

UNITED STATES DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

User oriented, interactive Multics computer programs
to create grid cell, contour, and perspective maps
using Surface Display Library

by

R. K. Mark, and E. B. Newman

Open-File Report 81-513

This report is preliminary and
has not been reviewed for conformity
with U.S. Geological Survey
editorial standards. Any use of
trade names is for descriptive
purposes only and does not imply
endorsement by the USGS.

Menlo Park, California

1981

CONTENTS

Abstract	3
Info Segs	4
sdl_drivers.info	4
Annotated terminal sessions	6
sdl_gridder	6
sdl_convue	7
sdl_mshvue	11
sdl_conmap	14
sdl_cellmap	17
sdl_names.info	20
Program source code	24
sdl_gridder.fortran	24
sdl_conmap.fortran	26
sdl_convue.fortran	30
sdl_mshvue.fortran	34
sdl_cellmap.fortran	38
sdl_device.fortran	42
sdl_grid_.fortran	43
sdlpop_device_.fortran	45
upper_case_.pl1	46
tkx_4025_setup.fortran	46
clctxt_.pl1	47
get_term_type_.pl1	48
query_.pl1	49
sdlpop.ec	49

ABSTRACT

Computer programs `sdl_conmap`, `sdl_convue`, `sdl_mshvue`, `sdl_cellmap`, and `sdl_gridder` are designed for users of the Honeywell Multics System who have a need for convenient generation of contour, perspective, or grid cell maps.

These programs utilize the Surface Display Library* and Surface Gridding Library* subroutines.

* Program packages developed by Dynamic Graphics Inc, Berkeley, Ca.

3/27/81 sdl_drivers.info

sdl_conmap sdl_convue sdl_mshvue sdl_cellmap sdl_gridder

Introduction:

The Surface Display Library (SDL) is a collection of Fortran subroutines for generating graphic representations of three-dimensional continuous surfaces, in the form of contour maps and full perspective views. It was developed by Dynamic Graphics, Inc., Berkeley, California, and is documented in the User Manual for the Surface Display Library. The Surface Gridding Library (SGL) is a collection of Fortran subroutines for producing gridded numerical representations of three-dimensional continuous surfaces from scattered point input. The rectangular grid it produces is suitable for use by SDL. It is also a product of Dynamic Graphics, Inc., and is documented in the Users Manual for the Surface Gridding Library.

A package of Fortran subroutines for producing cell maps were developed by Dynamic Graphics, Inc., under contract to the USGS. These are documented in `sdl_cellmap.info`.

The programs `sdl_conmap`, `sdl_convue`, `sdl_mshvue`, `sdl_cellmap`, and `sdl_gridder` are user oriented, interactive drivers for the above subroutine libraries. They are designed for ease of use, and do not provide the full range of options available through direct use of the SDL and SGL subroutines.

Plotting is currently supported on Tektronix, Versatec, and Houston, either directly or through the SDL postprocessor. The program prompts for the plotting device. An `exec_com` (`sdlpop.ec`) is available for generating the postprocessor plots.

Variable names:

The programs prompt for input using the standard SDL variable names. These names are defined in the manuals and in `sdl_names.info`.

Data files:

Data files, with the exception of the `sdl_cellmap` character array, are read with a list directed, free field format. Data entries are separated by spaces or a comma. One type of data file, used as input to `sdl_cellmap`, contains a grid of elevation values or characters. Each line is one row of the matrix. The other data file is used as input to `sdl_gridder` and/or to plot points on maps or views. It contains one data point per line defined by either `x,y` or `x,y,z`.

`sdl_gridder` produces an elevation grid in a standard format, which is directly accessible to `sdl_conmap`, `sdl_convue`, and `sdl_mshvue`. The input file contains all the parameters needed to define the grid. If

the grid file is not in this standard format, the programs will prompt for the needed parameters.

sdl_cellmap requires a grid file of digits and/or upper case characters which is read with a 400a1 format. The program will prompt for the required grid parameters.

The arrays are limited to a maximum of 400 by 400 entries. Data files are limited to 1000 entries.

