A COMPUTERIZED AUDIT PACKAGE
FOR CENTRAL POWER AND LIGHT COMPANY

Graduate Project
Presented in
Partial Fulfillment
of the Requirements
for the
Master of Science Degree

Computer Science Department
Corpus Christi State University

by
Mary E. Sullivan

July 23, 1982

Project Committee
David Perkins, Chairman
Dr. Roy Ellzey
Paul Cameron
A COMPUTERIZED AUDIT PACKAGE FOR
CENTRAL POWER AND LIGHT COMPANY

This project was completed in accordance with the specifications set out in my graduate proposal. Section A of this report will describe each of the programs written. A few minor deviations from the proposal were made, and these will also be explained. Section B contains a copy of the user's manual given to the Internal Audit Department. The manual contains some general instructions for using the computer terminal and ROSCOE, along with detailed instructions for running each program, a copy of the program as it is stored in the ROSCOE library and a sample program with output. Section C contains the project proposal.

As was proposed, most of the programs were written in EASYTRIEVE, an information retrieval language developed by Pansophic. EASYTRIEVE was designed to be a high-level language that required a minimum amount of programming on the part of the user. The logic behind EASYTRIEVE is slightly different from most programming languages. Records are "selected" for output only if they satisfy a conditional statement in the program. Calculations may be performed on the data, but again, a condition must be satisfied before the logic immediately following the condition is performed. At times it is necessary to "FLUNK" a record, which prevents it from being output. Read statements are not required, as a "read loop" is performed automatically.

The programs were written to be as flexible as possible for audit use, but yet simple enough for non-programmers to use. The detailed instructions, together with the sample program, show the auditor exactly how to customize these programs for each audit.
SECTION A
DESCRIPTION OF PROGRAMS

Each program will be described in the order presented in the proposal.

1. Voucher Sequence Check

A voucher file is created each month by keying in batches of paid vouchers. Distribution reports are run at the end of the month and then the same file is used for the next month's vouchers. This voucher file is used in this program and in others in this project. Before the sequence check is done, the file is sorted by voucher number and saved in a temporary file. Program SRTVOUCH is the JCL required for SYNSORT, the sort used here.

After the sorted file is created, program MSGNUM can be executed. The only input required by the user is the current date, which is only for documentation purposes. The program works by comparing each voucher number to the next one in the file. If the difference is more than one, both numbers are printed. Duplicate voucher numbers are allowed in this file, which had to be accounted for. The auditors also asked that the last number in the file be printed.

2. List Vouchers Exceeding $100,000

This program was written to use the sorted voucher file described previously. These vouchers will normally be pulled from a manual file for further examination, so sorting on voucher number is desirable. Program VOUCHERS was designed to list vouchers exceeding a user-specified amount, instead of coding the $100,000 threshold stated in the proposal. The general ledger account number was also added to the output.

3. Random Number Generator

This FORTRAN program, RANDNUM, uses a random-number generator algorithm to produce the number of random numbers needed. The user inputs the upper and
TABLE OF CONTENTS

Section A - Description of Programs

Section B - User's Manual

<table>
<thead>
<tr>
<th>Program</th>
<th>Tab No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRTVOUCH</td>
<td>1</td>
</tr>
<tr>
<td>MSGNUM</td>
<td>2</td>
</tr>
<tr>
<td>VOUCHERS</td>
<td>3</td>
</tr>
<tr>
<td>RANDNUM</td>
<td>4</td>
</tr>
<tr>
<td>PAYSAM1</td>
<td>5</td>
</tr>
<tr>
<td>CUSTSAM1</td>
<td>6</td>
</tr>
<tr>
<td>VOUSAM1</td>
<td>7</td>
</tr>
<tr>
<td>PAYSAM2</td>
<td>8</td>
</tr>
<tr>
<td>CUSTSAM2</td>
<td>9</td>
</tr>
<tr>
<td>VOUSAM2</td>
<td>10</td>
</tr>
<tr>
<td>SAMSIZE</td>
<td>11</td>
</tr>
<tr>
<td>SPEC</td>
<td>12</td>
</tr>
<tr>
<td>ADJ</td>
<td>13</td>
</tr>
<tr>
<td>INACT</td>
<td>14</td>
</tr>
<tr>
<td>CONTRACT</td>
<td>15</td>
</tr>
<tr>
<td>FLAT</td>
<td>16</td>
</tr>
<tr>
<td>NOREAD</td>
<td>17</td>
</tr>
<tr>
<td>LSTPGM</td>
<td>18</td>
</tr>
<tr>
<td>PRINTOFF</td>
<td>19</td>
</tr>
<tr>
<td>JCL-Customer Files</td>
<td>20</td>
</tr>
</tbody>
</table>

Section C - Graduate Project Proposal
lower limits of the required range and how many numbers are needed. The user
is asked to input the current date, which is used as the starting point in
the algorithm. The user also has the option of sorting the random numbers.

