008)
03	IACT-LAST-UPDATE-DATE	Y	GROUP	
04	IACT-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IACT-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IACT-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IACT-LAST-RACN	Y	COMP	9(004)
02	IACT-SORT-KEY	N	GROUP	
03	IACT-STYLE-CODE	N	DISPLAY	X(010)
03	IACT-COLOR-CODE	Y	DISPLAY	X(003)
03	IACT-SIZE-CODE	Y	DISPLAY	X(007)

03	IACT-SKU-NUMBER	Y	DISPLAY	X(008)
02	IACT-COLOR-DESCRIPTION	Y	DISPLAY	X(010)
02	IACT-ACT-SHIP-UNITS	Y	DISPLAY	9(006)
02	IACT-ACT-ORIG-COUNTRY	Y	DISPLAY	X(003)
02	IACT-ACT-SHIP-CARTONS	Y	DISPLAY	S9(05)
02	IACT-CARTON-BREAKDOWN	Y	GROUP	
03	IACT-CARTON-PACKING	Y	DISPLAY	9(3)V9(2)
02	IACT-RCPT-LOC-CODE	Y	DISPLAY	X(005)
02	IACT-UPC	Y	DISPLAY	X(015)
02	IACT-UNIT-FOB-COST	Y	DISPLAY	9(4)V9(4)
02	FILLER	Y	DISPLAY	X(20)

ISTY-REC

LEVEL	FIELD NAME	NULL	OCCURS	USAGE	PICTURE
02	ISTY-LAST-UPDATE-GRO UP	Y		GROUP	
03	ISTY-LAST-USER-ID	Y		DISPLAY	X(008)
03	ISTY-LAST-UPDATE-DATE	Y		GROUP	
04	ISTY-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	ISTY-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	ISTY-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	ISTY-LAST-RACN	Y		COMP	9(004)
02	ISTY-STYLE-CODE	N		DISPLAY	X(010)
02	ISTY-STYLE-DESC	Y		DISPLAY	X(025)
02	ISTY-EST-DUTY-PCT	Y		DISPLAY	9(2)V9(2)
02	ISTY-MASTER-PACK-NUMBER	Y		DISPLAY	9(005)
02	ISTY-INNER-PACK-NUMBER	Y		DISPLAY	9(003)
02	ISTY-UNIT-FOB-COST	Y		DISPLAY	9(4)V9(4)
02	ISTY-UNIT-RETAIL-COST	Y		DISPLAY	9(4)V9(2)
02	ISTY-USER-DATE-1	Y		GOUP	
03	ISTY-USER-DATE-1-YY	Y		DISPLAY	9(002)
03	ISTY-USER-DATE-1-MM	Y		DISPLAY	9(002)
03	ISTY-USER-DATE-1-DD	Y		DISPLAY	9(002)
02	ISTY-ACS-CLASS-NUMBER	Y	2	DISPLAY	9(003)
02	ISTY-ACS-CLASS-NAME	Y	2	DISPLAY	X(015)
02	ISTY-SEASON-CODE	Y		DISPLAY	X(005)
02	ISTY-ACS-COUNTRY-CODE	Y		DISPLAY	X(003)

IPRD-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IPRD-LAST-UPDATE-GROUP	Y	GROUP	
03	IPRD-LAST-USER-ID	Y	DISPLAY	X(008)
03	IPRD-LAST-UPDATE-DATE	Y	GROUP	
04	IPRD-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IPRD-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IPRD-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IPRD-LAST-RACN	Y	COMP	9(004)

02	IPRD-IX-ACCT	Y	GROUP	
03	IPRD-ACS-ACCT-NUMBER	Y	DISPLAY	9(006)
03	IPRD-ACS-DEPT-NUMBER	Y	DISPLAY	X(005)
03	IPRD-ACS-VENDOR-NUMBER	Y	DISPLAY	X(010)
03	IPRD-STYLE-CODE	Y	DISPLAY	X(010)
03	IPRD-CANCEL-DATE	Y	GROUP	
04	IPRD-CANCEL-DATE-YY	Y	DISPLAY	9(2)
04	IPRD-CANCEL-DATE-MM	Y	DISPLAY	9(2)
02	IPRD-IMPORTER-OFFICE	Y	DISPLAY	X(03)

