10/15/91 W.B. 1

9/18/91

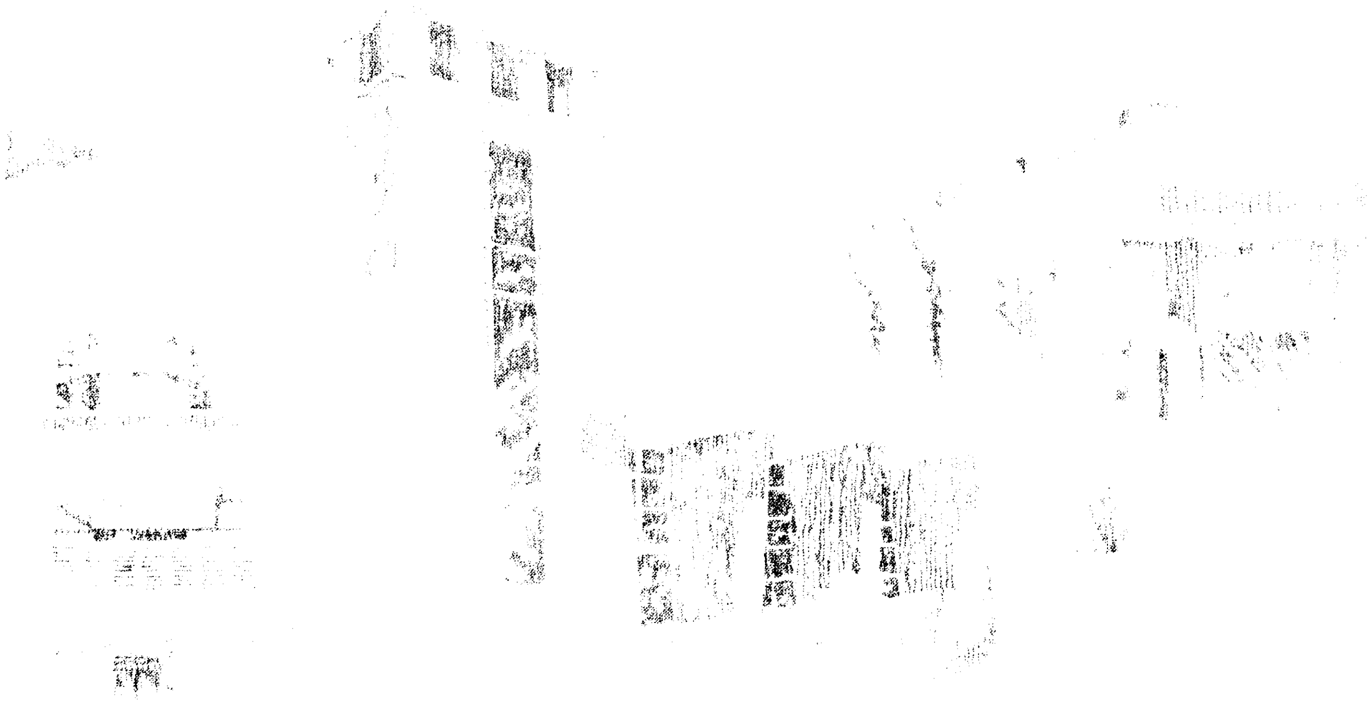
SANDIA REPORT

SAND91-1167 • UC-906
Unlimited Release
Printed August 1991

A Sandia Telephone Database System

Spencer D. Nelson, Lawrence F. Tolendino

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

SAND91-1167
Unlimited Release
Printed August 1991

A Sandia Telephone Database System

Spencer D. Nelson and Lawrence F. Tolendino
Computer Communications Design, Division 1934
Sandia National Laboratories, Albuquerque NM 87185

Abstract

Sandia National Laboratories, Albuquerque, may soon have more responsibility for the operation of its own telephone system. The processes that constitute providing telephone service can all be improved through the use of a central data information system. We studied these processes, determined the requirements for a database system, then designed the first stages of a system that meets our needs for work order handling, trouble reporting, and ISDN hardware assignments. The design was based on an extensive set of applications that have been used for five years to manage the Sandia secure data network. The system utilizes an Ingres database management system and is programmed using the Application-By-Forms tools.

Acknowledgement

The authors thank M. M. Miller of our division for his support in installing and maintaining the database system software.

Table of Contents

Introduction

Sandia National Laboratories (Sandia) is located on Kirtland Air Force Base (KAFB) in Albuquerque, New Mexico and as a base tenant shares the telephone system with the U. S. Air Force (USAF), Department of Energy (DOE) offices and other base tenants. The USAF is responsible for maintaining telephone service for the entire base; however, Sandia funds its portion of the system based on the number of lines installed and estimated usage.

Recently the USAF began a telephone system upgrade project that will bring digital telephone service to all of KAFB. As part of the system upgrade, switch hardware will be located in Sandia facilities allowing Sandia to exercise some administrative control of those portions of the telephone system. In order to perform administrative functions, it is necessary to have accurate information available regarding existing telephone services, available hardware, and cabling. The vehicle chosen to provide this information is a relational database system that will document existing services, track the progress of work orders, aid in resolving trouble reports, and help in assigning Integrated Services Digital Network (ISDN) hardware.

This paper outlines the process we used to create and implement the database system and highlights the basic elements of the database application program.

System Requirements

The first step in implementing the telephone system database was to determine the system requirements. Our preliminary examination of the processes that make up telephone service at Sandia showed us that the service work order process was the most visible and important to the telephone users and had the most promise for improvement. The conclusion was that a carefully planned telephone service work order system would improve customer service while reducing the efforts of the telephone service technicians. In addition to examining the service work order process, the telephone service record system and the trouble ticket systems were also identified as targets for improvement. The examination revealed that both of these processes could also benefit from an automated data information system.

Process Requirements in More Detail

Work Order Functions - Processing a work order, a request for some service by a telephone user, can be improved in several areas by use of a database system.

Using the Integrated Services Digital Network (ISDN) portions of an AT&T 5ESS telephone system requires complex and careful record keeping in the area of equipment module assignments. Telephone system design engineers, customer-site system administrators, and switch field engineers all tell us that an automated system external to the 5ESS is essential to effectively using ISDN, even for the assignment of secretarial coverage telephones.

Availability of complete cable and wire records can speed up the installation of telephones. Without complete records, installation of a telephone requires manual tracing of building wiring to locate the proper wire for a designated jack. Complete records can completely eliminate the wire tracing step in the installation. And that type of record-keeping is standard in commercial telephone operations today.

Perhaps most importantly, a computerized work order function can put information at the disposal of responsible individuals more quickly. The parallel work efforts needed in most jobs can be recognized. For example, a typical telephone installation will require several distinct functions such as 1) determining what equipment is needed and checking stock, 2) determining if the cables and jacks are in place and requesting plant work if they are not, 3) updating the database of the switching system, 4) doing any field connection work. A person doing any one of the individual functions often needs to know if one or more of the other functions are complete. Looking at an on-line record of the order is certainly faster than attempting to contact the other individuals one-by-one or waiting for a completed form to arrive in the office mail.

