

PROC
MGT

FILE COPY

SANDIA REPORT

SAND83-8243 • Unlimited Release • UC-13

Printed October 1983

Purchase Order Graphical Output (POGO)

OCT 12 1983

1-2	FILE	
	HARODDY	Har
	SCHUKNECHT	"
	HANSER	"
	WRIGHT	"
	BARHAM	"
	CENTRAL FILE	
	DESTROY	

C. T. Schafer, P. W. Zimmerman

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 the 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 or subcontractors.

Printed in the United States of America
Available from
National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161

NTIS price codes
Printed copy: A05
Microfiche copy: A01

PURCHASE ORDER GRAPHICAL OUTPUT (POGO)

C.T. Schafer 8443
P.W. Zimmerman 8453
Sandia National Laboratories, Livermore

ABSTRACT

The Purchase Order Graphical Output (POGO) system was created to provide up-to-date financial reporting and accounting of Direct Charge (D.C.) purchases, including emergency orders and excluding shop and plant expenditures, and to provide improved summaries in the form of bar charts, monthly plots, and tabular listings.

The budget and committed amounts can be kept up to date by entering purchase requisitions and budget changes on a daily basis. The money actually costed will be out of date by the amount of time required by SNLA financial accounting to make payouts, enter them into the computer, then transfer the data to SNLL.

The graphics are done through the 'TELL-A-GRAF' facility. Thus, the user has the flexibility to alter the format or text of a plot to suit individual needs. Furthermore, report quality reproductions are easily obtained.

TABLE OF CONTENTS

SYSTEM DESCRIPTION	
I. Programs and Files	11
II. File Maintenance	11
III. Login and Logout Error Messages	16
IV. Other Applications	17/18
 PROGRAM INPUT	
I. Input Procedure	19
II. Reclassification of a Purchase Order	19
III. Split Accounts	20
IV. Error Checks	20
V. Trouble Shooting	21
VI. Input Files	22
VII. Output Files	22
 PROGRAM STRIP	
I. Procedure	23/24
II. Input Files	23/24
III. Output Files	23/24
 PROGRAM COSTING	
I. Costing Procedure	25
II. Reclassifications	25
III. New Fiscal Year	26
IV. Input Files	26
V. Output Files	26
 PROGRAM REPORT	
I. Plots	27
II. Tabular Output	27
III. Report Specifications	27
IV. Prompts	28
V. Exit	28
VI. Input Files	28
VII. Output Files	29
 PROGRAM ACCOUNT	
	33/34
 OTHER PROGRAMS	
I. Programs CHECK and INCHECK	35/36
II. Programs POGO and MESSage	35/36
III. Program ERASER	35/36

USER'S GUIDE

INTERPRETATION OF PURCHASE.LIS	39
INSTRUCTIONS TO LOG PURCHASE ORDERS ONTO THE VAX	40
MONTHLY COSTING INSTRUCTIONS	42
PLOT INSTRUCTIONS	44
LIST INSTRUCTIONS	46
TAPE BACKUP PROCEDURE	47
SET-UP INSTRUCTIONS	49

APPENDIX A

LIST OF FILES	55
I. INFORMATION FILES	55
II. BACKUP DATA FILES	55
III. PROGRAM FILES	56
IV. WORKING DATA FILES	56
V. PERMANENT DATA FILES	57
VI. COMMAND FILES	57
FILE NAME ASSIGNMENTS	58

APPENDIX B

FILE FORMATS FOR THE POGO SYSTEM	61
----------------------------------	----

APPENDIX C

I. Command File For Monthly Costing Procedure	69
II. Command File to Obtain Plots by Division	70

APPENDIX D

CASE AND SUBCASE MANAGING	73/74
---------------------------	-------

ILLUSTRATIONS

Figure 1.	Flow Chart of POGO	12
Figure 2.	Tabular output from program 'REPORT' showing integrated contractor orders over \$100,000 for each division.	13
Figure 3.	Bar plot showing spent, committed, and costed information for each division.	14
Figure 4.	Monthly plot of DC purchases for the entire department.	15
Figure 5.	Level of commitment and spending by consultant	30
Figure 6.	A portion of the data used to produce the bar graph of figure 5.	31
Figure 7.	Summarized procedure to obtain figures 5 and 6.	32

SYSTEM DESCRIPTION

I. Programs and Files

The POGO system, illustrated in Figure 1, is initialized by creating support files ROSTER.LIS and ACCOUNT.LIS. Program 'INPUT' reads these files then opens a PURCHASE.LIS file into which the user enters purchases. An ACCOUNT.ERR file is written if there are accounts (case and division) in PURCHASE.LIS which do not appear in ACCOUNT.LIS. PURCHASE.TMP is a user requested backup file.

Program 'STRIP' reads an SNLA purchase file and accepts only those purchases whose charging organization matches one of the charging organizations in the account file, then writes the purchases sorted by purchase order number into file COSTING.DAT. (The MONTH.COM file written by 'STRIP' is a command file that can be used for the subsequent month's costing.)

Program 'COSTING' matches the purchases in COSTING.DAT to those in the PURCHASE.LIS file, enters the monthly costing data into the main data file and writes a new version of PURCHASE.LIS. The X.INP file contains purchases from the SNLA data that were not in the SNLL data and X.COS contains SNLL purchases that were not in the SNLA data.

Program 'REPORT' prepares the data summaries per user specifications and writes data output files for bar graphs, monthly graphs and tabular output. 'REPORT' summarizes committed and costed monies from the PURCHASE.LIS file and the budget monies from the ACCOUNT.LIS file. The output plot files are tailored for 'TELL-A-GRAF'. Figures 2 through 4 illustrate the results.

II. File Maintenance

The POGO programs and files of Figure 1 (they are also listed in detail in Appendix A) are protected against user and computer errors. The main data base, PURCHASE.LIS, is checked for errors by every program that uses it and at login and logout. Programs, executable files and important data files are kept in subdirectories to prevent a user from inadvertently deleting them or altering them with the editor.

A problem encountered on the VAX system that may cause trouble is overflow of allocated file space. If a file being written overflows the space, it may be written in a short form. The best prevention is to continuously remove obsolete files from the directory. The POGO system keeps the directories cleared by purging all but the last two versions of all files at logout. Files created by program 'REPORT' are automatically deleted after one month. Journal files are automatically deleted after two weeks. Object files are automatically deleted at logout.

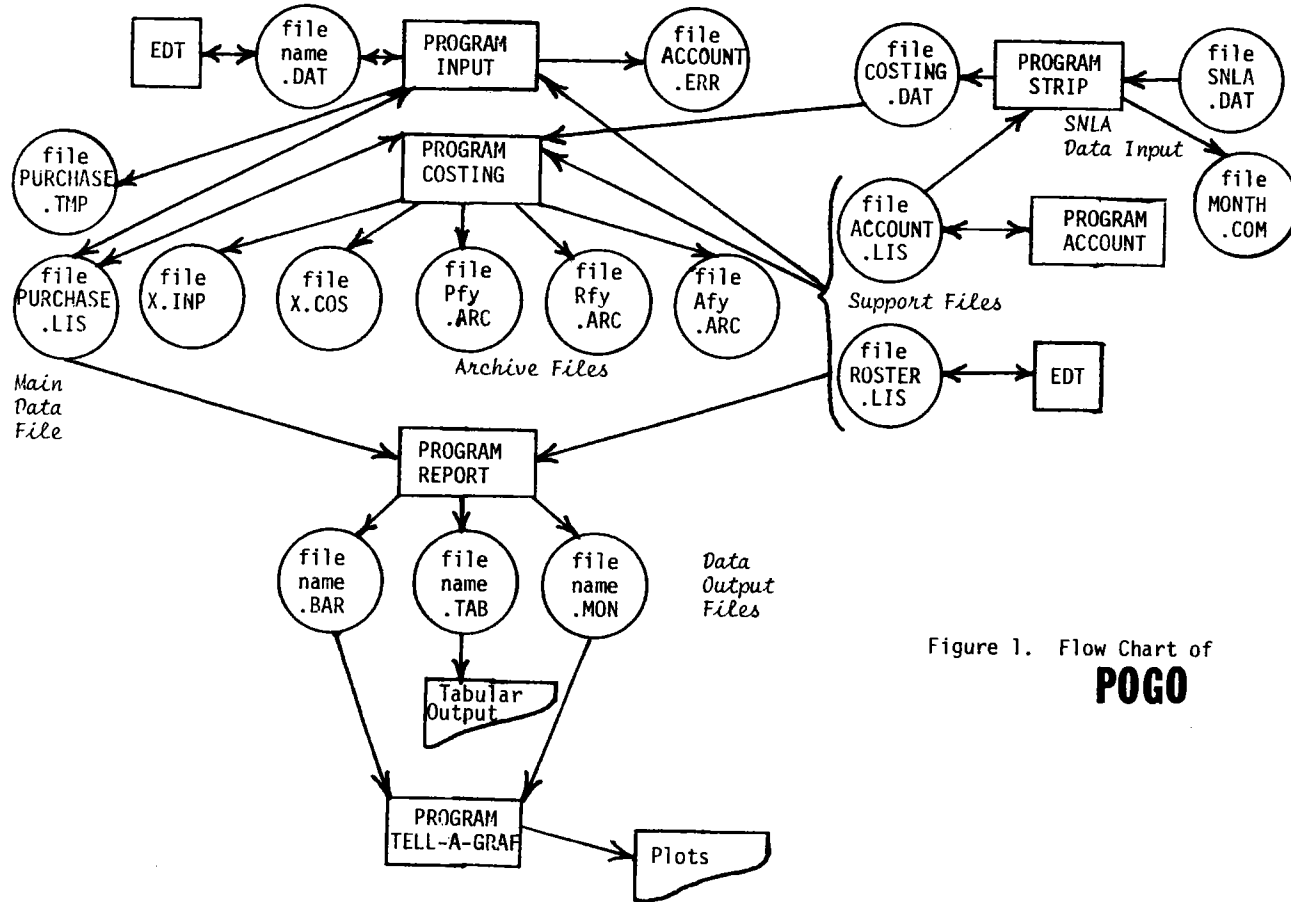


Figure 1. Flow Chart of **POGO**

**** DATE = 13-JUL-83 ****
 *** REPORT SPECIFICATIONS ***
 * PURCHASE ORDER 92 *
 * AMOUNTS OVER \$ 100.K *

P.O. NO.	DATE	CASE NO.	NAME	V.	ITEM	AMOUNT	BAL.	THIS FY	SUBS	FY
----- GROUP = 8441 -----										
92-9430	04	02/12/82	1292.200	RNG	TH NOT VERIFIE	277.3	44.0	44.0		0.0
			[BUDGET 1501.0]TOTALS.....		277.3	44.0	44.0		0.0
----- GROUP = 8442 -----										
92-8851	00	10/01/80	0720.270	CMHINC	RO UNKNOWN	300.5	-11.1	0.0		0.0
92-9298	03	09/25/81	0720.270	CMHINC	RO UNKNOWN	341.3	-17.1	0.0		0.0
92-9407	03	01/12/83	0560.100	JTCARR	RO MONITORING	254.6	9.5	9.5		0.0
92-9407	03	01/12/83	1293.200	JTCARR	RO MONITORING	102.8	8.7	8.7		0.0
92-9506	01	04/09/82	0720.270	JTCARR	RO R & D	140.0	10.4	10.4		0.0
92-9544	03	05/13/82	1292.200	EAENGL	RO NOT VERIFIE	197.9	168.7	168.7		0.0
			[BUDGET 2220.0]TOTALS.....		1337.1	169.1	169.1		0.0
----- GROUP = 8445 -----										
92-8578	07	10/20/82	0501.200	HHANSE	RO NOT VERIFIE	105.9	0.0	0.0		0.0
92-8716	05	06/06/80	0501.200	DRADOL	EI INCREASE FR	137.3	0.0	0.0		0.0
92-8725	08	11/03/82	0852.200	LABROW	RO TOTAL	157.6	1.6	1.6		0.0
92-8852	02	10/01/80	0440.200	DWPUTZ	RO CHANGE 2 TO	230.0	12.0	12.0		0.0
92-8930	11	01/03/83	0853.200	LWDERI	RO NOT VERIFIE	114.5	16.3	16.3		0.0
92-8948	01	12/22/80	0440.200	JLBOLE	RO NOT VERIFIE	294.5	-1.1	0.0		0.0
92-8949	01	12/23/80	0440.200	JLBOLE	RO NOT VERIFIE	118.7	-3.3	0.0		0.0
92-8951	01	12/23/80	0440.200	DWPUTZ	RO NOT VERIFIE	114.6	0.0	0.0		0.0
92-9247	00	08/19/81	0501.200	DRADOL	EI NOT VERIFIE	117.0	0.0	0.0		0.0
92-9380	04	12/02/81	0853.200	LWDERI	RO CO #4- INCR	175.0	-7.4	0.0		0.0
92-9409	02	01/25/82	0853.200	LWDERI	RO NOT VERIFIE	162.0	28.7	28.7		0.0
92-9443	04	05/14/82	1292.200	DRADOL	RO NOT VERIFIE	212.5	96.3	14.3		82.0
			[BUDGET 2068.0]TOTALS.....		1939.6	143.1	61.1		82.0

Figure 2. Tabular output from program 'REPORT' showing integrated contractor orders over \$100,000 for each division. The 'name.TAB' file was printed on a Diablo printer. 'V.' is the vendor column and the two 'FY' columns are forecasts. Name, Vendor and Item are shortened in the listing to fit all fields on one line.