IDTE-REC

LEVEL	FIELD NAME	NULL	USAGE	PICTURE
02	IDTE-LAST-UPDATE-GROUP	Y	GROUP	
03	IDTE-LAST-USER-ID	Y	DISPLAY	X(008)
03	IDTE-LAST-UPDATE-DATE	Y	GROUP	
04	IDTE-LAST-UPDATE-YY	Y	DISPLAY	9(002)
04	IDTE-LAST-UPDATE-MM	Y	DISPLAY	9(002)
04	IDTE-LAST-UPDATE-DD	Y	DISPLAY	9(002)
03	IDTE-LAST-RACN	Y	COMP	9(004)
02	IDTE-CALC-KEY	N	GROUP	
03	IDTE-ARRIVE-DEST	N	GROUP	
04	IDTE-ARRIVE-DEST-DATE	N	GROUP	
05	IDTE-ARRIVE-DEST-YY	N	DISPLAY	9(002)
05	IDTE-ARRIVE-DEST-MM	N	DISPLAY	9(002)
04	IDTE-ARRIVE-DEST-EA	N	DISPLAY	X(001)
03	IDTE-CUSTOMS-ENTRY	N	GROUP	
04	IDTE-CUSTOMS-ENTRY-DATE	N	GROUP	
05	IDTE-CUSTOMS-ENTRY-DD	N	DISPLAY	9(002)
04	IDTE-CUSTOMS-ENTRY-EA	N	DISPLAY	X(001)
04	IDTE-USER-DEFINED-3-EA	N	DISPLAY	X(001)
03	IDTE-CARGO-MISC-FIELD1	N	DISPLAY	X(012)
03	IDTE-NOTES-78	N	DISPLAY	X(078)
02	FILLER	Y	DISPLAY	X(018)

ISKU-REC

LEVEL	FIELD NAME	NULL	OCC URS	USAGE	PICTURE
02	ISKU-LAST-UPDATE-GROUP	Y		GROUP	
03	ISKU-LAST-USER-ID	Y		DISPLAY	X(008)
03	ISKU-LAST-UPDATE-DATE	Y		GROUP	
04	ISKU-LAST-UPDATE-YY	Y		DISPLAY	9(002)
04	ISKU-LAST-UPDATE-MM	Y		DISPLAY	9(002)
04	ISKU-LAST-UPDATE-DD	Y		DISPLAY	9(002)
03	ISKU-LAST-RACN	Y		COMP	9(004)
02	ISKU-COLOR-CODE	Y		DISPLAY	X(003)
02	ISKU-COLOR-DESCRIPTION	Y		DISPLAY	X(010)
02	ISKU-SIZE-CODE	Y		DISPLAY	X(007)
02	ISKU-SKU-NUMBER	Y		DISPLAY	X(008)
02	ISKU-UNPLANNED-SKU-FLAG	Y		DISPLAY	X(001)
02	ISKU-SKU-QTY	Y		DISPLAY	9(006)

System Implementation And Testing

5.1 SYSTEM IMPLEMENTATION

Implementation is the stage in which the theoretical design is transferred into a working system. This involves careful planning, investigation of current system and its constraints on implementation, design of methods to achieve the change over, training of staff in the change over procedures and evaluation of changeover methods.

Since the HR Department of three branches located at different geographical area uses the system, proper implementation planning is done for the methods and time scale. Once the planning has been completed, the system is tested so as to ensure that the programs in the system are working properly.

To achieve the objectives and benefits expected from the new system it is essential for the people who will be involved to be confident of their role in the new system. So at this stage the emphasis must be on training in new skills to the staff so that they can cope with the new system. Once the staff has been trained, the system and its whole environment are tested and the changeover can begin.

5.2 SYSTEM TESTING

System testing plays a major role before implementing the new system so as to ensure that the system work at all levels and is effective before live operation starts. The rest should take place as far as possible in the actual operating environment, and also should test the people as well as programs.

The logical design and the physical design should be thoroughly and continually examined on paper to ensure that they work successfully after implementation. Thus the system test in implementation should be a confirmation that all is correct and an opportunity to show the user that the system works. When we have tested each program individually, using the test data designed by us and have verified that these programs link together in the way specified in the program suite specification, the complete system and its environment was tested to the satisfaction of the system analyst and the user.

5.3 CHANGE OVER

Once all the proprietary work of implementation has taken place-the system has been tested and the staff trained. Changeover is the stage of moving over from the old to the new system. The application is put up on production and is tested for the accuracy, response time and other criteria that may affect when this is launched on production.