4. **File Sampling (1)**

Three programs were written for interval sampling. They are PAYSAM1,
CUSTSAM1 and VOUSAM1, which sample the payroll, customer and voucher files,
respectively. The user inputs the sample size and interval in the LIST statement.
This statement also specified what fields are to be printed, which the user
selects from the field names given in the program instructions.

5. **File Sampling (2)**

Three programs were also written for this type of sampling. They are
PAYSAM2, CUSTSAM2 and VOUSAM2. Records are selected for output if they satisfy
the condition specified by the user. This condition is an "IF" statement
which can test relations on any of the fields specified in the program instructions.
The fields to be printed are also user specified, and selected from the same
list.

6. **Determine Sample Size**

This FORTRAN program, SAMSIZE, determines a statistically valid sample
size for discovery sampling, which is frequently used in internal auditing.
The user determines the confidence level required, which the program translates
to a reliability factor. The user also inputs the highest exception occurrence
rate he will allow. The sample size is obtained by dividing the reliability
factor by the occurrence rate.

7. **Special Handle Accounts**

The customer record contains a field for "special handle" code. If this
code is "S" the customer's bill is sent to the local office for handling
instead of being mailed directly. This capability can be abused by local
office personnel, and a high special handle rate could signal a problem in
a town. The user inputs the town number to be examined, and records for this
town with a code of "S" are output by program SPEC. To print out only the
summary total, the user can delete the word "DETAIL" from the LIST statement.

8. Adjusted Bills

The customer record has a field for the previous twelve months' consumption
code. A code of "9" means an adjustment was made to a customer's account.
Repeated adjustments to an account may need to be investigated. Program
ADJ counts the number of adjustments to a record, compares this to the user-
input number of adjustments allowed, and prints the record if the allowed
number of adjustments is exceeded.

9. Inactive Accounts

The customer record has a status for active or inactive accounts.
Program INACT examines inactive accounts for the town specified. Accounts
are printed that still have an unpaid amount due or a deposit not refunded.
This information is extracted from the customer master file, rather than a
separate inactive customer file as stated in the proposal.

10. List Contracts

Program CONTRACT prints out information for accounts required to have
contracts on file for the town being audited. The rate codes requiring
contracts are 28, 29, 39, and 63. This information is extracted from the
customer master file rather than a separate contract file as stated in the
proposal.
11. **Check Flat Rates**

Program FLAT is used to verify flat rate billings. Certain rate codes have a fixed billing each month that is not dependent on kilowatt-hours used. The user inputs the rate he wants to examine, and all or a sample of the accounts on this rate with a flat rate amount will be printed.

12. **Unread Meters**

When a meter is not read during the month, the account is given a consumption code of "4". The bill must then be estimated, which changes this code to a "7". Program NOREAD looks for accounts with these codes for both of the two preceding months. The user inputs the town number desired.

13. **List Production Program**

Program LISTPCM is the JCL needed to obtain a copy of a specified production program source listing from LIBRARIAN.

14. **List ROSCOE Member**

PRINTOFF and JCLEND are the JCL used to obtain hard copy of a program stored in the user's ROSCOE library.
General Instructions For Executing A Program Under ROSCOE

Programs in this package will be edited and submitted at the terminal with the aid of ROSCOE, a software system for on-line program development. The programs are stored in the ROSCOE library under number CTS0071. Each ID-number has its own library. The "LIB" command will show you a list of programs in that library.

In order to sign on to the ROSCOE system, perform the following steps:

1. Type in "ROSS1".

2. The system will return a prompt for your key and will automatically position the cursor. Enter the user-ID number this package is stored under (i.e. CTS0071).

3. The system will return a prompt for your password. Enter the 4-character password associated with this number.

According to company security rules, all passwords must be changed every ten days. The system will tell you if you need to change your password. This is done by typing your old password, a slash (/), and your new password. You may also change your password anytime you wish by following this same procedure in place of step 3.

After the sign on process is complete, you are ready to modify and/or submit a job for execution. If you are submitting a program that requires no modifications, you may submit it directly from the ROSCOE library by entering "SUB program-name".