Making future improvements to the work order process will constantly require answers to questions like: How long does it take to get the equipment ready for an installation? Which part of the process is the most lengthy? What is the expected process interval for an order? Are we getting the order done on the day requested by the customer? Analysis of orders to get answers to those type of questions is much more easily done if the raw information is recorded as a standard part of the everyday working routine.

Trouble Tickets - When a customer has telephone problems and calls the extension assigned for trouble reports, he wants the problem solved as quickly as possible. To solve the problem fast, the answering technician may need information about the customer's line: location of the telephone, its corresponding line card location in the switch, cable pair information, service record, etc. Ideally, all that should be available knowing only the telephone number of the line in trouble.

How long does it take to solve the average problem? How many times does a customer need to call before the problem is completely resolved? Do we get many calls that map to lines on the same switch line card? How many calls require a technician to be dispatched to the field? These are questions that are more easily answered with an automated record system.

Conduit and Cable Assignments - If one duct in an 8-way conduit system is needed for supplying telephone service to a building to be completed a year from now, how can we be sure the duct will be available at that time? Are there wire pairs free for two more phones in an existing office? Complete central conduit and cable records annotated whenever any work is planned or completed can eliminate routine manual searches of wire closets and speed telephone installation service. In the case of a multi-use conduit and manhole system, a central record system tied to the plant drawings and accessible by all responsible individuals will allow engineers to find out if facilities are already assigned or if facilities are undergoing modification by another project team.

Switch Hardware Upgrades - Some small scale hardware upgrades that resulted in changing some telephone numbers to the 845 prefix from the 844 and 846 prefix, showed the importance of complete information. A lack of accurate records caused disruption of phone service to many of those affected customers. Because of poor records, it was difficult to notify customers of possible changes or verify the use of many lines; for example, certain facsimile-connected lines, lines in laboratory areas, and numbers recently changed. It is imperative that data be assembled and kept up-to-date for periodic hardware upgrades.

Build or Buy?

An extensive survey and trial of commercial software failed to turn up a system that would have cost or performance advantages over building a data information system based upon a similar system Sandia has been using to manage its secure data communications. The secure data communications system has the same core of customers - employees - as the telephone system, incorporates a copper and optical fiber distribution system as extensive as the telephone system, and has two large PBX systems as well as Ethernet and other high-speed data networking products.

The basic dilemma is that commercial systems are either too small, based on a single-user PC, or they are large systems that computerize paper-based systems and require more support staff than could be assigned given the service improvement that would be obtained.

Of the larger data systems, many of the functions are difficult to use and have software modules without good documentation and with many operational bugs. The implied method of usage for work orders would have a printed copy of the work order circulated to the technicians who do cable and telephone installation work, then passed back to someone to close the order in the database. One system, although it interfaces to the 5ESS for updating line features, does not read data from the switch for updating its information. This means that changes made on the normal switch consoles are not reflected in the database. This characteristic removes useful tools from the hands of trained switch field engineers. On the positive side, that system performs automatic equipment and cable assignments balancing the circuit load across the telephone switch hardware.

The typical commercial system would need the addition of several customized database tables and screen based applications to augment the system in the areas of tracking the progress of service requests, helping the customer service representatives in translating customer requests to hardware features, dealing with point-to-point communications circuits, and validating database information. Also, programming of specialized reports would be required. A staff of three people might be necessary to support the data information system in addition to maintenance from the supplying firm.

To gain our greatest process improvements we must 1) recognize parallel work efforts, 2) allow the switch and cable field engineers to use all tools at their disposal and make all decisions they can in work scheduling, 3) extract data from the switch and other records, 4) have an aid to ISDN hardware assignment. Our best way to reach those goals was to build a core database system by extensively drawing upon the five years of work in this area on the processing system used in the secure data communications area. The procedures used to process orders should start out rudimentary and should be augmented only while they are being used for at least a small portion of telephone service functions. As we learned in organizing the prior system, the people using the procedures and tools every day in their job must be able to give early feedback to the system designers.

Database System - Performance and Operational Goals

Given the database system requirements, the evaluation of commercial products, and previous experience with the development of a database to support data communications circuits, a set of system goals was developed. The first, and in most ways the most important goal, is to develop a system that provides the responsible individuals with the means to document work as it is being performed. A second goal is to create a system that allows easy information exchange through the computer rather than via paperwork passed hand-to-hand. A third goal for the system is to create a database that is reconciled with the telecommunications switch by downloading information from the switch. Instead of loading data from the database to the switch and assuming that the switch and database are reconciled, the database must query the switch for information and modify the database to match. The telecommunications switch is the ultimate authority on the network configuration. Combining these goals yields a system with unique characteristics.

There are two additional performance considerations. One is to have a system that is accessible to many technicians **simultaneously.** Another is that system must have foundation management software that insures that we **cannot** lose any transactions that have been entered by the technicians.

Sandia Developed Telephone Database System

The Sandia telephone database system was designed to meet our operational requirements and goals. The system is based on the five years of experience gained with a similar database system used to document the operation of the Sandia secure data communications network. A Digital Equipment Corporation VAX 8350 running VMS and the Ingres relational database software served as the development platform and the application software was developed using the Ingres product, ABF (Applications-by-Forms), which is a 4GL implementation. The tables and database queries were created using SQL, the IBM developed Standard Query Language.

Structurally, the database system consists of the database application program and the set of database tables. The application and the database are named PHONEDB. Functionally, the database application can be divided into three parts: work order processing, trouble report processing, and direct manipulation of system records. See Figure 1.

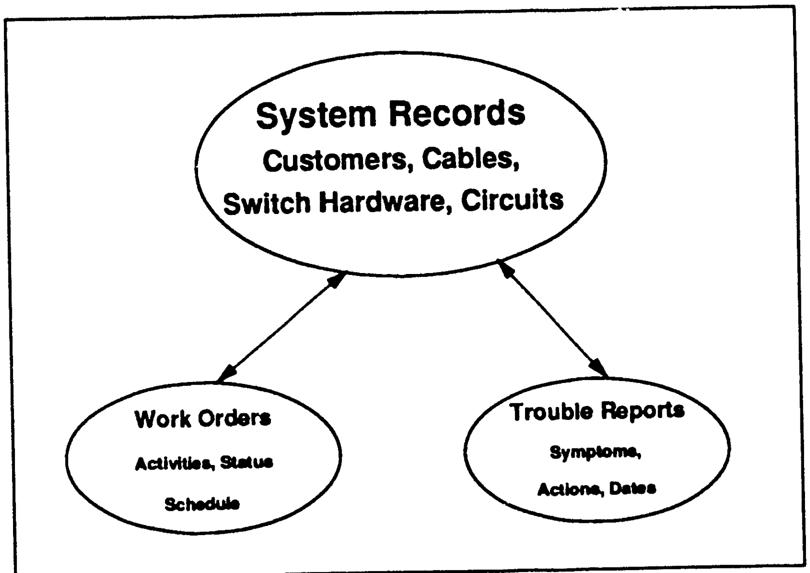


Figure 1 - Database Architecture

Work Order Processing

Work order processing consists of a set of database tables and database application screens that provide the environment to record customer service requests. Once a request is entered, progress can be monitored. The primary work order entry screen provides a structured environment for recording the customers' requirements without resorting to paper forms. This activity is meant to be an interactive process between the customer and technician accepting the order. Such a process serves to eliminate ambiguities and provide the customer with positive feedback.