6. CONCLUSION

The tool named **DATA MIGRATION** ~~**TOOL TO PORT DATA FROM PRODUCTION TO TEST**~~ has almost got rid of the nightmare of the programmers of reporting the bug after the program has been moved to the production region. It will definitely help to avoid the bugs related to data validation, etc. which are related to the database. The testing programmer can use the tool to download the data from the database, which is of need by giving the calc key in the input file. The tool is already being used for downloading database and testing certain programs.

Scope for Future Development

7. SCOPE FOR FUTURE DEVELOPMENT

In the present tool the input to the tool has to be feed to the input file. This can be a tiresome process for a person new to the M.V.S. environment. This problem can be rectified, by making the program online by using Customer Information Control System (CICS). This will definitely help the new programmers at ease with the tool by inputting through the screens.

Bibliography

1. **MAINFRAME CONCEPTS :-**

   Covansys Training Materials.

2. **COBOL PROGRAMMING: -**

   Stern and Stern.

   D. Gosh Dastidar.

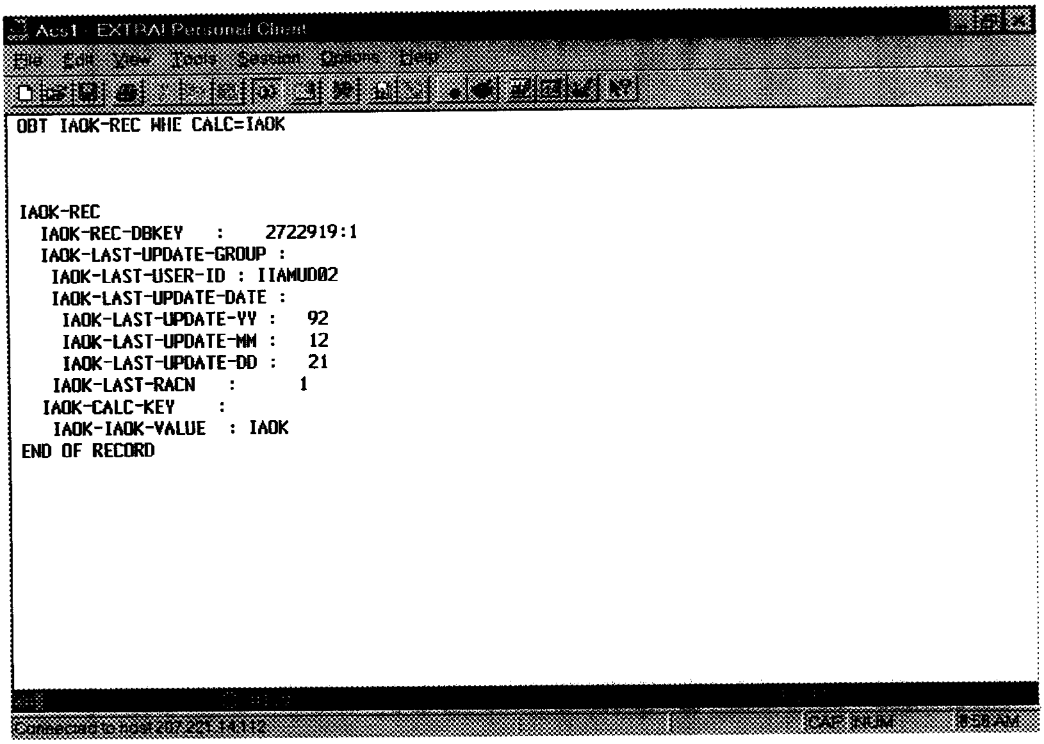
   M. K. Roy.

3. **INTEGRATED DATABASE MANAGEMENT SYSTEM(IDMS):-**

   Online Tool by First Class System Corporation.

