7N-61-TM
253004
67P.

A Compendium of Utility Software Available on the Lewis TSS/370 Computer

(NASA-TM-101926) A COMPENDIUM OF UTILITY SOFTWARE AVAILABLE ON THE LEWIS TSS/370 COMPUTER (NASA) 67 p

N90-70513

Unclas

00/61 0253004

James J. Pelouch, Jr.
Lewis Research Center
Cleveland, Ohio

February 1982

NASA

CONTENTS

	Page
SUMMARY	1
BIGPRINT (Release 12-23-81)	3
CHART . (Release 08-13-81)	9
EVALUATE (Release 09-23-80)	13
HUGE . . (Release 09-29-81)	23
LABELS . (Release 03-19-81)	25
PCSR . . (Release 01-22-82)	27
PRINT90 (Release 12-02-81)	33
PULL . . (Release 01-28-82)	37
RUNOFF90 (Release 12-02-81)	39
SMARTX . (Release 01-28-82)	41
APPENDIX -A-	43
APPENDIX -B-	51
APPENDIX -C-	59
APPENDIX -D-	65
APPENDIX -E-	69
APPENDIX -F-	71
APPENDIX -G-	73
APPENDIX -H-	75

A COMPENDUM OF UTILITY SOFTWARE AVAILABLE ON THE LEWIS TSS/370 COMPUTER

James J. Pelouch, Jr.
Lewis Research Center

SUMMARY

This report consists of descriptions of software developed for use on the Lewis TSS/370 System. The report was prepared because of the high degree of general utility of these software items, and the increasing instances of Lewis-wide usage of them.

The software items were developed to allow the TSS/370 to assist in the accomplishment of management and administrative tasks such as status reporting, preparing presentations, and performing analyses. The software employs the TSS Graphics Package and its peripherals, and the IBM3800 printer. Special stroke and font tables have been developed and are used to drive these hardware devices to produce unique output such as large characters, special style characters, and characters that are rotated sideways.

Descriptions of the software items in this report appear in alphabetical order by name. A summary of these items by name and function is given below.

BIGPRINT.....Viewgraph Word-Chart making on the IBM3800 printer.

CHART.....Viewgraph, Slide, Poster, and Movie presentation artwork using TSS/370 Graphics.

EVALUATE.....Direct solution of mathematical expressions and display in graphical or tabular form.

HUGE.....Headline size (12 X standard size) printing on the IBM3800 printer.

LABELS.....Crosshair placement of labels, symbols, and other artwork on TSS/370-generated plots.

PCSR.....Generation of Procurement Cycle Status Reports in bar-chart format using the IBM3800 printer.

PRINT90.....Sideways (90-degree rotated) printing on the IBM3800 printer.

PULL.....Making a copy of a single byte of alphameric data from an INTEGER*4 array.

RUNOFF90.....RUNOFF with sideways printed output on the IBM3800 printer.

SMARTX.....User-Friendly input of program variables from a dataset or terminal.

BIGPRINT COMMAND DESCRIPTION
(Replaces CHART <DS>, DEVICE=P3800)

BIGPRINT is an easy-to-use command that produces high-quality word-containing viewgraphs and signs using text from an input VI dataset. BIGPRINT causes the text from the input dataset to be printed on the IBM3800 printer with characters that are either two or three times normal size. A sample of this output is shown in Appendix H. Because of the large character size, the output is useful as viewgraphs and signs. Instructions that control the placement and size of the text to be printed can be inserted in the input dataset. BIGPRINT operates on these instructions in a manner similar to RUNOFF.

BIGPRINT contains a preview feature that displays a scaled-down facsimile of the output on a local TEK4014 scope. This allows the user to rapidly review and change the arrangement of the output before actually printing it on the IBM3800. The user can also initially set the output character size and left margin without inserting instructions in the input dataset.

Operations with BIGPRINT are described in FUNCTIONAL DESCRIPTION, below. The instructions in the input dataset that BIGPRINT recognizes are described in BIGPRINT INSTRUCTIONS, below.

Operation	Operand
BIGPRINT	DSNAME = VI data set name, COPIES = number of copies.

DSNAME is the fully-qualified of the input VI dataset. If DSNAME is defaulted, BIGPRINT will terminate. If the input dataset is other than VI, or if the input dataset does not exist, BIGPRINT will terminate.

COPIES is the number of BIGPRINT output copies desired. Default is 1.

FUNCTIONAL DESCRIPTION

BIGPRINT functions interactively with the user. Immidiately after BIGPRINT is initiated, the user is asked whether he desires a preview of the output on a TEK4014 scope. If the user responds 'y' (for Yes), the preview feature is invoked. Following this, BIGPRINT examines the first few records of the input dataset for instructions. If it finds none, it asks the user whether he wishes to set the character size or the left margin on the output. If the user responds 'n' (for No) to these questions, BIGPRINT will default to Size=2 (2 X normal size) and Left Margin=1 (one space from the left border). If the user responds 'y' to these questions, he will be prompted for the desired settings.

After the proper settings have been established, BIGPRINT prepares the output for printing on the IBM3800. If the user has not specified a TEK4014 preview, and if no errors are detected in the input dataset (such as unintelligible instructions or lines that are too long for the output), the output is printed

on the IBM3800 and BIGPRINT terminates. If the TEK4014 preview feature is specified, or if errors are detected, the errors (and corresponding input dataset line numbers) are listed and the user is asked whether he wishes to enter REDIT to make modifications to his input dataset. If the user responds 'n' to this, BIGPRINT prints the output on the IBM3800 and terminates. If the user responds 'y', BIGPRINT automatically invokes REDIT on the input. Upon filing and quitting REDIT, the user is asked if he wishes to re-run the modified input (in order to effect the changes on the output). BIGPRINT continues in this manner until the user is satisfied with the output, whereupon he may direct the output to the IBM3800 printer.

BIGPRINT output appears in upper-case only regardless of whether the input data is upper-case or lower-case. BIGPRINT instructions are recognizable in either upper-case or lower-case. In addition to A-Z and 0-9, the following characters can be included in the input text for processing by BIGPRINT.

. ; : + - / * ( ) = , ' $? ! # ¬ < > and "

All other characters come out as blanks. The '#' character comes out looking like a bullet. The '¬' character comes out looking like a cents sign.

THE PREVIEW FEATURE

The preview feature of BIGPRINT eliminates the wait for copy from the IBM3800 printer as part of the trial-and-error process of developing an acceptable arrangement on the output. The preview feature, if selected by the user immidiately after initiating BIGPRINT causes a 2/3-scale display of the output to be made on a local TEK4014 scope. The following items are presented on the TEK4014 screen for each page of output.

1) A 2/3-scale picture of an 8-1/2 X 11-1/2 inch IBM3800 printer page, with corner markers for a 7-1/2 X 9-1/2 inch viewgraph chart frame added.

2) Text from the input dataset, scaled to the proper size and position, is located on the IBM3800 printer page.

3) All of the lines from the input dataset used to make the output page are listed next to the output page.

4) Any input dataset errors detected while constructing the output page are listed below the output page.

BIGPRINT INSTRUCTIONS

As was mentioned above, any VI dataset can be processed by BIGPRINT, regardless of whether it contains instructions or not. However, only minimal control over the placement and size of the text is possible without inserting BIGPRINT instructions in the input dataset.

The BIGPRINT instructions that are inserted in the input dataset are similar in apparence and function to RUNOFF commands. They are recognized as instructions (not text) because they begin with a period (in column one). They can be entered one to a line, or they can be strung out on a line and

separated with semicolons, as shown in the following examples.

```
0000300 .ul on
0000400 .sp 2; si 3; ce
0000500 .lm 10;rm 80
```

The BIGPRINT instructions are recognizable in either upper or lower case, but, unlike RUNOFF, only their abbreviated form is recognizable (eg. .UL but not .UNDERLINE). As with RUNOFF, the BIGPRINT instructions do not appear (are not printed) on the output.

Nine BIGPRINT instructions are recognized in the input dataset. They are

.SI	(Size)	.EC	(End Chart)	.LM	(Left Margin)
.RM	(Right Margin)	.RA	(Right Adjust)	.CE	(Center)
.UL	(Underline)	.LI	(Line)	.SP	(Space)

Instructions other than the above or text beginning with a period in column one will be interpreted by BIGPRINT as an errant instruction and skipped. Each of the nine BIGPRINT instructions is described below.

.SI <2 or 3>
Sets the CHARACTER SIZE for all subsequent lines of text, and remains in effect until another .SI instruction is encountered. .SI 2 is twice the size of standard IBM3800 print. .SI 3 is three times the size of standard IBM3800 print. The default setting for .SI is 2. Standard size print (which would be .SI 1) is not available when using BIGPRINT. Since the IBM3800 page can hold 60 lines of 132-character printing, the page capacity for .SI 2 print is 30 lines of 66 characters each, and for .SI 3 print, the page capacity is 20 lines of 44 characters each. Output lines are truncated when the length of the input lines exceed the values given above. Input lines greater in number than can fit on a page will cause a new output page to be started.

.EC
Causes an END CHART (begins a new chart) at the point the instruction is encountered in the input dataset. .EC is necessary when it is desired to start a new output page at a particular point. If .EC instructions are not included in the input dataset, new output page(s) are automatically started when previous pages become full. It is not necessary to insert an .EC instruction at the end of the input dataset.

.LM <n>
Sets the LEFT MARGIN for all subsequent lines of text, and remains in effect until another .LM instruction is encountered. <n> is the number of standard printer spaces to insert to the left of each output line of text. A standard printer space is 1/12-inch wide. <n> must be specified in the range 1 to 132. Values outside of this range, or values that are greater than or equal to the current .RM setting will cause BIGPRINT to disregard the .LM instruction. If <n> is not specified, an .LM 1 instruction will be assumed. An .LM 1 instruction will cause each line of output to appear 1/2-inch from the left edge of the IBM3800 output page. This is the left-most printing position possible with BIGPRINT. If the .LM setting is increased, the number of characters that can fit on an output line will be reduced.

.RM <n>
Sets the RIGHT MARGIN for all subsequent lines of text, and remains in effect until another .RM instruction is encountered. .RM effects the subsequent placement of Right Adjusted amd Centered text. <n> is the number of standard printer spaces to insert between the end of a line and the right border of the IBM3800 printer page. <n> must be specified in the range 2 to 133. Values outside this range or values that are less than or equal to the current .LM setting will cause BIGPRINT to disregard the .RM instruction. If <n> is not specified, an .RM 133 instruction will be assumed. A .RM 133 instruction will cause Right Adjusted output to appear 1/2-inch from the right edge of the IBM3800 output page. If the .RM setting is decreased, the number of characters that can fit on an output line will be reduced.

.RA
RIGHT ADJUSTs the next line of text such that it will end at the current Right Margin (RM) setting. Lines of text after that are not affected by the .RA instruction unless they too are preceeded by their own .RA instructions.

.CE
CENTERs the next line of text between the current Left Margin (LM) and Right Margin (RM) settings. Lines of text after that are not affected by the .CE instruction unless they too are preceeded by their own .CE instructions.

.UL <ON or OFF>
Causes UNDERLINEs to be drawn under all subsequent lines of text, and remains in effect until another .UL instruction is encountered. UL ON turns the underlining feature on. .UL OFF turns it off. .UL is the same as .UL ON. If .UL is ON, all non-blank characters are underlined. Each blank space is also underlined if non-blanks exist on either side of it. If a blank space is accompanied by one or more adjacent blank spaces none of them are underlined. The underline itself requires one extra standard IBM3800 print line, reducing the number of lines that can fit on a page.

.LI
Causes a LINE to be drawn across the output page at the point the .LI instruction is encountered in the input dataset. The line is similar in appearance to an underline except that it extends across all blanks, from the left edge to the right edge of the IBM3800 page, regardless of current .LM or .RM settings. .LI is useful for separating material on viewgraphs. .LI requires one standard IBM3800 print line.

.SP <n>
Causes SPACES (blank lines) to be inserted in the output page at the point the .SP instruction is encountered in the input dataset. <n> is the number of blank lines to insert. .SP is equivalent to .SP 1. The blank lines are standard IBM3800 printer lines which are 1/8-inch high. Thus, .SP 8 will cause a one inch space to be inserted in the output. Spacing can also be accomplished by inserting a line of text containing 'nothing' except for a few blank characters. If this is done when .SI is set to 2, two blank lines will be inserted. If .SI is set to 3, three blank lines will be inserted.

MAKING VIEWGRAPHS WITH BIGPRINT

The output from BIGPRINT that is printed on the IBM3800 printer is of exceptional quality for making viewgraph transparencies. The transparency quality is best if the copy from the IBM3800 is used directly on THERMOFAX machines (i.e. without first making a Xerox copy of it). If the THERMOFAX you use is in good working order, and if the transparency stock is fresh, you will find that the transparency quality approaches that available with photographic processes.

The normal line length output by BIGPRINT is, however, 11-inches wide. This output is too wide to fit into a standard 9-1/2 inch wide viewgraph frame. Therefore, when making viewgraphs, it is advisable to use the .LM and .RM instructions to set the margins such that the output is centered on the IBM3800 page and short enough to fit inside the 9-1/2 inch viewgraph frame. The following table serves as a guide for .LM and .RM settings for viewgraphs.

MARGIN SETTINGS	LINE CAPACITY (CHARACTERS)		LINE LENGTH (INCHES)	VIEWGRAPH UTILITY
	.SI 2	.SI 3		
.LM 1 .RM 133	66	44	11	NONE - Exceeds Width of Viewgraph Frame
.LM 10 .RM 124	57	38	9.5	Line Length Equals Width of Viewgraph Frame
.LM 13 .RM 121	54	36	9	1/4-inch Margin Each Side Viewgraph Frame
.LM 16 .RM 118	51	34	8.5	1/2-inch Margin Each Side Viewgraph Frame

In addition to the .LM and .RM settings for viewgraphs, it is desirable to insert a blank line or two at the beginning of each page; otherwise the output will begin right against the top of the viewgraph frame. Leave room at the bottom also. Twenty-six Size 2 lines or seventeen Size 3 lines will fit nicely in the frame, leaving 1/2-inch top and bottom margins.

GETTING ACCESS TO BIGPRINT

```
FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO:  mergepro viewgraf.procdefs
```

CHART COMMAND DESCRIPTION

CHART operates on a lined (VI) dataset in a manner similar to RUNOFF to produce presentation-quality computer-generated color artwork in the form of posters, viewgraph transparencies, 35MM projection slides, or 16MM movies. CHART is used to create text (words), symbols, geometric shapes, and the composition of these on the artwork, including their position, style, size, and color. The artwork is composed and edited using a local black-and-white graphics terminal such as the TEK4014. Pending satisfactory results on the local device, the output can then be automatically re-run on either black-and-white or color on one of four remote devices for automatic processing into the final product form.

While CHART uses the TSS Graphics Package, the User need have no knowledge of it in order to produce poster, viewgraph, projection slide, or movie artwork.

Details on the construction of the VI dataset on which CHART operates are given in Appendix A. Supporting information pertaining to the CHART generating instructions which appear in the VI dataset is given in Appendix B. Examples are given in Appendix C. Information on CHART display editing (ECFLOW) is given in Appendix D. Notes on the use of the .PA (pause) CHART instruction are given in Appendix E.

Getting access to CHART is explained at the end of this section of the report.