Plotting points on map or view:

The programs will, optionally, plot points on maps or views. Maps require a data file of x and y values in the grid unit system. Views require a data file with x, y, and z values (ie. the same format as input to sdl_gridder). There should be one data point per line.

sdl_gridder annotated terminal session:

sdl_gridder

See: sdl_names.info or the Surface Gridding Library manual for definitions of variable names.

help sdl_names -section <name>

input data file (x,y,z): blanco.data

output grid file: blanco.2m_by_10m.grid

132 data points read.

enter grid parameters

inmxcl: 12 [12 columns in output grid]

inmyrw: 41 [41 rows in output grid]

xgdmin: -60 [coordinates of grid corners]

xgdmax: 50

ygdmin: -40

ygdmax: 40

grid report to terminal? (y or n): n [an extensive report is generated]

grid report in file grid_report

grid written to blanco.2m_by_10m.grid [following parameters in grid file]

inmxcl = 12, inmyrw = 41, iaxorn = 1

xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.

zgdnul = -9999.

STOP

r 06:54 32.381 336

sdl_convue annotated terminal session:

sdl_convue

If using a Tektronix, the execution log is in sdl_log.

See: sdl_names.info or the SDL manual for definitions of variable names.

help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files
To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid

inmxcl = 12, inmyrw = 41, iaxorn = 1

xgdmn = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.

zgdnul = -9999.

null data? (y or n): n [should grid points with value zgdnul be ignored]

Do you want to plot data points? (y or n): y

data points (x,y,x) file name: blanco.data

132 data points read.

caption (in quotes): "convue of penasco blanco site"

contour control variables

zvlref: 0 [reference level]

zvlint: .25 [contour interval]

ithbld: 4 [every 4th contour bold]

ithlab: 4 [every 4th contour labeled]

itptxt (1-6): 2 [label size]

ideplc: -1 [places after decimal point; -1 = no point]

scale elvarr? (y or n): y

zprfac: 7 [vertical exaggeration]

default perspective is: datsph=10., thdinc=20., phidxy=325.

changes? (y or n): y

datsph: , [retain default distance]

thdinc: , [retain default inclination]

phidxy: 145 [view from the opposite side]

depression ticks? (y or n): n

Do you want to change anything before plotting? (y or n): n

device (t, v, h, p, or?): ?

t=Tektronix, v=Versatec

h=Houston, p=postprocessor

device (t, v, h, p, or?): p

I EXECUTION REPORT LOG

I SURFACE DISPLAY LIBRARY

I VERSION SDL1, RELEASE 3.0I

I DYNAMIC GRAPHICS, INC.

I DATE WEDNESDAY 25 MARCH 1981

I TIME 2.52.36 P.M.

I

C 1 BGNPLT

C 2 SETXYG INMXCL= 12 XGDMIN= -60.000 XGDMAX= 50.000

```

+          INMYRW=      41      YGDMIN= -40.000      YGDMAX=  40.000
C  3 SETNUL IFLNUL=      0      ZGDNUL= -9999.0      EPSNUL= 0.10000E-04
C  4 SETPRP IFLPRP=      1      XPRFAC=  1.0000      YPRFAC=  1.0000
+          ZPRFAC=  7.0000
C  5 SETORN IFLORN=      1      IAXORN=      1
C  6 SETPAG XPGLEN=  10.000      YPGLEN=  10.000      XPGOFF=  1.0000
+          YPGOFF=  1.0000
C  7 SETWIN XWNLEN=  8.0000      YWNLEN=  8.0000      XWNOFF=  1.0000
+          YWNOFF=  1.0000
C  8 SETCON ZVLREF=  0.00000      ZVLINT=  .25000      ITHBLD=      4
+          ITHLAB=      4      ITPTXT=      2      IDCPLC=     -1
C  9 SETTIK IFLTIC=      0      ITHTIK=      1      TIKLEN= 0.40000E-01
+          TIKSEP=  .10000
C 10 SETVUE DATSPH=  10.000      THDINC=  20.000      PHIDXY=  145.00
C 11 CONVUE ELVARR= *ARRAY*      IDMXCL=   400      IDMYRW=   400
+          IWKARR= *ARRAY*      IDMWRK= 100000
+          M31 ELAPSED TIME IN ROUTINE=  8.0060      SECONDS.
+          (RESOLUTION OF TIMER IS SECS/1000.)
+          M32 THIS DISPLAY REQUIRED  337 ELEMENTS OF THE WORK ARRAY.
C 145 SETRPL IFLRPL=      3      IUNRPL=      6
C 146 CLCTSZ ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      29
+          TXTLEN=*RESULT*      TXTHGT=*RESULT*
+          M21 STRING PARAMETER 2=CONVUE OF PENASCO BLANCO SITE
+          M23 RESULT PARAMETER 4=  8.6000      .
+          M23 RESULT PARAMETER 5=  .35000      .
C 147 CLCTSZ ITPTXT=      5      ICHARR= *ARRAY*      ICHCNT=      29
+          TXTLEN=*RESULT*      TXTHGT=*RESULT*
+          M21 STRING PARAMETER 2=CONVUE OF PENASCO BLANCO SITE
+          M23 RESULT PARAMETER 4=  6.8800      .
+          M23 RESULT PARAMETER 5=  .28000      .
C 148 CAPTXT IFLSID=      1      OFFCAP=  .36000      IFLJST=      2
+          ITPTXT=      5      ICHARR= *ARRAY*      ICHCNT=      29
+          M21 STRING PARAMETER 5=CONVUE OF PENASCO BLANCO SITE