Appendices

RECORDS FROM TEST DATABASE
IAOK-REC

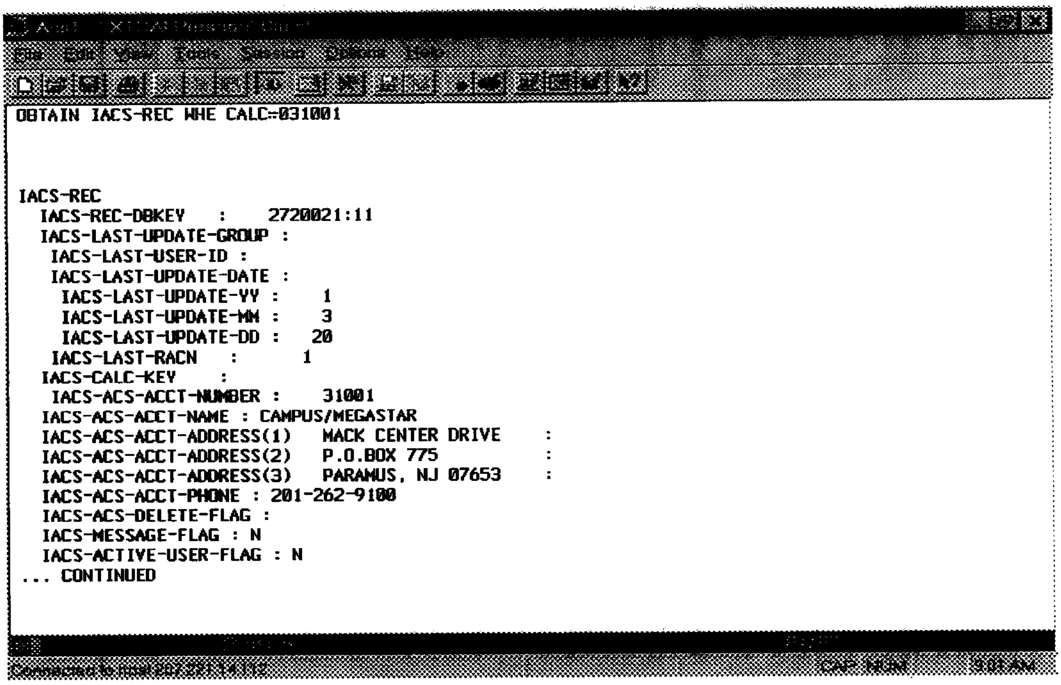

Acs1 - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 207.221.14.112
CAP NUM
8:58 AM

```
OBT IAOK-REC WHE CALC=IAOK



IAOK-REC
  IAOK-REC-DBKEY    :     2722919:1
  IAOK-LAST-UPDATE-GROUP :
   IAOK-LAST-USER-ID : IIAMUD02
   IAOK-LAST-UPDATE-DATE :
    IAOK-LAST-UPDATE-YY :    92
    IAOK-LAST-UPDATE-MM :    12
    IAOK-LAST-UPDATE-DD :    21
   IAOK-LAST-RACN    :       1
  IAOK-CALC-KEY     :
   IAOK-IAOK-VALUE   : IAOK
END OF RECORD
```