Operation	Operand
CHART	DSNAME = VI dataset name, DEVICE = (LOCAL | ZETA36 | ZETA12 | 35MM | 16MM), NCHART = Number of the chart in the VI dataset, AUTOCOPY = (Y | N), ECFLOW = (DIRECT | EDIT | RERUN), CORNERS = (Y | N), FRAMES = Number of 16MM frames per chart.

DSNAME

is the fully-qualified name of a VI dataset that contains a mixture of chart generating instructions and lines of text to be charted. Details on the construction of the VI dataset, including the kinds of instructions contained therein which CHART recognizes, is given in Appendix A.

If DSNAME is not given (defaulted), the CHART command will terminate.

DEVICE

directs the output of CHART to one of four hardcopy units or to the user's local graphics terminal itself.

If DEVICE=ZETA36, the output will be printed using the 36-inch ZETA plotter. The output is scaled up from the 7.5 X 9.5 inch viewgraph

size to 22.5 X 28.5 inch size. DEVICE=ZETA36 is useful for making paper posters and signs with CHART.

If DEVICE=ZETA12, the output will be printed using the 12-inch ZETA plotter, and the size will be the standard 7.5 X 9.5 inch viewgraph size. If DEVICE=ZETA12, the output is normally printed on white paper, but the output can also be printed on acetate (clear) film upon reuqest to the ZETA operator. DEVICE=ZETA12 is useful for making viewgraph transparencies and full-size viewgraph hardcopy with CHART.

If DEVICE=35MM, the output will be image processed on 35MM unsproketed film. DEVICE=35MM is useful for making 35MM projection slides with CHART.

If DEVICE=16MM, the output will be image processed on 16MM sprocketed film. DEVICE=16MM is useful for making 16MM still-image movie strips using CHART. See FRAMES to make repetitive frames of each chart image on 16MM film.

If DEVICE=LOCAL, the user will be prompted for a graphics terminal-ID on which to display the output. The TEK4014 is an acceptable local device for CHART output.

The default value of DEVICE is LOCAL.

NCHART

is an integer number which permits individual charts to be output from the VI dataset. For example, if NCHART=3, output of the first two charts in the VI dataset will be supressed, output of the third chart will take place, and output of the forth through the last charts will be suppressed.

If NCHART is defaulted, all of the charts in the VI dataset will be output.

AUTOCOPY

if set to Y will cause automatic copies to be made of each chart if the local graphics terminal is connected to a hardcopy device (such as the TEK4631 Hardcopy Printer). When AUTOCOPY=Y, the flow from CHART requires no user attention (responses to prompts), and therefore is useful when many charts are being produced from one DSNAME.

If AUTOCOPY=N, no automatic hardcopies will be made, and the user will be prompted for the display of each chart at the terminal. All prompting is done while CHART is in the display mode, such that the user only need use the RETURN button to shift from chart to chart.

NOTE - AUTOCOPY has no meaning if DEVICE is set to other than LOCAL.

The default value for AUTOCOPY is N.

ECFLOW

enables each chart to be edited (using local terminals which have

crosshairs such as the TEK4014) while the chart remains displayed on the screen.

If ECFLOW=EDIT, CHART will, upon encountering each end-chart (.EC) instruction in the input VI dataset, display the chart together with a list of single-key commands. You may then invoke these single-key commands, together with the crosshairs, to move, copy, erase, or add to the items on the chart. In addition, these single-key commands allow you to create geometric shapes such as circles, ellipses, rectangles, and arrows, and to move, copy or erase these as well. You can also change the style, size or color of subsequent items.

NOTE - ECFLOW=EDIT will only work if DEVICE=LOCAL, otherwise CHART will terminate.

If ECFLOW=RERUN, CHART will automacilly reproduce each chart by including the actions you previously took while using CHART with ECFLOW=EDIT on the same input VI dataset. If, for instance CHART XYZ,EXFLOW=EDIT were issued, CHART would make a dataset named CHART.XYZ which would contain all of your edit actions. If then CHART XYZ,ECFLOW=RERUN were issued, CHART would read CHART.XYZ and would automatically reproduce these actions. This is useful when the charts are first edited locally and then re-run on a remote device.

If ECFLOW=DIRECT, CHART will not attempt to EDIT or RERUN, instead it will directly display each chart upon encountering an .EC instruction in the input VI dataset.

The default value for ECFLOW is DIRECT.

Additional information on ECFLOW operations is given in Appendix D.

CORNERS

if set to N will turn off all information on the screen except the chart being displayed, ie. no chart corner markers, chart sequence numbers, or user prompts will appear with the chart.

If CORNERS=Y, all of this information will appear with the chart.

The default value for CORNERS is Y.

FRAMES

is an integer number in the range 1 to 511 which allows each chart image to be displayed repetitively to make strips of 16MM movie film. For example, if DEVICE=16MM and about 10 seconds of movie are desired for each chart (at 16 frames/second), set FRAMES=160 when issuing CHART and 160 movie frames of each chart will be produced.

NOTE - FRAMES should be specified only when DEVICE=16MM, otherwise FRAMES will default to 1.

The default value for FRAMES is 1.

Appendix A contains information on the construction of input VI datasets.

Appendix B contains a description of character style and size which are available when using CHART.

Appendix C contains Examples of the use of special character styles with the CHART command.

Appendix D contains information on ECFOLW operations.

Appendix E contains notes on the use of the .PA (pause) instruction.

GETTING ACCESS TO CHART

```
FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO:  mergepro viewgraf.procdefs
```

EVALUATE COMMAND DESCRIPTION

EVALUATE causes an algebraic function of the form Y=f(X) or f(X) to be plotted or tabled over a specified range of X without writing a program to calculate the function and without knowledge of the TSS Graphics Package. The function can be entered as part of the EVALUATE command, or EVALUATE can read it from a VI dataset record. In its simplest form, EVALUATE can be issued with only the FUNCTION operand, eg. EVALUATE (1.0/(X*COS(X)). In its more complex forms, EVALUATE can plot multiple functions on one graph, determine and plot derivatives or integrals of functions, locate roots, maxima and minima, and note them on the plot. Getting access to EVALUATE is explained at the end of this report.

Operation	Operand
EVALUATE	FUNCTION=(FORTRAN IV functional expression | VI DSNAME containing a FORTRAN IV functional expression), XMIN=Minimum independent variable value, XMAX=Maximum independent variable value, VILINE=Line number in a VI dataset containing the functional expression, OUTPUT=(PLOT | TABLE | Name of an output VI dataset), CONTROL=(BEGIN | CONTINUE | END), POINTS=Number of points to be tabled or plotted, SEARCH=(ZEROY | ZERODYDX | BOTH | DYDX | INTEGRAL | DASHES), YMIN=MINIMUM Y-AXIS VALUE, YMAX=MAXIMUM Y-AXIS VALUE.

FUNCTION

is either an algebraic expression or the fully-qualified name of a VI dataset which contains the algebraic expression. The value of VILINE is used to specify whether FUNCTION is an algebraic expression or a VI dataset name (see VILINE below). Examples of valid algebraic expressions are-

```
'Y=X**(SIN(X))'
'POINT(N)=(VALUE(M)+1)/(VALUE(M)**3.2)'
 3*Z**3-6*Z**2+4*Z-16
'IDEAL(N,M)**3+((16*IDEAL(N,M)-44)/100)'
```

The following rules apply to construction of the algebraic expression.

1) It must be recognizable by the ftn compiler.

2) All data on the left of and including the equal sign are optional. EVALUATE disregards this stuff prior to processing. Thus,- '100 Y=2*X-X**2' becomes '2*X-X**2'

3) All variables on the right side of the equal sign are interpreted as the same variable 'X'. Thus-

'Y=A+B+C(1,2)-C(4,4)' becomes 'A+A+A-A'. The variables may be subscripted with signed or unsigned integers. The variables can either be real or integer (i.e., names beginning with A-Z and no longer than 6 characters).

4) Expressions in VI datasets may begin in any column but must reside on only one line (record) of 120 characters or less. FORTRAN statement numbers and IF's before the Y=f(X)-type statement are acceptable. Thus- ' 250 IF(JJ.GE.1000)XMIN=1.0/EXP(1.0-FLIGHT(4,2))' is interpreted as '1.0/EXP(1.0-X)'. FORTRAN functions that EVALUATE understands are EXP, SIN, COS, TAN, ERF, ALOG, SQRT, ATAN, SINH, COSH, TANH, ERFC, ARSIN, COTAN, GAMMA, ALOG10, and ALGAMA. VI datasets which are read by EVALUATE are not altered.

5) Expressions which are entered as part of the EVALUATE command and which contain embedded commas or an equals sign must be enclosed in quotes. Thus- EVALUATE Y(I,J)**SIN(Y(I,J)) will be improperly interpreted as-
FUNCTION=Y(I, XMIN=J)**SIN(Y(I, XMAX=J)) Instead, use quotes- EVALUATE 'Y(I,J)**SIN(Y(I,J))'

Default of FUNCTION will cause EVALUATE to terminate.

XMIN

is an integer or real number which represents the minimum, or starting value of 'X'.

Default of XMIN will cause a prompt to be issued for it at the terminal.

XMAX

is an integer or real number which represents the maximum, or finishing value of 'X'.

Default of XMAX will cause a prompt to be issued for it at the terminal.

VILINE

is the line number in a VI dataset which contains the algebraic expression to be evaluated. If VILINE is specified (not defaulted), the value of FUNCTION is interpreted as the VI dataset name.

Default of VILINE will cause the value of FUNCTION to be interpreted as the algebraic expression itself.

OUTPUT

Controls whether the output of EVALUATE is to consist of a plot or a table.

If OUTPUT=PLOT, a plot of the algebraic expression will be produced.

If OUTPUT=TABLE, a table of the values of the algebraic expression

will be produced on the terminal.

If OUTPUT=Some output dataset name, a table will be produced on the output dataset.

Default of OUTPUT will cause OUTPUT=PLOT.

CONTROL

allows the user to issue EVALUATE repetitively to generate multiple-curve plots. CONTROL has no meaning if OUTPUT is set to specify table output.

If CONTROL=BEGIN, the resulting output is treated as the first of a family of curves on the graph.

If CONTROL=CONTINUE, the resulting output is treated as an inclusive member of a family of curves on the graph.

If CONTROL=END, the resulting output is treated as the last of a family of curves on the graph.

Physical output of the graph will not occur until the user specifies CONTROL=END.

NOTE - On multiple curve plots, the first issuance of EVALUATE (with CONTROL=BEGIN) scissors all subsequent curves to within this plotting space.

Default of CONTROL causes the output to be treated as the only curve on the graph. Default of CONTROL will cause the physical output of the graph containing the single curve.

POINTS

is an integer number in the range 2 to 1001 which indicates the number of plotted or tabled values of X equally spaced in the XMIN to XMAX interval.

NOTE - Increasing the number of points will increase the resolution of plots of very 'wiggly' functions. If POINTS is specified outside the range of 2 to 1001, the EVALUATE function will terminate.

Default of POINTS will cause 101 values of X to be used for plotted outputs, or 11 values of X to be used for tabular outputs.

SEARCH

causes dashed crosshairs to be placed on the plot which pertain to particular characteristics of the function to be plotted, or causes the function to be differentiated or integrated prior to plotting. SEARCH has no meaning if OUTPUT is set to specify table output.

If SEARCH=ZEROY, crosshairs will be placed on each point where f(X)=0 occurs. This is useful in locating roots of transcendental expressions. For instance, expressions in X such as f(X)=g(X) can be rewritten as f(X)-g(X)=0. Plotting the function Y=f(X)-g(X) with SEARCH=ZEROY will locate the root(s) (the value(s) of X which cause

Y=0).

If SEARCH=ZERODYDX, crosshairs will be placed on each point where df(X)/dX=0 occurs, e.g., all local maxima and minima.

If SEARCH=DYDX, synthetic differentiation of f(X) will be performed over the range XMIN to XMAX and the derivative of f(X) will be plotted (using a dashed line) instead of f(X).

If SEARCH=INTEGRAL, synthetic integration of f(X) will be performed in the interval XMIN to XMAX and the integral of f(X) will be plotted (using a dashed line) instead of f(X).

If SEARCH=DASHES, f(X) will be plotted using a dashed line instead of a solid line. None of the other SEARCH features will be in effect.

NOTE - To cause f(X) and the derivative (or integral) of f(X) to appear as two curves on the same graph, issue EVALUATE f(X) twice, once with CONTROL=BEGIN, and once with CONTROL=END and SEARCH=DYDX (or SEARCH=INTEGRAL).

Default of SEARCH will cause the plot to be displayed without any of the SEARCH features.

YMIN

is an integer or real number which establishes the minimum value for the Y-axis scale of the plot. YMIN has no meaning if OUTPUT is set to specify table output.

Default of YMIN (or YMAX) will cause EVALUATE to select values for the Y-axis scale which encompass f(X) over the range of XMIN to XMAX.

YMAX

is an integer or real number which establishes the maximum value for the Y-axis of the plot. YMAX has no meaning if OUTPUT is set to specify table output.

Default of YMAX (or YMIN) will cause EVALUATE to select values for the Y-axis scale which encompass f(X) over the range of XMIN to XMAX.

The following pages contain examples of the use of EVALUATE.

EXAMPLE 1) - Simple graphical output

```
u:   evaluate function=x**sin(x)
s/u: XMIN=0.0
s/u: XMAX=20.0
```

....causes X**SIN(X) to be plotted over the X-interval 0,20

EXAMPLE 2) - tabular output on TABLEOUT of a function listed on line number 0001200 in SOURCE.DELTAV

```
u:   evaluate source.deltav,-45,45,1200,tableout
```

....causes the desired functional values to be written into TABLEOUT over the X interval -45.0,+45.0

EXAMPLE 3) - multiple curve plotting of functions listed in GRAPH.INPUT

```
u:   default function=graph.input
u:   default xmin=1.3e-6
u:   default xmax=2.6e-6
u:   default control=continue
u:   evaluate  ,,,600,begin
u:   evaluate  ,,,700
u:   evaluate  ,,,800
u:   evaluate  ,,,900,end
```

....causes functions which appear on lines 600 through 900 in graph.input to be plotted as four curves on one graph over the X-interval 1.3e-6 to 2.6e-6.

Examples of tabular and plotted outputs are shown below.