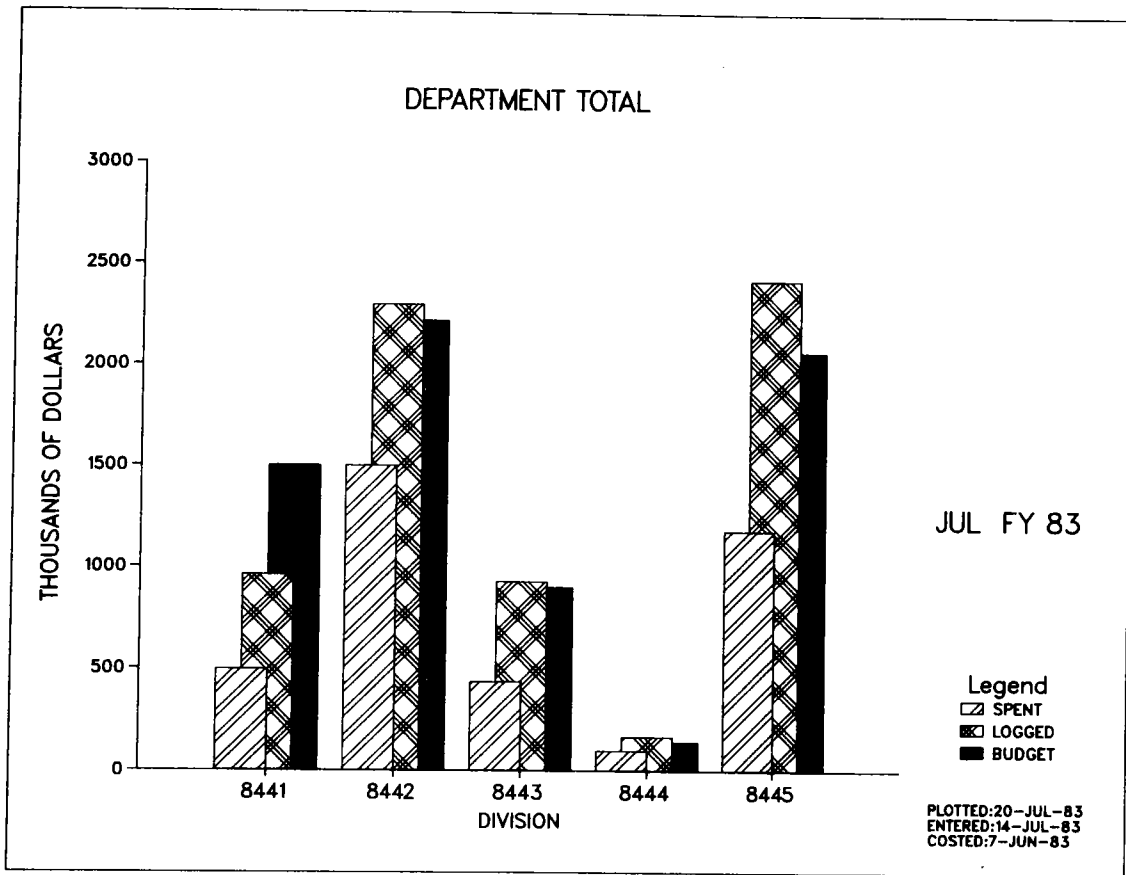


Figure 3. Bar plot showing spent, committed, and costed information for each division. (Copy obtained from a film plot)

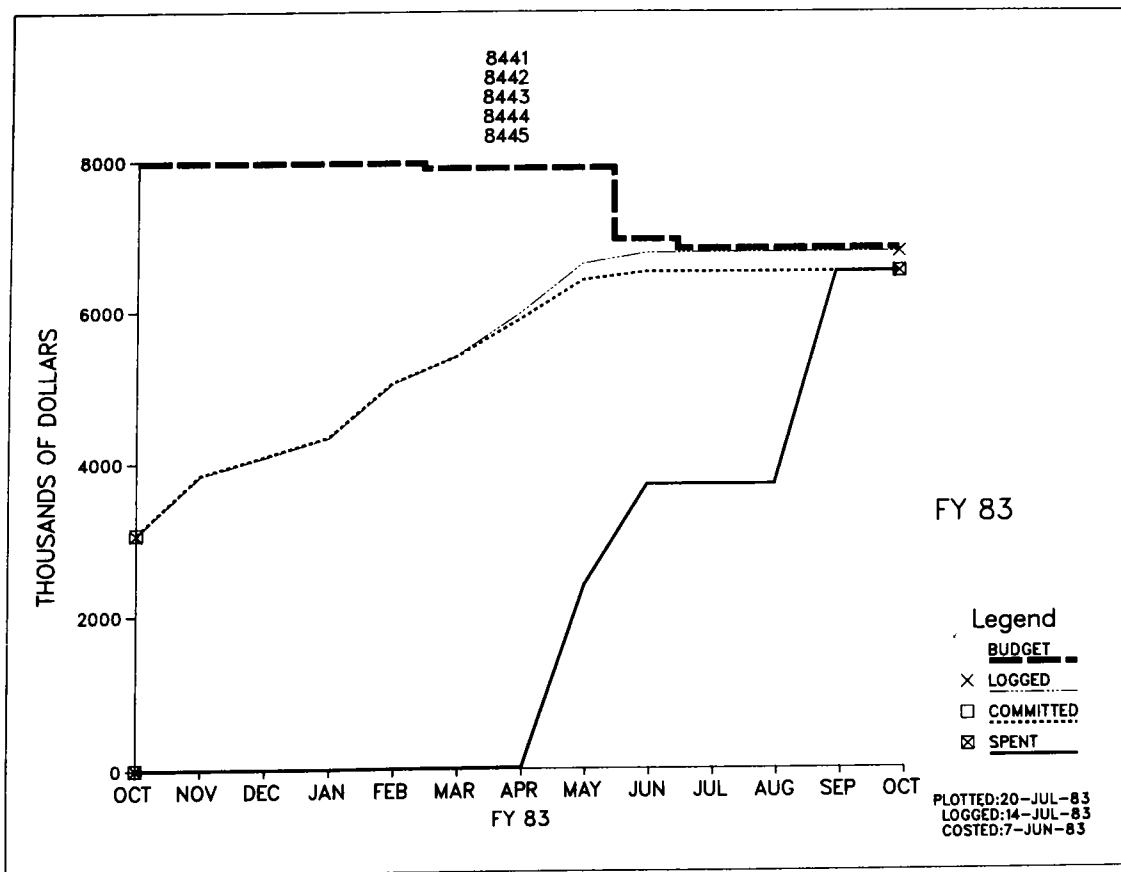


Figure 4. Monthly plot of DC purchases for the entire department.
 (Copy obtained from a film plot)

Another possible source of an error is computer down time. If files are lost or mutilated as a result of a computer system failure, the user may recover from a backup tape. This tape must be updated periodically to facilitate recovery in the event of a system failure.

If the PURCHASE.LIS file is manipulated using 'EDT', it may cause a problem since any slight error in the PURCHASE.LIS data base could produce erroneous output. It is strongly recommended that the users never modify PURCHASE.LIS using 'EDT', but rather use the 'INPUT' program.

III. Login and Logout Error Messages

The LOGIN and LOGOUT procedures check the PURCHASE.LIS file for errors. If an error is found, a message is written on the screen and the backup file [POGO.BACKUP]PURCHASE.BCK (normally written at LOGIN) is not written. First you should check the message for a problem description then act accordingly.

'FILE PURCHASE.LIS CURRENTLY LOCKED BY ANOTHER USER'

Do nothing. The error checks are not done in this case.

'ERROR ON PURCHASE.LIS OPEN STATEMENT. CODE = n'

This indicates a serious problem. The code number should be looked up in a Fortran manual for further assistance.

'PROCEED BY INITIALIZING A PURCHASE.LIS FILE'

No PURCHASE.LIS file was found. Simply proceed if this is the first data entry session. If this message appears after a PURCHASE.LIS file has been made, that requires troubleshooting.

'P.O. nn-nnnn OUT OF ORDER'

Since the programs put purchases in order, the most probable cause of this error is altering the PURCHASE.LIS file with 'EDT'. Since none of the programs will run with out-of-order data, it must be corrected. If this error was caused by 'EDT', the culprit should be warned to prevent future problems.

'RECLASSIFICATION ERROR ON PO= nn-nnnn'

The programs handle reclassification flags internally. Therefore, this also is an error that is most probably caused by someone in the editor. Handle in the same way as PO OUT OF ORDER.

'PURCHASE.LIS SHOULD HAVE n RECORDS
CURRENT COUNT IS m RECORDS'

If m is less than n, a short file was probably written. Generally, this problem is cured by deleting the newest version with the following commands:

```
$DELETE [POGO.LEDGER]PURCHASE.LIS;i (where i is the highest  
                                     numbered version)
```

```
$PURGE[POGO.LEDGER]PURCHASE.LIS
```

\$PURGE

If someone was running 'INPUT' in prompt mode and the computer quit or 'INPUT' was terminated abnormally and backup files PURCHASE.TMP had been made, the latest backup file may be brought in as the main data file with the following command:

```
$COPY [POGO]PURCHASE.TMP [POGO.LEDGER]PURCHASE.LIS
```

If you can't seem to find a good file, you may go back and pick up the backup file, which will be good, with the following command.

```
$COPY [POGO.BACKUP]PURCHASE.BCK [POGO.LEDGER]PURCHASE.LIS
```

The error can also be caused by someone using 'EDT' on the PURCHASE.LIS data base. IT IS VERY IMPORTANT TO STAY OUT OF THE DATA BASE WITH 'EDT'.

You may check to make sure whatever file you recovered is good by instituting the LOGIN check procedure with the following commands:

```
$D  
$@LOGIN
```

IV. Other Applications

Although untested, POGO can be set up to track D.C purchases for case or subcase managers instead of D.C. purchases for department managers and division supervisors. Appendix D explains the changes required.

PROGRAM INPUT

I. Input Procedure

Program 'INPUT' enters purchase orders into the data base and allows viewing of and changes to the data base. The user has a choice of entry methods 'prompt mode' and 'line mode'.

Prompt Mode

In prompt mode, by hitting the 'return' key, the user may scroll through the data to view individual purchases. Also in prompt mode program 'INPUT' queries for p.o. number. If the p.o. number is not already in the data base, the user is then prompted for date, division, case, consultant, vendor, item description and contract amount. Each entry is checked for errors before the next prompt. After the information for a p.o. is entered, it is displayed and can be corrected or changed. If there are no changes, the record is entered into the data base and the user is given a choice of entering more data or exiting the program. If he chooses to continue entering data, he can specify whether or not a backup file PURCHASE.TMP is written of the current status of the data base. This may be a valuable option especially if the computer goes down since the work is less likely to be lost if it's backed up. If the p.o. number entered exists in the data base, it is displayed in the upper scrolling window and the user is given the option of adding to the p.o. as in the event of a split case, deleting the p.o., changing any information in the record, reclassifying the order, or returning to the beginning of the program.