This process flow is meant to be essentially paperless with the customer request being placed by phone and entered directly into the database. A confirmation memo can be sent to the customer to provide feedback. Also, a technician may print working papers, but these can be discarded when the task is done.

Trouble Report Processing

The screens supporting trouble reports are intended to be a fast tool for the use of someone answering a trouble report phone number. Calling customer number and circuit identification are entered, additional information is extracted from the database, and trouble symptom and action is chosen from a menu. The call can be closed immediately if the problem is resolved, or a trouble ticket can be assigned to someone for further work. The resulting records can be sifted and searched in various ways. Each open report is closed by the assigned individual when the trouble is cleared.

Records - Circuit, Hardware, and Customer Information

Many individualized screens aid in the entry and viewing of the elements that make up the telephone system. Customers are uniquely identified by a customer number. We have initially used the company assigned employee number with the understanding that this will change to social security number soon. The system can assign a number for customers who do not already have a unique number. Cables are recorded with a name, pair or fiber number range, and end locations. A line class of service (COS) list is defined to identity the various types of telephone service as well as specialized circuits (for example, fire alarm circuits). The telephone switch hardware configuration is enterer into tables recording the telephone numbers themselves, line equipment number (LEN), type of equipment, protocol handlers, etc.

Future Directions

Further development of the database system may proceed in several directions. First the system will be enhanced to meet Sandia's AT&T 5ESS telephone switch needs for both the Albuquerque and Livermore sites. A direct connection from the database system to a 5ESS man-machine port will provide a method of reconciling the database to the 5ESS data. Using 5ESS configuration reports, the switch specific database tables will be updated with the latest information automatically. Periodic updates will be scheduled to ensure database accuracy. As development progresses, a method will be added to update the 5ESS from database tables.

A second area of further development would add increased functionality to the application programs. In addition to work order and trouble call processing, the database could support telephone directory, billing, and inventory functions. These functions could easily be added to the system as the basic records data provides the necessary information.

We also expect to move a copy of the application programs and database to a small Unix computer sometime in the Fall of 1991.

Appendix I -- System Data Entry Screens

A few of the system's screens are represented here to illustrate the general layout of the presentation to the user. Additional screens are shown in the "Getting Started" appendix.

```
(Application Main Menu Frame)

                         +--------------------------------+
                         |            PhoneDB             |
                         |Telecommunications Data System  |
                         +--------------------------------+


       Select ->  RECORDS        to view or update circuit, customer,
                                 cable entries, & switch configuration


                  WORK ORDERS    to enter, update, or close work orders


                  TROUBLES       to enter, update, or close trouble reports



                     (c) 1991 Sandia National Laboratories
```

```
(System Records -- Main Menu Frame)

                           +---------------+
                           |System Records|
                           +---------------+

select->     CUSTOMERS to search/modify customer descriptions.

             LINES to search/modify circuit descriptions.

             CABLES to search/modify cable descriptions.

             SWITCH to search/modify the switch configuration

             E_CABLES to search/modify special entry cable connections

             COS to search/modify line classes of service
```

(Search/Modify Telephone Circuit Information Frame)

```
                     +----------------------------------------+
                     |Circuit Customer, Location, & Equipment|
                     +----------------------------------------+

Circuit Id: 00400012          Telephone No: 8444448       Location: AREA IV
Customer No: E00021367      Station Equip: STD            Building: 980
Name: NELSON, SPENCER D.         Line Cos: I1                 Room: 105
                                                              Jack: 1469

                     +----------------------------------+
+------------------+ |                                  |------------+------------+
|Keyword/Cable Name|        +------------------+        |ol 3 / Pr 3|Col 4 / Pr 4|
|------------------|        |ISDN Line Details|        |------------+------------+
|                  |                                    |            |            |
|                  | len (circ id): 00400012           |            |            |
|                  |  phone number: 8444448            |            |            |
|                  |PSU shlf PH ch: 000004             |            |            |
|                  |         DPIDB: 11                 |            |            |
|                  |            TS: 1                  |            |            |
|                  +-----------------------------------+            |            |
|                    |            |              |            |            |
+--------------------+------------+--------------+------------+------------+
```

(Select/Define Line Class of Service Frame)

```
+---------------------------+ +-----------------+
|Line COS - Class of Service| |Line Cos: I1     |  Card Type: U
+---------------------------+ +-----------------+
 Description:
  ISDN telephone, voice only, 2 wire U card, D channel used for
  dialing, no data module.

          pps D: 9                              pps B:
 Usage percent D: 5                   Usage percent B:

process: ISDN telephone only, no data port
```

Appendix II -- Database System Development Environment

The Sandia Telephone Database System is being developed using the Ingres relational database product running on a DEC Vax 8350 using the VMS operating system.

PHONEDB Account and Database General Information

Database PHONEDB is owned by user PHONEDB.

The file TABLES_DEF.TXT contains the commands to create all the tables in PHONEDB.

Source code for the applications is in directory ud:[phonedb] on VMS VAX SAV65. Temporarily some source code was developed on a 11/750 Ultrix VAX (directory /usr/user/phonedb of account phonedb). Moving the entire application and database between the Ultrix and VMS environments was accomplished in a matter of a few hours using the Ingres Unloaddb and Copydb commands. The query language is SQL.

The planned application, written using Ingres ABF tools, is called phonedb and initially includes records (circuit and administrative information on the phone connections), work orders (work order entry and tracking), and trouble reports (trouble ticket entry and tracking).

The long term plan is to do software development on one VAX (SAV65) with users and the actual phone data operating on another VAX (SAV80).

References for requirements

Memo dated January 23, 1991, to M. O. Vahle, 2934 from S.D. Nelson, 2934 "The Telephone Service Work Order Process and Some Suggested Improvements"

File report of S. D. Nelson, 2934, "The Telephone Data System Project" dated 1/4/91.

Ascii versions of the above references are on SAV65 in the phonedb directory as filenames

woproces.txt
dbargu.txt

Conventions for programming

Keep column names and table names directed toward a general telephone application rather than Sandia specific.

Make operation of forms as close as possible to TCCDATA and DCRS of the TCCDATA application.

Allow cables to run from one location to many locations, but with transition to only two ends per cable name.

Allow a "circuit" to have no phone number to allow for roving ISDN telephones. Allow for multiple number appearances per phone and a phone number appearing on more than one phone instrument.