```

Do you want another plot? (y or n): n

C 149 ENDPLT

The SDL independent plot file is finished [file sdl_plot created]

S

S SUMMARY OF EXECUTION

S SDL TOTAL CALLS= 149

S CALLS WITH ERRORS= 0

S SDL TOTAL FRAMES= 1

S FRAMES WITH ERRORS= 0

S TOTAL TIME IN SDL CALLS= 9.8060 SECONDS.

S (RESOLUTION OF TIMER IS SECS/1000.)

S END OF SUMMARY

STOP

r 14:54 13.327 756

ec sdlpop [use SDL postprocessor]

device (t, v, or h): v

MAPPED - VECTOR

DEFAULT VALUES OF ALL PARAMETERS WILL BE USED

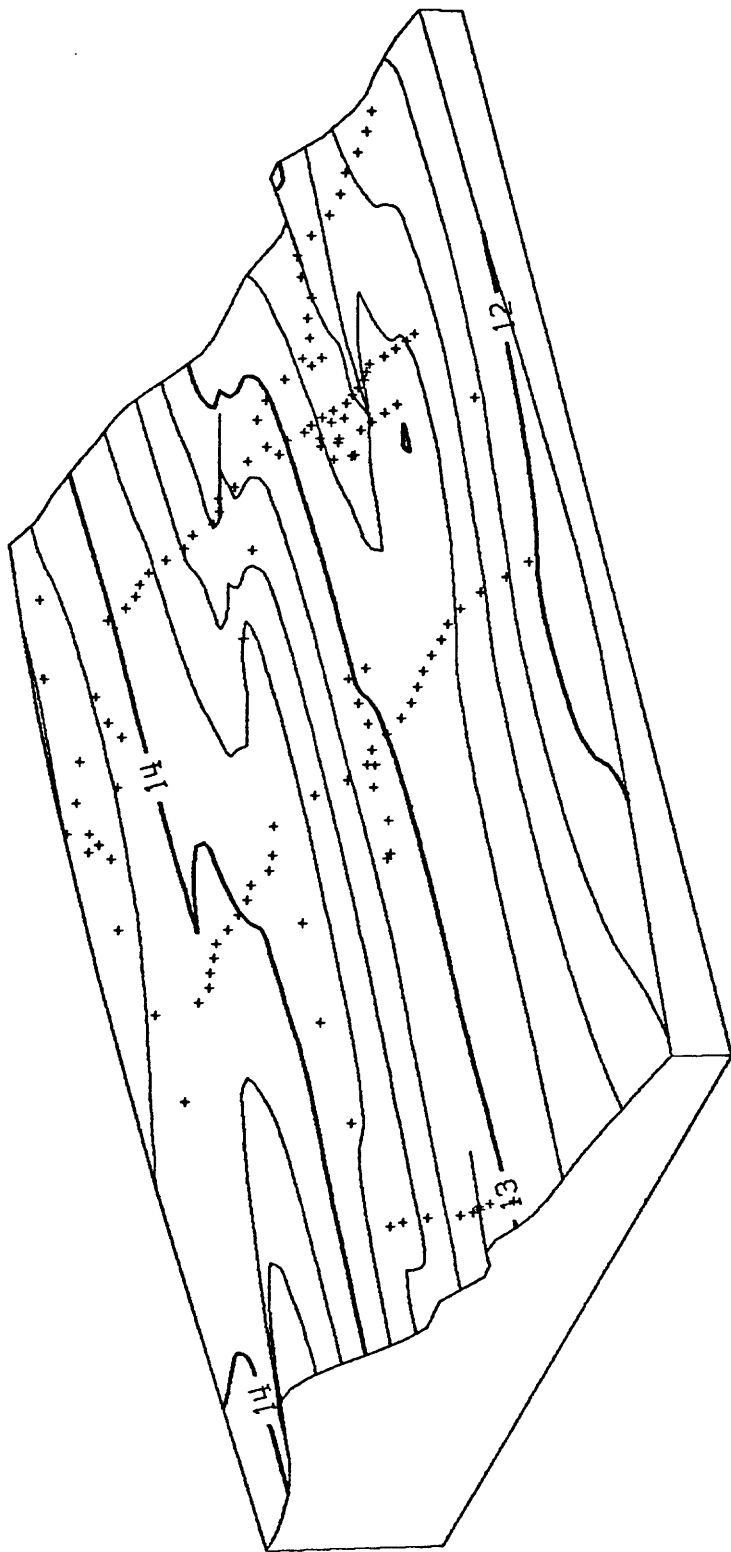
*****PLOT OPTIONS IN EFFECT*****

MODEL	=	8242	,XMIN	=	0.00	,XMAX	=	100.00
YMIN	=	0.00	,YMAX	=	81.90	,MSGVLV	=	1
XSTART	=	0.00	,YSTART	=	0.00	,SCALE	=	1.00
XFACT	=	1.00	,YFACT	=	1.00	,UNITS	=	1.00
STRIP	=	40.96	,STRIPO	=	0.00	,SPACE	=	6.00
NSCAN	=	382	,OUT	=	-1.00	,LYNES	=	5000

0 VECTORS LOST
27 ACTIVE LINES USED

The plotting is finished.

delete sdl_plot ? no
r 15:00 21.576 1409



CONVUE OF PENASCO BLANCO SITE

sdl_mshvue annotated terminal session:

sdl_mshvue

If using a Tektronix, the execution log is in sdl_log.

See: sdl_names.info or the SDL manual for definitions of variable names.

help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files
To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid

inmxcl = 12, inmyrw = 41, iaxorn = 1

xgdmin = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.

zgdnul = -9999.

null data? (y or n): n [should grid points with value zgdnul be ignored]

Do you want to plot data points? (y or n): y

data points (x,y,x) file name: blanco.data

132 data points read.

caption (in quotes): "mshvue of penasco blanco site"

scale elvarr? (y or n): y

zprfac: 7 [vertical exaggeration]

default perspective is: datsph=10., thdinc=20., phidxy=325.

changes? (y or n): y

datsph: , [retain default distance]

thdinc: , [retain default inclination]

phidxy: 145 [view from the opposite side]

Do you want to change anything before plotting? (y or n): n

device (t, v, h, p, or?): p

I EXECUTION REPORT LOG

I SURFACE DISPLAY LIBRARY

I VERSION SDL1, RELEASE 3.0I

I DYNAMIC GRAPHICS, INC.

I DATE THURSDAY 26 MARCH 1981

I TIME 8.44.07 A.M.