Line Mode

To use line mode, the user must prepare an input file using the VAX 'EDT' program. Purchases are input with an entry for each of the eight fields, p.o through contract amount, with at least one space or tab between each field. Program INPUT will read the input file, perform the same error checks on each field as prompt mode does and put acceptable purchases into the data base and put erroneous entries with a description of the error into a new version of the input file. The user should correct the errors and resubmit the file into line mode. The error messages are preceded by three asterisks (***) so that the program will ignore these lines when the file is resubmitted. The asterisks also precede input purchases that match p.o. number, case and division of a purchase already in the main data base.

II. Reclassification of a Purchase Order

Reclassification of a purchase order cannot be done on purchases that have not entered the SNLA data base; these are indicated in the SNLL data by a plus (+) sign in the third column of the p.o. number. The reason for this is that a purchase that has not reached SNLA can simply

be changed in the SNLL data base. (The SNLL data base p.o. is changed from a plus status when the p.o. appears on the SNLA costing data. This is handled by program 'COSTING'.)

To reclassify a purchase order, simply respond to the program prompts. When completed, the purchase order will appear at least twice, once with an @ symbol in the fifth column of the case number which indicates what SNLA believes is the account to be charged and again with an exclamation point (!) in the fifth column of the case number(s) of the following purchase(s) indicating what SNLL believes is/are the account(s) to be charged.

III. Split Accounts

When a purchase order is charged against more than one account (case and division), it must appear in the data base at least once for each account. In prompt mode, the program leads the user through this procedure. In line mode, to enter the same p.o. number more than once, it must be ten characters long which includes the two digit prefix, a dash (-), the four digit suffix, a blank or letter designator in the eighth column, and finally a two digit version number. The rest of the data fields are as previously described.

IV. Error Checks

Program 'INPUT' performs numerous error checks on each field of input data to insure that the data base is kept accurate.

The purchase order number is checked for length which can range from seven to ten characters. The third character must be a dash (-) and characters 1, 2, 9, and 10 and 4 to 7 must be digits. The eighth character may be a blank or a letter. The program cannot determine whether or not digits are transposed.

The date must have a slash (/) between digits. The numbers for the dates must be digits or an error message is printed. The numbers for month and day cannot be zero or exceed 12 and 31 respectively. The number for year cannot be zero and cannot be less than the current year minus one or greater than the current year plus one.

The division entry must be four characters long and consist of digits.

The case number must be eight characters long with the fifth character a period (.) and all others digits. Next, the division/case combination must match one of the accounts (division/case) in the account file to pass to the next field.

The consultant (synonyms: requester, originator) can be any length up to ten alpha characters. The entry is compared to the roster file. If there is not a match, the roster is printed and the user may select a

name from it. Initials of the consultant may be entered and the program will find the requester(s) with those initials. In the case of more than one person with the same initials, the user must enter enough characters to make the name unique.

Five characters are allowed for the vendor and they can be any character except blank or tab. After a purchase appears in the SNLA data, the first five characters of the vendor listed in the SNLA data are put into the SNLL data file.

The item description can have any characters including blanks up to a maximum of twenty five characters.

The contract amount must be digits and in the case of a no cost purchase, a zero (0) must be entered.

There are error checks on all file output write statements so that if an error occurs (e.g. over running the allocated disc space) the program will delete the file that it was writing when the error occurred and abort with an error message alerting the user.

The input is also error protected and if an error is encountered, a descriptive message is written.

V. Trouble Shooting

There are cases in which the program cannot distinguish between errors and gives the same error message for them.

UNABLE TO PARSE INPUT is written from line mode into the error file when the line of data had insufficient number of fields, too many fields, or no slashes in the date field. There may be others, but these are the most likely.

DATE NOT COMPATIBLE WITH THE JULIAN CALENDAR is written from line mode into the error file when the number for month is greater than 12, the number for day is greater than 31, the number for year is less than the current year minus one or greater than the current year plus one, and when month, day or year are not represented with two digits.

The other error messages are very specific and in most cases the solution is easily implemented.

VI. Input Files (Formats in Appendix B)

[POGO.LEDGER]PURCHASE.LIS	Main data base file.
[POGO.LEDGER]ACCOUNT.LIS	Account file containing budget information.
[POGO]ROSTER.LIS	Personnel roster.
[POGO]name.DAT	Input file for line mode.

VII. Output Files (Formats in Appendix B)

[POGO.LEDGER]PURCHASE.LIS	New version of main data base file.
[POGO]ACCOUNT.ERR	List of accounts found in PURCHASE.LIS not found in ACCOUNT.LIS.
[POGO]name.dat	Lines of erroneous input data with a message for each line explaining the error written from line mode.
[POGO]PURCHASE.TMP	Temporary backup of PURCHASE.LIS that can be written during work in prompt mode.

PROGRAM STRIP

I. Procedure

The financial data obtained from a UNIVAC computer at Sandia Labs in Albuquerque is transferred to Livermore, put into a format that is compatible with the VAX computer, then processed by program 'STRIP'. Acceptable purchases are those which have a charging organization that matches one of the divisions in the ACCOUNT.LIS file and are subsequently written into the COSTING.DAT file in numerical order by purchase order number.

Program 'STRIP' condenses the costing information to conserve file space. 'STRIP' adds together the accrued (ACCR) and paid current fiscal year (PDCFY) fields from the SNLA.DAT file to produce the "PAID CURRENT FISCAL YEAR" field on the COSTING.DAT file. It also adds the next fiscal year forecast (NXTFYF) and subsequent fiscal year forecast (SUBFYF) fields on the SNLA.DAT file to produce the "SUBSEQUENT FISCAL YEAR FORECAST" field in the COSTING.DAT file. Program 'STRIP' carries forward unchanged, the contract amount, remaining balance, current month spent, and current fiscal year forecast. Monthly forecasts are ignored.

If the paid previous fiscal year (PRPRFY) field is greater than or equal to the purchase order amount (POAMNT) field in the SNLA.DAT file, 'STRIP' marks the purchase with a '#' sign in the third character of the p.o. number on the COSTING.DAT file. This indicates that the p.o. was paid off in a previous fiscal year or reclassified.

Finally, 'STRIP' writes a command file for the next months costing procedure which contains the commands to copy SNLA tape data, to run 'STRIP', to run 'COSTING', and to run 'INPUT'.

II. Input Files (Formats in Appendix B)

[POGO]SNLA.DAT	Produced in Albuquerque and copied into the directory.
[POGO.LEDGER]ACCOUNT.LIS	Account file for finding acceptable divisions.

III. Output Files (Formats in Appendix B)

[POGO]COSTING.DAT	Contains updated SNLA financial data.
[POGO]MONTH.COM	Command file for subsequent month costing.

PROGRAM COSTING

I. Costing Procedure

The 'COSTING' program compares purchases in file PURCHASE.LIS to purchases from SNLA in file COSTING.DAT, then adds the SNLA financial information to matching purchase orders in PURCHASE.LIS.

Program 'COSTING' assumes that the SNLA purchase order data is correct and enters it into PURCHASE.LIS. 'COSTING' does not make any change to Purchase Orders on the old PURCHASE.LIS file that do not appear in the SNLA data base.

Purchases are deleted that have been replaced by higher versions of the same purchase order number. Purchases that have been marked with a pound (#) sign in file COSTING.DAT by 'STRIP' are also deleted from PURCHASE.LIS.

Purchases that are on the SNLA data file but are not in the POGO data file are written in an X.INP file in a format that is compatible with the line mode of program 'INPUT'. These purchases should be entered into the PURCHASE.LIS file using 'INPUT', then costed by re-running 'COSTING'. If the SNLA data contains more than one version of a purchase order and the POGO data contains only one version, 'COSTING' adds the missing version(s) to the POGO data automatically without writing it/them on the X.INP file.

Purchases that are in the POGO data base but do not appear in the SNLA data are written into an X.COS file. Typically, these purchases are the new requisitions that have not been entered into the SNLA data base yet.

II. Reclassifications

On SNLL logged reclassified orders that use the '@' and '!' signs, 'COSTING' updates the records with '@' symbols to be consistent with the convention that the @ represents information that SNLA is aware of and the ! represents information that SNLL is aware of. 'COSTING' only removes these symbols after the reclassification has been entered into the system and the case and division numbers from the SNLA records for that purchase match the case and division numbers for the purchase in the POGO data base.

In the event that a reclassification is not logged into the POGO PURCHASE.LIS file but is entered into the system and shows up on the SNLA data, 'COSTING' will make the PURCHASE.LIS file agree with the SNLA information.

III. New Fiscal Year

The new fiscal year is defined by 'INPUT' in October after the September SNLA data has entered the data base. 'COSTING' then writes the old fiscal year data onto archive files. When the October SNLA data is received in November, 'COSTING' will remove all purchases defined by the tape as PAID PRIOR FISCAL YEAR.

IV. Input Files (Formats in Appendix B)

[POGO]COSTING.DAT	Produced by program STRIP and has the costing data from the SNLA data base.
[POGO.LEDGER]PURCHASE.LIS;n	Current SNLL data base.
[POGO.LEDGER]ACCOUNT.LIS	Only for archiving.
[POGO]ROSTER.LIS	Only for archiving.

V. Output Files (Formats in Appendix B)

[POGO.LEDGER]PURCHASE.LIS;n+1	Updated version of current SNLL database.
[POGO]X.INP	List of purchases in COSTING.DAT that were not in PURCHASE.LIS, written in a format compatible with LINEMODE of program INPUT. Also, includes a listing of purchase requisitions that disappeared from the SNLA data base.
[POGO]X.COS	Listing of purchases in PURCHASE.LIS that were not in COSTING.DAT; written in the same format as COSTING.DAT.
[POGO.BACKUP]Pfy.ARC	Archive file of old PURCHASE.LIS.
[POGO.BACKUP]Afy.ARC	Archive file of old ACCOUNT.LIS.
[POGO.BACKUP]Rfy.ARC	Archive file of old ROSTER.LIS.

PROGRAM REPORT

Program 'REPORT' sorts the purchase order data per user specifications and builds files of tabular listings and instruction files that can be used by program 'TELL-A-GRAF' to produce monthly plots and bar graph summaries of the data. Upon entry into program 'REPORT', the user is given opportunity to specify whether or not plots or tables are desired, after which he is prompted for report specifications.

I. Plots

Monthly graphical output has four lines (e.g. Figure 4); one is a heavy dashed line for the budget, (2) a thin chain dashed line with an X at each end indicating SNLL known commitments, (3) a moderate dashed line indicating SNLA knowledge of SNLL commitments and (4) a moderate solid line showing SNLA costing. The last month of the plot shows the SNLA spending forecast for the fiscal year. The bar plot uses a solid bar to indicate the budget, a plaid bar to show SNLL known commitments, and a striped bar to show SNLA costing. The bars are cumulative from the beginning of the fiscal year through the month shown on the right side of the plot and, therefore, correspond to the data for the same month on the monthly plot. On some plots, the budget will show as zero because 'REPORT' only recognises budgets for divisions and/or case numbers.

II. Tabular Output

The tabular data shows one purchase per line with po number, date, division, case, consultant, vendor, item description, contract amount, remaining balance, current fiscal year forecast, and subsequent fiscal year forecast. To fit all categories on one line, the consultant name, vendor name and item description are shortened. When grouping on a certain field, that field is not printed to allow more room for the other fields.

Note that the sum of the contract amounts will not necessarily be equal to the sum of the committed amounts since many contracts span fiscal years.

III. Report Specifications

At any prompt, the user has a choice of making a selection or, by hitting the return key, passing to the next prompt.

Some prompts include a choice to 'group'. Grouping is taking each entry in a category and listing the purchases which apply under their respective entries. For example, if at the case number prompt the user responded by typing 'group', the result would be a listing of purchases for each case with the case numbers in numerical order. The user may

also type in a few case numbers then type 'group' and the result would be the same except that the grouping would be limited to the case numbers that were specified. Thus, a user can get the equivalent of the monthly SNLA printout by entering his division number at the division prompt and typing 'group' at the case number prompt. The user can enter his division number at the division prompt and type 'group' at the requester prompt and look at the purchases that the members of the division have generated. Figure 5 shows this information summarized in a bar chart, Figure 6 shows a tabular listing of some of the data used in the plot and Figure 7 summarizes the procedure to obtain the plot. If a user wants to look at integrated contractor orders, he may specify 92 at the purchase order prompt.