The following definitions are the single Frskey definitions to use for the indicated menu items if they are present:

Menu Key	menu	= pf1
	help	= frskey2
	quit	= frskey3
	Get_no	= frskey9
	InRow	= frskey10
	DelRow	= frskey11
	Add, Save	= frskey13
	Drop	= frskey14

```
Change        = frskey15
Clear         = frskey16
Select, Pick= frskey17
Find          = frskey18
```

The following menu items should appear in the order given AT THE END of all menu items for the screen (if they appear at all).

Get_no
InRow
DelRow
Select
Find
Add or Save
Drop
Change
Clear
Help
Quit

Planned Application Development Order

Basic tables definition
Rudimentary "records"
Rudimentary "work orders"
Rudimentary ISDN line assignments
Rudimentary "trouble"
Update all functions
Continuing support

Maintaining Database Integrity During Circuit Assignment

While entering circuit information into the database, through database administrative screens or work order screens, it is imperative that the operator not destroy the database integrity. During the circuit assignment process various resources are selected for the circuit one-at-a-time. However, the resources are not "tied" to the circuit until the operator writes the information to the database with a "Save" command. This type of operation leads to the possibility of making partial assignments and creating orphaned resources. The need to allow multiple users to access the database simultaneously serves to exacerbate the situation.

In order to overcome these difficulties, the resource selection screens limit what the operator can do. The operator will only be able to select one resource of each type for any particular circuit. If a second resource of the same type is chosen the operator will be queried as to whether he wishes to change the resource assignment from the original. If the assignment is changed, the original resource will be returned to the status of "FREE". Once a resource is chosen it is marked "RESERVED" in the database. This "RESERVED" status is only intended to last for the duration of this work session and serves to provide concurrent operation for several people.

The "RESERVED" status of the selected resources will be reset to "FREE" if the operator aborts the session with a Clear or Quit command before using the Save command. The Save command replaces the "RESERVED" status with the circuit ID (usually the line equipment number) in each of the resource tables. The software determinations as to which resources should be returned to "FREE" status are made by examining the list of resources maintained in hidden fields. In case of a session abort those resources listed in the hidden fields that have the status "RESERVED" are returned to "FREE". These hidden resource fields would include:

htn	telephone number
hlen	line equipment number
hdpidb	hidden dpidb
hts	time slot
hqtr_ts	quarter time slot
hpsu	shelf, protocol handler, and member

hsm	switch module
hislu	ISDN line unit

The following figure illustrates the circuit assignment/resource selection process from the viewpoint of an operator working with the PHONEADM screen. For clarity not all user commands have been included. The example illustrates the various operations possible when choosing a telephone number for a circuit. A similar process would be used for choosing any other resource such as an LEN or protocol handler.