A command is entered by typing it at the top of the screen and hitting the "ENTER" button. Most of the time you will need to modify the program to fit a particular audit. To do this you must first bring a copy of the program into the Active Work Space (AWS) with the command "FETCH program-name". FETCH may be abbreviated by "F". The system will return a blank screen. To display the AWS on the screen, enter "ATTACH" or "A". To page through
the program, refer to the description of PF keys at the end of this section.

Refer to specific program instructions to determine if modifications need to be made to a program before submitting it. If so, when the correct page is on the screen, position the cursor and make a change by typing in new characters over old characters. If it is necessary to insert additional lines in the program, position the cursor at the line you wish to insert after. Hit the PF12 key to get a screen showing the line the cursor was on, followed by blank lines. Type your additional lines, then hit ENTER to add this to the program.

After all modifications have been made, the program can be submitted with the command "SUB program-name". When this command is used without specifying a program name, the contents of the AWS is submitted by default. In order to make the modification permanent, type "U program-name" to update the existing program in the ROSCOE library. This can be done before or after submitting a program for execution.

The status of a submitted job can be determined by depressing the PF4 key. Each job submitted will be identified on the terminal with a unique job number next to its status. When the status of a job is "on output queue", you can view the output at the terminal by depressing the PF5 key.

The output for all EASYTRIEVE programs is divided into four files:
1. Job statistics
2. JCL
3. Diagnostics
4. Program listing and output

In the two FORTRAN programs, RANDNUM and SAMSIZE, output will be in File 6. Under the output mode, the PFn key will give you page 1 of file N. The return key will also allow you to page through all four files. Page 1
of file 1 will initially be shown on the terminal. On this page is a return code, identified as RC, with a number under it. A return code of 0 generally means the program worked, while a return code of 16 signals a problem with the EASYTRIEVE coding. Should you get a return code of 16, go to file 4 and look for an error flagged in the program. Once you find the error, end the output and re-edit and submit the job.

You can "end" the output without having anything printed by entering "END" at the top of the screen. To print the output, enter "PRINT ALL CLASS=A".
DESCRIPTION OF PF KEYS

Command Mode

PF1 - Splits screen for viewing two different modules.
PF2 - Swaps cursor between split screens.
PF3 - Gets out of split screen.
PF4 - Shows status of jobs submitted.
PF5 - Displays program output (puts you in output mode).
PF6 - Repeats last command executed.
PF7 - Scroll backward one page.
PF8 - Scroll forward one page.
PF9 - Continues execution of a paused RPF program.
PF10 - Gets you to top of module.
PF11 - Gets you to bottom of module.
PF12 - Allows you to insert additional lines of code.

Output Mode

In this mode, the PFn key takes you to the top of output file N. Paging is accomplished by hitting the ENTER key.
<table>
<thead>
<tr>
<th>Program Name</th>
<th>Section</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>13</td>
</tr>
<tr>
<td>CONTRACT</td>
<td>15</td>
</tr>
<tr>
<td>CUSTSAM1</td>
<td>6</td>
</tr>
<tr>
<td>CUSTSAM2</td>
<td>9</td>
</tr>
<tr>
<td>FLAT</td>
<td>16</td>
</tr>
<tr>
<td>INACT</td>
<td>14</td>
</tr>
<tr>
<td>JCLCORP</td>
<td>20</td>
</tr>
<tr>
<td>JCLFIELD</td>
<td>20</td>
</tr>
<tr>
<td>JCLTAPE</td>
<td>20</td>
</tr>
<tr>
<td>LISTPGM</td>
<td>18</td>
</tr>
<tr>
<td>MSGNUM</td>
<td>2</td>
</tr>
<tr>
<td>NOREAD</td>
<td>17</td>
</tr>
<tr>
<td>PAYSAM1</td>
<td>5</td>
</tr>
<tr>
<td>PAYSAM2</td>
<td>8</td>
</tr>
<tr>
<td>PRINTOFF</td>
<td>19</td>
</tr>
<tr>
<td>RANDNUM</td>
<td>4</td>
</tr>
<tr>
<td>SAMSIZE</td>
<td>11</td>
</tr>
<tr>
<td>SPEC</td>
<td>12</td>
</tr>
<tr>
<td>SRTVOUCH</td>
<td>1</td>
</tr>
<tr>
<td>VOUCHERS</td>
<td>3</td>
</tr>
<tr>
<td>VOUSAM1</td>
<td>7</td>
</tr>
<tr>
<td>VOUSAM2</td>
<td>10</td>
</tr>
</tbody>
</table>