IV. Prompts

Some prompts, like the purchase order prompt, have special features. The purchase order prompt allows a sort on as few or as many characters of a purchase order as are entered. The third column with its special symbols (see User's Guide for symbol interpretation) is included if the two digit prefix and a dash, slash, question mark or plus, is entered. The special symbols are ignored if less than three or more than three characters of a p.o. are entered.

The case number prompt allows the user to enter the first four digits and the period of the case number and the program will sort purchases by the subcases within that case. This will not occur if 'grouping' is specified on a subsequent prompt.

The rest of the prompts, division, month, day, year, requester, item, vendor, and minimum purchase amount have no special features but allow the user to obtain the information desired.

V. Exit

The program will loop back to the beginning at the users request and more reports can be obtained. When the program is terminated, a message reminds the user of any plot or tabular files created as a result of running the program.

VI. Input Files (Formats in Appendix A)

[POGO.LEDGER]PURCHASE.LIS	Main data base file.
[POGO.LEDGER]ACCOUNT.LIS	Account file with budget information.
[POGO]ROSTER.LIS	Personnel roster.

[POGO.BACKUP]Pnn.ARC

Archive data base file of
previous fiscal year.
(nn is the year)

[POGO.BACKUP]Ann.ARC

Archive account file with
previous fiscal year
budget data. (nn = year)

[POGO.BACKUP]Rnn.ARC

Archive roster file.
(nn = year)

VII. Output Files (Formats in Appendix A)

[POGO]name.MON

Monthly plot file.

[POGO]name.BAR

Bar plot file.

[POGO]name.TAB

Tabular copy file

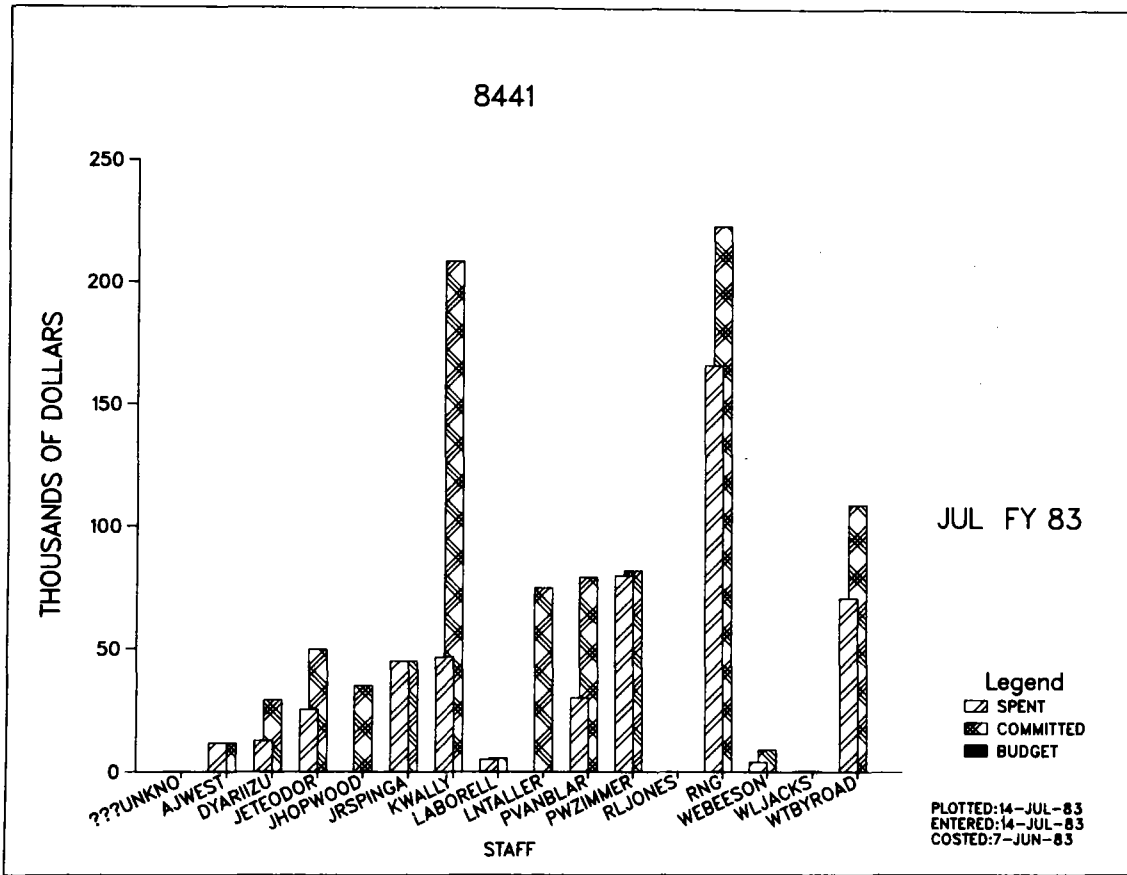


Figure 5. Level of commitment and spending by consultant. To obtain this plot, the TAG data file was edited to put all budget data to zero, to remove negative numbers in committed and spent, and to remove inappropriate names. (Copy obtained from a film plot)

**** DATE = 14-JUL-83 ****
 *** REPORT SPECIFICATIONS ***
 * DIVISION 8441 *

P.O. NO.	DATE	DIV.	CASE NO.	V.	ITEM	AMOUNT	BAL.	THIS FY	SUBS	FY
----- GROUP = KWALLY										
80-9549A04	04/11/83	8441	0193.200	DF	PD POWDER	5.8	0.0	0.0	0.0	0.0
80-9549A04	04/11/83	8441	0193.700	DF	PD POWDER	5.8	0.0	0.0	0.0	0.0
80-9549B02	04/11/83	8441	0193.200	EN	PD POWDER	3.5	0.0	0.0	0.0	0.0
80-9549B02	04/11/83	8441	0193.700	EN	PD POWDER	3.5	0.0	0.0	0.0	0.0
80-9549C01	05/09/83	8441	0193.200	EN	PD POWDER	2.5	0.0	0.0	0.0	0.0
80-9549C01	05/09/83	8441	0193.700	EN	PD POWDER	2.5	0.0	0.0	0.0	0.0
91?3915	00	05/03/83	8441	0193.600	UN %?=PURCHASE R	10.0	10.0	10.0	10.0	0.0
92-0061	00	03/29/83	8441	0033.760	MO FILTER ASSY S	25.0	21.0	21.0	21.0	0.0
92-0062	00	03/29/83	8441	0033.760	MO LASER RECLAM	30.0	27.0	27.0	27.0	0.0
92-0063	00	03/29/83	8441	0033.760	MO TRIAL LASER R	20.0	17.0	17.0	17.0	0.0
92-9644	01	07/23/82	8441	0193.600	MO NOT VERIFIED	15.0	0.0	0.0	0.0	0.0
92-9906	01	12/20/82	8441	0193.600	MO PCT WORK	53.3	37.0	37.0	37.0	0.0
92-9986	00	02/11/83	8441	0033.760	EI RECLAMATION	50.0	50.0	50.0	50.0	0.0
			[BUDGET	0.0].....TOTALS.....	227.1	161.9	161.9	161.9	0.0
----- GROUP = LABORELLO										
81-5007	00	06/04/82	8441	0193.200	HE INFORMATION S	1.0	0.0	0.0	0.0	0.0
81-5064	00	09/07/82	8441	0720.270	NI NOT VERIFIED	0.2	0.0	0.0	0.0	0.0
81-5065	00	06/29/82	8441	0560.100	TE LAB HARDWARE	0.3	0.0	0.0	0.0	0.0
81-8060	00	09/13/82	8441	0720.270	HE BLANK DISCS	0.1	0.0	0.0	0.0	0.0
81-8827	00	11/12/82	8441	0720.200	AP GRINDING WHEE	0.1	0.0	0.0	0.0	0.0
81-8833	00	11/15/82	8441	0853.200	HE THERMAL PAPER	0.4	0.0	0.0	0.0	0.0
81-8834	00	11/15/82	8441	0853.200	TE PRESSURE REGU	1.2	0.0	0.0	0.0	0.0
81-8894A00	12/21/82	8441	1292.200	GE NOT VERIFIED	0.1	0.0	0.0	0.0	0.0	0.0
81-8894B00	01/26/83	8441	1292.200	GE NOT VERIFIED	0.1	0.0	0.0	0.0	0.0	0.0
91-1411	00	01/27/83	8441	1292.200	BA O-RINGS	0.0	0.0	0.0	0.0	0.0
91-1418	00	02/01/83	8441	1292.200	CE JAM NUTS	0.5	0.0	0.0	0.0	0.0
91-1429	00	02/18/83	8441	1292.200	GE NOT VERIFIED	0.3	0.0	0.0	0.0	0.0
91-1433	00	02/22/83	8441	1292.200	SN NOT VERIFIED	0.5	0.0	0.0	0.0	0.0
91-1436	00	02/25/83	8441	1292.200	OA NOT VERIFIED	0.3	0.0	0.0	0.0	0.0
91-1437	00	02/25/83	8441	1292.200	CO NOT VERIFIED	0.1	0.0	0.0	0.0	0.0
91-1462	00	03/18/83	8441	1292.200	OM THERMOCOUPLE	0.3	0.3	0.3	0.3	0.0
91-1489	00	04/08/83	8441	1292.200	HE INTERFACE BOA	0.3	0.3	0.3	0.3	0.0
			[BUDGET	0.0].....TOTALS.....	5.8	0.6	0.6	0.6	0.0
----- GROUP = LNTALLERIC										
91+3363	00	04/19/83	8441	0501.200	ME COLDWIRE FEED	50.0	50.0	50.0	50.0	0.0
91+3363	00	04/28/83	8441	0864.200	ME COLD WIRE FEE	25.0	25.0	25.0	25.0	0.0
			[BUDGET	0.0].....TOTALS.....	75.0	75.0	75.0	75.0	0.0

Figure 6. A portion of the data used to produce the bar graph of Figure 5.

\$REPORT

1 MONTHLY PLOT none.MON
2 BAR PLOT none.BAR
3 TABULAR OUTPUT none.TAB

ENTER THE NUMBER OF THE ITEM YOU WISH TO CHANGE.
(none INDICATES NO FILE WRITTEN)

2

ENTER A NAME TO REPLACE none IN FILE none.MON OR [RETURN]
PLOT

1 MONTHLY PLOT none.MON
2 BAR PLOT PLOT.BAR
3 TABULAR OUTPUT none.TAB

ENTER THE NUMBER OF THE ITEM YOU WISH TO CHANGE.
(none INDICATES NO FILE WRITTEN)

.

Specify 3 and replace 'none' to get a tabular output file.

.

ENTER DIVISION NUMBER, GROUP, OR RETURN

8441

ENTER DIVISION NUMBER, GROUP, OR RETURN [RETURN]

.

.

ENTER REQUESTERS FIRST AND SECOND INITIALS WITH SURNAME, OR GROUP,
OR RETURN.

GROUP

.

Hit return on all other prompts.

.

FILE PLOT.BAR WRITTEN

FILE PLOT.TAB WRITTEN

THAT WAS FUN!!! CAN WE DO IT AGAIN? Y/N

N

LAST BAR PLOT FILE PLOT.BAR

LAST TABULAR FILE PLOT.TAB

\$

Edit the PLOT.BAR file to replace negative numbers with
zeros, remove inappropriate names with their respective
SPENT, COMMITTED and BUDGET amounts and set the first
budget number (division 8441 budget) to zero.

\$TAG

Figure 7. Summarized procedure to obtain Figures 5 and 6.

PROGRAM ACCOUNT

Program 'ACCOUNT' reads and writes file ACCOUNT.LIS which contains the budget information and must be run when the budget changes, to update the file.

The program checks for errors on data input. A division must have four characters which are digits. A case number must be eight characters, seven of which must be digits, with the fifth character a period. A budget amount must be a non-negative number.