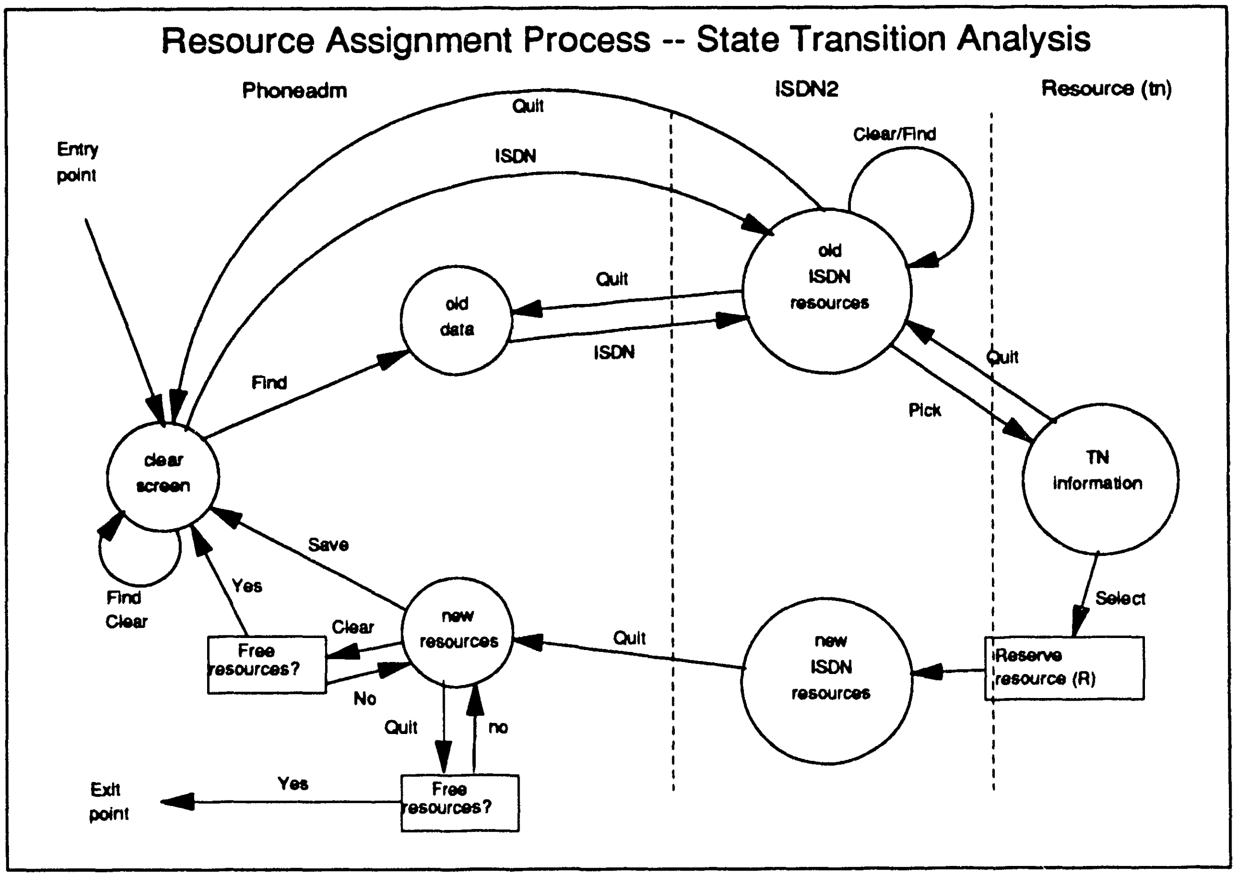


Figure 2 - Resource Assignment Process

Appendix III — Relational Database Table Structures

The following definitions are documented in a form compatible with the Ingres SQL query language and could be extracted from this document and passed to the Ingres interactive monitor program for execution. These procedures would create all the tables required by the database system.

```
/* TABLES_DEF */
/* Definitions to create ALL the tables for the database PHONEDB */


commit;
\g

create table customers
        (customer_no char(9) NOT NULL, /* customer number, ie. SSN */
        name char(24) NOT NULL,
        organization char(5),
        category char(4),
        classification char(10),
        tn char(7)) with noduplicates, journaling;
\g

commit;
\g

/* modify customers to isam on customer_no; */
/* create index name_index on customers(name) with structure=isam; */


commit;
\g

create table phones
        (len char(15) NOT NULL,  /* switch line equipment number OR unique ID*/
        tn char(10),             /* telephone number if assigned */
        customer_no char(9),
        line_cos char(5),        /* line class of service; ISDN, analog, etc. */
        location char(10),       /* geographic location */
        building char(6),
        room char(6),
        jack char(6),
        station_equip char(10)) /* station equipment model #, etc. */
         with noduplicates, journaling;
\g

commit;
\g

/* modify phones to isam on len; */
/* create index phone_no on phones(tn) with structure=isam; */
/* create index phone_cust on phones(customer_no) with structure isam; */


commit;
\g

create view cust_phones (
  len,
  tn,
  customer_no,
  line_cos,
  location,
  building,
  room,
  jack,
  station_equip,
  name,
  organization)
```

```
   as select phones.len, phones.tn, phones.customer_no,
             phones.line_cos, phones.location, phones.building,
             phones.room, phones.jack, phones.station_equip,
             customers.name, customers.organization
   from phones, customers
   where phones.customer_no = customers.customer_no;
\g

commit;
\g




create table cables
        (name char(20) NOT NULL, /* cable name */
        first_pair integer NOT NULL,
        last_pair integer NOT NULL,
        end1_loc  char(10),
        end1_bldg char(10),
        end1_room char(10),
        end1_cab char(8),
        end2_loc char(10),
        end2_bldg char(10),
        end2_room char(10),
        end2_cab char(8),
        type char(10) )
          with noduplicates, journaling;

\g

/* modify cables to isam on name; */

commit;
\g


create table conductors
        (name char(20) NOT NULL,
        element integer,
        len char(15))
          with noduplicates, journaling;
\g

/* modify conductors to isam on name, element */

commit;
\g


create table circuits
        (len char(15) NOT NULL,
        seq_no smallint,
        item char(20),
        col1 char(15),
        col2 char(15),
        col3 char(12),
        col4 char(12))
          with noduplicates, journaling;
\g

/* modify circuits to isam on len, seq_no; */
commit;
\g


/* The work order table has fewer fields than the work_order screen since
```

```
** some fields of the screen are extracted from the customers and phones
** tables using the fields that are in work_orders.  For example, org,
** name, and phone for the user are extracted from customers using the
** key user_cus_no as the customer number.
*/
create table work_orders (
order_no       integer NOT NULL,
pkg_no         integer NOT NULL,
request_type   char(25),    /* A choice from the wo_types table */
entry_date     date,        /* Date the new work order was entered */
sch_rout       char(1),     /* Scheduled or Routine (S/R) work order */
complete_by    date,
user_cus_no    char(9),     /* A customer number for the end user */
req_cus_no     char(9),     /* A customer number for the requester */
cur_tn         char(10),    /* Telephone number currently operating */
cur_len        char(15),
new_tn         char(10),    /* Telephone number to be assigned */
new_loc        char(10),    /* Location, to go into the phones table .. */
new_bldg       char(6),
new_room       char(6),
new_jack       char(6),
new_len        char(15),
new_cos        char(5),
new_model      char(10),
comments       char(210),
entry_by       char(8)      /* From login info when new entry was entered */
)
    with noduplicates, journaling;
\g
commit;
\g

create table wo_actions (
order_no       integer NOT NULL,
seq_no         smallint NOT NULL,
item           char(30) NOT NULL, /* An action item to-do */
assigned_to    char(20),   /* The person or group assigned to do the item */
assigned       date,
assigned_by    char(8),   /* Who put the date into assigned; from login info */
done           date,
done_by        char(8)    /* Who put the done date in; from login info */
)
    with noduplicates, journaling;
\g
commit;
\g

/* actionlist keeps the possible items that can go into the wo_actions table.
** The seq_no column is used to assign a number that will determine the
** relative order of display of the items on the work_orders screen.
*/
create table actionlist (
seq_no         smallint NOT NULL,
item           char(30) NOT NULL
)
   with noduplicates, journaling;
\g
commit;
\g

/* Table for work order type (used by work orders frame)
** and the actions to take in
** updating phones table according to
** work order request type (See notes in procedure close_wo.osq)
*/
create table wo_types (
request_type   char(25),
cur_action     char(8),   /* What to do for current len, tn */
new_action     char(8)    /* What to do for new len, tn */
)
  with noduplicates, journaling;
\g
commit;
\g

/* Miscellaneous integers used to form work order numbers, temporary
** customer numbers, trouble ticket numbers, etc. are held here.
*/
create table numbers (
type        char(10) NOT NULL,
numbers     integer
)
  with noduplicates, journaling;
commit;
\g

/* Table for closed work orders */
create table closed_wo (
order_no        integer NOT NULL,
pkg_no          integer NOT NULL,
request_type    char(25),     /* A choice from the wo_types table */
entry_date      date,         /* Date the new work order was entered */
sch_rout        char(1),      /* Scheduled or Routine (S/R) work order */
complete_by     date,
user_cus_no     char(9),      /* A customer number for the end user */
req_cus_no      char(9),      /* A customer number for the requester */
cur_tn          char(10),     /* Telephone number currently operating */
cur_len         char(15),
new_tn          char(10),     /* Telephone number to be assigned */
new_loc         char(10),     /* Location, to go into the phones table .. */
new_bldg        char(6),
new_room        char(6),
new_jack        char(6),
new_len         char(15),
new_cos         char(5),
new_model       char(10),
comments        char(210),
entry_by        char(8)       /* From login info when new entry was entered */
)
  with noduplicates, journaling;
\g
commit;
\g

create table clo_actions (
order_no        integer NOT NULL,
seq_no          smallint NOT NULL,
item            char(30) NOT NULL, /* An action item to-do */