IACS-REC

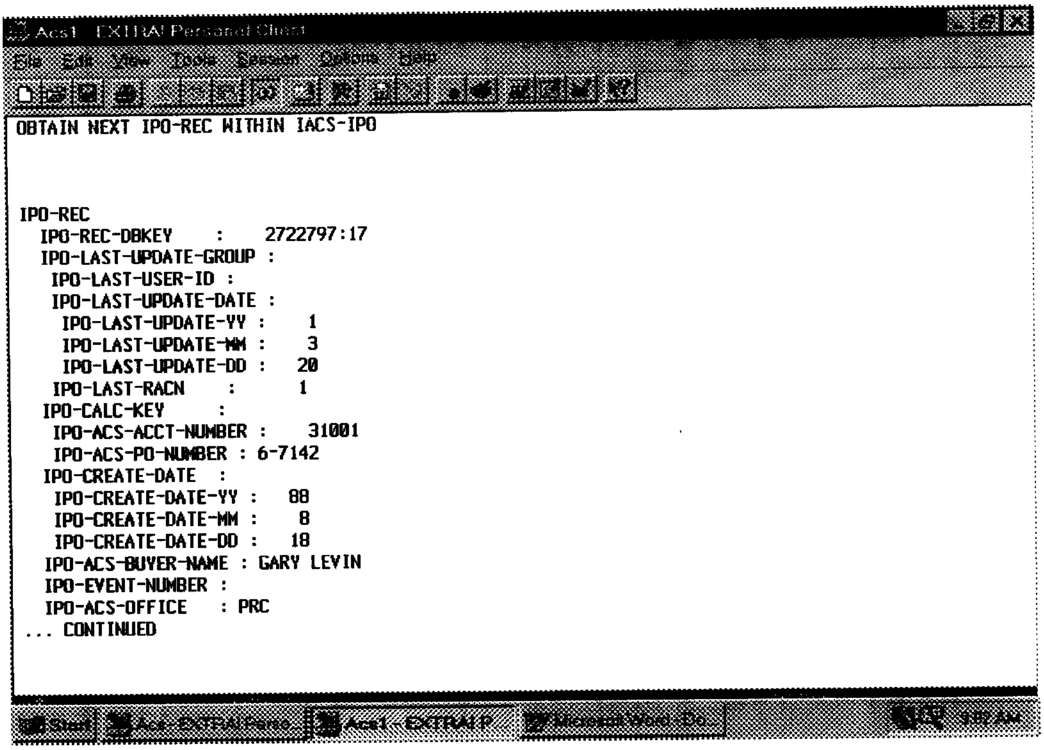

Acs1 - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 207.221.14.112
CAP NUM
9:01 AM

```
OBTAIN IACS-REC WHE CALC=031001



IACS-REC
  IACS-REC-DBKEY    :     2720021:11
  IACS-LAST-UPDATE-GROUP :
   IACS-LAST-USER-ID :
   IACS-LAST-UPDATE-DATE :
    IACS-LAST-UPDATE-YY :     1
    IACS-LAST-UPDATE-MM :     3
    IACS-LAST-UPDATE-DD :    20
   IACS-LAST-RACN    :       1
  IACS-CALC-KEY     :
   IACS-ACS-ACCT-NUMBER :     31001
  IACS-ACS-ACCT-NAME : CAMPUS/MEGASTAR
  IACS-ACS-ACCT-ADDRESS(1)    MACK CENTER DRIVE      :
  IACS-ACS-ACCT-ADDRESS(2)    P.O.BOX 775            :
  IACS-ACS-ACCT-ADDRESS(3)    PARAMUS, NJ 07653      :
  IACS-ACS-ACCT-PHONE : 201-262-9100
  IACS-ACS-DELETE-FLAG :
  IACS-MESSAGE-FLAG : N
  IACS-ACTIVE-USER-FLAG : N
... CONTINUED
```

IPO-REC

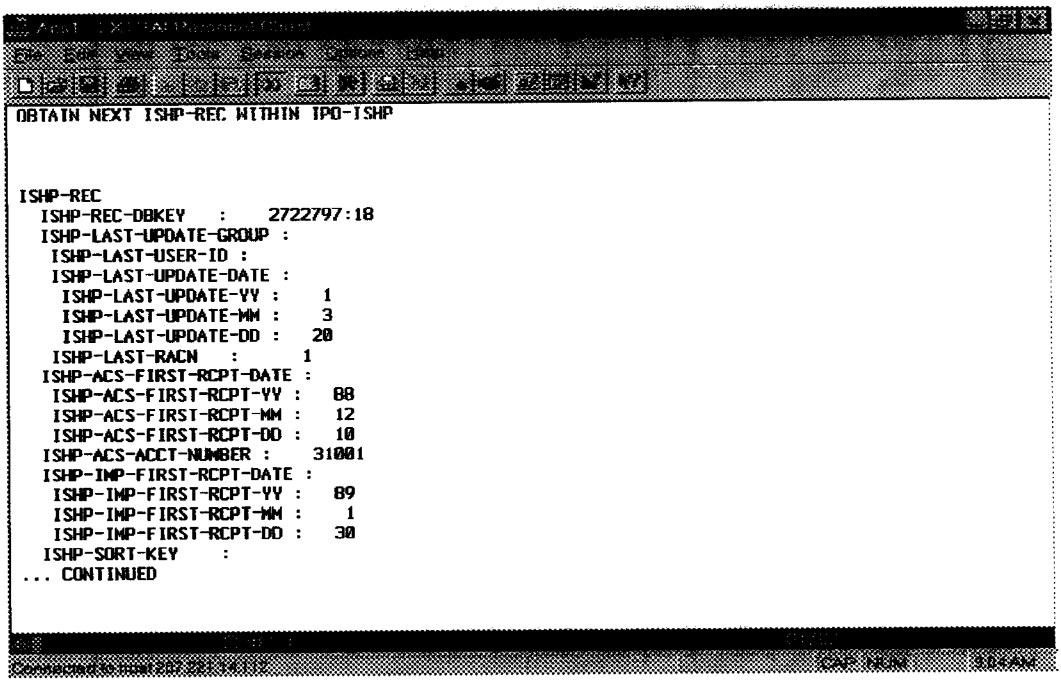


```
OBTAIN NEXT IPO-REC WITHIN IACS-IPO



IPO-REC
  IPO-REC-DBKEY      :    2722797:17
  IPO-LAST-UPDATE-GROUP :
   IPO-LAST-USER-ID :
   IPO-LAST-UPDATE-DATE :
    IPO-LAST-UPDATE-YY :    1
    IPO-LAST-UPDATE-MM :    3
    IPO-LAST-UPDATE-DD :   20
   IPO-LAST-RACN      :     1
  IPO-CALC-KEY       :
   IPO-ACS-ACCT-NUMBER :    31001
   IPO-ACS-PO-NUMBER : 6-7142
  IPO-CREATE-DATE   :
   IPO-CREATE-DATE-YY :   88
   IPO-CREATE-DATE-MM :    8
   IPO-CREATE-DATE-DD :   18
  IPO-ACS-BUYER-NAME : GARY LEVIN
  IPO-EVENT-NUMBER :
  IPO-ACS-OFFICE    : PRC
... CONTINUED
```