There are twelve fields that carry budget information for each account. These fields correspond to the months of the fiscal year with October as the first month. When there is a budget change during the fiscal year, the new budget amount is entered in the appropriate month and the program extends the new number into the months following. Thus, when a monthly plot is done, the budget line will reflect the changes that have occurred during the fiscal year.

To run 'ACCOUNT', use the command

\$ACCOUNT

Input File

ACCOUNT.LIS

Output File

ACCOUNT.LIS

OTHER PROGRAMS

I. Programs CHECK and INCHECK

'INCHECK' is run at login and displays the message 'PURCHASE FILE VERIFICATION ... ONE MOMENT PLEASE'. The program checks the length of the file PURCHASE.LIS against the number of records listed on the third line of PURCHASE.LIS, looks for reclassification errors, and makes sure that the purchase order numbers are in strict numerical order. Upon successful completion of the checks, a backup file [POGO.BACKUP] PURCHASE.BCK is written. If any errors are encountered, an error message is printed and the backup file is not written. If the PURCHASE.LIS file is in use by another user, the program says so, the error checks are not done, and the backup file not written.

'CHECK' is run at logout and is identical to 'INCHECK' except that 'CHECK' does not write a backup file and if an error is encountered, the user cannot logout until the error is corrected. The exception to this is system logout if the terminal has been unused for 20 minutes.

II. Programs POGO and MESSAGE

'POGO' writes the warning and bold characters at login. 'MESS' writes a special message if there is a MESSAGE.LIS file.

III. Program ERASER

'ERASER' is run at logout to clear the screen to facilitate terminal security. The screen is not cleared if the system automatically logs the user off after 20 minutes of no use.

SPECIAL CHARACTERS FOUND IN THE PO FIELD

There are special characters found in the third character of the purchase order number field.

- Normal character, purchase order found on SNLA data base.
- + Purchase Requisition in the SNLL data, not yet in SNLA (INPUT).
- ? Purchase Requisition found on the SNLA data base (STRIP).
- # Character indicating that the PO was paid off in a previous fiscal year or was reclassified by SNLA. (STRIP).
The PO will be removed from PURCHASE.LIS by COSTING.
- & Indication that an SNLA PR disappeared from the data base without ever appearing as a PO (COSTING).
- / Indication that a PO was once found on the SNLA tape but has now disappeared, probably because it was paid off (COSTING).
These records are automatically deleted if replaced by higher version numbers.

SPECIAL CHARACTERS FOUND IN CASE NUMBER FIELD

There are special characters found in the fifth character of the Case Number field.

- . Normal character.

Reclassification Symbols

- @ Indicates the case(s) and division(s) that SNLA believes the PO is charged to.
- ! Shows the SNLL reclassification intent. Indicates the case(s) and division(s) that SNLL wants the PO charged against.

Reclassification symbols are removed when SNLA agrees with SNLL.

SPECIAL CHARACTER FOUND IN THE ITEM DESCRIPTION FIELD

A '%' sign in the first character of the item description field indicates that the purchase order was found first on the SNLA tape before it was installed as SNLL data.

INSTRUCTIONS TO LOG PURCHASE ORDERS ONTO THE VAX (HOWINPUT.LIS 5/25/83)

1.0 LOG ON

```
BREAK [ return ]                Terminal power on
enter classVAX2 [ return ]
[ return ]
USERNAME:user [ return ]
PASSWORD:      [ return ]
CLASSIFICATION:U [ return ]
ARE YOU ON A VT100?Y [ return ]
$                                Dollar sign indicates ready
```

2.0 PO INPUT, PROMPT MODE

(To add, change, reclassify(see 6.0) or delete PO's)

```
$INPUT [ return ]
  INPUT DATA FILE NAME OR [ P ] FOR PROMPT MODE
  P [ return ]
```

2.1 If the PO entered matches one in the data file, the PO will be displayed in the upper window and the user has the option of deleting or editing the PO.

2.2 A backup file can be written on PURCHASE.TMP after every PO entry.

3.0 PO INPUT, LINE MODE

(To add many PO's)

```
$EDT [name].DAT [ return ]      Create a data file with
                                your name as the filename
                                (Example: $EDT PAUL.DAT)
```

```
INPUT FILE DOES NOT EXIST (first time message from EDT)
[EOB]                            Type in PO data in the
                                correct format (See 4.0)
```

```
[ CTRL Z ]
*EXIT [ return ]
DRB1:[user]name.DAT;N M LINES
$INPUT [ return ]
ENTER DATA FILE NAME OR [ P ] FOR PROMPT MODE
name.DAT [ return ]
FILE name.DAT CONTAINS 3 ERRORS (or how many errors were found)
***** SEE YOU NEXT TIME *****
EDT name.DAT [ return ]          Correct errors by editing
                                the name.DAT file then
                                re-running INPUT using
                                name.DAT
```

4.0 FORMAT FOR DATA FILE USING LINE MODE (SEE SECTION 3.0)

```
**PO#.... /MM/DD/YY/DIV./CASE#..... /STAFF..... /VEND./ITEM..... /AMOUNT*  
92-8116 06/23/82 8441 0720.270 PVANBLARIGA RF IRON FORGINGS 30000.  
-OR- the following line would produce identical results in the file  
92-8116 00 6/23/82 8441 0720.270 Pv RF iron forgings 30000
```

NOTE: Order and number of categories are important and the amount must be a number without commas. There must be at least one space or tab between categories. Acceptable separator between month, day and year is [/]. Leading zeros are not necessary in the month and day in line mode (they are required in PROMPT MODE).

The only fields that accept embedded blanks are the item and purchase order fields. In the purchase order field, a single blank is accepted between the seven character PO and its two digit change order number (synonym: version number).

The program converts all letters to upper case letters.

SHORT CUT: Type a single slash [/] for date, as shown, to automatically enter the date from the computer.
92-8116 / 8441 0720.270 Pv RF iron forgings 30000

5.0 SCREEN SPLIT IN PROMPT MODE (SEE SECTION 2.0)

THE SCREEN IS SPLIT INTO TWO WINDOWS

The UPPER window displays a PO in the current data base. To display other PO's, hit RETURN. To change the direction of the scroll, hit the \ key and RETURN. A line of either +'s or -'s indicates scroll direction. A <BEEP> indicates top or bottom of list and change of scroll direction.

The LOWER window displays the commands being entered.

6.0 RECLASSIFICATION OF A PURCHASE ORDER (Prompt mode only)

Reclassification of purchases may only be done on those purchases that have appeared in the SNLA data tape (third character of po is not a plus (+) sign). To reclassify, enter the purchase order number at the enter-a-PO prompt. The program will find the purchase and offer a choice of action. Respond by typing an 'R', then division, case and amount. If a purchase is being reclassified to more than one account, answer yes at 'DO YOU WISH TO ADD ANOTHER LINE?'

Reclassification is only permanent if the account (case/division) is being changed. If only the amount is being changed, 'COSTING' will put the amount from the next month SNLA data into the PO and remove the reclassification status.

Once a month, SNLA sends a 9 track ASCII, 1600 BPI data tape containing 499 character records (see Appendix B) of purchase orders. After receiving the data tape from SNLA, it must be named and entered into the VAX2 tape library against the user's account before the costing procedure is done.

The tape is named P_mon, for example, P_AUG. The month is the month to which the data applies. The tape received in September contains the August end-of-month data and, therefore, is named P_AUG.

I. Automated Method

- 1.0 After the SNLA data tape is in the VAX2 library, the costing procedure (steps 1.0 to 4.0 of Manual Method) is started with the following command (ref. Appendix C)

```
$@MONTH
```

This command will require some minutes to execute and, when complete, will inform the user, via message on the screen, whether or not an X.INP file exists.

- 2.0 If there is an X.INP file, follow step 5.0 of II. Manual Method.

II. Manual Method

- 1.0 Begin by copying the SNLA data from tape to disc on the VAX2.

```
$ MOUNT/FOREIGN MF: P_mon TAPE
```

(Where 'P_mon' is the name assigned to the tape and 'mon' are the first three letters of the month to which the data applies.)

```
$ TAPECOPY
TAPE
function? REWIND
function? SKIP
how many records? 1 (this is the title block)
function? READ
    enter output file  SNLA.DAT
    length of records  499
    default on everything else
function? STOP
```

```
$ DISMOUNT TAPE
```


- 2.0 After copying the data, run program 'STRIP' to write the file COSTING.DAT.
- ```
$ STRIP
$ DELETE SNLA.DAT (optional to conserve file space)
```
- 3.0 The SNLA costing data is entered into file PURCHASE.LIS by program 'COSTING' run by the command
- ```
$COSTING
```
- 4.0 If 'COSTING' writes an X.INP file (indicated by a message on the screen at the end of the run), the purchases in the X.INP file are entered into PURCHASE.LIS by running 'INPUT' in line mode.
- ```
$INPUT
 ENTER DATA FILE NAME OR [P] FOR PROMPT MODE.
 X.INP [return]

 FILE X.INP CONTAINS n ERRORS (n is the number of errors)
 ***** SEE YOU NEXT TIME *****
```
- \$  
If there is no error message, simply repeat step 3.0.
- 5.0 If the X.INP error file is written from 'INPUT', edit the file, correct the errors, and re-run 'INPUT'. Repeat this step until there are no errors, then repeat step 3.0.
- 6.0 When the costing is completed, the data should be backed up on tape so that it can be recovered if the computer crashes (See Tape Backup Procedure).

PLOT INSTRUCTIONS (Stored as HOWPLOT.LIS on 5/25/83)

Plotting can be done on a VT100 Retro-Graphics or on a Tektronix terminal. Department total bar chart by division and monthly plot file for each division can be obtained quickly by use of the command '\$@PLOT' (see Appendix C) instead of step 1.0.

1.0 To produce a plot, you must run the REPORT program.

\$ REPORT [ return ]

There are two types of plots that can be produced. Specify name.BAR for bar graphs and/or name.MON for monthly plots.

2.0 Start the TELL-A-GRAPH program with a TAG command.

\$TAG [return ]

If TAG requests a device number, enter the appropriate response.

2.1 Enter the name of the file produced by REPORT.

EXAMPLES= SPECIFY FILES: name.BAR [ return ]

-or-

SPECIFY FILES: name.MON [ return ]

--or--

SPECIFY FILES: name.BAR (XEQ [ return ]

-or-

SPECIFY FILES: name.MON (XEQ [ return ]

2.2 If the (XEQ qualifier is used, TAG will plot on the screen and go to section 2.4. If the qualifier is not used, TAG will then request a command. Enter one of the following commands at the

GENERATE LEVEL..ENTER: prompt.

GEN HIDDEN BAR. [ return ]

-or-

GEN MONTHLY PLOT. [ return ]

NOTE: ALL TAG commands MUST end with a period.

2.3 TAG will prepare a plot file and return for another command. Enter one of the following commands at the ENTER: prompt.

GO. [ return ] (To plot on the device you are using.)

-or-

SEND. [ return ] (To plot film on the dicomed. See 2.6)  
(Program may take quite a long time to execute GO or SEND)

2.4 After viewing or copying the plot, type any key to clear the screen.  
If the (XEQ qualifier was used, the program will terminate. If the qualifier was not used, TAG will return you to step 2.2 above where you may specify plotting additional plots.

2.4.1 To switch to a bar chart after plotting a monthly plot, at the prompt = GENERATE LEVEL..ENTER:, type DATA FILE "name.BAR". [ return ]

2.4.2 To switch to a monthly plot after plotting a bar chart, at the prompt = GENERATE LEVEL..ENTER:, type DATA FILE "name.MON". [ return ]

2.5 If you are finished, at the prompt = GENERATE LEVEL..ENTER:, enter QUIT. [ return ] [ return ]

2.5.1 VT100 Retro-Graphics: press [ SET-UP ], [ RESET (on the zero key) ] to get out of graphics mode.

2.5.2 Tektronix terminals: press page to clear the screen.

2.6 To obtain report quality graphs, execute 2.3 [ SEND ], then exit the 'TAG' program as described in 2.5 then enter the command  
\$DICOMED METAFIL. [ return ]

To preview the metafile, enter PVIEW command.

\$PVIEW

enter device number 1 [ return ] (even on a VT100)

enter metafile name METAFIL [ return ] (no period)

(Use [ SPACE BAR ] [ return ] to view each plot; use

END [ return ] to terminate.