assigned_to     char(20),   /* The person or group assigned to do the item */
assigned        date,
assigned_by     char(8),  /* Who put the date into assigned; from login info */
done            date,
done_by         char(8)   /* Who put the done date in; from login info */
)
  with noduplicates, journaling;
\g
commit;
\g

/* Table of 5ESS preconstructed features extracted from the CD-ROM
** documentation, Recent Change, Table J
*/
create table all_features (
feature  char(11),
description  char(198)
)
  with noduplicates, journaling;
\g
```

```
create table tn ( /* phone numbers and if they are assigned */
  tn      char(10),
  status  char(3)      /* F=free, A=active assigned, R=Reserved  */
  )
   with noduplicates, journaling;
\g
/* modify tn to btree on tn; */


create table len (  /* Line Equipment, Line Cards  */
   /* len for ISDN is sssillcc where sss=sm, i=islu, ll=lgc, cc=card          */
   /* len for analog is ssslcgwv sss=sm (1-192), l=lu (0-7), c=conc (0-9),    */
   /*                      g=swg (0-1), w=sw (0-7),v=lvl(0-3)                 */
 len       char(15) NOT NULL,
 card_type char(3),          /* T, U, Z, A=analog card channel */
 status    char(3)           /* F=free, A=active, assigned, R=reserved  */
 )
   with noduplicates, journaling;
\g
/* modify len to btree on len; */

/* Define a view that allows finding the used or free len for the 5ESS
   grouped by lgc or switch group - possible 32 lines each.
*/
create view len_trunc (len_trunc, card_type, status)
  as select
    left(len,6), card_type, status
    from len;
\g
commit;
\g
/* Get the number used or free with a query like this:
  select len_trunc, card_type, count(len_trunc)
   from len_trunc
   where status='A'  /* or 'F' for free */
   group by len_trunc, card_type
   order by len_trunc;
*/


create table dpidb (
 sm         smallint,  /* 1-192 */
 islu       smallint,  /* 0-7   */
 dpidb      smallint,  /* 2-11  */
 ts         smallint,  /* 0-31  */
 qtr_ts     smallint,  /* 0-3   */
 lgc        smallint,  /* Line Group Controller for time slot */
 len        char(15)   /* len if assigned, R=reserved for len in      */
                       /*  qtr_ts above, F=free                       */
 )
  with noduplicates, journaling;
 \g

 create table ph (   /* Protocol Handlers  */
  sm         smallint,  /* 1-192 */
  islu       smallint,  /* 0-7   */
  shelf      smallint,  /* 0-4   */
  ph         smallint,  /* 0-15  */
  ph_member  smallint,  /* 0-127 but only each 4th entered initially: 0,4,8 .. */
  len        char(15),  /* len if assigned, F=free */
  pps_t      float4,    /* pps x usage estimated; see line_cos table;0 if free */
  type       char(4)    /* D for D channels, B for on-demand B, A for all  */
  )
   with noduplicates, journaling;
 \g
 /* Get usage per ph with a query like:
   select sm, islu, shelf, ph, xx=sum(pps_t)
```

```
    from ph
    group by sm, islu, shelf, ph
    order by sm, islu, shelf, ph
*/


create table line_cos (
 line_cos     char(5) NQT NULL,  /* Line class of service: e.g. A1, I2  */
 reqd         integer,           /* Required service: see cos_procs tbl.*/
                                 /* 1-99  reserved for ISDN             */
                                 /* 101-199 for LU POTS lines           */
 pps_d        integer,           /* packets per second estimated        */
 usage_d      smallint,          /* percent of time-in-use estimated    */
 pps_b        integer,           /* packets per second estimated        */
 usage_b      smallint,          /* percent of time-in-use estimated    */
 card_type    char(3),           /* T, U, Z, or A (A=analog card)       */
 description  char(195)          /* Text like 'ISDN telephone, 4 wire T card */
                                 /* local power at phone, no data module */
 ) with noduplicates, journaling;
\g

commit;
\g

create table cos_procs (
  reqd        integer,           /* See reqd in line_cos table          */
  comment     char(70)           /* Text that displays instead of the   */
                                 /* reqd number for line_cos   */
  ) with noduplicates, journaling;
\g
commit;
\g

/* Special table for entry cables that map to logical/other cables
   to make the transition to a cable scheme of interconnected real
   cable (segments) from the scheme of logical cables that have a
   single name for a cable that has pairs peeling off to various buildings.
   See the table e_cablemap for the physical-logical mapping.
*/
create table e_cables
        (e_name char(20) NOT NULL, /* cable name */
        first_pair integer NOT NULL,
        last_pair integer NOT NULL,
        end1_loc  char(10) with null,
        end1_bldg char(10) with null,
        end1_room char(10) with null,
        end1_cab char(8) with null,
        end2_loc char(10) with null,
        end2_bldg char(10) with null,
        end2_room char(10) with null,
        end2_cab char(8) with null,
        type char(10) with null )
          with noduplicates, journaling;

\g
commit;
\g

create table e_cablemap
(     e_name char(20) NOT NULL,    /* from table e_cables   */
      e_pair1 integer NOT NULL,
      e_pair2 integer NOT NULL,
      name char(20) NOT NULL,     /* from table cables     */
      pair1 integer NOT NULL,
      pair2 integer NOT NULL)
       with noduplicates, journaling;
\g
commit;
\g
```

```
create table troubles (
number smallint,
len char(15),
customer_no char(9),
tn char(10),
date_reported date,
date_cleared date,
assigned_to char(24),
symptom char(30),
action char(30),
notes vchar (150),
comments vchar(150)) with noduplicates, journaling;
\g

create table trobl_act (action char(30) not null)
   with noduplicates, journaling;
\g

create table trobl_symp (symptom char(30) not null)
   with noduplicates, journaling;
\g

create table trobl_auto (number smallint, symptom char(30),
 action char(30)) with noduplicates, journaling;
\g


/* Table of miscellaneous information used as starting points for
   searching for resources in large tables, to spread resource
   usage across the switch, etc.
*/
create table misc (
  item char(15) not null with default,
  v1 integer not null with default,
  v2 float4 not null with default,
  v3 char(20) not null with default
);
\g


grant all on  customers to public;
grant all on  phones to public;
grant all on  cables to public;
grant all on  conductors to public;
grant all on  circuits to public;
grant all on  work_orders to public;
grant all on  wo_actions to public;
grant all on  actionlist to public;
grant all on  wo_types to public;
grant all on  numbers to public;
grant all on  closed_wo to public;
grant all on  clo_actions to public;
grant all on  all_features to public;
grant all on  tn to public;
grant all on  len to public;
grant all on  dpidb to public;
grant all on  ph to public;
grant all on  line_cos to public;
grant all on  cos_procs to public;
grant all on  e_cables to public;
grant all on  e_cablemap to public;
grant all on  troubles to public;
grant all on  trobl_act to public;
grant all on  trobl_symp to public;
grant all on  trobl_auto to public;
grant all on misc to public;
\g
commit;
\g
```