ISHP-REC

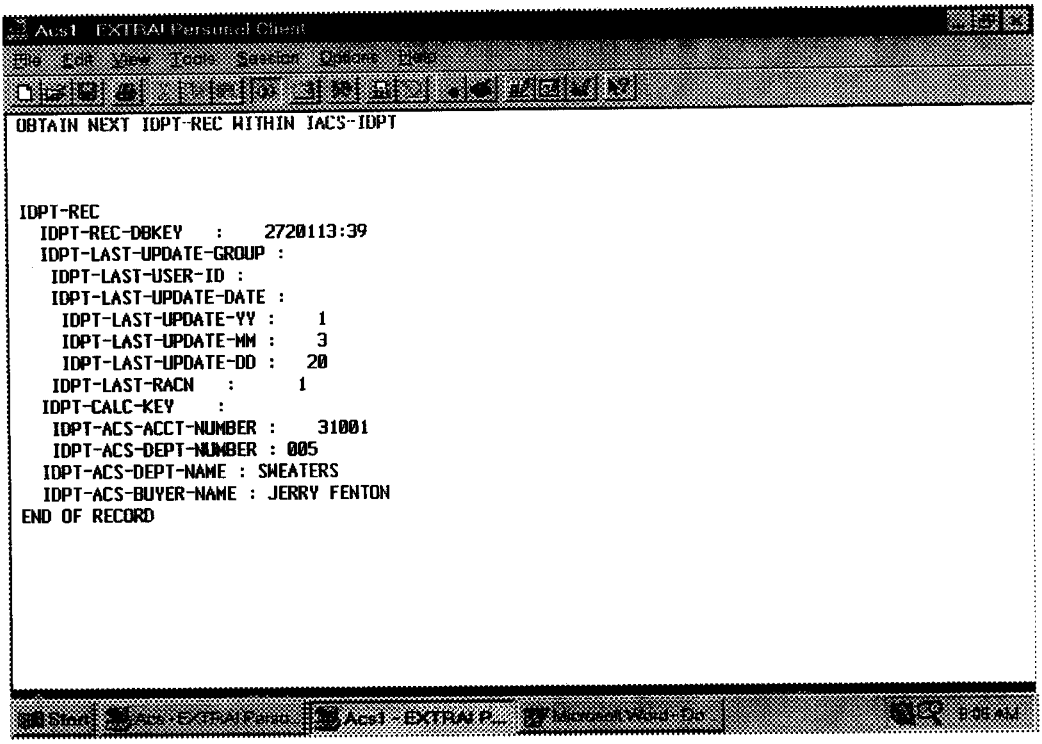


```
OBTAIN NEXT ISHP-REC WITHIN IPO-ISHP



ISHP-REC
  ISHP-REC-DBKEY     :    2722797:18
  ISHP-LAST-UPDATE-GROUP :
   ISHP-LAST-USER-ID :
   ISHP-LAST-UPDATE-DATE :
    ISHP-LAST-UPDATE-YY :    1
    ISHP-LAST-UPDATE-MM :    3
    ISHP-LAST-UPDATE-DD :   20
   ISHP-LAST-RACN     :     1
  ISHP-ACS-FIRST-RCPT-DATE :
   ISHP-ACS-FIRST-RCPT-YY :   88
   ISHP-ACS-FIRST-RCPT-MM :   12
   ISHP-ACS-FIRST-RCPT-DD :   10
  ISHP-ACS-ACCT-NUMBER :    31001
  ISHP-IMP-FIRST-RCPT-DATE :
   ISHP-IMP-FIRST-RCPT-YY :   89
   ISHP-IMP-FIRST-RCPT-MM :    1
   ISHP-IMP-FIRST-RCPT-DD :   30
  ISHP-SORT-KEY      :
... CONTINUED
```