Pick up film at the I/O desk, then take the film to drawing reproduction to obtain report quality plots.

3.0 If, on the VT100 Retro-Graphics, the plot is scribbley, do the following;

\$GRAPH [ return ]

fx

At the [ fx ] press [ SET-UP ] then [ SET-UP A/B ] (the 5 key)

Next press the [ TRANSMIT SPEED ](7 key) and

[ RECEIVE SPEED ](8 key) to change the transmit and receive speeds to 4800. Finally press [ SET-UP ] then go to 2.0.

## 5.0 LIST INSTRUCTIONS (Stored as HOWLIST.LIS on 5/25/83)

To produce a list, you must run the REPORT program.

```
$ REPORT [return]
```

To produce a listing on the Diablo or the line printer, you must specify a tabular file name [ name.TAB ] at the beginning of the REPORT program.

## 5.1 DIABLO PRINTING (if your terminal has a diablo assigned to it)

```
$REP [return]
```

Follow directions of the program. Typically, just use default values by hitting [ return ] when appropriate.

## 5.2 LINE PRINTING

```
$PRINT name.TAB [return]
```

(Line printer output is delivered to an assigned box in the computer room.)

## TAPE BACKUP PROCEDURE

It is important to backup the files in the POGO system to tape. This should be done after every group of major changes but certainly not less than once a month. The best time to do a backup is just before running the monthly 'COSTING'. The following procedure assumes that the user

has obtained two permanent tape assignments from the computer center and has changed the command files as described in the 'SET-UP INSTRUCTIONS'.

### I. Writing a Backup Tape

Command files have been prepared to do the backups so the procedure is relatively easy. Use the following commands:

\$D SOURCE

\$EDT DIARY.LIS

Go to the end of the diary and see which tape was used for the last backup. There are two tapes, S and B. Select the one which was not used in the last backup and write a message on the diary indicating which tape you are using for backup.

Use one of the following two commands for your backup, depending on which tape you are using for backup:

\$\$@SAVES

- OR -

\$\$@SAVEB

When the backup is finished (which will take quite a long time depending on how fast the operator gets your tape mounted), return to the main directory using the following command:

\$D

### II. Recovering Data From a Backup Tape

If you do not know which is the latest backup tape, it will be necessary to load and read the directory on both tapes to find out which is the newest. The best files to look at are the [POGO.SOURCE]DIARY.LIS or [POGO.LEDGER]PURCHASE.LIS.

TO LOOK AT THE TAPE DIRECTORY USE THE FOLLOWING COMMANDS:

\$D SOURCE

\$\$@TDIRS

Look at the directory dates. If you miss them you may look again while the tape is still mounted with,

\$DSD TAPE -OR- \$DSD TAPE:file\_name

\$\$@DISMOUNT

\$@TDIRB

Look at the directory dates. If you miss them you may look again while the tape is still mounted with,

\$DSD TAPE -OR- \$DSD TAPE:file\_name

\$@DISMOUNT

TO WRITE ALL OF THE FILES FROM A BACKUP TAPE INTO THE DIRECTORY USE:

\$D SOURCE

\$@GETS -OR- \$@GETB

TO GET ALL VERSIONS OF SPECIFIC FILES FROM A BACKUP TAPE USE:

\$D SOURCE

\$@TDIRS -OR- \$@TDIRB

\$COPY TAPE:file\_name;\* [POGO.dir]\*

EXAMPLE: COPY TAPE: PURCHASE.LIS;\* [POGO.LEDGER]\*

\$@DISMOUNT

## SET-UP INSTRUCTIONS

These instructions indicate how the POGO system can be set up in your file space to operate with your organization. First you must have file space assigned to you on VAX2 (or you may transfer the programs to another VAX with tape commands instead of step 2). The instructions below assume that you can set up on VAX2. You must have a directory space that is at least 10,000 blocks and you should substitute your username in all of the following steps.

### I. Create Subdirectories

Create the following sub-directories in your directory space using CREATE/DIR commands.

```
CREATE/DIR [username.SOURCE]
CREATE/DIR [username.BACKUP]
CREATE/DIR [username.LEDGER]
```

It is not necessary that you use these sub-directories but this procedure assumes these directories. You may use your own choices of where to place files by proper changes in step 6 and step 9.

### II. Copy Files

Copy all files and directories over into your file space with the following commands.

```
$COPY [POGO]*.*;* [username]*
$COPY [POGO.SOURCE]*.*;* [username.SOURCE]*
```

### III. Delete and Purge Files Unique to 8440

Delete and purge files with the following commands.

```
$DELETE [username]*.dat;*
$DELETE [username]*.err;*
$DELETE [username]X*.*;*
$DELETE [username]*.MON;*
$DELETE [username]*.BAR;*
$DELETE [username]*.TAB;*
$DELETE [username.SOURCE]*.LIS;*
$PURGE/KEEP=1
$PURGE/KEEP=1 [username.SOURCE]
```

The above procedure may not have eliminated all of the files unique to 8440. You may safely delete any that you find.

#### IV. Set LOGIN.COM and LOGOUT.COM Files

Use 'EDT' to edit the files  
\$EDT LOGIN.COM

```
change @[POGO.SOURCE]ALOGIN.COM
to @[username.SOURCE]ALOGIN.COM
```

Repeat the edit procedure for the LOGOUT.COM file.

#### V. Set Up ALOGIN.COM File

Move to SOURCE sub-directory with the following command.  
\$D SOURCE

Load into 'EDT' the login co5i

```
delete $ASSIGN [POGO.DOC] MY$DICT:

change $RUN [POGO.SOURCE]INCHECK
to $RUN [username.SOURCE]INCHECK

change $PURGE/KEEP=1 [POGO.BACKUP]
to $PURGE/KEEP=1 [username.BACKUP]

change $INPUT:==RUN [POGO.SOURCE]INPUT
to $INPUT:==RUN [username.SOURCE]INPUT

change $STRIP:==RUN [POGO.SOURCE]STRIP
to $STRIP:==RUN [username.SOURCE]STRIP

change $COSTING:==RUN [POGO.SOURCE]COSTING
to $COSTING:==RUN [username.SOURCE]COSTING

change $ACCOUNT:==RUN [POGO.SOURCE]ACCOUNT
to $ACCOUNT:==RUN [username.SOURCE]ACCOUNT

change $REPORT:==RUN [POGO.SOURCE]REPORT
to $REPORT:==RUN [username.SOURCE]REPORT

change $RUN [POGO.SOURCE]MESS
to $RUN [username.SOURCE]MESS
```



## VI. Set Up ALOGOUT.COM File

Load into 'EDT' the logout commands with the following command.

```
$EDT ALOGOUT.COM
```

Change all occurrences of [POGO] to [username].

```
EXAMPLE: change $RUN [POGO.SOURCE]CHECK
 to $RUN [username.SOURCE]CHECK
```

## VII. Set Up Command Files

Edit the [username.SOURCE]FILE.COM file using 'EDT' to change [POGO] to [username] in every line of that file.

```
$D SOURCE
$EDT FILE.COM
```

```
EXAMPLE: change $ASSIGN [POGO.LEDGER]PURCHASE.LIS F10
 to $ASSIGN [username.LEDGER]PURCHASE.LIS F10
```

Edit all [username]any.COM files in the same way.

## VIII. Backup Tapes

The following command files on the SOURCE sub-directory make and read backup tapes. Obtain your own backup tape assignments from the computer center and substitute your tape numbers into these command files using 'EDT'. The numbers that must be changed are "V240" and "V241".

```
GETS.COM;*
SAVES.COM;*
TDIRS.COM;*
GETB.COM;*
SAVEB.COM;*
TDIRB.COM;*
```

Return to the main directory and exercise the login command file with the following commands to initialize the tape assignments.

```
$D
$@LOGIN
```

## IX. Open ROSTER.LIS File

Make sure that you are in the main directory with the command

```
$D
```

Use 'EDT' to create a roster file. Enter first and second initial(s) (if there is a second initial) WITHOUT periods or blanks and the surname with only one name per line. Do not enter the same name twice.

```
$EDT ROSTER.LIS
```

## X. Open ACCOUNT.LIS File

Run the 'ACCOUNT' program to create the accounts file. To properly initialize the file, a list of budget numbers by case and division is required.

```
$ACCOUNT
```

The programs are now set up to enter data using the 'INPUT' program using either the prompt mode or line mode.

## XI. Arrange For Data Transfer From SNLA to SNLL

Division 2625, Financial Systems Division in Albuquerque, will cull data for the user from the main data base from the UNIVAC computer upon user request. Be sure to specify sort on charging organization only.

Section 2632-5, Production Control Section in Albuquerque, may also need a request to have the data written onto a tape then mailed to the user.

## XII. Notes

This documentation is stored on UDO:[POGO.DOC]\*.DOC. For documentation customized to your department, you may copy the '\*.DOC' files into your directory then edit the copy, changing POGO to 'your name' (except where POGO means program 'POGO', in SET-UP INSTRUCTIONS, and FILE NAME ASSIGNMENTS in Appendix A).

When you implement POGO in your department, please send a mail message to user POGO on VAX2 so that your department can be kept informed of any changes in the system.

## LIST OF FILES

### I. Information Files

|                          |                                                                                       |
|--------------------------|---------------------------------------------------------------------------------------|
| [POGO]HOWINPUT.LIS       | INSTRUCTIONS FOR 'INPUT' PROGRAM.<br>( 'TYPE HOWINPUT.LIS' TO SEE).                   |
| [POGO]HOWREAD.LIS        | INSTRUCTIONS FOR INTERPRETING THE<br>SPECIAL CHARACTERS IN THE<br>PURCHASE.LIS FILE.  |
| [POGO]HOWLIST.LIS        | INSTRUCTIONS FOR LISTING DATA<br>WITH 'REPORT' PROGRAM.                               |
| [POGO]HOWPLOT.LIS        | INSTRUCTIONS FOR PLOTTING DATA<br>WITH 'REPORT' PROGRAM.                              |
| [POGO]HOWSTRIP.LIS       | INSTRUCTIONS FOR PROCESSING DATA<br>FROM SNLA DATA BASE USING<br>THE 'STRIP' PROGRAM. |
| [POGO]HOWCOST.LIS        | INSTRUCTIONS FOR COSTING PROCEDURE.                                                   |
| [POGO]DIR.LIS            | THIS LIST OF FILES.                                                                   |
| [POGO.SOURCE]DIARY.LIS   | DIARY OF BACKUP TAPES AND PROGRAM<br>CHANGES.                                         |
| [POGO.SOURCE]MESSAGE.LIS | SPECIAL MESSAGE AT LOGIN.                                                             |

### II. Backup Data Files

|                           |                                                                                      |
|---------------------------|--------------------------------------------------------------------------------------|
| [POGO.BACKUP]PURCHASE.BCK | BACKUP FILE OF PURCHASE.LIS<br>PRODUCED BY PROGRAM 'INCHECK'.                        |
| [POGO.BACKUP]Pfy.ARC      | ARCHIVE FILE OF PURCHASE.LIS<br>PRODUCED BY PROGRAM 'COSTING'<br>AT NEW FISCAL YEAR. |
| [POGO.BACKUP]afy.ARC      | ARCHIVE FILE OF ACCOUNT.LIS<br>PRODUCED BY PROGRAM 'COSTING'<br>AT NEW FISCAL YEAR.  |
| [POGO.BACKUP]rfy.ARC      | ARCHIVE FILE OF ROSTER.LIS<br>PRODUCED BY PROGRAM 'COSTING'<br>AT NEW FISCAL YEAR.   |

### III. Program Files

|                          |                                  |
|--------------------------|----------------------------------|
| [POGO.SOURCE]INPUT.FOR   | DATA INPUT PROGRAM.              |
| [POGO.SOURCE]REPORT.FOR  | LISTING AND PLOTTING PROGRAM.    |
| [POGO.SOURCE]COSTING.FOR | COSTING PROGRAM.                 |
| [POGO.SOURCE]ACCOUNT.FOR | ACCOUNT PROGRAM.                 |
| [POGO.SOURCE]STRIP.FOR   | EDIT SNLA DATA BASE FILE.        |
| [POGO.SOURCE]INCHECK.FOR | CHECKS PURCHASE.LIS AT LOGIN.    |
| [POGO.SOURCE]CHECK.FOR   | CHECKS PURCHASE.LIS AT LOGOUT.   |
| [POGO.SOURCE]ERASER.EXE  | ERASES SCREEN AT LOGOUT.         |
| [POGO.SOURCE]POGO.FOR    | WRITES FIXED LOGIN COMMENTS.     |
| [POGO.SOURCE]MESS.FOR    | WRITES SPECIAL MESSAGE AT LOGIN. |