Appendix IV -- 5ESS ISDN Hardware Assignment Rules

Outline notes from various sources and observations:

Rules for the system configuration and equipped hardware:

Max of 16 LGCs per ISLU
Max of 32 Line Cards per LGC
Max of 12 PIDBs per ISLU (but one source also says 16)
Max of 12 DPIDBs per ISLU
Even number of PIDBs or DPIDBs
Max of 5 PSU shelves per ISLU
Max of 16 PHs per PSU shelf
Min of 8 DPIDBs per fully equipped ISLU
Max of 16 PIDBs per fully equipped ISLU with 8 DPIDBs
Max of 11, Min of 2 = assigned number of a DPIDB assigned to an ISDN line card

Rules for assigning equipment to a line:

Odd numbered (11, 09, 07, ..) DPIDBs allocated to LGCs 0 to 7 (ISLU half 1)
Even numbered (10, 08,06, ..) DPIDBs allocated to LGCs 8 to 15 (ISLU half 0)
B-PHs for 64kbps B-channels with X.25 or D-channels with >8 dpps
D-PHs for D-channels with <8 dpps or D-channels with Q.931 signalling
Max of 128 logical channels per B-channel
Max of 16 logical channels per D-channel
Max of 128 logical channels per PH
Max of 50 pps for 5E5 PH2
Max of 160 pps for 5E6 PH3
Max of 4 time slots from each LGC to each DPIDB
LGC to DPIDB time slot allocation according to formula below
Max of 5 DPIDBs with access to an LGC
Max of 20 time slots per LGC to the PSU (from 4ts/DPIDB x 5DPIDBs/LGC)
1/4 time slot used by packet D-channel (assumption for 5E4; <8 dpps)
Max of 32 B-channels per PH allowed by original ILAS assignment system
37 dpps equated to a time slot weight of 1 in first uses
Each forth PH member assigned by some systems (0, 4, 8 ... 32 max channels)

Goals for line assignments:

Get SM, ISLU, LGC, Card; example=004-0-00-01=00400001
Get Shelf, PH, PH Member ; example 0-02-000=002000
Get DPIDB, TS, Qtr TS; example 11-1-0

5ESS Recent Change Screen 23.2 selected lines:

```
line 10          =  SM [1-192], ISLU [0-7], LGC [00-15], Card [00-31]
line 13   100047 =  PSU Shelf [0-4], CHL Group [00-15], PH Channel [000-127]
line 23 DPIDB 11 =  [1-11]
line 24 TS     0 =  [0-31]
line 25 QTR TS 0 =  [0-3]
```

Formula for DPIDB TS to LGC assignments:

```
 lgc=( ((mod(DPIDB+1,2))*8)+mod(TS+7+8-(11-DPIDB-mod(DPIDB+1,2))/2,8))

 where DPIDB in range [11-2]
 where TS in range [0-31]
 mod(n,b) = n modulo b
```

Appendix V — Getting Started

Basic Instructions on Using the Albuquerque Version of Phonedb:

1. Connect to the VMS VAX SAV80 by:

a. Use the IBX by
1. Turning on a terminal hooked to an ADI data set.
2. Type - = until the prompt ENTER AUTHORIZATION CODE appears.
3. Enter your authorization code <return>.
4. At the ENTER DESTINATION NUMBER prompt enter SAV80 <return>.
5. At DATA CALL INITIATED press <return>.

OR

b. Connect through the Macintosh network

2. Log into SAV80:
a. At Username: type your username <return>, for example SNELSON.
b. At Password: type your password <return>.
c. Be sure that your Ingress terminal type is correct. There are many choices, but for most uses from VT100 type terminals the command

```
SAV80> define term_ingres vt100f <return>
```

executed from the prompt as shown or in your login command file, will set things up ok. If you are in doubt, type the above command.

(For a vt200 or vt300 terminal running in vt220 mode, use the command: define term_ingres vt220)

3. At the SAV80> prompt type PHONEDB <return>.
That starts up the database application program and may take 12 seconds.

4. Go to the circuit record screen:
a. Type R <return> to bring up the Records menu.
b. Type L <return> to bring up the Lines or circuit record screen. Bringing-up the Lines screen the first time may take 4 seconds; subsequent moves to the Lines screen will be much faster.

5. To find the record for a 5ESS circuit on LEN 00425120:
a. Note that for 5ESS circuits the circuit id is the LEN (Line Equipment Number) which is the same as LCEN and OE for the database. The leading zeroes must be typed.
b. Type 00425120 in the Circuit ID field (no need to press <return>).
c. Press the PF1 key to move the cursor to the menu line. On a VT100 terminal this key is labeled PF1.
d. Type F <return> to activate the menu Find function.

6. To clear the screen fields for new entry:
a. Press the terminal PF1 key.
b. Type CL <return> to activate the screen Clear function.

7. To find the record for a 5ESS circuit with telephone number 8457271:
a Clear the screen fields according to step 6 above.
b. Press the TAB key to move the cursor to the telephone number field.
c. Type 8457271 (no need to press <return>).
d. Press the PF1 key to move the cursor to the menu line.
e. Type F <return> to activate the menu Find function.

8. To find the record for a circuit with id C123:

a. Clear the screen fields according to step 6 above.
b. Type C123 in the Circuit ID field (no need to press <return>).

c. Press the terminal PF1 key to move the cursor to the menu line.
d. Type F <return> to activate the menu Find function.
e. The circuit should be displayed.

9. To Quit and logoff:
a. Press the terminal PF1 key.
b. Type Q <return> to activate the screen Quit function.
c. Repeat step b until the SAV80> prompt appears.
d. Type LO <return> to log off.

10. Some SHORTCUTS:
a. On a VT100 terminal (or emulated VT100 terminal), instead of the sequence of PF1 F <return> you can just hit the "enter" key with the cursor anywhere on the screen. This will activate the menu Find function. The "enter" key is the key on the lower right on the keypad.
b. Instead of the sequence PF1 CL <return> you can just hit the keypad "0" key. This will activate the menu Clear function.
c. Instead of the sequence PF1 Q <return> you can just hit the keypad PF3 key.
d. Other speed function keys are indicated in parenthesis after the function on the menu line. You can scroll the menu line horizontally by repeatedly pressing the PF1 key.

11. Correcting the information typed in fields and moving between fields:
a. The VT100 "del" key deletes the character preceding the cursor.
b. All the standard Ingres forms movement functions are used. The arrow keys move the cursor within a field. TAB moves to the next field. On a VT100, CTRL-P (control P), moves to the previous field.
c. To scroll tablefields (like where cable circuit entries are done and in big help files):
i. Press the keypad "-" key to scroll down.
ii. Press the keypad "," key to scroll up.
d. Additional functions can be checked by activating the HELP menu item on most screens.
i. Press PF2.
ii. Type K <return> to look at the key definitions.
iii. PF1 E <return> to escape HELP.

12. Customer number and customer name:

a. A customer number or the word SPARE is required for every circuit. A unique number is used to identify each customer to get around the problems of people having the same name, the same name being entered in various ways (Mike, Michael, M.), and to allow bulk corrections of organization and contact phone number from the corporate database.

b. A customer number is required for each circuit so we can identify the resources controlled by each user or organization and so that we can get in touch with the "owner" of a circuit in case a circuit change needs to be made.

c. For special use, group identifications, such as the word SPARE, can be entered into the system as a customer and used in the circuit records.

d. Customer number is initially the E number for Sandia employees or an N number for those contract people who have an N number. Social security number can be used also (Sandia is planning to switch to this in 1992), or the database system can assign a unique number from within the Customers screen with the menu item Get_No.