I

C 1 BGNPLT

C 2 SETXYG INMXCL= 12 XGDMIN= -60.000 XGDMAX= 50.000

+ INMYRW= 41 YGDMIN= -40.000 YGDMAX= 40.000

C 3 SETNUL IFLNUL= 0 ZGDNUL= -9999.0 EPSNUL= 0.10000E-04

C 4 SETPRP IFLPRP= 1 XPRFAC= 1.0000 YPRFAC= 1.0000

+ ZPRFAC= 7.0000

C 5 SETORN IFLORN= 1 IAXORN= 1

C 6 SETPAG XPGLEN= 10.000 YPGLEN= 10.000 XPGOFF= 1.0000

+ YPGOFF= 1.0000

C 7 SETWIN XWNLEN= 8.0000 YWNLEN= 8.0000 XWNOFF= 1.0000

+ YWNOFF= 1.0000

C 8 SETVUE DATSPH= 10.000 THDINC= 20.000 PHIDXY= 145.00

C 9 MSHVUE ELVARR= *ARRAY* IDMXCL= 400 IDMYRW= 400

+ IWKARR= *ARRAY* IDMWRK= 100000

```

+      M31 ELAPSED TIME IN ROUTINE=  7.0570      SECONDS.
+      (RESOLUTION OF TIMER IS SECS/1000.)
+      M32 THIS DISPLAY REQUIRED  337 ELEMENTS OF THE WORK ARRAY.
C 143 SETRPL IFLRPL=      3      IUNRPL=      6
C 144 CLCTSZ ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      29
+      TXTLEN=*RESULT*      TXTHGT=*RESULT*