EXAMPLE 4) - A table of values for ETA=1.094/(1+(6.99E6/(ISP**2))) is desired in the range of ISP from 2000 to 4000.

```
u:   evaluate 'eta=1.094/(1+(6.99e6/(isp**2)))',2000,4000,output=table
s:   EVALUATE FUNCTION=ETA=1.094/(1+(6.99E6/(ISP**2))), XMIN=2000, XMAX=4000,
s:            VILINE=, OUTPUT=TABLE, CONTROL=
s:            POINTS=, SEARCH= .....IS COMPLETE

s:    ISP              1.094/(1+(6.99E6/(ISP**2)))

s:       2000.           0.3982
s:       2200.           0.4476
s:       2400.           0.4942
s:       2600.           0.5379
s:       2800.           0.5784
s:       3000.           0.6158
s:       3200.           0.6502
s:       3400.           0.6818
s:       3600.           0.7107
```

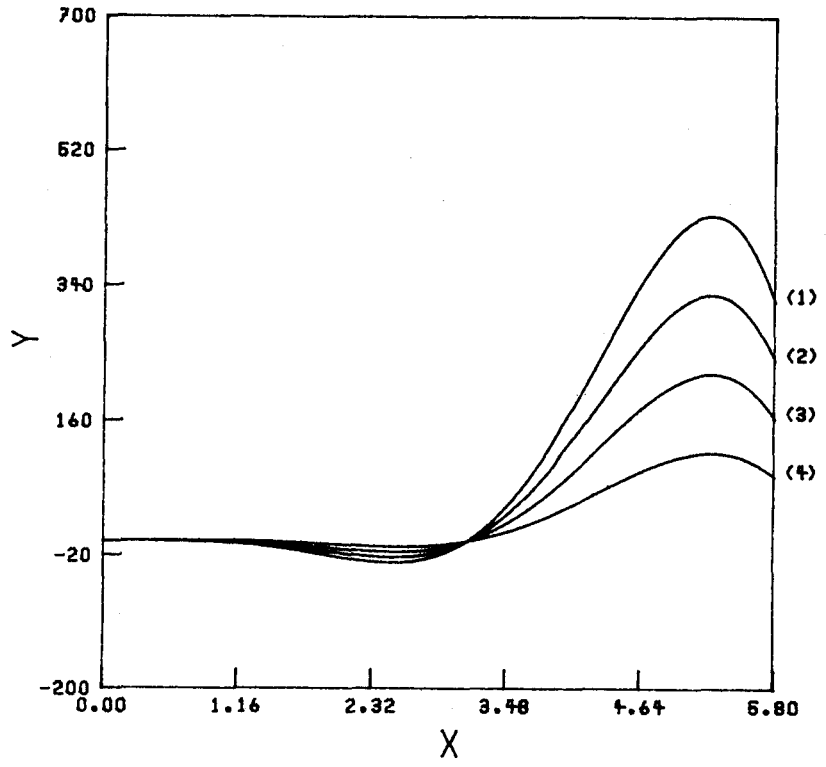
```
s:      3800.          0.7372
s:      4000.          0.7614
```

EXAMPLE 5) - A series of four equations listed in a VI dataset called EQUATION.N001 are to be plotted in the range of 'x' from 0 to 5.800

```
u:   redit equation.s001
s:   LOADING EQUATION.N001
s:   EDIT
u:   n;p100
s:   0000100 Y(I)=(-SIN(X(I)))*(4.0*(X(I)**3)-3.0*(X(I)**2)+2.0*X(I)-1.0)
s:   0000200 Y(I)=(-SIN(X(I)))*(3.0*(X(I)**3)-2.0*(X(I)**2)+1.0*X(I)-2.0)
s:   0000300 Y(I)=(-SIN(X(I)))*(2.0*(X(I)**3)-1.0*(X(I)**2)+0.0*X(I)-3.0)
s:   0000400 Y(I)=(-SIN(X(I)))*(1.0*(X(I)**3)-0.0*(X(I)**2)-1.0*X(I)-4.0)
s:   EOF
u:   q
u:   default function=equation.n001
u:   default xmin=0.0
u:   default xmax=5.8
u:   default control=continue
u:   evaluate ,,,100,,begin;evaluate ,,,200;-
u:   evaluate ,,,300;evaluate ,,,400,,end
s:   GRAPHICS DEVICE NOT DEFINED BY DDEF.
s:   ENTER UNIT NAME.  DEFAULT TO CANCEL.
u:   calcomp
s:   EVALUATE FUNCTION=EQUATION.N001, XMIN=0.0, XMAX=5.8,
s:            VILINE=100, OUTPUT=, CONTROL=BEGIN
s:            POINTS=, SEARCH= .....IS COMPLETE

s:   EVALUATE FUNCTION=EQUATION.N001, XMIN=0.0, XMAX=5.8,
s:            VILINE=200, OUTPUT=, CONTROL=CONTINUE
s:            POINTS=, SEARCH= .....IS COMPLETE

s:   EVALUATE FUNCTION=EQUATION.N001, XMIN=0.0, XMAX=5.8,
s:            VILINE=300, OUTPUT=, CONTROL=CONTINUE
s:            POINTS=, SEARCH= .....IS COMPLETE

s:   EVALUATE FUNCTION=EQUATION.N001, XMIN=0.0, XMAX=5.8,
s:            VILINE=400, OUTPUT=, CONTROL=END
s:            POINTS=, SEARCH= .....IS COMPLETE

u:   gterm
s:   BSN=0323
```

The above yields the CALCOMP plot shown on the next page.

```
(1)    (-SIN(X(I)))*(4.0*(X(I)**3)-3.0*(X(I)**2)+2.0*X(I)-1.0)
(2)    (-SIN(X(I)))*(3.0*(X(I)**3)-2.0*(X(I)**2)+1.0*X(I)-1.0)
(3)    (-SIN(X(I)))*(2.0*(X(I)**3)-1.0*(X(I)**2)+0.0*X(I)-1.0)
(4)    (-SIN(X(I)))*(1.0*(X(I)**3)-0.0*(X(I)**2)-1.0*X(I)-1.0)
```

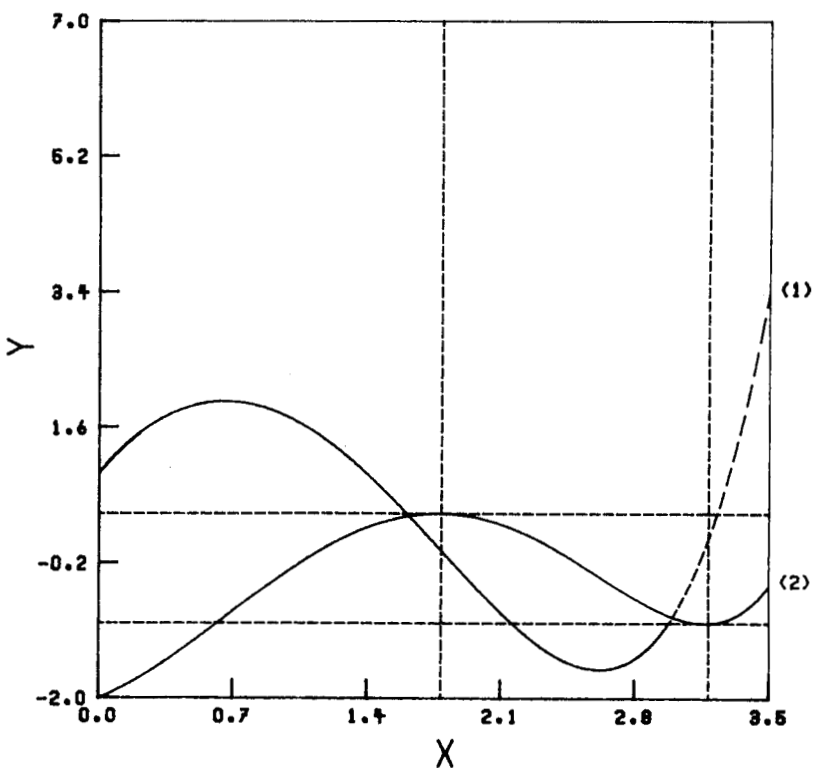


EXAMPLE 6) - A plot of the function 'EXP(X)-(X**3)+(X**2)-3.0' together with a plot of the derivative of this function is desired. In addition, crosshairs are to be placed on each local maxima and minima (points where the derivative equals zero. The function is listed on line 100 of a dataset called INPUT.

```
u:   evaluate input,0,3.5,100,control=begin,search=dydx
s:   GRAPHICS DEVICE NOT DEFINED BY DDEF.
s:   ENTER UNIT NAME.  DEFAULT TO CANCEL.
u:   calcomp
s:   EVALUATE FUNCTION=INPUT, XMIN=0, XMAX=3.5,
s:            VILINE=100, OUTPUT=, CONTROL=BEGIN,
s:            POINTS=, SEARCH=DYDX .....IS COMPLETE

u:   evaluate input,0,3.5,100,control=end,search=zerodydx
s:   EVALUATE FUNCTION=INPUT, XMIN=0, XMAX=3.5,
s:            VILINE=100, OUTPUT=, CONTROL=END,
s:            POINTS=, SEARCH=ZERODYDX .....IS COMPLETE

u:   gterm
s:   PLOT BSN 0378 ASSIGNED
```

The above yields the CALCOMP plot shown below.

```
(1)  d/dx EXP(X)-(X**3)+(X**2)-3.0
(2)       EXP(X)-(X**3)+(X**2)-3.0
```



GETTING ACCESS TO EVALUATE

```
FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO:  mergepro viewgraf.procdefs
```

NOTE: If you are going to assign default values to the parameters of evaluate, as in Examples 3) and 5), you must do so prior to issuing the MERGEPRO GRAPHPAK command, otherwise the default values you assign will be blanked after the first time EVALUATE is issued.

HUGE SUBPROGRAM DESCRIPTION

HUGE is a FORTRAN IV callable subprogram that writes an input string into an output dataset such that the characters in the output dataset will be printed in headline (12 times normal) size on the IBM3800 printer. An example of this output is shown in Appendix H.

HUGE is called in the form HUGE(LEAD,IWORDS,NDEV).

LEAD
is an INTEGER*4 word which contains the number of standard size (1/12-inch) spaces to put before the string on the output. LEAD must be in the interval 1 to 124, otherwise it will be changed to 124.

IWORDS
is an INTEGER*4 array of dimension 4 containing the string to be printed. IWORDS should be filled in 4A4 format. Normally, only 11 characters can be printed (11 X 12 = 132, which is the maximum length of IBM3800 output). However, some characters such as an 'I' require only half of the 12 printing spaces, hence more space is provided for in IWORDS.

NDEV
is the device number on which the output is to be written. NDEV must be in the interval 1 to 99, otherwise HUGE will terminate without writing any output.

PROGRAMMING NOTES

HUGE output must be printed using a special SYSUCS dataset called VIEWGRAF.P38001. The following example shows how your dataset must be printed.

```
PRINT <YOUR DSNAME>,,,EDIT,FORM=FULL,SYSUCS=VIEWGRAF.P38001
```

HUGE output can appear in upper/lower case. The dataset on which you write HUGE output can also contain other output (ie. output from FORTRAN WRITE statements). This output will appear normal size.

GETTING ACCESS TO HUGE

In order to access HUGE and VIEWGRAF.P38001, do

```
share viewgraf,spjjp,viewgraf
ddef xyz,vp,viewgraf.objlib,option=joblib
```

LABELS COMMAND DESCRIPTION

LABELS permits the user to employ the TEK4014 crosshair controls and keyboard to locate and enter alphameric labels, symbols, and vectors (lines) directly on a graph or on a clean screen while the TEK4014 remains continuously in the display mode. In addition, (1) each label, symbol, or vector so entered can be erased, and (2) character and symbol size can be changed while the TEK4014 remains continuously in the display mode. LABELS is entered before the TEK4014 is switched to the display mode, or it is entered on another terminal to which the TEK4014 is DDEF'ed.

Getting access to LABELS is explained at the end of this section of the report.

Operation	Operand
LABELS	DSNAME = VI dataset name

DSNAME
is the fully qualified name of a VP dataset which will be used by LABELS to store the work done while in the display mode. Work thus stored is given ID=100, such that the graphics command RETRIEVE 100,DSNAME will automatically reproduce the results at a later time or on another device.

If DSNAME is defaulted, the work done will not be saved.

FUNCTIONAL DESCRIPTION

DDEF XXX,VP,GRAPHICS,OPTION=JOBLIB must be done before LABELS is entered. The user must also SHARE GRAPHICS,SYSGRAPH,GRAPHICS one time prior to DDEFing GRAPHICS.

When LABELS is entered, it first makes a display of the users plot (or a clean screen if no prior plotting was done). The display will contain a set of instructions which appear to the right of the 10X10-inch plot region. These instructions will tell you how to enter labels, symbols, vectors with or without arrowheads, change character size, erase entries, re-display to observe cumulative results of your work, and stop the LABELS program. The instructions are self-explanatory, but a few important points are listed below.

(1) In order to raise the crosshairs, press 'X', then press the 'RETURN' key.

(2) Characters are transmitted in both upper and lower case. No lower-to-upper case conversion is made as is the case when the terminal is in the command mode. For upper case only, press the 'TTY LOCK' before preceeding.

(3) The 'RETURN' key must be pressed to transmit EACH character, one at a

time. The crosshair location is also transmitted each time, so if it is important in the transmission, position it before pressing the 'RETURN' key. The crosshairs will dissappear with each keystroke and re-appear with each 'RETURN' stroke.

(4) Certain keys have special meaning to LABELS. They are ; _ > # @ * ? and !. When these keys are transmitted, LABELS precesses the crosshair location and previous key transmissions in particular ways; therefore these keys cannot be used in alphameric labels.

GETTING ACCESS TO LABELS

FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO: mergepro viewgraf.procdefs

PCSR SOFTWARE DESCRIPTION

This section describes the entry of data and commands for using this data to prepare a PROCUREMENT CYCLE STATUS REPORT (PCSR) on the IBM/370 computer. The computer-prepared report is printed in bar-chart format on the IBM3800 printer. An example of the report is shown in Appendix H.

The command for data entry is called PREDIT DSNAME=<DS>. It is identical to REDIT, except that it automatically sets case to lower and does not permit data entry beyond column 65.

The command for report generation is called PRRUN DSNAME,ORDER,KOPIES. PRRUN is described later in this report.

DATA ENTRY

Data for the PCSR is entered in a VI (lined) dataset. A portion of this type of dataset is shown below.

```
0000049 ----------------------------------------------------------------
0000050  REPORTING DIVISION NAME=6100/Space Propulsion Division
0000051 ----------------------------------------------------------------
0000052                                        REPORT CUTOFF DATE=01-21-82
0000053 ----------------------------------------------------------------
0000054
0000055 a) . . . . . . . . . . . ENTER ASTERISK TO INCLUDE FOLLOWING ITEM=*
0000056 b) . . . . . . . . . . . . . . . . . . . . . . . . . . RTOP=179-80-51
0000057 c) . . . . . . . . . . . . . . . . . . . . . . . . . .TASK=Y0S1472
0000058 d) . . . . . . . . . . . . . . . . . . . . . . . . . . .PR=378062
0000059 e) . . . . . . . . . . . . . . . . . . . . . . . . . . VALUE=40K
0000060 f) . . . . . . . . . . . . . . . . . . . . . . . . . . FISCAL YEAR=82
0000061 g) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . TYPE=S
0000062 h) . . . . . . . . . . . . . . . . . . . . . . .ENGINEER=DeWitt
0000063 i) . . . TITLE=Particle Cloud Combustion
0000064 j) . . . . . . . . . . . . . . . . . . . . . . . . . BUYER=Burke
0000065 k) . . . . . . .  TECH PKG. TO DIV. CHIEF  . . (1) PLANNED=03-02-81
0000066 l)                                        (1) ACCOMPLISHED=03-02-81
0000067 m) . . . . . . . . . .  PR TO PROCUREMENT  . . (2) PLANNED=03-20-81
0000068 n)                                        (2) ACCOMPLISHED=03-20-81
0000069 o) . . . . . . . . . . . . . . . RFP ISSUED  . . (3) PLANNED=04-05-81
0000070 p)                                        (3) ACCOMPLISHED=04-05-81
0000071 q) . . . . . . . . . . PROPOSALS RECEIVED  . . (4) PLANNED=05-25-81
0000072 r)                                        (4) ACCOMPLISHED=05-25-81
0000073 s) . . . . . . . . . TECH. EVAL. COMPLETE  . . (5) PLANNED=08-03-81
0000074 t)                                        (5) ACCOMPLISHED=08-12-81
0000075 u) . . . . . . . . . NEGOGIATION COMPLETE  . . (6) PLANNED=11-09-81
0000076 v)                                        (6) ACCOMPLISHED=
0000077 w) . . . . . . . . . . . . . . . . . . AWARD  . . (7) PLANNED=12-26-81
0000078 x)                                        (7) ACCOMPLISHED=
0000079 y) . . . . . . . . . . . . . . . . . . . . . . . . . . . . PROGRESS=7
0000080 z)  COMMENTS=
0000081
```

The first 48 lines each contain only a few blank spaces and are therefore not shown above. Data is entered to the right of the equals (=) signs. The equals signs must remain in the columns shown. Data must not extend beyond column 65. Data is entered with the REDIT 'change' command. For example, the data on line 50, above, was entered by 'c*=*=6100/Space Propulsion Division*'.