13. Finding a customer number when you only know the name:

a. Get to the Customer screen. From the Circuit screen, press PF1, then type CU <return>.

b. Move the cursor to the name field (if it is not already there) by pressing TAB once.

c. Type in enough of the name as you think will make it unique and unambiguous followed by the wildcard character *. See the sample in the figure below where "DOE, J*" was typed to find the name JOHN DOE.

d. Activate the menu item Find by pressing PF1 followed by F <return>.

e. If more than one name matches the search, the menu items NEXT and END appear. Activate NEXT until the proper name appears, then activate END.

f. Choosing Select from the menu will move the customer number and name to the Circuit screen. Choosing Quit will return you to the Circuit screen without transferring the customer number.

```
                    +---------------------------------------+
                    |Circuit Customer, Location, & Equipment|
                    +---------------------------------------+

Circuit Id:                  Telephone No:           Location:
Customer No:+-----------------------------------------------------------+
Name:       |                                                           |
            |          +----------------+                               |
            |          |System Customers|                               |
            |          +----------------+                               |
+-----------|Customer No:                             Organization:     |+
|Keyword/Cab|      Name: DOE, J*                         Telephone:     ||
|-----------|                                                           ||
|           |Category:         Classification:                          ||
|           +-----------------------------------------------------------+|
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
|                 |               |               |            |         |
+-----------------+---------------+---------------+------------+---------+
```

14. Class of Service, Line Class of Service, and COS:

Class of Service is the designator that allows identification of the detailed purpose of a circuit. Some examples are FIRE for fire alarms circuits, CS1 for analog telephone circuits, PORT for IBX asynchronous data ports. Having a detailed class of service allows us to take special action in the database software. We would only assign ISDN switch resources to circuits with ISDN classes of service, and so on.

15. Circuit ID, CLEN, LEN, and Port:

These synonymous terms become the basic identifier for a circuit in the database. The conductors for each cable are internally marked with the Circuit ID of the circuit using each.For circuits where there is no "line equipment" number (LEN), or port number, any unique circuit identifier can be entered or the database system can assign a number with the circuit screen menu item Get_No.

16. Searching for a circuit when you only know the customer:

a. Clear the Circuit screen field.

b. Find and fill-in the customer number.

c. Activate the menu item Find.

d. If the customer has only one circuit, that one will appear. But if the customer has multiple circuits, the Search screen will be presented as shown in the example below. Select the circuit you wish to view in detail like this:
   i. Move the cursor to the table by pressing CTRL-P (control P) or pressing TAB 11 times (easier to push CTRL-P).
   ii. Move the cursor to the line of interest using the up and down arrow keys.
   iii. Activate the Select menu item with PF1 S <return>.

e. The Search screen can be cleared and then any combination of information (including the wildcard *) can be filled-in the top portion to formulate criteria for a Find.

```
                             +---------------------+
                             |Circuit Search Screen|
                             +---------------------+

Circuit Id:                   Telephone No:           Location:
Customer No:                 Station Equip:           Building:
Name:                             Line Cos:               Room:
Org:                                                      Jack:

+---------------+----------+-----------------------+-----------+------------+
|Circuit ID     |Phone No  |Name                   |Customer No|Organization|
|---------------+----------+-----------------------+-----------+------------|
|00400119       |8450006   |DOE, JOHN B.           |E00012345  |04587       |
|C28            |          |DOE, JOHN B.           |E00012345  |04587       |
|765B           |          |DOE, JOHN B.           |E00012345  |04587       |
|               |          |                       |           |            |
|               |          |                       |           |            |
|               |          |                       |           |            |
|               |          |                       |           |            |
|               |          |                       |           |            |
|               |          |                       |           |            |
|               |          |                       |           |            |
+---------------+----------+-----------------------+-----------+------------+
```

17. Cable and conductor information in the Circuit screen:

The table in the lower part of the Circuit screen shows the wiring for the circuit. The elements that make up a circuit are listed in the order in which they are connected. See the sample circuit below. It shows that cable 910ESS05 pair 19 is jumpered to cable 910ESS0101 pair 804. Likewise 910ESS05 pair 20 is jumpered to cable 910ESS0101 pair 805. Cable 910ESS0501 pair 806 also goes to the destination jack. (The keyword "BDF POWER" or a similar term can be put in a line between the two cables, but some sites may decide to skip that addition.)

To add wiring information in the circuit screen:

a. Find the proper circuit using the Find menu item.

b. Move the cursor to the tablefield using the TAB key or CTRL-P, whichever is quicker.

c. Type the cable name in the first column, then press <return> to move to the second column.

d. Type-in first cable pair number. Press <return> to move to subsequent columns if more pairs of the same cable are used.

e. If more than one cable is involved in the circuit, move to the next line by pressing <return> until the cursor moves to the first column of the next line. Repeat steps c and d until the entire circuit is shown.

f. Activate the Save menu item with PF1 SA <return> or by pressing the designated single stroke function key.

g. Note that as you are typing in the tablefield, if the cable name you type is not already in the system cable records, an error message will appear. You can only put in cable names or keywords that the system already knows about. Keywords can be viewed by activating the menu item Keywords. Cable names must be looked-up from the cable manipulation screens.

```
                +--------------------------------------+
                |Circuit Customer, Location, & Equipment|
                +--------------------------------------+

Circuit Id: 00220006        Telephone No: 2949381      Location: LIVERMORE
Customer No: E00054777     Station Equip: 7506         Building: 910
Name: SMITH, MICHAEL T.         Line Cos: I1               Room: 107
                                                           Jack:

+--------------------+--------------+--------------+------------+------------+
|Keyword/Cable Name  |Col 1 / Pair 1 |Col 2 / Pair 2 |Col 3 / Pr 3|Col 4 / Pr 4|
|--------------------+--------------+--------------+------------+------------|
|910ESS05            |19            |20            |            |            |
|910ESS0501          |804           |805           |806         |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
|                    |              |              |            |            |
+--------------------+--------------+--------------+------------+------------+
```

18. Common error messages and how to deal with them:

a. The most common error messages are issued when a mistake is made activating a menu item. The first line of the error message is displayed. If you know what has happened (like you pressed X instead of C), just press <return> to clear the message and try again. If you want to read the details of the message, press the "h" key, read the message, then press <return>.

A common problem is not entering enough of menu item's name to allow the system to determine which item to activate. For example, some menus have the items "customers" and "clear". Entering PF1 C <return> is ambiguous. You must enter PF1 CL <return> or PF1 CU <return> to differentiate the two. Alternately, use the predefined one-stroke function keys to reduce keystrokes.

b. The system's internal verification of the values in fields occasionally results in a message. These messages will occur when an out-of-range value is entered in some screen field. The messages are cleared by pressing <return>. If more details of the message are available, the prompt will have an option to press h for "more" as in the menu error messages.

DISTRIBUTION:

3141 S. A. Landenberger (5)
3145 Document Processing
for DOE/OSTI (8)
3151 G. C. Claycomb (3)
8523-2 Central Technical Files

DOE/AL/IMD W. L. Heatherly
2910 D. L. Crawford
2930 D. C. Jones
2934 S. D. Nelson (25)
2934 L. F. Tolendino (25)
2934 M. O. Vahle
2935 D. M. Darsey
2935 F. H. Jones
2935 J. P. Sena (5)
2935 D. W. Tapscott
8273 G. Gutierrez (5)

Mr. Oscar Suarez (5)
AT&T Federal Systems
1001 Menaul Blvd. NE
Albuquerque, NM 87107

END

DATE FILMED

11 / 05 / 91