+      M21 STRING PARAMETER 2=MSHVUE OF PENASCO BLANCO SITE
+      M23 RESULT PARAMETER 4=  8.6000      .
+      M23 RESULT PARAMETER 5=  .35000      .
C 145 CLCTSZ ITPTXT=      5      ICHARR= *ARRAY*      ICHCNT=      29
+      TXTLEN=*RESULT*      TXTHGT=*RESULT*
+      M21 STRING PARAMETER 2=MSHVUE OF PENASCO BLANCO SITE
+      M23 RESULT PARAMETER 4=  6.8800      .
+      M23 RESULT PARAMETER 5=  .28000      .
C 146 CAPTXT IFLSID=      1      OFFCAP=  .36000      IFLJST=      2
+      ITPTXT=      5      ICHARR= *ARRAY*      ICHCNT=      29
+      M21 STRING PARAMETER 5=MSHVUE OF PENASCO BLANCO SITE

```

Do you want another plot? (y or n): n

C 147 ENDPLT

The SDL independent plot file is finished

S

S SUMMARY OF EXECUTION

S SDL TOTAL CALLS= 147

S CALLS WITH ERRORS= 0

S SDL TOTAL FRAMES= 1

S FRAMES WITH ERRORS= 0

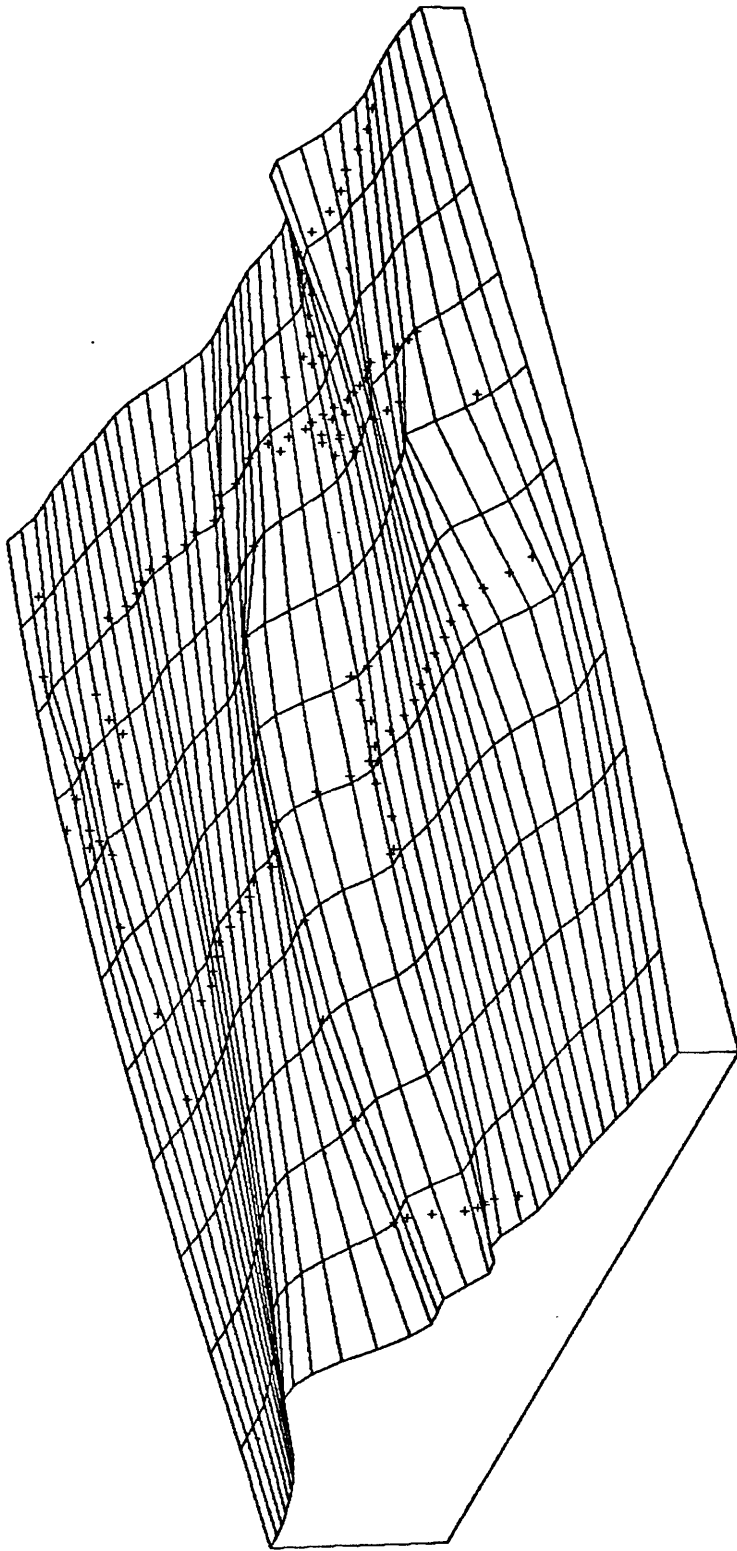
S TOTAL TIME IN SDL CALLS= 8.7490 SECONDS.