The name of the Responsible Division is entered on line 50 (20 characters max.).

The REPORT CUTOFF DATE shown on line 52, above, is entered as two digits, a dash, two digits, a dash, and two digits. This date is printed on the PCSR and is used to position the vertical dashed line on the PCSR schedule. All procurement progress is reported with respect to this date, therefore it should be equal to or greater than the date when procurement data was last entered or updated.

The data for procurement items starts on line 55. Each procurement item requires 27 lines in the dataset. The last line for each procurement item (such as line 81, above) contains only a few blanks, and is used only to separate the items. The above example shows the 26 lines of data, labelled a) through z), for 1 procurement item, followed by the blank line. The data for subsequent procurement items follows this as additional blocks of 27 lines. These additional blocks have an identical format to lines 55 through 81, above, except, of course, their data content is different because the procurements which they represent are different. Data for up to 240 procurement items can be included in the dataset. The dataset can contain more blocks of 27 lines than are necessary for the procurement items to be reported. Each block of 27 lines, however, must be in the exact sequence shown, and the position of the equals signs must be in the exact columns shown.

A model of this type of dataset is available to you. It is called REPORT.INPUT.MODEL. It contains room for 80 procurement items. Make a copy of it and enter your data on this copy in order to generate PCSR's.

The individual data fields for each procurement item are labelled a) through z) as in the above example. They are discussed individually below.

<u>a)</u>
Enter an asterisk (*) or some other character to the right of the (=) sign if you want this procurement item included in the PCSR. If left blank, the item will not appear in the PCSR.

<u>b)</u>
Enter three digits, a dash, two digits, a dash, and two digits corresponding to the RTOP number.

<u>c)</u>
Enter the 7-character task number. Use upper case A-Z. Leave blank if unknown.

<u>d)</u>
Enter the Purchace Request number. Leave blank if unknown.

e)
Enter the amount on the Purchase Request. Leave blank if unknown.

f)
Enter two digits for the Fiscal Year from which the funds for the procurement are obtained.

g)
Enter ONLY the upper-case characters C, S, G, A, I, or O to specify the type of procurement. The characters stand for

C - Competitive	S - Sole Source	G - Grant
A - Interagency	I - Incremental	O - Overrun

h)
Enter the name of the Technical Project Manager using upper/lower case, last name first. Leave blank if unknown.

i)
Enter the Title for the procurement in upper/lower case.

j)
Enter the name of the Acquisition Division Specialist responsible for the procurement, using upper/lower case, last name first. Leave blank if unknown.

k) through x)
Enter the PLANNED and ACCOMPLISHED milestone dates as two digits, a dash, two digits, a dash, and two digits. Data for 7 milestones can be entered, in the sequence (1) PLANNED, (1) ACCOMPLISHED, (2) PLANNED, (2) ACCOMPLISHED, (3) PLANNED, etc. Enter all of the PLANNED milestone dates before the procurement cycle begins for the item. PLANNED milestone dates can (and in some cases should) extend into the future, beyond the REPORT CUTOFF DATE on line 52. If only milestone (1) PLANNED is entered (as on line 55, above) a triangle will appear on the PCSR instead of a bar for this item. If specific estimates for PLANNED dates are not available, use the table in Appendix -G- of this report. Notice in this table that the number of PLANNED milestone dates to enter and the time between PLANNED milestones is dependent on the type of procurement. Enter each ACCOMPLISHED milestone date when, and only when, the accomplishment occurs. Do not skip ACCOMPLISHED milestone date entries if corresponding PLANNED date entries have been made. IT IS IMPORTANT TO ACCURATELY ADD TO THE ACCOMPLISHED DATE ENTRIES ON A PERIODIC BASIS. ACCOMPLISHED milestones cannot be greater than (in the future with respect to) the REPORT CUTOFF DATE on line 52. When milestone (7) ACCOMPLISHED is finally entered, the bar for this item will disappear on the PCSR and the message 'Contract Awarded on MM-DD-YY' will appear instead. Do not change any of the PLANNED date entries on the basis of when the ACCOMPLISHED milestones occured. The Report Generating Program will automatically adjust the remainder of the PLANNED milestones on the basis of the ACCOMPLISHED milestones. Change the PLANNED date entries only if the estimated time to accomplish future milestones has changed.

y)
Enter a blank (or leave blank), or enter a single digit in the range 0 to 9. PROGRESS is used to show the progress between milestones by controlling the amount that the bar is to be shaded in the PCSR. If PROGRESS is left blank, the bar will be shaded to the REPORT CUTOFF DATE, showing that this item is on schedule. PROGRESS cannot be left blank (showing 'on time') if PLANNED dates which are earlier than the REPORT CUTOFF DATE do not have any corresponding ACCOMPLISHED date entries. If PROGRESS is entered as a digit in the range 0 to 9, it will control the degree of progress which is shown from the last ACCOMPLISHED milestone date to the next PLANNED milestone date. For instance, in the above example, PROGRESS=7, the last ACCOMPLISHED milestone date is 08-12-81 (on line 74), and the next PLANNED milestone date is 11-09-81 (on line 75). The bar would be shaded 70-percent of the distance between 08-12-81 and 11-09-81, which would be to about 10-13-81. The single digit number represents the fraction of 10 in the progress between milestones, such that PROGRESS=0 means 'no progress', PROGRESS=3 means '30-percent progress', etc. It is important that when no ACCOMPLISHED milestone dates have been entered, PROGRESS must be entered as '0' (no progress). The exception to this is when all PLANNED milestone dates are in the future (beyond the REPORT CUTOFF DATE), whereupon PROGRESS may be entered as a blank (on schedule).

z)
Enter any comments you may have for the procurement item, such as reasons for schedule slippage, relationships to other procurements, FAR's, etc. Leave blank if none.

REPORT GENERATING COMMAND (PRRUN)

The command which causes the Procurement Cycle Status Report to be generated from the input data described above is called PRRUN DSNAME,ORDER,COPIES. An example of how the command is actually entered is PRRUN FY82,ORDER=RTOP,KOPIES=3.

DSNAME is the name of the VI dataset containing the procurement input data, described in the previous section.

ORDER is the particular sequence you desire for the procurement items to be listed in the PCSR. If ORDER is not specified (eg. PRRUN FY82,KOPIES=2) the items in the PCSR will be in the same sequence as they are listed in the input dataset. ORDER can be specified as RTOP, TASK, PR, TYPE, ENGINEER, or BUYER. In these cases, the PCSR will be prepared in the sequence specified. For instance, if PRRUN FY82,ORDER=ENGINEER were entered, the items in the resulting PCSR would be arranged in accordance with engineer's names in alphabetical order. In addition, a second, third, and forth order hierarchy is also produced. In the above example, the procurement items for each engineer (if he has more than one) are sorted by RTOP, then by TASK, and then by PR. Additional ordering hierarchy is also produced for other ORDER fields. The following table summarizes this hierarchy.

ORDER	HIERARCHY			
	(1)	(2)	(3)	(4)
RTOP	FY	RTOP	TASK	PR
TASK	FY	TASK	PR	
PR	FY	PR		
TYPE	TYPE	RTOP	TASK	PR
ENGINEER	ENGINEER	RTOP	TASK	PR
BUYER	BUYER	RTOP	TASK	PR

Notice in the above table that if ORDER is entered as RTOP, TASK, or PR, the first hierarchy is FY, or Funding Year. This will separate the PCSR into several sections, one for each Funding Year.

KOPIES is the number of copies of the PCSR needed. This number of copies will automatically be printed on the IBM3800 printer when PRRUN completes. If KOPIES is not specified (eg. PRRUN FY82,ORDER=RTOP), one copy will be printed.

PRRUN FUNCTIONAL DESCRIPTION

PRRUN first reads the input dataset to obtain all of the data for the PCSR. PRRUN automatically inspects this data for formatting accuracy as it reads it. Errors, if any, will be printed on the terminal. The error messages will show the line number in the input dataset where the error exists and the type of error which was detected. If any errors are detected, PRRUN will terminate without printing the PCSR. The data errors must then be corrected and PRRUN must be entered again in order to obtain the PCSR. If no errors are detected, PRRUN writes the data as a series of records, one record per procurement item. The format of these records is defined in Appendix -F-. PRRUN then sorts these records and uses them in the sorted sequence to prepare the PCSR.

PRRUN detects 16 kinds of data formatting errors. The messages for these errors is given below.

LINE NNNNNNN DATA CUTOFF DATE INCORRECTLY ENTERED (Should Be of Form MM-DD-YY)

LINE NNNNNNN EQUALS SIGN MOVED OR ERASED

LINE NNNNNNN DATA EXTENDS BEYOND 65 CHARACTER LIMIT

LINE NNNNNNN RTOP NUMBER INCORRECTLY ENTERED (Should Be of Form NNN-NN-NN)

LINE NNNNNNN FISCAL YEAR INCORRECTLY ENTERED (Should Be of Form NN, eg. 82)

LINE NNNNNNN CONTRACT TYPE INCORRECTLY ENTERED (Sould Be Either C, S, G, A, I, or O)

LINE NNNNNNN DATE INCORRECTLY ENTERED (Should Be of Form MM-DD-YY)

LINE NNNNNNN MILESTONE (1) PLANNED DATE NOT ENTERED BUT SUBSEQUENT DATE(S) ENTERED

LINE NNNNNNN ACCOMPLISHED DATE ENTERED, BUT CORRESPONDING PLANNED DATE NOT ENTERED

LINE NNNNNNN PLANNED DATE IS EARLIER THAN PREVIOUS PLANNED DATE

LINE NNNNNNN ACCOMPLISHMENT DATE IS LATER THAN DATA CUTOFF DATE

LINE NNNNNNN ACCOMPLISHMENT DATE IS EARLIER THAN PREVIOUS ACCOMPLISHMENT DATE

LINE NNNNNNN PROGRESS IMPROPERLY ENTERED (Should be Blank or a Digit in the Range 0 to 9)

LINE NNNNNNN PROGRESS ENTERED AS 'N', BUT PROCUREMENT NOT YET STARTED

LINE NNNNNNN PROGRESS ENTERED AS 'N', BUT PLANNED DATE(S) PAST DUE WITH NO ACCOMPLISHMENTS

LINE NNNNNNN PROGRESS ENTERED AS 'ON TIME', BUT (N) PLANNED PAST DUE WITH NO ACCOMPLISHMENTS

GETTING ACCESS TO PCSR

Programs and commands for the PCSR capability are owned by SPR6100. In order to access them, do SHARE REPORT,SPR6100,REPORT, then do MERGEPRO REPORT.PROCS.

PRINT90 COMMAND

PRINT90 works like the PRINT command except that the output is printed sideways on the IBM3800 printer. Up to 66 lines of 90 characters each can be printed per page. The entire Full-Form upper/lower case character set can be printed. The sprocket holes can be chopped off what is now the top and bottom of the print to create 8 1/2 by 11 inch pages. Underlining is available if the dataset is first processed through RUNOFF to create output for sideways printing. Any dataset that can be printed with the PRINT command can be sideways printed with the PRINT90 command, however, the PRINT90 command will take a little longer to run, especially on large datasets.

Operation	Operand
PRINT90	DSNAME = data set name, STARTNO = starting position, ENDNO = ending position, PRTSP = (EDIT | 1 | 2 | 3), HEADER = H, LINES = lines per page, PAGE = P, ERASE = (Y | N), COPIES = number of copies, OFFSET = number of blank spaces printed to the left of each line

The operands work exactly as they do for the PRINT command except for the following changes in the permitted ranges.

OPERAND	PERMITTED RANGE	DEFAULT
STARTNO	1 to 130	1
ENDNO	1 to 210	90
PRTSP	1 to 3	1
LINES	22 to 66	54

OFFSET is used with PRINT90 even though it is not an operand associated with PRINT. It is used to position each printed line more to the right, thereby leaving more of a left margin, but also reducing the number of characters which can be printed on a line. The permitted range of OFFSET is 0 to 24. The default for OFFSET is 0.

FUNCTIONAL DESCRIPTION

PRINT90 prints a dataset in a 90-degree clockwise-rotated orientation on the IBM3800 printer. Output is printed up to 90 characters per line at 12 characters per inch (maximum text width = 7.5 inches), and up to 66 lines per page at 6 lines per inch (maximum text length = 11 inches).

Underlining is accomplished by using PRINT90 with PRTSP=EDIT. The dataset to be printed should contain separate lines with the underline characters in the proper columns and with the carriage control byte set to '+' for these lines.

When the dataset is printed with PRINT90, these '+' carriage controls are recognized and underlined characters are printed in these positions. RUNOFF90 will produce this carriage control format for underlining.

NOTE - The first page of the print of your dataset made with PRINT90 (the page with your USERID in big characters) will appear a little wierd. Also, you will not be able to easily read the dataset name, the time and date in the little starred box on the bottom of this page. This information is therefore re-written sideways on the next page, and then the print of your dataset begins on the subsequent page.

CHARACTER TABLE FOR PRINT90

16	; 5E	' 7D	{ 8B	q 98	v A5	² B2	– BF	M D4	Y E8	5 F5
40	¬ 5F	= 7E	≤ 8C	r 99	w A6	³ B3	A C1	N D5	Z E9	6 F6
¢ 4A	- 60	" 7F	' 8D	┬ 9A	x A7	⁴ B4	B C2	O D6	∩ EA	7 F7
. 4B	/ 61	a 81	+ 8E	) 9B	y A8	⁵ B5	C C3	P D7	∪ EB	8 F8
< 4C	⅛ 6A	b 82	┼ 8F	¤ 9C	z A9	⁶ B6	D C4	Q D8	/ EC	9 F9
(4D	, 6B	c 83	| 90	' 9D	├ AA	⁷ B7	E C5	R D9	\ ED	' FA
+ 4E	6C	d 84	j 91	± 9E	└ AB	⁸ B8	F C6	⊥ E0	⊆ EE	∪ FB
| 4F	_ 6D	e 85	k 92	■ 9F	┌ AC	⁹ B9	G C7	S E2	⊃ EF	' FC
& 50	> 6E	f 86	l 93	¯ A0	{ AD	┤ BA	H C8	T E3	0 F0	\ FD
! 5A	? 6F	g 87	m 94	° A1	≥ AE	⁺ BB	I C9	U E4	1 F1	⊇ FE
$ 5B	: 7A	h 88	n 95	s A2	• AF	┐ BC	J D1	V E5	2 F2	⊃ FF
* 5C	# 7B	i 89	o 96	t A3	⁰ B0	] BD	K D2	W E6	3 F3	
) 5D	@ 7C	⊥ 8A	p 97	u A4	¹ B1	≠ BE	L D3	X E7	4 F4	

MAKING THE NASA WORM

PRINT90 will print the NASA logo when it encounters the hexidecimal string

'EAEBECEDEEEFECED'

in your dataset. This string contains characters that are not normally printable on most terminals, so in order to input them, you must use the REDIT 'translate' feature.