### IV. Working Data Files

|                        |                                                                                                     |
|------------------------|-----------------------------------------------------------------------------------------------------|
| [POGO]filename.DAT;n   | USER NAMED FILE FOR DATA INPUT<br>WHERE 'filename' IS ANY NAME.                                     |
| [POGO]SNLA.DAT         | FILE CREATED BY PROGRAM 'TAPECOPY'<br>THAT HAS DATA COPIED VERBATIM<br>FROM THE SNLA DATA TAPE.     |
| [POGO]COSTING.DAT      | FILE CREATED BY PROGRAM 'STRIP'<br>THAT HAS DATA STRIPPED AND<br>REFORMATTED FROM THE SNLA.DAT.     |
| [POGO]PURCHASE.TMP     | INTERIM FILE CREATED BY USER<br>DURING A RUN OF PROGRAM 'INPUT'                                     |
| [POGO]filename.DAT;n+1 | ERROR FILE WRITTEN FROM PROGRAM<br>'INPUT' (USER NAMED FILE.)                                       |
| [POGO.LEDGER]X.COS     | FILE CREATED BY PROGRAM 'COSTING'.<br>LISTS RECORDS IN PURCHASE.LIS<br>THAT ARE NOT IN COSTING.DAT. |
| [POGO.LEDGER]X.INP     | FILE CREATED BY PROGRAM 'COSTING'.<br>LISTS RECORDS IN COSTING.DAT<br>THAT ARE NOT IN PURCHASE.LIS. |

|                          |                                           |
|--------------------------|-------------------------------------------|
| [POGO]name.BAR           | BAR GRAPH FILE USED BY TELL-A-GRAF.       |
| [POGO]name.MON           | MONTHLY PLOT FILE USED BY<br>TELL-A-GRAF. |
| [POGO]name.TAB           | TABULAR LISTING.                          |
| [POGO.SOURCE]MESSAGE.LIS | SPECIAL MESSAGE FILE.                     |

#### V. Permanent Data Files

|                           |                                                       |
|---------------------------|-------------------------------------------------------|
| [POGO.LEDGER]PURCHASE.LIS | MAIN DATA BASE.                                       |
| [POGO]ROSTER.LIS          | STAFF ROSTER.                                         |
| [POGO.LEDGER]ACCOUNT.LIS  | ACCOUNTS BY DIVISION AND CASE<br>WITH BUDGET AMOUNTS. |

#### VI. Command Files

|                          |                                                                                          |
|--------------------------|------------------------------------------------------------------------------------------|
| [POGO]LOGIN.COM          | CALLS [POGO.SOURCE]ALOGIN.COM.                                                           |
| [POGO.SOURCE]ALOGIN.COM  | CALLS 'INCHECK' PROGRAM, DOES<br>LOGIN COMMANDS AND THEN<br>CALLS [POGO.SOURCE]FILE.COM. |
| [POGO.SOURCE]FILE.COM    | ASSIGNS FILE NAMES.                                                                      |
| [POGO]LOGOUT.COM         | CALLS [POGO.SOURCE]ALOGOUT.COM.                                                          |
| [POGO.SOURCE]ALOGOUT.COM | CALLS 'CHECK' PROGRAM AND<br>ERASES SCREEN.                                              |
| [POGO.SOURCE]SAVES.COM   | WRITES A BACKUP TAPE OF ALL FILES<br>ON TAPE V241.                                       |
| [POGO.SOURCE]SAVEB.COM   | WRITES A BACKUP TAPE OF ALL FILES<br>ON TAPE V242.                                       |
| [POGO.SOURCE]GETS.COM    | READS A BACKUP TAPE OF ALL FILES<br>FROM TAPE V241.                                      |
| [POGO.SOURCE]GETB.COM    | READS A BACKUP TAPE OF ALL FILES<br>FROM TAPE V242.                                      |
| [POGO.SOURCE]TDIRS.COM   | MOUNTS TAPE V241 AND LISTS FILES.<br>DOES NOT DISMOUNT.                                  |

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| [POGO.SOURCE]TDIRB.COM    | MOUNTS TAPE V242 AND LISTS FILES.<br>DOES NOT DISMOUNT. |
| [POGO.SOURCE]DISMOUNT.COM | DISMOUNTS AND DEALLOCATES TAPE.                         |
| [POGO]MONTH.COM           | PERFORMS MONTHLY COSTING.                               |
| [POGO]PLOT.COM            | PREPARES PLOT FILES BY DIVISION.                        |

FILE NAME ASSIGNMENTS (FROM [POGO.SOURCE]FILE.COM)

```

$ASSIGN [POGO.LEDGER]PURCHASE.LIS F10
$ASSIGN [POGO.LEDGER]PURCHASE.LIS F20
$ASSIGN [POGO]PURCHASE.TMP F21
$ASSIGN [POGO.BACKUP]PURCHASE.BCK F12
$ASSIGN [POGO.BACKUP]PURCHASE.BCK F22
$ASSIGN [POGO]COSTING.DAT F14
$ASSIGN [POGO]COSTING.DAT F24
$ASSIGN [POGO.LEDGER]ACCOUNT.LIS F13
$ASSIGN [POGO.LEDGER]ACCOUNT.LIS F23
$ASSIGN [POGO]ROSTER.LIS F15
$ASSIGN [POGO]SNLA.DAT F16
$ASSIGN [POGO]SNLA.DAT F26
$ASSIGN [POGO]ACCOUNT.ERR F33
$ASSIGN [POGO.BACKUP]P83.ARC P83
$ASSIGN [POGO.BACKUP]A83.ARC A83
$ASSIGN [POGO.BACKUP]R83.ARC R83
$ASSIGN [POGO.BACKUP]P84.ARC P84
$ASSIGN [POGO.BACKUP]A84.ARC A84
$ASSIGN [POGO.BACKUP]R84.ARC R84
$ASSIGN [POGO.BACKUP]P85.ARC P85
$ASSIGN [POGO.BACKUP]A85.ARC A85
$ASSIGN [POGO.BACKUP]R85.ARC R85
$ASSIGN [POGO.BACKUP]P86.ARC P86
$ASSIGN [POGO.BACKUP]A86.ARC A86
$ASSIGN [POGO.BACKUP]R86.ARC R86
$ASSIGN [POGO.BACKUP]P87.ARC P87
$ASSIGN [POGO.BACKUP]A87.ARC A87
$ASSIGN [POGO.BACKUP]R87.ARC R87
$ASSIGN [POGO.BACKUP]P88.ARC P88
$ASSIGN [POGO.BACKUP]A88.ARC A88
$ASSIGN [POGO.BACKUP]R88.ARC R88
$ASSIGN [POGO.SOURCE]MESSAGE.LIS F1
$ASSIGN [POGO]X.INP F2
$ASSIGN [POGO]X.COS F3

```

FILE FORMATS FOR THE POGO SYSTEM

FILES: [POGO.LEDGER]PURCHASE.LIS  
 [POGO]PURCHASE.TMP  
 [POGO.BACKUP]PURCHASE.BCK  
 [POGO.BACKUP]Pfy.ARC (created by 'COSTING' for archive)

CREATED BY: PROGRAM 'INPUT' OR PROGRAM 'COSTING'  
 USED BY: 'INPUT', 'COSTING', 'REPORT', 'CHECK', 'INCHECK'

NOTE: All programs using PURCHASE.LIS for input check the length  
 (RECORD 3) against the number of records, and check to  
 verify the file is in order by PO number after file OPEN.

RECORD CONTENTS

- 1 \*\*\* DATE UPDATED = dd-mmm-yy FY 83
- 2 \*\*\* DATE COSTED = dd-mmm-yy -- COSTING PROGRAM RUN ON:dd-mmm-yy
- 3 \*\*\* # OF RECORDS = nnnn

| RECORD | FIELD/LINE | CHARS. | CONTENTS                   |
|--------|------------|--------|----------------------------|
| 4 to   | 1/1        | 1-10   | PO/PR NUMBER               |
| nnnn+3 | 2/1        | 12-17  | DATE                       |
|        | 3/1        | 19-22  | ORGANIZATION               |
|        | 4/1        | 24-31  | CASE NUMBER                |
|        | 5/1        | 33-42  | NAME OF REQUESTER          |
|        | 6/1        | 44-48  | VENDOR                     |
|        | 7/1        | 50-74  | DESCRIPTION                |
|        | 1/2        | 1-9    | AMOUNT                     |
|        | 2/2        | 10-18  | REMAINING BALANCE          |
|        | 3/2        | 19-27  | TO BE PAID THIS FY         |
|        | 4/2        | 28-36  | TO BE PAID SUBSEQUENT FY'S |
|        | 5/2        | 37-45  | OCTOBER-PAID               |
|        | 6/2        | 46-54  | NOVEMBER-PAID              |
|        | 7/2        | 55-63  | DECEMBER-PAID              |
|        | 8/2        | 64-72  | JANUARY-PAID               |
|        | 1/3        | 1-9    | FEBRUARY-PAID              |
|        | 2/3        | 10-18  | MARCH-PAID                 |
|        | 3/3        | 19-27  | APRIL-PAID                 |
|        | 4/3        | 28-36  | MAY-PAID                   |
|        | 5/3        | 37-45  | JUNE-PAID                  |
|        | 6/3        | 46-54  | JULY-PAID                  |
|        | 7/3        | 55-63  | AUGUST-PAID                |
|        | 8/3        | 64-72  | SEPTEMBER-PAID             |

FILES: ALBUQUERQUE DATA BASE  
[POGO]SNLA.DAT

CREATED BY: SNLA FINANCIAL PROGRAMMING DIVISION  
USED BY: 'STRIP' (SNLA.DAT ONLY)  
FIRST RECORD: ID (SKIPPED BY 'TAPECOPY' BEFORE 'STRIP')  
SNLA.DAT DOES NOT HAVE THIS ID RECORD

SUBSEQUENT RECORDS: 499 CHARACTER RECORDS WITH FOLLOWING FORMAT)

| CHAR/LNG | DESCRIPTION    | FIELDS USED BY 'STRIP'  |
|----------|----------------|-------------------------|
| 1/2      | POPR-WCLS      |                         |
| 3/7      | POPR-CAS.      | CASE NUMBER             |
| 10/7     | POPR-JO.       |                         |
| 17/7     | POPR-SO.       |                         |
| 24/1     | POPR-ORCODE    | PO/PR FLAG              |
| 25/5     | POPR-PONUM     | PO NUMBER               |
| 32/2     | POPR-CHGNM     | PO NUMBER-CHANGE NUMBER |
| 34/4     | POPR-CHORG.    | CHARGING ORGANIZATION   |
| 38/4     | POPR-CSMGR     |                         |
| 42/4     | POPR-SCSMGR    |                         |
| 46/4     | POPR-JOBMGR    |                         |
| 50/4     | POPR-SVCORDMGR |                         |
| 54/4     | POPR-JREQORG   |                         |
| 58/4     | POPR-SREQORG   |                         |
| 62/30    | POPR-CASNAM    |                         |
| 92/30    | POPR-SCASNAM   | DESCRIPTION             |
| 122/30   | POPR-SSCASNAM  |                         |
| 152/20   | POPR-JOBNAM    |                         |
| 172/20   | POPR-SVCNAM    |                         |
| 192/7    | POPR-CAP-CAS   |                         |
| 199/1    | POPR-CAP-FLAG  |                         |
| 200/18   | POPR-CASMGRNAM |                         |
| 218/18   | POPR-SCASMGRNM |                         |
| 236/3    | POPR-CONSL     | REQUESTER INITIALS      |
| 238/23   | POPR-DES       |                         |
| 262/6    | POPR-SUPPC     |                         |
| 268/24   | POPR-SUPPNAM   | VENDOR NAME             |
| 292/3    | POPR-BUYER     |                         |
| 295/7    | POPR-STATC.    |                         |
| 302/6    | POPR-DDATE     | DATE                    |
| 308/4    | POPR-EDATE     |                         |
| 312/1    | POPR-SUPP-TYPE |                         |
| 313/6    | POPR-POAMT     | PURCHASE ORDER AMOUNT   |
| 319/6    | POPR-OBLFD     |                         |
| 325/6    | POPR-PDCURMO   |                         |
| 331/6    | POPR-PDCURFY   | PAID CURRENT FY         |
| 337/6    | POPR-ACCR      | ACCRUAL                 |
| 343/6    | POPR-PDPRFY    | PAID PRIOR FISCAL YEAR  |
| 349/6    | POPR-REMBAL    | REMAINING BALANCE       |



|       |             |                        |
|-------|-------------|------------------------|
| 355/6 | POPR-DISCT  |                        |
| 361/6 | POPR-CURFYF | CURRENT FY FORECAST    |
| 367/6 | POPR-CURMOF |                        |
| 373/6 | POPR-2MOFCS |                        |
| 379/6 | POPR-3MOFCS |                        |
| 385/6 | POPR-4MOFCS |                        |
| 391/6 | POPR-5MOFCS |                        |
| 397/6 | POPR-6MOFCS |                        |
| 403/6 | POPR-NXTFYF | NEXT FY FORECAST       |
| 409/6 | POPR-SUBFYF | SUBSEQUENT FY FORECAST |

FILES: [POGO]name.BAR  
[POGO]name.MON

CREATED BY: PROGRAM 'REPORT' (can be edited with 'EDT')  
USED BY: 'TELL-A-GRAF' [ 'TAG' ]

BAR AND MONTHLY PLOT FILES IN 'TAG' FORMAT.