S (RESOLUTION OF TIMER IS SECS/1000.)

S END OF SUMMARY

STOP

r 08:45 11.269 811



MSHVUE OF PENASCO BLANCO SITE

sdl_conmap annotated terminal session:

sdl_conmap

If using a Tektronix, the execution log is in sdl_log.

See: sdl_names.info or the SDL manual for definitions of variable names.

help sdl_names -section <name>

CAUTION: this program overwrites existing default plot files

To save plot files, rename them before execution.

Data matrix file name: blanco.2m_by_10m.grid

grid read from blanco.2m_by_10m.grid

inmxcl = 12, inmyrw = 41, iaxorn = 1

xgdmn = -60., xgdmax = 50., ygdmin = -40., ygdmax = 40.

zgdnul = -9999.

null data? (y or n): n [should grid points with value zgdnul be ignored]

Do you want to plot data points? (y or n): y

data points (x,y) file name: blanco.data

132 data points read.

xpglen: 12 [page size in inches]

ypglen: 9

xwnlen: 7.22 [window size in inches; determines map scale]

ywnlen: 5.25

caption (in quotes): "penasco blanco site"

contour control variables

zvlref: 0 [reference level]

zvlint: .25 [contour interval]

ithbld: 4 [every 4th contour bold]

ithlab: 4 [every 4th contour labeled]

itptxt (1-6): 2 [label size]

ideplc: -1 [places after decimal point; -1 = no point]

depression ticks? (y or n): n

Do you want to change anything before plotting? (y or n): n

device (t, v, h, p, or?): p

I EXECUTION REPORT LOG

I SURFACE DISPLAY LIBRARY

I VERSION SDL1, RELEASE 3.0I

I DYNAMIC GRAPHICS, INC.

I DATE THURSDAY 26 MARCH 1981

I TIME 8.26.52 A.M.