PRINT90 produces the NASA logo as shown below.

NASA

GETTING ACCESS TO PRINT90

```
FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO:  mergepro viewgraf.procdefs
```

PULL SUBPROGRAM DESCRIPTION

PULL is a FORTRAN IV callable subprogram that copies a single byte of alphameric data from an array into another word.

PULL is called in the form PULL(IWORDS,NDIM,NBYTE,JWORD).

IWORDS
is an INTEGER*4 array of alphameric data.

NDIM
is the dimension of IWORDS.

NBYTE
is the byte number of IWORDS to copy the alphameric contents from.

JWORD
is a single INTEGER*4 word into which PULL installs the byte it copies from IWORDS. Upon returning from PULL, the first byte of JWORD will contain the alphameric contents of the NBYTE'th byte of IWORDS. The last three bytes of JWORD will contain blanks (Hexadecimal 40's).

GETTING ACCESS TO PULL

In order to access PULL, do

```
share viewgraf,spjjp,viewgraf
ddef xyz,vp,viewgraf.objlib,option=joblib
```

RUNOFF90 COMMAND

RUNOFF90 invokes RUNOFF on your dataset and then prints the result sideways on the IBM3800 printer. RUNOFF90 saves a certain amount of handwork which must otherwise be done on the RUNOFF output prior to printing it sideways.

Operation	Operand
RUNOFF90	DSNAME = data set name, COPIES = number of copies, OFFSET = number of blank spaces printed to the left of each line

COPIES is used to control the number of sideways-printed copies of the output. The default value for COPIES is 1.

OFFSET is used to position the RUNOFF90 output more to the right on the sideways printed pages, thereby leaving more of a left margin, but also reducing the number of characters which can be printed on a line. The permitted range of OFFSET is 0 to 24. The default value for OFFSET is 0.

OFFSET	MARGIN FROM EDGE OF PAGE	MAXIMUM REMAINING LINE LENGTH
0	1/2 inch	90 characters
6	1 inch	84 characters
12	1-1/2 inches	78 characters
18	2 inches	72 characters
24	2-1/2 inches	66 characters

USAGE NOTES

If the RUNOFF 'PRINT OFFSET' command is inserted in the input dataset, the effect will be additive with the OFFSET operand in the RUNOFF90 command. For example, if .PO 12 were to exist in the input dataset and then RUNOFF90 DSNAME,OFFSET=6 were issued, the output would be offset by a total of 18 characters.

Be sure to use the RUNOFF 'PAPER LENGTH' command in the input dataset. Set it to 66, i.e. insert a .PL 66 command in the input dataset. You'll get a RUNOFF error message doing this, but it will set the output on the 66-line page centers necessary for sideways printing. Control the top and bottom space settings for your printed pages by using the RUNOFF .TS and .BS commands on your input dataset. Control underlining by using the RUNOFF .UL command in your input dataset or by using the REDIT 'PB' command.

GETTING ACCESS TO RUNOFF90

```
FIRST DO: share viewgraf,spjjp,viewgraf
THEN DO:  mergepro viewgraf.procdefs
```

SMARTX SUBPROGRAM DESCRIPTION

SMARTX is a FORTRAN IV callable subprogram that reads a number or numbers from a terminal or a dataset into an X-array. The number(s) can be entered in F, I, D, or E. formats, separated by blanks, slashes, or any other character that is not a part of the number(s) themselves. Furthermore, the numbers need not occupy any particular columns. SMARTX, for instance, will convert input which looks like

```
4.0/3*2.0e-2/5/-99
```

into the X-values

```
4.000    0.020    0.020    0.020    5.000  -99.000
```

SMARTX is called in the form SMARTX(NDEV,X,NDX,NX)

NDEV
is the device number on which the input is to be read. If NDEV is not DDEF'ed, the input will be read from the terminal. NDEV must be in the range 1 to 99, otherwise SMARTX will return with an error message.

X
is the array of variables which will contain the input numbers upon return from SMARTX.

NDX
is the dimension of X.

NX
controls the number of X-values to be read. NX must be a signed integer, ie. don't CALL SMARTX(NDEV,X,NDX,6); instead set NX=6, then CALL SMARTX(NDEV,X,NDX,NX). If NX is sent as zero, SMARTX will continue to read X-values until a line that contains nothing is encountered. NX will then be returned as the number of X-values read. If the user attempts to input more than NDX values, SMARTX will return at the point NDX values are read. If NX is sent as a number greater than zero, SMARTX will continue to read until NX values of X are encountered, whereupon it will return with the value of NX left unchanged. If NX is sent as a number less than zero, or if NX is sent as a number greater than NDX, SMARTX will print an error message and return without finding any X-values.

RESTRICTIONS

Maximum input record length - 120 bytes.

Maximum input total data length - 12,000 bytes.

EXAMPLE

```
0000100 C          THIS DATASET IS CALLED SOURCE.RUNX
0000200 C
0000300       DIMENSION X(100)
0000400 C
0000500       PRINT 100
0000600   100 FORMAT(' ENTER NUMBER OF X-VALUES TO BE READ')
0000700 C
0000800 C          USE SMARTX TO ENTER NX
0000900 C
0001000       NX=1
0001100       CALL SMARTX(10,X,100,NX)
0001200       NX=X(1)
0001300       PRINT 200,NX
0001400   200 FORMAT(' ENTER',I4,' VALUES OF X')
0001500 C
0001600 C          USE SMARTX TO ENTER X
0001700 C
0001800       CALL SMARTX(10,X,100,NX)
0001900 C
0002000 C          DISPLAY RESULTS
0002100 C
0002200       PRINT 300
0002300   300 FORMAT(' X-VALUES READ AS FOLLOWS')
0002400       PRINT 400,(X(I),I=1,NX)
0002500   400 FORMAT(5F12.4)
0002600       STOP
0002700       END
```

When RUNX is run, the terminal looks like this.

```
runx
ENTER NUMBER OF X-VALUES TO BE READ
9
ENTER   9 VALUES OF X
-1, 3*44.2, 5.0e-2, 22, 23.3, 88, 100.2
X-VALUES READ AS FOLLOWS
    -1.0000     44.2000     44.2000     44.2000      0.0500
    22.0000     23.3000     88.0000    100.2000
TERMINATED: STOP
```

GETTING ACCESS TO SMARTX

SMARTX is in FTNPAK, belonging to SPJJP. To use, do

```
share ftnpak,spjjp,ftnpak
ddef xyz,vp,ftnpak,option=joblib
```

APPENDIX -A-

CONSTRUCTION OF INPUT DATASETS FOR USE BY 'CHART'

In order for CHART to produce a viewgraph, it requires input from a VI dataset in the form of lines of text (words to go on the chart) and instructions which are interspersed between these lines of text which tell CHART how to control the text (character style, size, placement on chart, etc.). The following example illustrates this principle -

The VI dataset has 9 lines as follows

```
0000100 .si 3
0000200 .st i
0000300 .sp 4
0000400 .ce
0000500 THIS IS MY TITLE
0000600 .sp 2
0000700 .ra
0000800 By Joe Dokes
0000900 .ec
```

....which CHART interprets as follows -

```
0000100  Set character size to 3
0000200  Set character style to italic
0000300  Insert 4 blank lines
0000400  Center the next line of text
0000500  'THIS IS MY TITLE' is thus displayed on the chart
0000600  Insert 2 more blank lines
0000700  Right adjust (move to right margin) the next line of text
0000800  'By Joe Dokes' is thus displayed on the chart
0000900  End the chart (begin the next chart)
```

As can be seen above, instructions are distinguished from text because of the period. This period at the beginning of the instruction signals CHART that this is an instruction (not text). Instructions control what happens to the subsequent line or lines of text. The line of text is controlled by the instructions above it regardless of the sequence or amount of these instructions. In the example, lines 100 through 400 control line 500, and lines 100 through 400 could have been entered in any sequence.

Instructions are necessary only when you want to control the text in special ways. For instance, if all of the instructions (except for the last instruction) were removed from the above example dataset, the resulting chart would be produced with the following characteristics -

1) Standard character size (Size 2)
2) Standard character style
3) No spaces between lines
4) both lines left adjusted

RULES PERTAINING TO CHART INSTRUCTIONS

The CHART instructions can appear in the VI datasets in either upper or lower case. The instruction must appear in its abbreviated form (eg. '.CE', not 'CENTER'). The period must appear in column one of the line. More than one instruction may be placed on one line provided they are separated by semicolons. If more than one instruction is placed on a line, only the first instruction need contain (begin with) a period. The following are examples of valid lines of CHART instruction in the input VI dataset.

```
0000100 .si 3
0000200 .st 1;lm 10;ds
0000300 .rm 90; .ce
```

Eighteen CHART instructions are available. They are .SI, .ST, .LM, .RM, .CE, .RA, .SP, .SS, .HS, .DS, .TS, .VP, .EC, .PA, .LS, .CO, .TW and .GR. Some of these instructions effect only the next line of text. Others effect all subsequent lines of text until they are re-issued with a different setting. If an invalid instruction is encountered, CHART will issue an error message and skip (disregard) the errant instruction. If text is encountered which will not fit on the viewgraph, CHART will issue an error message which will indicate how it will dispose of the text (either by truncating or by skipping).

MAKING THE VI DATASETS

The VI datasets can be produced in upper/lower case using either of the TSS Editors EDIT or REDIT. It is assumed that the User either knows or can recieve instruction on how to use these editors.

Table A1 describes the CHART instructions and their effect.

TABLE A1 - CHART INSTRUCTION SYNOPSIS (1 of 5)

INSTRUCTION/OPERAND		EFFECT
.SI	1 to 9	Sets CHARACTER SIZE. Sizes range from 1 (small) to 9 (large). Examples of the sizes are given in Table B2. Physical size dimensions are given in Table B3. The .SI instruction will remain in effect until '.SI' is re-issued. Default value of .SI is 2. A size of 1 is not recommended for vu-charts except for page numbers and the like, because it is about equal to the size of typewriter type.
.ST	S, I, L, B, G, O, P, M, Y, Z, or <DSNAME>	Sets CHARACTER STYLE. S is the standard CHART style and is available in full upper-lower case. I is the italic version of S. L is logo style (similar to the new NASA emblem) and text is converted to upper case for display on the chart. B is box style, which consists of characters which are useful to create box, bracket, arrow, and line construction on charts. G is the TSS Graphics Package style (the one that is commonly employed in plotting). O is open style, wherein the characters are formed by outlining them. O is available in full upper-lower case. P is the italic version of S. M is mathematical style, containing mathematical and greek symbols. Y is the same as S except the characters are repositioned slightly to the right and up and restruck in four intervals to increase the width of the lines which draw the character. Z is a similar restrike of the O style. The Y and Z styles are effective only in the larger sizes (6 through 9), and because of repeated restrikes, take 5 times as much time to be completed on the chart. The various character styles are shown in Table B1. The .ST instruction will remain in effect until '.ST' is reissued. If .ST <DSNAME>, i.e. if .ST is supplied with the dsname of a stroke table you wish to use in place of the S, I, L, B, O, P, M, Y, or Z character styles, then that stroke table will be implemented and the resulting character style will be in accordance with that stroke table until .ST is again issued. Default value of .ST is S.
.LM	0 to 95	Sets LEFT MARGIN. The number (0 to 95) corresponds to the distance (in tenths of an inch) from the left frame of the vu-chart to the left margin of each subsequent line of text. The default value of .LM is 5 (or 0.5 of an inch).

TABLE A1 - CHART INSTRUCTION SYNOPSIS (2 of 5)

INSTRUCTION/OPERAND		EFFECT
.RM	0 to 95	Sets RIGHT MARGIN. The number (0 to 95) corresponds to the distance (in tenths of an inch) from the left frame of the vu-chart to the right margin of each subsequent line of text. Text which is too long for the LM. and .RM settings will be truncated. The default value of .RM is 90 (or 9.0 inches). Thus with the default .LM and RM. settings, the printing field on the chart is 9.0 - 0.5, or 8.5 inches wide. If .LM 0 and .RM 95, the printing field is increased to the maximum of 9.5 inches, but no space is left between the printing and the edges of the viewgraph frame.
.CE	<none>	CENTERs next line of text. The next line of text will be centered between the current .LM and .RM settings. Subsequent lines will be left justified to the current setting of .LM unless .CE or .RA is invoked on them also.
.RA	<none>	RIGHT ADJUSTs next line of text. The next line of text will be right justified such that its end will correspond to the current .RM setting. Subsequent lines will be left justified to the setting of .LM unless .RA or .CE is invoked on them also.
.SP	-9 to 9, T or H	Inserts SPACEs prior to next line of text. If .SP is set to a positive number, that number of blank lines will be inserted before the next line of text. .SP has the same meaning as .SP 1. If .SP is set to a negative number, the next line of text will be 'backed up' by the specified number of lines. If .SP -1, the previous line will be overwritten by the next line (useful for underlining). If .SP -2, the next line will be written one space above the previous line, and so forth. If .SP H, a half space will be inserted before writing the next line of text. If .SP T, the next line will be written at the top of the chart. Each subsequent line will be written beneath the the line which is affected by the .SP instruction. .SP overrides the carrage settings .SS, .HS, .DS, and .TS described below for the next line of text only.
.SS	<none>	Sets 'carrage' on automatic SINGLE SPACING. .SS prevails over all subsequent lines of text unless reset with the .HS, .DS, .TS instructions or overridden by the .SP instruction. .SS is the default carrage setting.

TABLE A1 - CHART INSTRUCTION SYNOPSIS (3 of 5)

INSTRUCTION/OPERAND		EFFECT
.HS	<none>	Sets 'carrage' on automatic ONE-AND-ONE-HALF SPACING. .HS prevails over all subsequent lines of text unless reset or overridden in a manner similar to the .SS instruction.
.DS	<none>	Sets 'carrage' on automatic DOUBLE SPACING. .DS prevails over all subsequent lines of text unless reset or overridden in a manner similar to the .SS instruction.
.TS	<none>	Sets 'carrage' on automatic TRIPLE SPACING. .TS prevails over all subsequent lines of text unless reset or overridden in a manner similar to the .SS instruction.
.VP	0 to 75	Establishes the VERTICAL POSITION of the next line of text. The number (0 to 75) corresponds to the distance (in tenths of an inch) from the bottom frame of the vu-chart to the base of the next line of text. Subsequent lines of text will be placed beneath the line of text which is affected by the .VP instruction, unless they to are affected by their own .VP instruction or by an .SP instruction. The .VP instruction can be used to place lines of text above, below, or on top of previous lines of text. If the .VP instruction is used in conjunction with the .LM (Left Margin) instruction, lines of text can be positioned anywhere and in any sequence on the 7-1/2 inch high by 9-1/2 inch wide vu-chart. .VP issued without an operand is equivalent to issueing .VP 0, whereupon the next line of text will be placed on the bottom of the vu-chart.
.EC	<none>	Signals END-of-CHART and causes display of the chart, followed by processing of the next chart in the VI dataset. .EC is inserted at the point in the VI dataset where one chart ends and another begins. Insertion of .EC at the end of the VI dataset is optional, because CHART will interpret the VI dataset end as the end of the last chart, regradless of whether an .EC is there or not. Therefore, VI datasets which contain only one chart need not contain an .EC instruction.
.PA	<none>	Causes a PAUSE in the chart construction. .PA allows the user to display his own graphical information (X-Y plots, etc.) on the emerging vu-chart. The User must invoke his own plotting procedues at this pause point in order to do this. See Appendix E for more on .PA.