IDPT-REC

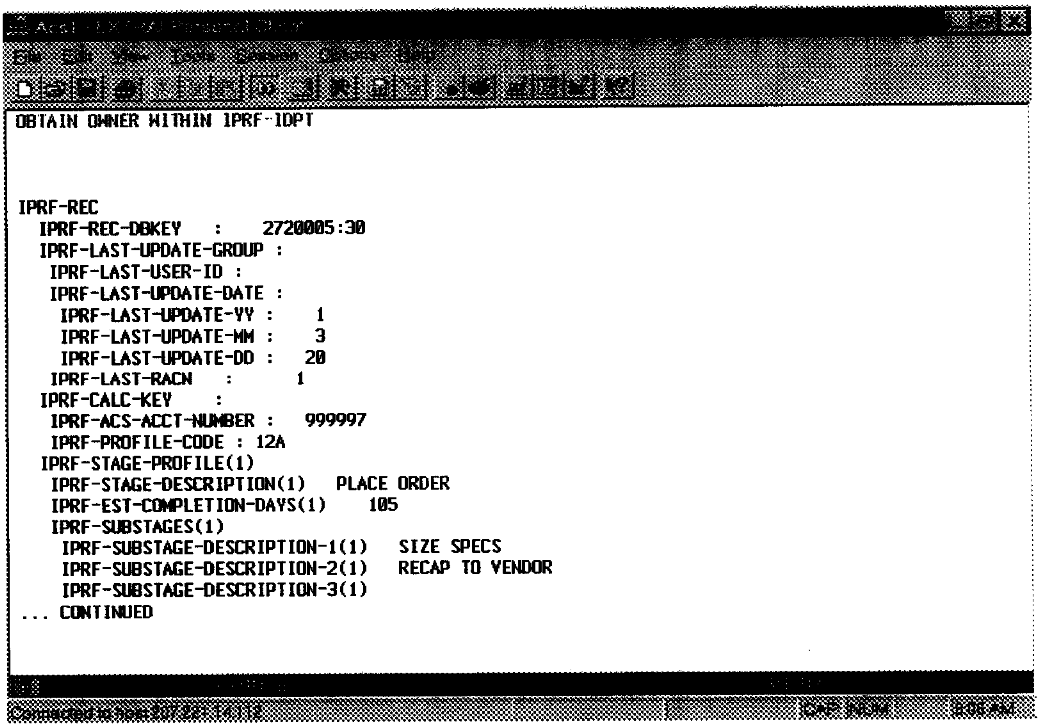


```
OBTAIN NEXT IDPT-REC WITHIN IACS-IDPT



IDPT-REC
  IDPT-REC-DBKEY    :     2720113:39
  IDPT-LAST-UPDATE-GROUP :
   IDPT-LAST-USER-ID :
   IDPT-LAST-UPDATE-DATE :
    IDPT-LAST-UPDATE-YY :     1
    IDPT-LAST-UPDATE-MM :     3
    IDPT-LAST-UPDATE-DD :    20
   IDPT-LAST-RACN    :       1
  IDPT-CALC-KEY     :
   IDPT-ACS-ACCT-NUMBER :    31001
   IDPT-ACS-DEPT-NUMBER : 005
  IDPT-ACS-DEPT-NAME : SWEATERS
  IDPT-ACS-BUYER-NAME : JERRY FENTON
END OF RECORD
```