FILES: [POGO]name.TAB

CREATED BY: PROGRAM 'REPORT'  
USED BY: USER FOR 'TYPE', 'PRINT', OR OTHER LISTING.

PURCHASE.LIS DATA LISTING IN VARIABLE FORMAT DEPENDING ON THE  
TYPE OF REPORT GENERATED.

FILES: [POGO.SOURCE]MESSAGE.LIS

CREATED BY: 'EDT'  
USED BY: 'MESS.FOR'

SPECIAL MESSAGE PRINTED AT LOGIN.

FILES: [POGO.LEDGER]COSTING.DAT  
[POGO]X.COS

CREATED BY: PROGRAM 'STRIP'  
USED BY: 'COSTING'

RECORD CONTENTS

1 dd-mmm-yy  
{RECORD 1 IS THE DATE THAT THE STRIP PROGRAM WAS RUN}

| RECORD | FIELD/LINE | CHARS. | CONTENTS                            |
|--------|------------|--------|-------------------------------------|
| 2 to   | 1/1        | 1-10   | PO/PR NUMBER                        |
| end    | 2/1        | 12-19  | DATE                                |
|        | 3/1        | 21-24  | ORGANIZATION                        |
|        | 4/1        | 26-33  | CASE NUMBER                         |
|        | 5/1        | 35-37  | STAFF NAME                          |
|        | 6/1        | 39-43  | VENDOR                              |
|        | 7/1        | 45-64  | DESCRIPTION                         |
|        | 8/1        | 66-72  | AMOUNT                              |
|        | 1/2        | 7-13   | PAID CURRENT FY + ACCRUAL           |
|        | 2/2        | 14-20  | REMAINING BALANCE                   |
|        | 3/2        | 21-27  | FORECAST TO BE PAID THIS FY         |
|        | 4/2        | 28-34  | FORECAST TO BE PAID SUBSEQUENT FY'S |

FILES: [POGO.LEDGER]ACCOUNT.LIS (BUDGET NUMBERS IN THOUSANDS OF DOLLARS)

[POGO.BACKUP]afy.ARC (created by 'COSTING' for archive)

CREATED BY: PROGRAM 'ACCOUNT'  
USED BY: 'INPUT', 'STRIP', 'REPORT', 'COSTING'

| RECORD | FIELD/LINE | CHARS. | CONTENTS         |
|--------|------------|--------|------------------|
| ALL    | 1/1        | 4      | DIVISION         |
|        | 2/1        | 8      | CASE             |
|        | 1/2        | 6      | SEPTEMBER BUDGET |
|        | 2/2        | 6      | OCTOBER BUDGET   |
|        | 3/2        | 6      | NOVEMBER BUDGET  |
|        | 4/2        | 6      | DECEMBER BUDGET  |
|        | 5/2        | 6      | JANUARY BUDGET   |
|        | 6/2        | 6      | FEBRUARY BUDGET  |
|        | 7/2        | 6      | MARCH BUDGET     |
|        | 8/2        | 6      | APRIL BUDGET     |
|        | 9/2        | 6      | MAY BUDGET       |
|        | 10/2       | 6      | JUNE BUDGET      |
|        | 11/2       | 6      | JULY BUDGET      |

FILES: [POGO]name.DAT

CREATED BY: USER WITH 'EDT'

USED BY: 'INPUT'

NOTE: FREE FIELD WITH BLANKS BETWEEN FIELDS

| RECORD | FIELD/LINE | CONTENTS                                                  |
|--------|------------|-----------------------------------------------------------|
| 1 to n | 1/1        | PO/PR NUMBER (nn-nnnn nn) OR (nn-nnnnLnn)<br>OR (nn-nnnn) |
|        | 2/1        | DATE (mm/dd/yy)                                           |
|        | 3/1        | ORGANIZATION (nnnn)                                       |
|        | 4/1        | CASE NUMBER (nnnn.nnn)                                    |
|        | 5/1        | NAME OF REQUESTER (initial(s) plus surname)               |
|        | 6/1        | VENDOR (5 characters, no blanks)                          |
|        | 7/1        | DESCRIPTION (25 characters)                               |
|        | 8/1        | AMOUNT (dollars, no commas)                               |

FILES: [POGO]name.DAT  
[POGO]X.INP

CREATED BY: PROGRAM 'INPUT' -or- PROGRAM 'COSTING'

USED BY: 'INPUT', 'EDT'

NOTE: FREE FIELD WITH BLANKS BETWEEN FIELDS

| RECORD | FIELD/LINE | CONTENTS                                                  |
|--------|------------|-----------------------------------------------------------|
| 1 to n | 1/1        | *** (BEGINS WITH THREE ASTERISKS)                         |
|        | 2/1        | DESCRIPTION OF ERROR FOUND IN NEXT LINE                   |
|        | 1/2        | PO/PR NUMBER (nn-nnnn nn) OR (nn-nnnnLnn)<br>OR (nn-nnnn) |
|        | 2/2        | DATE (mm/dd/yy)                                           |
|        | 3/2        | ORGANIZATION (nnnn)                                       |
|        | 4/2        | CASE NUMBER (nnnn.nnn)                                    |
|        | 5/2        | NAME OF REQUESTER (initial(s) plus surname)               |
|        | 6/2        | VENDOR (5 characters, no blanks)                          |
|        | 7/2        | DESCRIPTION (25 characters)                               |
|        | 8/2        | AMOUNT (dollars, no commas)                               |

FILES: [POGO]ROSTER.LIS  
[POGO.BACKUP]Rfy.ARC (created by 'COSTING' for archive)

CREATED BY: USER WITH 'EDT'  
USED BY: 'INPUT', 'REPORT', 'COSTING'

NOTE: One name per line.

| RECORD | FIELD/LINE | CHARS.               | CONTENTS                                                                                                                                         |
|--------|------------|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| MOST   | 1/MOST     | VARIABLE,<br>10 MAX. | CONSULTANT NAME, FIRST AND<br>SECOND (if applicable) INITIAL(S)<br>AND SURNAME. NO BLANKS, PERIODS,<br>OR APOSTROPHES.<br>(EXAMPLE: PWZIMMERMAN) |
| SOME   | 1/SOME     | 4                    | [ OPTIONAL ] DIVISION NUMBERS CAN<br>BE USED TO BREAK UP THE LIST.<br>THEY ARE IGNORED BY THE PROGRAMS.                                          |

## I. Command File For Monthly Costing Procedure

For successful tapecopy, there must be an SNLA data tape in the computer center with the same name as specified in the MOUNT/FOREIGN command (i.e. P\_mon). If you do not know the month to replace 'mon' with, type the command file and look at the name contained in it with the command

```
$TYPE MONTH.COM
```

Program 'STRIP' writes a new MONTH.COM file each time it is run, and writes the first three letters of the month in which it was run into the tape name in the MOUNT command.

The following commands, necessary to copy an SNLA tape, run program 'STRIP', run program 'COSTING', and if an X.INP file is written from 'COSTING', run program 'INPUT' one time, are contained in the command file MONTH.COM and can be exercised with the following single command,

```
$@MONTH
```

### Listing of File MONTH.COM

```
$DELETE SNLA.DAT;*,COSTING.DAT;*,X.*;*
$OPEN/ERROR=OPEN_ERROR POGO [POGO.LEDGER]PURCHASE.LIS
$MOUNT/FOREIGN MF: P_mon TAPE
$TAPECOPY
TAPE
REWIND
SKIP
1
READ
SNLA.DAT
499
```

(These three blank lines represent [ return ])

```
STOP
$DISMOUNT TAPE
$STRIP
$DELETE SNLA.DAT;*
$CLOSE POGO
$COSTING
$INPUT
X.INP
$D
$EXIT
$OPEN_ERROR:
$WRITE SYS$OUTPUT "PURCHASE.LIS NOT AVAILIABLE; PROCEDURE ABORTED"
```

## II. Command File to Obtain Plots by Division

This file contains the command to run 'REPORT' and properly answer the prompts to obtain files for a bar plot of spending by division, and a monthly plot of spending for each division. The user may desire other plots and can use this file as a model to build his own file for them.

The command procedure is exercised by the single command,

```
$@PLOT
```

Listing of File PLOT.COM

```
$REPORT
```

```
1
```

```
D8440
```

```
2
```

```
D8440
```

This, and the following blank lines represent [ return ].

```
Y
```

```
GROUP
```

```
N
```

```
$D
```

## CASE AND SUBCASE MANAGING

The POGO system is set up to handle purchase accounting by charging organization. It can, theoretically, be altered to handle case and subcase D.C. purchases as well, but has not been tested. To do this, requires at least three steps, (1) obtain SNLA data sorted on case and/or subcase manager organization instead of charging organization, (2) put the proper accounts into the account file, and the correct personnel into the roster file and, (3) go into STRIP.FOR with the editor and after line

```
500 FORMAT
```

add the line

```
 ORG2=LINE (38:41)
```

then after line 550, change the IF statement from  
 IF(ORG.EQ.DIVGOOD(I))GO TO 560

to

```
 IF(ORG2.EQ.DIVGOOD(I))GO TO 560 (Case Manager)
```

or

```
 IF(ORG3.EQ.DIVGOOD(I))GO TO 560 (Subcase Manager).
```

There may be other changes required.

UNLIMITED DISTRIBUTION

R. S. Claassen, 8000; Attn: D. M. Olson, 8100  
A. N. Blackwell, 8200  
D. L. Hartley, 8300

R. C. Wayne, 8110  
W. E. Alzheimer, 8150  
D. E. Gregson, 8160  
J. H. Cordial, 8161-1  
P. G. Dominguez, 8161-1  
B. I. Matter, 8161-1  
P. L. Mattern, 8350  
C. W. Robinson, 8360  
L. Gutierrez, 8400  
R. A. Baroody, 8410  
A. G. Schuknecht, 8420  
D. E. Clay, 8423  
W. W. Hollis, 8423  
V. A. Common, 8440  
H. Hanser, 8440  
A. J. West, 8441  
P. A. Fleming, 8442  
W. R. Hoover, 8442  
P. D. Gildea, 8443  
B. J. Goodman, 8443  
C. T. Schafer, 8443 (25)  
M. G. Delgado, 8444  
A. R. Willis, 8444  
E. T. Cull, 8445  
J. B. Wright, 8450  
P. W. Zimmerman, 8453 (10)  
J. F. Barham, 8460  
Publications Division, 8265/Technical Library Processes Division, 3141  
Technical Library Processes Division, 3141 (3)  
M. A. Pound, 8214 for Central Technical Files (3)  
Publications Division, 8265, for TIC (27)