TABLE A1 - CHART INSTRUCTION SYNOPSIS (4 of 5)

INSTRUCTION/OPERAND		EFFECT
.LS	Y	Controls the LINE SPACING for the current SIZE setting. Y is a real floating point number between 1.0 and 10.0 which is the ratio of the distance between each single space line and the size (height) of the characters. For example, in Table B3 for character size 1, the default value of Y is 0.167/0.106, or 1.575. If .LS 1.0 were issued, the line spacing would equal the character size and no room would be left between the bottom of the characters on one line and the top of the characters on the subsequent line. .LS controls all subsequent lines of text until an .SI instruction is encountered, whereupon the normal or default value of Y in accordance with Table B3 is again assumed. .LS is useful if room for a few extra lines is needed on a chart because it can be used to slightly compress the line-to-line distance.
.CO	0 to 6	Establishes the COLOR for the subsequent text. CO remains in effect until another .CO instruction is encountered. The integer operand in the range 0 to 6 which accompanies the .CO instruction is the TSS Graphics Package color code. This code is repeated below in the context of the CHART program.

DEVICE	LOCAL	ZETA36 or ZETA12	35MM or 16MM
BACKGROUND COLOR	GREEN	WHITE (if paper) CLEAR (if acetate)	BLACK
COLOR CODE (with .CO)	COLOR OF SUBSEQUENT TEXT		
.CO 0	WHITE	BLACK	CLEAR
.CO 1	WHITE	RED	RED
.CO 2	WHITE	BLUE	BLUE
.CO 3	WHITE	GREEN	GREEN
.CO 4	WHITE	BLACK	YELLOW
.CO 5	WHITE	BLACK	MAGENTA
.CO 6	WHITE	BLACK	CYAN

Note that the effect of .CO depends on what DEVICE you have specified in the CHART command.

TABLE A1 - CHART INSTRUCTION SYNOPSIS (5 of 5)

INSTRUCTION/OPERAND		EFFECT
.TW	A	TWISTS the next line of text by "A" degrees. "A" is an integer number in the range 0 to 360. The next line of text will be rotated counter-clockwise by A degrees. The center of rotation will be where the line of text intersects the .LM (left margin) setting.
.GR	<none>	Superimposes a GRID on the chart. GR remains in effect only on the chart in which the .GR instruction is encountered. The grid spacing is 0.10 inches in the horizontal and vertical directions. Inch markers are drawn along the bottom and left edge of the chart. GR is useful as an aid for positioning items on the chart. In order to remove the grid, the .GR instruction must be deleted from the input VI dataset and the CHART program run again.

APPENDIX -B-

CHART CHARACTER STYLE AND SIZE INFORMATION

Table B1 shows the various character styles which are available when the .ST instruction is invoked.

Table B2 shows examples of the nine sizes which are available when the .SI instruction is invoked.

Table B3 lists the character and line spacing dimensions which pertain to the various character sizes.

Tables B1 and B2 were produced by CHART from datasets which are called VIEWGRAF.TTABLEB1 and VIEWGRAF.TABLEB2. Take a look at there datasets (using REDIT or PRINT) to see how CHART operates on them to produce the two tables.

TABLE B1 Page 1 of 9

STANDARD Style (.st s)

Key on left - Chart character on right

. .	**	__	" "	hh	pp	xx	FF	NN	VV	33
<<	))	>>	aa	ii	qq	yy	GG	OO	WW	44
((	;;	??	bb	jj	rr	zz	HH	PP	XX	55
++	¬¬	::	cc	kk	ss	AA	II	QQ	YY	66
||	--	#●	dd	ll	tt	BB	JJ	RR	ZZ	77
&&	//	@@	ee	mm	uu	CC	KK	SS	OO	88
!!	,,	''	ff	nn	vv	DD	LL	TT	11	99
$$	%%	==	gg	oo	ww	EE	MM	UU	22	

TABLE B1 Page 2 of 9

ITALIC Style (.st i)

Key on left - Chart character on right

. .	**	__	" "	h*h*	p*p*	x*x*	F*F*	N*N*	V*V*	3*3*
<<	))	>>	a*a*	i*i*	q*q*	y*y*	G*G*	O*O*	W*W*	4*4*
((	;;	??	b*b*	j*j*	r*r*	z*z*	H*H*	P*P*	X*X*	5*5*
++	¬¬	::	c*c*	k*k*	s*s*	A*A*	I*I*	Q*Q*	Y*Y*	6*6*
||	--	#●	d*d*	l*l*	t*t*	B*B*	J*J*	R*R*	Z*Z*	7*7*
&&	//	@@	e*e*	m*m*	u*u*	C*C*	K*K*	S*S*	O*O*	8*8*
!!	,,	''	f*f*	n*n*	v*v*	D*D*	L*L*	T*T*	1*1*	9*9*
$$	%%	==	g*g*	o*o*	w*w*	E*E*	M*M*	U*U*	2*2*	

TABLE B1 Page 3 of 9

LOGO Style (.st l)

Key on left - Chart character on right

. .	* *	_	" "	h H	p P	x X	F F	N N	V V	3 3
< (	))	>)	a A	i I	q Q	y Y	G G	O O	W W	4 4
((	; ;	? ?	b B	j J	r R	z Z	H H	P P	X X	5 5
+ +	¬	: :	c C	k K	s S	A A	I I	Q Q	Y Y	6 6
| |	- -	# ●	d D	l L	t T	B B	J J	R R	Z Z	7 7
&	/ /	@	e E	m M	u U	C C	K K	S S	0 0	8 8
! !	, ,	' '	f F	n N	v V	D D	L L	T T	1 1	9 9
$	%	= =	g G	o O	w W	E E	M M	U U	2 Z	

TABLE B1 Page 4 of 9

BOX Style (.st b)

Key on left - Chart character on right

.	* ↓	_ □	"	h ┘	p △	x 8	F └	N	V ├	3 ┐
< ◄	)	> ►	a ┌	i ┬	q 1	y 9	G ┴	O ┌	W ┤	4 ┤
(	;	?	b ┬	j ├	r 2	z ▲	H ┘	P ┐	X ═	5 ┤
+ ┼	¬	:	c ┐	k ┤	s 3	A ┌	I ┬	Q	Y ├	6 └
| |	- —	# ┼	d ├	l ┴	t 4	B ┬	J ├	R	Z ┤	7 ┴
&	/	@	e ┤	m ┼	u 5	C ┐	K ┤	S	0	8 ┘
! |	, ═	'	f └	n	v 6	D ├	L ┴	T |	1 ┌	9
$ ↑	%	= —	g ┴	o	w 7	E ┤	M ┼	U	2 ┬	

Page 5 of 9

TABLE B1
OPEN Style (.st o)

Key on left - Chart character on right

. .	* *	_ _	" "	h h	p p	x x	F F	N N	V V	3 3
< <	))	> >	a a	i i	q q	y y	G G	O O	W W	4 4
((	; ;	? ?	b b	j j	r r	z z	H H	P P	X X	5 5
+ +	¬ ¬	: :	c c	k k	s s	A A	I I	Q Q	Y Y	6 6
| |	- -	# #	d d	l l	t t	B B	J J	R R	Z Z	7 7
& &	/ /	@ @	e e	m m	u u	C C	K K	S S	0 0	8 8
! !	, ,	' '	f f	n n	v v	D D	L L	T T	1 1	9 9
$ $	% %	= =	g g	o o	w w	E E	M M	U U	2 2	

Page 6 of 9

TABLE B1
OPEN-ITALIC Style (.st p)

Key on left - Chart character on right

. .	* *	_ _	" "	h *h*	p *p*	x *x*	F *F*	N *N*	V *V*	3 *3*
< <	))	> >	a *a*	i *i*	q *q*	y *y*	G *G*	O *O*	W *W*	4 *4*
((	; ;	? ?	b *b*	j *j*	r *r*	z *z*	H *H*	P *P*	X *X*	5 *5*
+ +	¬ ¬	: :	c *c*	k *k*	s *s*	A *A*	I *I*	Q *Q*	Y *Y*	6 *6*
| |	- -	# #	d *d*	l *l*	t *t*	B *B*	J *J*	R *R*	Z *Z*	7 *7*
& &	/ /	@ @	e *e*	m *m*	u *u*	C *C*	K *K*	S *S*	0 *0*	8 *8*
! !	, ,	' '	f *f*	n *n*	v *v*	D *D*	L *L*	T *T*	1 *1*	9 *9*
$ $	% %	= =	g *g*	o *o*	w *w*	E *E*	M *M*	U *U*	2 *2*	

Page 7 of 9

TABLE B1
MATH Style (.st m)

Key on left - Chart character on right

..	*·	_—	”¨	hh	p{	xx	Fγ	Nν	Vψ	33
<<	))	>>	aa	ii	q}	yy	Gγ	Oπ	Wω	44
((	;;	??	bb	j]	rr	zz	Hη	PΠ	Xξ	55
++	¬∫	::	cc	kk	ss	Aα	Iδ	Qφ	Yυ	66
|∃	--	#≤	dd	ll	tt	Bβ	Jω	Rρ	Zζ	77
&≠	//	@∇	ee	mm	uu	Cθ	Kλ	Sσ	00	88
!≅	,	’·	ff	nn	vv	DΔ	LΛ	Tτ	11	99
$≥	%∑	==	g[	oo	ww	Eε	Mμ	Uχ	22	

Page 8 of 9

TABLE B1
STANDARD-BUZZ Style (.st y)

Key on left - Chart character on right

..	**	__	””	hh	pp	xx	FF	NN	VV	33
<<	))	>>	aa	ii	qq	yy	GG	OO	WW	44
((	;;	??	bb	jj	rr	zz	HH	PP	XX	55
++	¬¬	::	cc	kk	ss	AA	II	QQ	YY	66
||	--	#●	dd	ll	tt	BB	JJ	RR	ZZ	77
&&	//	@@	ee	mm	uu	CC	KK	SS	00	88
!!	,,	’’	ff	nn	vv	DD	LL	TT	11	99
$$	%%	==	gg	oo	ww	EE	MM	UU	22	

Page 9 of 9

TABLE B1

OPEN-BUZZ Style (.st z)

Key on left - Chart character on right

. .	* *	_ _	" "	h h	p p	x x	F F	N N	V V	3 3
< «	))	> »	a a	i i	q q	y y	G G	O O	W W	4 4
((	; ;	? ?	b b	j j	r r	z z	H H	P P	X X	5 5
+ +	¬ ¬	: :	c c	k k	s s	A A	I I	Q Q	Y Y	6 6
| |	- -	# O	d d	l l	t t	B B	J J	R R	Z Z	7 7
& &	/ /	@ @	e e	m m	u u	C C	K K	S S	0 0	8 8
! !	, ,	' '	f f	n n	v v	D D	L L	T T	1 1	9 9
$ $	% %	= =	g g	o o	w w	E E	M M	U U	2 2	

TABLE B2 - EXAMPLES OF CHARACTER SIZES

SIZE 1
SIZE 2
SIZE 3
SIZE 4
SIZE 5
SIZE 6
SIZE 7
SIZE 8
SIZE 9

|◄1 inch►|

TABLE B3 - CHARACTER SIZE DIMENSIONS

SIZE CODE [1]	CHAR SIZE (in)	LINE SPAC -ING (in)	LINES PER INCH	CHARS PER 8.5" LINE [2]	CHARS PER 9.5" LINE [3]	LINES PER CHART
1	0.106	0.167	6.0	80	89	45
2	0.157	0.250	4.0	54	60	30
3	0.211	0.333	3.0	40	45	22
4	0.270	0.400	2.5	32	35	18
5	0.376	0.500	2.0	23	25	15
6	0.533	0.667	1.5	16	17	11
7	0.604	0.750	1.33	14	15	10
8	0.813	1.000	1.00	10	11	7
9	1.238	1.500	0.67	6	7	5

NOTES -

[1] Size 1 not recommended except for chart serial numbers, etc. - too small for clarity when projected. Sizes 6 through 9 are headline size and should be used principally for this purpose (for titles, etc.).

[2] This line width obtained by accepting Default .LM and .RM settings (5 and 90, respectively).

[3] This line width obtained by setting .LM 0 and .RM 95.

APPENDIX -C-

BOX AND MATH CHARACTER CONSTRUCTION WITH CHART

BOX CHARACTER CONSTRUCTION

The instruction .ST B allows the user to produce graphical construction (lines, boxes, bars, arrows, textures, etc.) on the chart by using the characters available on the keyboard. The 87 keyboard characters will subsequently employed as elements of graphical construction, instead of as the characters themselves. The .ST B elements which correspond to the keyboard characters are designed such that they will "fit together" from side to side, top to bottom, or superimposed on one-another to produce continuous graphical forms. Alphabetical characters can be overlaid on this graphical construction by setting .ST to other than B (e.g. .ST S), then using the .SP instruction to move back up on the chart, and then contonuing with the lines of text to produce the characters. For example, the following simple sequence in the VI dataset willproduce a box inside of which the message BOO! will appear.

```
0000100 .st b
0000200 a====c
0000300 !    !
0000400 f====h
0000500 .st s
0000600 .sp -2
0000700  BOO!
0000800 .ec
```

In this example the characters a, =, c, !, f, and h beneath the .st b instruction are interpreted as lines and corners, and then, after .st s is encountered, the string "BOO!" is interpreted as BOO! and positioned in the box because of the .sp -2 instruction.

With a little practice, .ST B can be employed to generate very elegant and professional-looking charts with much less effort than would be required to illustrate the chart by hand.

The 87 graphical construction elements are grouped into 5 types. Any of the 87 elements can be used in conjunction with any of the other 87 elements. The 5 types are discussed below. The keyboard characters and their corresponding elements are given together with examples in Table C1.

HISTOGRAMS AND SCHEDULE ELEMENTS - are used to create shaded or unshaded bars of various lengths in schedule charts and histograms. The bars can be capped at either end. Various shadings can be inserted in open bars by using .SP -1. The bars can begin or end on heavy vertical lines. The bar length can be adjusted to within one-half of a character space. The bars can also be used as heavy underlines.

THIN AND HEAVY BOX ELEMENTS - are used to construct boxes, table grids, brackets, flow diagrams, etc. to emphasize a piece of text or to produce a table or schematic. All of the elements are designed to intersect one-another. Heavy-line, thin-line, and combinations of heavy and thin line

construction can be made.

SCHEDULE AND MILESTONE ELEMENTS - are used to produce open, shaded and numbered triangles on schedules. Numbers are inserted in open triangles by using .SP -1.

CHARACTER POSITIONING GUIDES - are useful for chart layout purposes. By using strings of these guides, the position of subsequent text can be determined. They can also be used in the design of forms which would be filled with EDP data.

TEXTURE/RASTER ELEMENTS - provide "wallpaper" or continuous tones for title page backgrounds, borders, dividers, etc. Since they take time to produce, they should be used with discretion. They also tent to be garish if over-used.

MATH CHARACTER CONSTRUCTION

If .ST M is issued in the VI dataset, the following lines of text will be used by chart to display mathematical symbols. For this purpose, the entire keyboard can be used to produce an array of numbers 0 through 9, lower-case a through z, algebraic symbols such as integral signs, and greek letters by using capitol A through Z.

Table B1 (Page 7 of 9) shows the correspondance between the keyboard characters and the .ST M characters.

Table C2 illustrates .ST M.