IPRF-REC

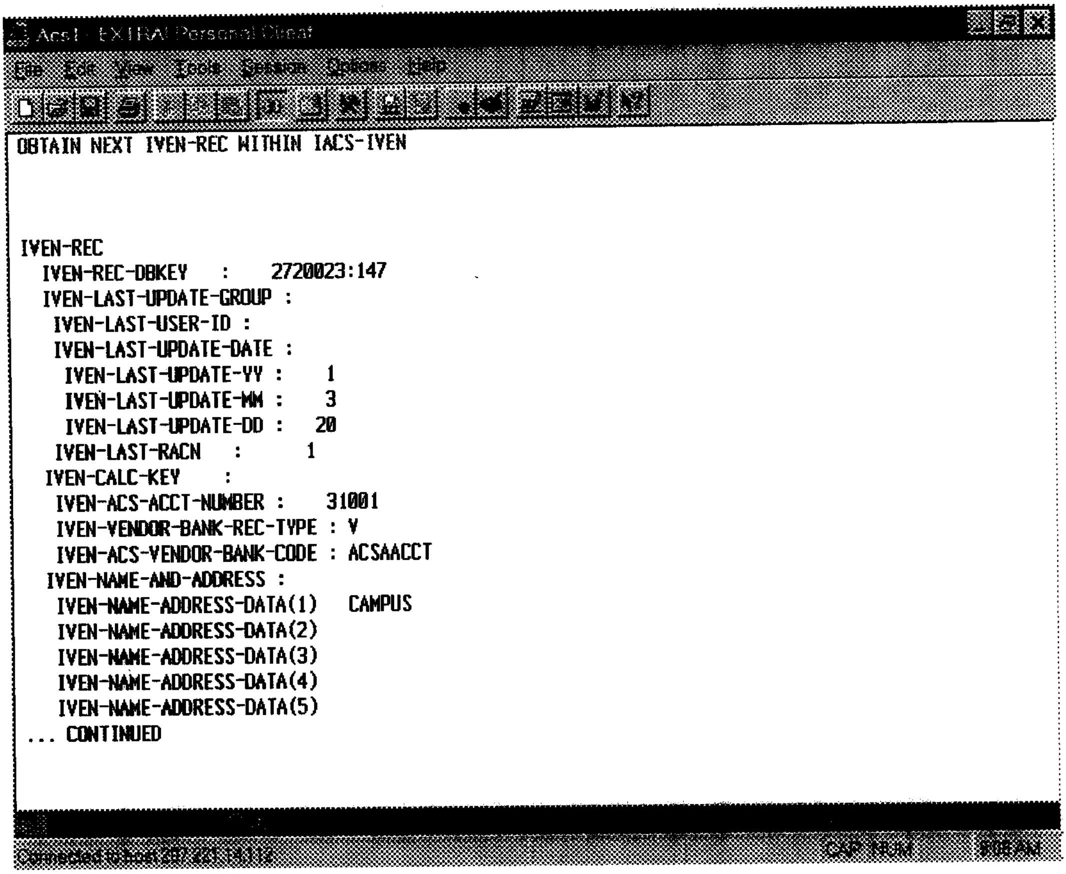

Acs1 - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 207.221.14.112
CAP NUM
9:08 AM

```
OBTAIN OWNER WITHIN IPRF-IDPT



IPRF-REC
  IPRF-REC-DBKEY    :     2720005:30
  IPRF-LAST-UPDATE-GROUP :
   IPRF-LAST-USER-ID :
   IPRF-LAST-UPDATE-DATE :
    IPRF-LAST-UPDATE-YY :     1
    IPRF-LAST-UPDATE-MM :     3
    IPRF-LAST-UPDATE-DD :    20
   IPRF-LAST-RACN    :       1
  IPRF-CALC-KEY     :
   IPRF-ACS-ACCT-NUMBER :    999997
   IPRF-PROFILE-CODE : 12A
  IPRF-STAGE-PROFILE(1)
   IPRF-STAGE-DESCRIPTION(1)   PLACE ORDER
   IPRF-EST-COMPLETION-DAYS(1)     105
   IPRF-SUBSTAGES(1)
    IPRF-SUBSTAGE-DESCRIPTION-1(1)   SIZE SPECS
    IPRF-SUBSTAGE-DESCRIPTION-2(1)   RECAP TO VENDOR
    IPRF-SUBSTAGE-DESCRIPTION-3(1)
... CONTINUED
```

IVEN-REC



Acs1 - EXTRA! Personal Client
File Edit View Tools Session Options Help
Connected to host 207.221.14.112
CAP NUM
9:08 AM

```
OBTAIN NEXT IVEN-REC WITHIN IACS-IVEN



IVEN-REC
  IVEN-REC-DBKEY    :    2720023:147
  IVEN-LAST-UPDATE-GROUP :
   IVEN-LAST-USER-ID :
   IVEN-LAST-UPDATE-DATE :
    IVEN-LAST-UPDATE-YY :    1
    IVEN-LAST-UPDATE-MM :    3
    IVEN-LAST-UPDATE-DD :   20
   IVEN-LAST-RACN    :      1
  IVEN-CALC-KEY     :
   IVEN-ACS-ACCT-NUMBER :    31001
   IVEN-VENDOR-BANK-REC-TYPE : V
   IVEN-ACS-VENDOR-BANK-CODE : ACSAACCT
  IVEN-NAME-AND-ADDRESS :
   IVEN-NAME-ADDRESS-DATA(1)   CAMPUS
   IVEN-NAME-ADDRESS-DATA(2)
   IVEN-NAME-ADDRESS-DATA(3)
   IVEN-NAME-ADDRESS-DATA(4)
   IVEN-NAME-ADDRESS-DATA(5)
... CONTINUED
```