I

C 1 BGNPLT

C 2 SETXYG INMXCL= 12 XGDMIN= -60.000 XGDMAX= 50.000

+ INMYRW= 41 YGDMIN= -40.000 YGDMAX= 40.000

C 3 SETNUL IFLNUL= 0 ZGDNUL= -9999.0 EPSNUL= 0.10000E-04

C 4 SETORN IFLORN= 1 IAXORN= 1

C 5 SETPAG XPGLEN= 12.000 YPGLEN= 9.0000 XPGOFF= 1.0000

+ YPGOFF= 1.0000

C 6 SETWIN XWNLEN= 7.2200 YWNLEN= 5.2500 XWNOFF= 2.3900

+ YWNOFF= 1.8750

```

C 7 SETCON ZVLREF= 0.00000      ZVLINT= .25000      ITHBLD=      4
+           ITHLAB=      4      ITPTXT=      2      IDCPLC=     -1
C 8 SETTIK IFLTIK=      0      ITHTIK=      1      TIKLEN= 0.40000E-01
+           TIKSEP= .10000
C 9 CONMAP ELVARR= *ARRAY*      IDMXCL=      400      IDMYRW=      400
+           IWKARR= *ARRAY*      IDMWK=      11000
+           M31 ELAPSED TIME IN ROUTINE= 13.930      SECONDS.
+           (RESOLUTION OF TIMER IS SECS/1000.)
+           M32 THIS DISPLAY REQUIRED      15 ELEMENTS OF THE WORK ARRAY.
C 143 SETRPL IFLRPL=      3      IUNRPL=      6
C 144 CLCTSZ ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      19
+           TXTLEN=*RESULT*      TXTHGT=*RESULT*
+           M21 STRING PARAMETER 2=PENASCO BLANCO SITE
+           M23 RESULT PARAMETER 4= 5.6000      .
+           M23 RESULT PARAMETER 5= .35000      .
C 145 CAPTXT IFLSID=      1      OFFCAP= .76250      IFLJST=      2
+           ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      19
+           M21 STRING PARAMETER 5=PENASCO BLANCO SITE

```

Do you want another plot? (y or n): n

C 146 ENDPLT

The SDL independent plot file is finished

S

S SUMMARY OF EXECUTION

S SDL TOTAL CALLS= 146

S CALLS WITH ERRORS= 0

S SDL TOTAL FRAMES= 1

S FRAMES WITH ERRORS= 0

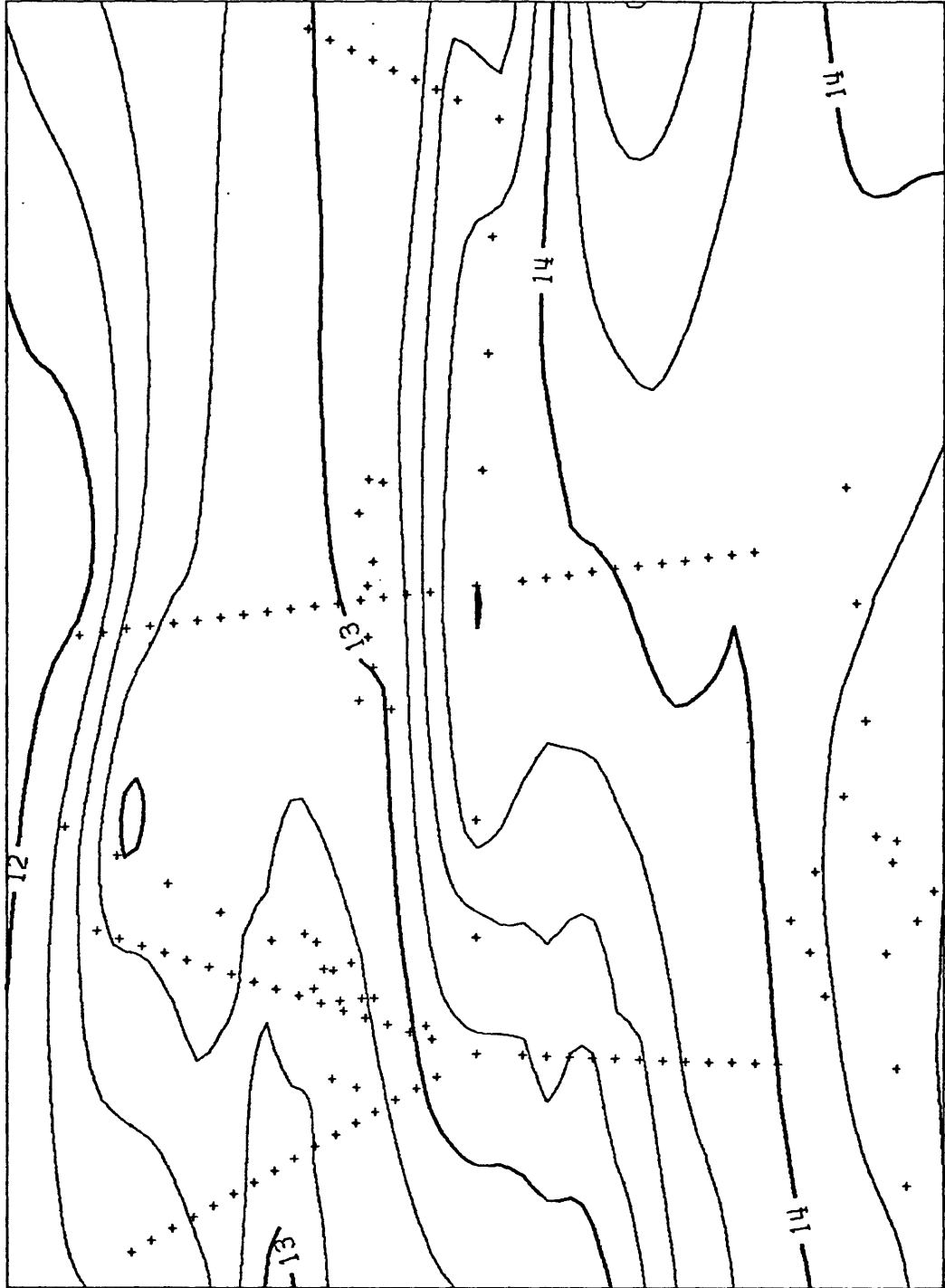
S TOTAL TIME IN SDL CALLS= 15.634 SECONDS.