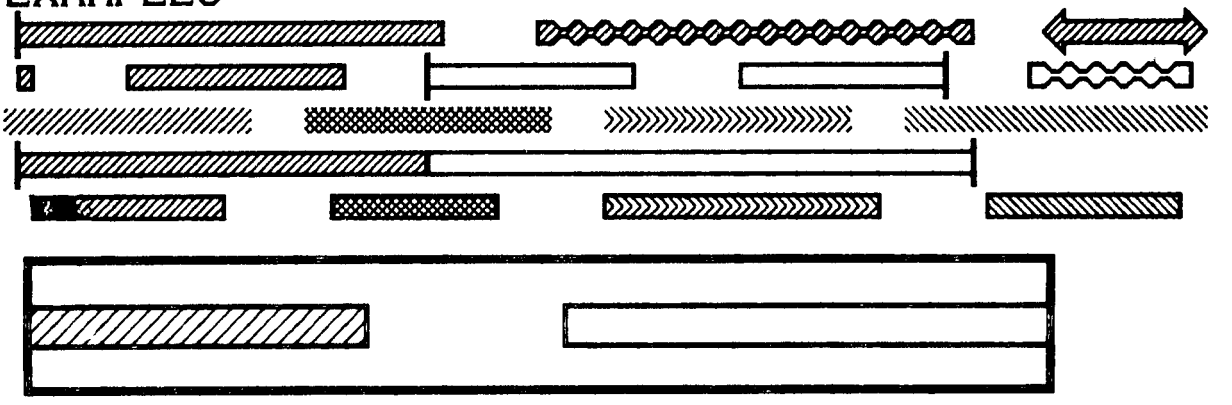
The VI datasets which were used to produce Tables C1 and C2 are called VIEWGRAF.TABLEC1 and VIEWGRAF.TABLEC2.

TABLE CI - BOX STYLE ELEMENTS

.ST B HISTOGRAMS AND SCHEDULE ELEMENTS

( ) n o N O P Q R S T U V W X Y Z 0 : ?

EXAMPLES



.ST B THIN AND HEAVY LINE BOX ELEMENTS

+ | - A B C D E F G H I J k l m I J K L M <

! $ * > # = a b c d e f g h 1 2 3 4 5 6 7 8

EXAMPLES

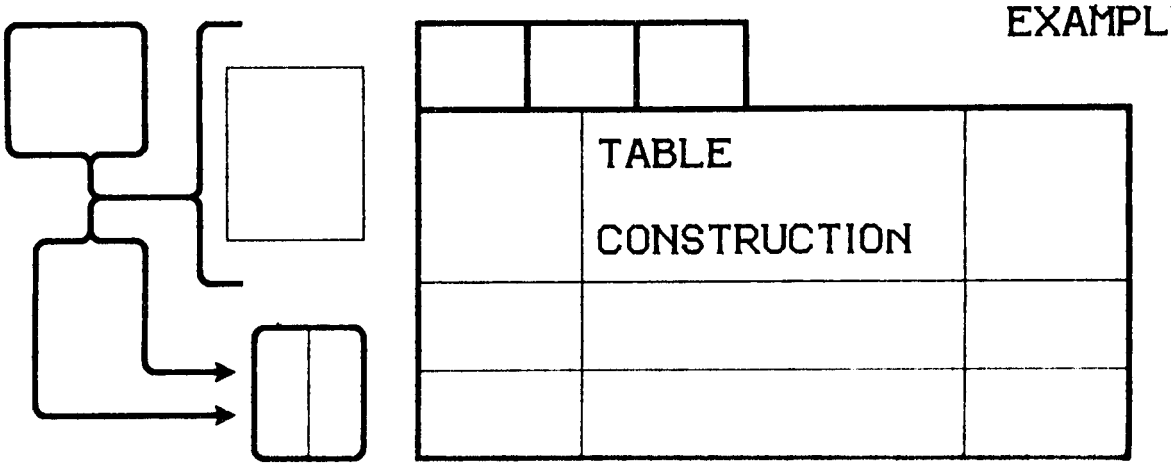


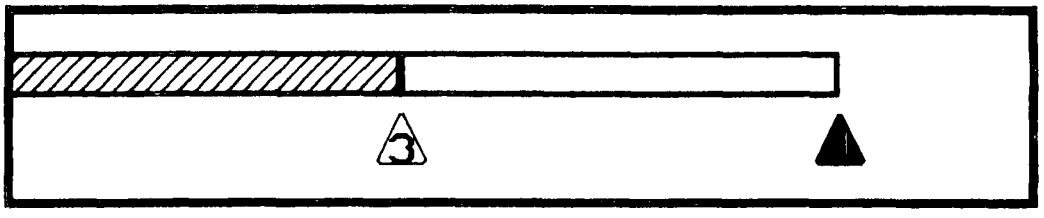
TABLE C1 - (Continued)

.ST B SCHEDULE MILESTONE ELEMENTS

p q r s t u v w x y z

△ 1 2 3 4 5 6 7 8 9 ▲

EXAMPLE



.ST B CHARACTER POSITIONING GUIDES

, % _

EXAMPLES

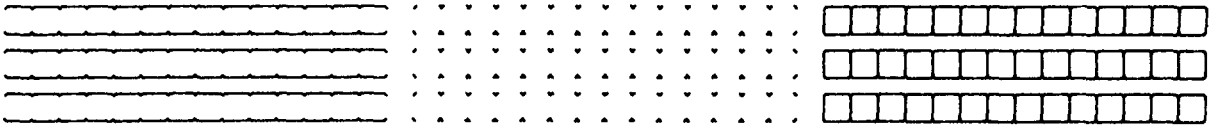


TABLE C1 - (Concluded)

.ST B TEXTURE/RASTER ELEMENTS

& . ; ¬ / @ ' " 9

EXAMPLES

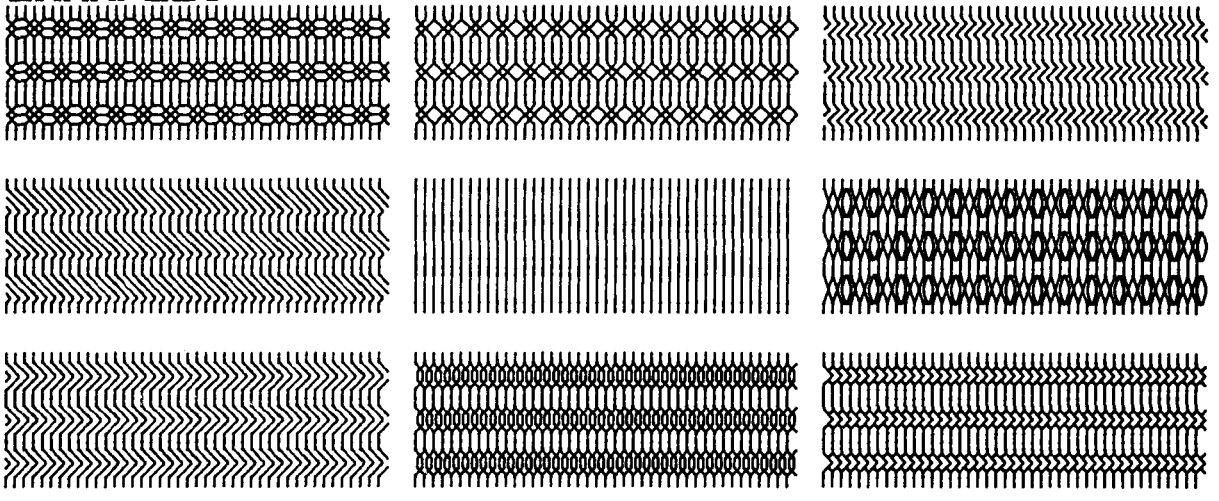


TABLE C2 - EXAMPLES USING STYLE=MATH

$$x(n) = \sum_{n=1}^{5} \frac{(n+1)^2}{n-1} \qquad (1)$$

$$\dot{\beta} = \int_{x=0}^{5.0} \phi(x)\,dx \qquad (2)$$

APPENDIX -D-

CHART COMPOSITION USING ECFLOW

The ECFLOW=(DIRECT | EDIT | RERUN) operand, when issued with the CHART command, provides the user a method of chart composition in addition to the mixture of CHART instructions and text in the input VI dataset. The advantages of ECFLOW are 1) the user composes/edits the chart while he is viewing it and thus has an obvious frame of reference for placement of items, and 2) the user can create geometric shapes which are not possible (or practical) via the instructions in the input VI dataset. The disatvantages of ECFLOW (at the present time) are 1) entry of text (words) is clumsy, and 2) the results of each composing/editing step are not apparent until the user "manually-refreshes" the chart by using one of the single-key ECFLOW=EDIT commands.

CONTROL OF OPERATIONS WITH ECFLOW

As was mentioned before, ECFLOW=EDIT provides for user composition/editing at each point an .EC is encountered in the input VI dataset; ECFLOW=RERUN automatically includes prior editing that was done with ECFLOW=EDIT; and ECFLOW=DIRECT (or default ECFLOW) by-passes both the EDIT and RERUN features of ECFLOW (ie. the chart will consist of only what is in the input VI dataset).

The following are examples of CHART operational sequences and their effects.

CHART XYZ,ECFLOW=RERUN
will default to ECFLOW=DIRECT because no prior CHART XYZ,ECFLOW=EDIT was done.

CHART XYZ,ZETA12,ECFLOW=EDIT
will terminate because user editing/composition cannot be done on a remote device such as the ZETA12.

CHART XYZ,ECFLOW=EDIT
CHART XYZ,ECFLOW=EDIT
CHART XYZ,DEVICE=16MM,ECFLOW=RERUN,FRAMES=200
The second set of EDIT actions will cause the first set to be erased. The third issuance of CHART will run XYZ on 16MM movie film and will incorporate the second set of edit actions.

CHART ZYZ,ECFLOW=EDIT
CHART XYZ
CHART XYZ,ECFLOW=RERUN
The second CHART XYZ will not incorporate and of the EDIT actions, but the third CHART XYZ will.

CHART XYZ,ECFLOW=EDIT
REDIT XYZ (to perform modifications to XYZ)
CHART XYZ,ECFLOW=RERUN
The second CHART XYZ will incorporate the EDIT actions but on the REDIT-modified dataset XYZ. Care must be taken when doing this because, for

example, you might be trying to EDIT action (with ECFLOW=RERUN) on an item of text you erased with REDIT.

ECFLOW=EDIT SINGLE KEY COMMANDS

When CHART is issued with ECFLOW=EDIT, each chart will be displayed with a menu of single key commands printed to the right of the chart. When this display is complete, crosshairs will appear. The crosshairs will dissappear as you enter commands, or during displays or making hardcopies, but they will re-appear when these actions are complete. If the crosshairs do not re-appear, press "X", then press RETURN. If they still do not re-appear, a system error has occured and an error message is waiting in command-display mode. You must then press TAB to exit from graphics display mode, whereupon the error message will appear. In this event, you cannot continue with CHART, so do an ABEND and start over.

The menu of 18 single-key commands which appears to the right of the chart is repeated below.

KEY	EFFECT
;	CHARACTER DRAWING
%	PARALLELOGRAM DRAWING
&	ELLIPSE/CIRCLE DRAWING
_	VECTOR DRAWING
>	ARROW DRAWING
/	DASHED-VECTOR DRAWING
<	DASHED-ARROW DRAWING
*	SYMBOL DRAWING
'	CHANGE CHARACTER STYLE
"	CHANGE CHARACTER SIZE
)	CHANGE COLOR
:	COPY ITEM
@	MOVE (RELOCATE) ITEM
(	ERASE ITEM
?	DISPLAY RESULTS
$	DISPL. W/ITEM LOCATORS
#	CLEAR SEQUENCE
!	RETURN TO CHART

A description of how to issue each single-key command is available by placing the crosshairs on the command for which information is desired and transmitting a "?" (ie. pressing the "?" key, then pressing the RETURN key, and then pressing the RETURN key again when the curser starts to blink). For example, if more information is desired on "& ELLIPSE/CIRCLE DRAWING", place the crosshairs on the "&" in this menu item and transmit "?".

In general, the single-key commands need information to go along with them. This information is supplied via the keyboard (and in some cases, the crosshairs) prior to transmitting the command key. For example, with ") CHANGE COLOR", CHART needs to know what color to change to. This is done as follows. If a change to color code 1 (red) is desired, the user presses the "1" key, presses RETURN, presses the ")" key, and then presses RETURN again. The color for all subsequent items is thus changed to 1 (red). This example shows that transmitting the sequence "!",")" to change the color

consists of pressing a keyboard character (key) and then pressing RETURN, repeating this until the entire sequence is complete. The user may then proceed immidiately to transmit another command sequence. Note that RETURN must be pressed after each key is pressed in order to individually transmit each character needed in the command sequence.

It is suggested that you practice a bit with these single key commands in order to develop familiarity with them. Make a VI dataset with a few lines of text and do CHART DSNAME,ECFLOW=EDIT on it. When the display is complete, try moving, copying, and erasing these lines, then try drawing ellipses and rectangles around them, arrows from one to another, change style and size, and then add new text.

APPENDIX -E-

USING THE CHART .PA (PAUSE) INSTRUCTION - SPECIAL RESTRICTIONS

The .PA instruction is intended to allow the user to invoke his own graphics program(s) at some point in the chart construction process. If, for example, it is desired to insert an X-Y plot on the chart, the input VI dataset would contain a .PA instruction at some convenient location. When CHART encounters this .PA instruction, it executes a FORTRAN pause. The user would then invoke his own program(s) to produce the X-Y plot on the chart, then enter GO to allow CHART to resume to the completion of the chart. The net effect, therefore, would be a chart which contains an X-Y plot.

Normally, the user can accomplish this without problems. However, it is necessary that the user understand the environment created by CHART that exists at the time the FORTRAN pause takes place. It is important not to disturb this environment when the users program(s) are run, otherwise problems will occur when the CHART program resumes.

The following specific conditions exist at the time the FORTRAN pause takes place.

1) CHART has DDEF'ed the following.

DDNAME	DSNAME	FUNCTION
FT91F001	CHARTEMP.ITEMS(IDnnnn)	RESERVED FOR ECFLOW=EDIT OPERATIONS
FT92F001	CHARTEMP.ITEMS(IDnnnn)	RESERVED FOR ECFLOW=EDIT OPERATIONS
FT93F001	CHARTEMP.ERRORS	VSAM ERROR MESSAGE OUTPUT BUFFER
FT94F001	VIEWGRAF.INTART	INTERACTIVE ARTWORK INSTRUCTIONS
FT95F001	CHART.<DSNAME>.PAGEnn	EDIT/RERUN I/O (FOR ECFLOW)
FT96F001	VIEWGRAF.GRIDS	UNFORMATTED (BINARY) DS CONTAINING GRID COORDINATES FOR .GR INSTRUCTION
FT97F001	SYSOUT	
FT98F001	CHARTEMP.IOBUF	VSAM TEMPORORY I/O BUFFER
FT99F001	<DSNAME>	VISAM "RUNOFF STYLE" DATASET CONTAINING INSTRUCTIONS AND TEXT TO BE CHARTED
DDNAME1	VIEWGRAF.OBJLIB	VP JOBLIB WHICH CONTAINS THIS AND OTHER PROGRAMS
DDNAME2	GRAPHICS	TSS GRAPHICS PACKAGE
CZAPPnnn	VIEWGRAF.PROCDEFS	VP DATASET WHICH CONTAINS THE PROCS FOR RUNNING THIS PACKAGE

Do NOT perform your own DDEF's using these DDNAMEs or DSNAMEs. Also note that the TSS Graphics Package is already DDEF'ed, so don't try to DDEF it

again.

2) The following names are used by CHART.

NAME	NAME TYPE
CALLDUPE	PROCDEF NAME
CGRID	SUBPROGRAM ENTRY NAME
CHART	PROCDEF NAME
CPRESS	SUBPROGRAM ENTRY NAME
DUPE	SUBPROGRAM ENTRY NAME
INTART	SUBPROGRAM ENTRY NAME
READV	SUBPROGRAM ENTRY NAME
RECORD	PROCDEF NAME
RUNART	SUBPROGRAM ENTRY NAME
RUNDUPE	MAIN PROGRAM NAME
SCGRID	SUBPROGRAM COMPILED NAME
SCPRESS	SUBPROGRAM COMPILED NAME
SDUPE	SUBPROGRAM COMPILED NAME
SINTART	SUBPROGRAM COMPILED NAME
SRUNART	SUBPROGRAM COMPILED NAME
STURN	SUBPROGRAM COMPILED NAME
SVIDSIN	SUBPROGRAM COMPILED NAME
TFREAD	SUBPROGRAM ENTRY NAME
TURN	SUBPROGRAM ENTRY NAME
VIDSIN	SUBPROGRAM ENTRY NAME
VIEWNEWS	PROCDEF NAME
VUCHART	MAIN PROGRAM NAME
WRITEREC	PROCDEF NAME

Do not invoke programs of your own that have these names. Better still, don't use these names, ie. think of other names for your programs and procdefs. Just to be on the safe side, release your JOBLIB when you are done running your programs and before issuing GO to resume with CHART.

3) The CHART program supresses the axes and corners that identify the extremities of the 10X10-inch plotting field. CHART establishes all locations using screen relative units. The user, therefore, must make calls to axis subprograms as part of generating his plot. When CHART resumes, it will automatically reestablish its own profile.

4) The character style which was set with the .ST instruction remains in effect while the user's program is running. Thus, calls to PRCHARS, GLABEL, and TITLE will result in displays in this style. Styles other than .ST G (standard graphics package) style will sometimes produce unpredictable results when used in PRCHARS, GLABEL, or TITLE. To avoid problems, use only .ST G, .ST S, or .ST I. If S or I styles are used, avoid using lower case.

5) Items that you add to the chart by using your own program(s) cannot be moved, copied, or erased when ECFLOW=EDIT. You may, however, add items with ECFLOW=EDIT directly over (superimposed upon) these items. This is especially useful if you have created a multiple-curve plot and you then use ECFLOW=EDIT with crosshair support to position and add word and symbol items to the curves on the plot.

APPENDIX -F-

PCSR SORT RECORD FORMAT - PROGRAMMING INFORMATION

The PCSR sort records are 276 bytes long and are written into a VS dataset. Twenty-seven data fields are contained in each of these records. The name and position of these fields is given below.

FIELD NAME	LENGTH	OFFSET
INCLUDE ITEM	4	0
RTOP	12	4
TASK	8	16
PR	8	24
VALUE	8	32
FY	4	40
TYPE	4	44
ENGINEER	12	48
TITLE	50	60
BUYER	12	110
(1) PLANNED	6	122
(1) ACCOMPLISHED	6	128
(2) PLANNED	6	134
(2) ACCOMPLISHED	6	140
(3) PLANNED	6	146
(3) ACCOMPLISHED	6	152
(4) PLANNED	6	158
(4) ACCOMPLISHED	6	164
(5) PLANNED	6	170
(5) ACCOMPLISHED	6	176
(6) PLANNED	6	182
(6) ACCOMPLISHED	6	188
(7) PLANNED	6	194
(7) ACCOMPLISHED	6	200
PROGRESS	4	206
COMMENTS	56	212
ITEM KEY	8	268

274 TOTAL BYTES

The DATE fields (1) PLANNED through (7) ACCOMPLISHED are in the form MMDDYY, therefore, to sort on dates, it is necessary to sort on MMDD first and then to sort on YY.

APPENDIX -G-

STANDARD LEAD TIME FOR R and D PROCUREMENTS FOR USE WITH PCSR SOFTWARE

A Table has been prepared that can be used to establish dates for PLANNED procurement milestones. This Table is quite long and is therefore not included in this report. To get a copy of it, contact the Space Propulsion Division Office at the Lewis Research Center. To use the table, look up the date for milestone (1) in the left column, then go across until you find the column of dates pertaining to the TYPE of procurement and use these dates for the PLANNED procurement milestones. The Table assumes the following elapsed times for the various types of procurements.

COMPETITIVE	SOLE SOURCE	GRANTS
1 to 2 3 wks.	1 to 2 3 wks.	1 to 2 3 wks.
2 to 3 4 wks.	2 to 3 4 wks.	2 to 7 4 wks.
3 to 4 5 wks.	3 to 4 4 wks.	
4 to 5 6 wks.	4 to 5 2 wks.	
5 to 6 8 wks.	5 to 6 6 wks.	
6 to 7 3 wks.	6 to 7 3 wks.	

INTERAGENCY	INCREMENTAL	OVERRUNS
1 to 2 3 wks.	1 to 2 3 wks.	1 to 2 3 wks.
2 to 7 5 wks.	2 to 3 2 wks.	2 to 5 1 wk.
	3 to 7 4 wks.	5 to 6 8 wks.
		6 to 7 2 wks.

APPENDIX -H-

SAMPLE OUTPUTS FROM THE IBM3800 PRINTER

The following figures are examples of IBM3800 printer output from the varions software items described in this report. Before proceeding to them, however, it must be mentioned that this entire report is an example of the RUNOFF90 and PRINT90 software previously described. In addition, the Tables in Appendix B and Appendix C are examples of CHART output.

Figure H1 is an example of BIGPRINT output at full scale.

Figure H2 is an example of HUGE output at full scale.

Figure H3 is an example of PCSR output at slightly reduced scale (to fit on the page).

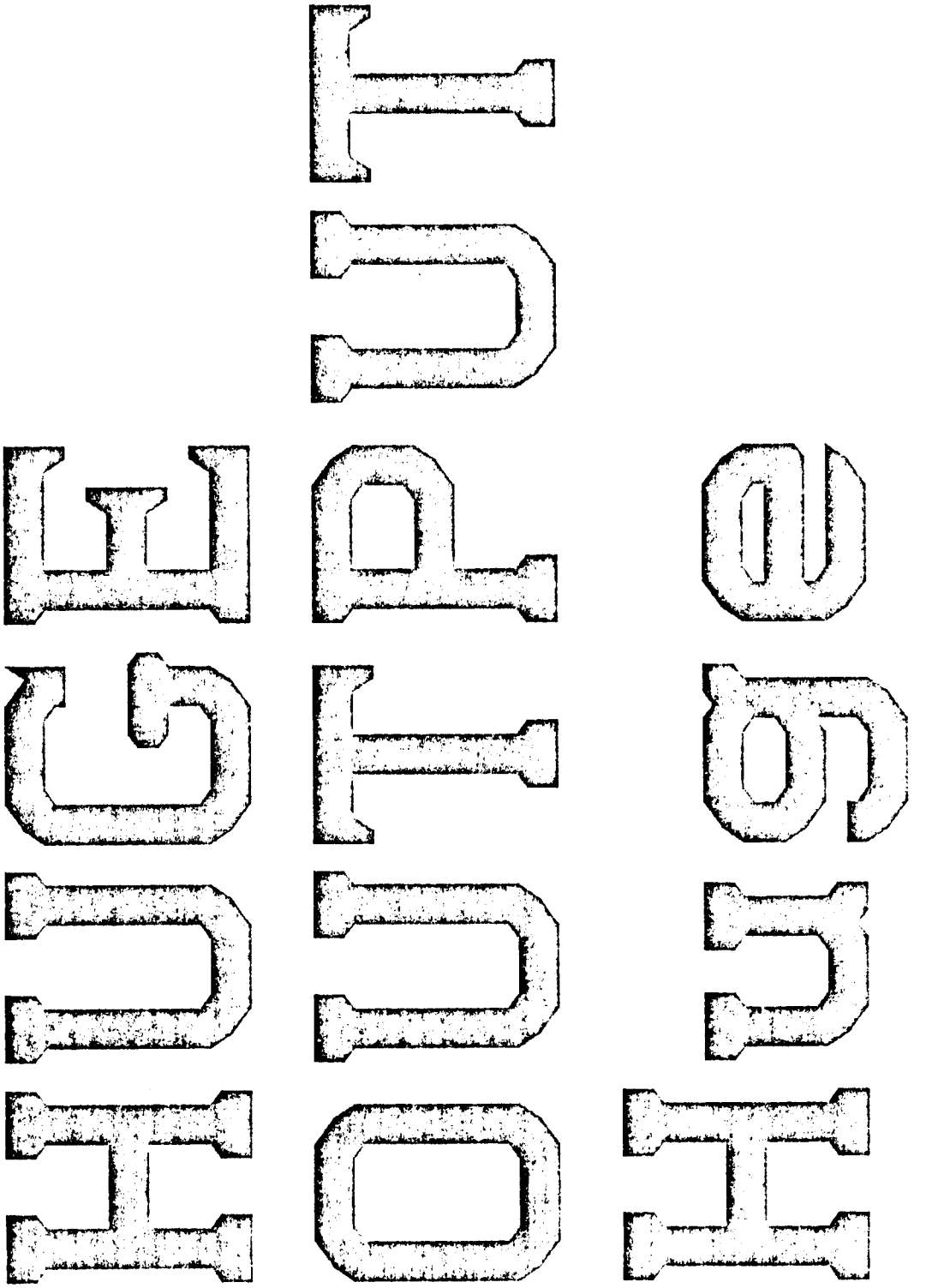
SYSTEMS INTEGRATION

OBSERVATIONS

- SYSTEMS INTEGRATION CONTINUALLY INCREASES IN COMPLEXITY AND COST.
  - MORE SYSTEMS - MORE INTERFACES THAN BEFORE
  - SYSTEMS MORE COMPLEX THAN BEFORE
  - MORE CONFORMANCE CRITERIA THAN BEFORE
- LSS FURTHER AGGREVATES THIS PROBLEM.
- LSS MISSIONS POSSIBLY PREVENTED BY IT.

FIGURE H1 - BIGPRINT OUTPUT SAMPLE (FULL SCALE)

FIGURE H2 - HUGE OUTPUT SAMPLE (FULL SCALE)



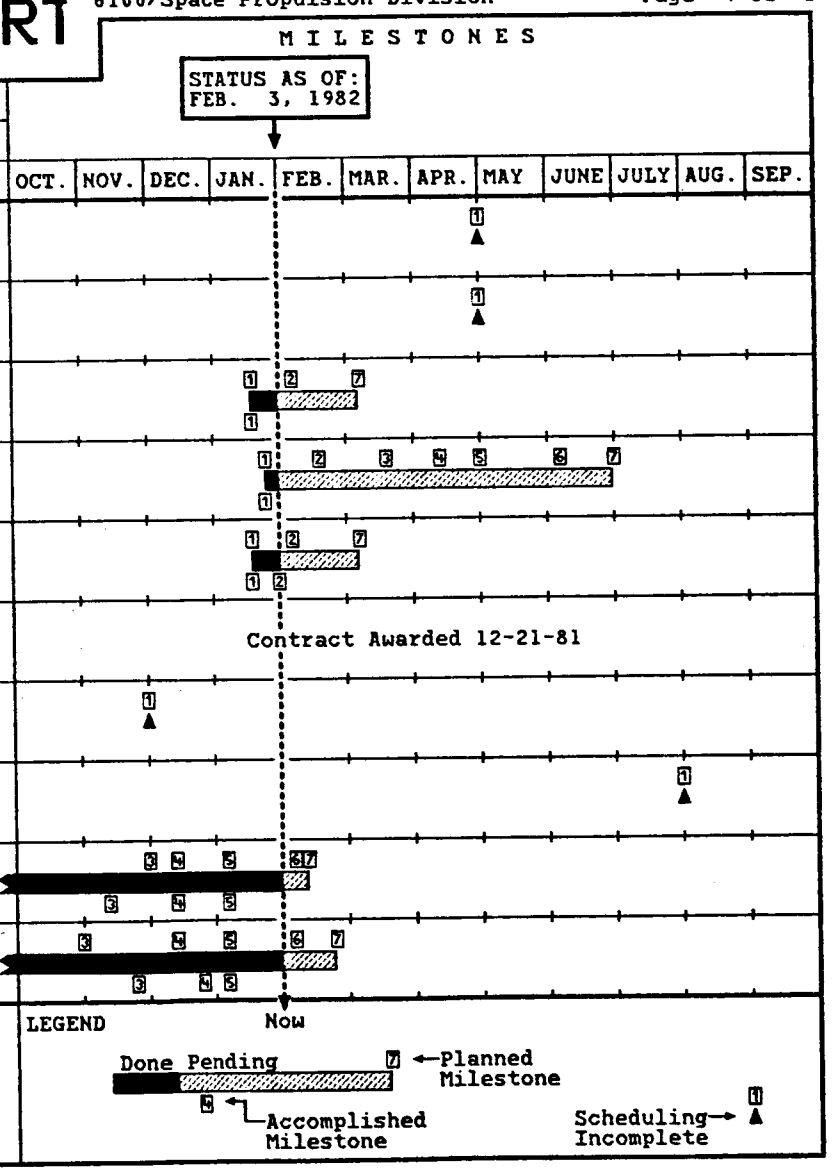
PROCUREMENT STATUS REPORT

6100/Space Propulsion Division

Page 4 of 8

Sorted by RTOP NUMBER

ITEM	RTOP / TITLE / COMMENT	TASK	PR	VALUE	FY	TYPE	ENGINEER / BUYER
31	506-52-42 P & W OTV Tech Phase 2	S1851		500K	82	S	Priem 0000595
32	506-52-42 AGC OTV Tech Phase 2	S1851		500K	82	S	Priem 0000622
33	506-52-42 Symposium of Combustion Inst. Milestone 2 accomplished 2-4-82	S2197	361931	15K	82	S	Priem Shinn 0001810
34	506-52-42 Thrust chamber life prediction	S2428	417945	70K	82	S	Kasper 0001945
35	506-52-42 Thermal Conductivity	S5689	417944	50K	82	A	Priem Cochran 0001837
36	506-52-42 Heat Transfer in Porous	S6995	417924	41K	82	G	Aukerman Bane 0000730
37	506-52-62 Reusuable Engine Maintenance Milestone 1 by 3-11-82 or cancel	S0571		80K	82	C	Masters 0000757
38	506-52-62 Hybrid Bearing	S1370		246K	82	I	Hannum 0000811
39	506-55-22 8 cm Thruster Limits	S0490	353967	155K	82	S	Mantenieks Boyer 0000892
40	506-55-22 RF Ion Generation	S0492	429361	200K	82	S	Nakanishi Burke 0000946



TYPE OF PROCUREMENT
- C-Competitive
- S-Sole Source
- G-Grant
- A-Interagency
- I-Incremental
- O-Overrun

MILESTONES
- 1-Tech. Pkg. to Div. Chief
- 2-PR to Procurement
- 3-RFP Issued
- 4-Proposals Received
- 5-Tech. Eval. Complete
- 6-Negogiation Complete
- 7-Award

FIGURE H3 - PCSR OUTPUT SAMPLE (SLIGHTLY REDUCED)