S (RESOLUTION OF TIMER IS SECS/1000.)

S END OF SUMMARY

STOP

r 08:28 18.771 748



PENJASRAN RI ONISMA STTE

sdl_cellmap annotated terminal session:

print smile.grid

smile.grid

03/26/81 0748.8 pst Thu

```
AAAAAAAAAA
AAAAAAAAAA
AABBAABBAA
AABBAABBAA
AAAAAAAAAA
AAAAAAAAAA
AACAAAACAA
AAACAACAAA
AAAACCAAAA
AAAAAAAAAA
```

r 07:48 0.070 6

sdl_cellmap

Data matrix of digits and upper case characters

Data matrix file name: smile.grid

inmxcl: 10 [10 columns]

inmyrw: 10 [10 rows]

Do you want to plot data points? (y or n): n

xpglen: 8.5 [page size in inches]

ypglen: 11

xwnlen: 5 [window size in inches; determines map scale]

ywnlen: 5

caption (in quotes): "smile"

iaxorn (1-4): 2 [this is the orientation in which map looks like grid]

distance (in no. of cells) between standard labels: 3

Do you want to change anything before plotting? (y or n): n

device (t, v, h, p, or?): p

```
I EXECUTION REPORT LOG
I SURFACE DISPLAY LIBRARY
I VERSION SDL1, RELEASE 3.0I
I DYNAMIC GRAPHICS, INC.
I DATE THURSDAY 26 MARCH 1981
I TIME 7.51.42 A.M.
I
```

```
C 1 BGNPLT
C 2 SETXYG INMXCL= 10 XGDMIN= 0.00000 XGDMAX= 5.0000
+ INMYRW= 10 YGDMIN= 0.00000 YGDMAX= 5.0000
C 3 SETORN IFLORN= 1 IAXORN= 2
C 4 SETPAG XPGLEN= 8.5000 YPGLEN= 11.000 XPGOFF= 1.0000
+ YPGOFF= 1.0000
C 5 SETWIN XWNLEN= 5.0000 YWNLEN= 5.0000 XWNOFF= 1.7500
+ YWNOFF= 3.0000
C 6 CLCTSZ ITPTXT= 6 ICHARR= *ARRAY* ICHCNT= 1
+ TXTLEN=*RESULT* TXTHGT=*RESULT*
+ M21 STRING PARAMETER 2=A
```

```

+      M23 RESULT PARAMETER 4=  .20000  .
+      M23 RESULT PARAMETER 5=  .35000  .
maximum label size (itptxt) that fits in one cell is 6
itptxt (0-6): 3          [I want a smaller label; 0 = no label]
C  7 CLCTSZ ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      5
+      TXTLEN=*RESULT*      TXTHGT=*RESULT*
+      M21 STRING PARAMETER 2=SMILE
+      M23 RESULT PARAMETER 4=  1.4000  .
+      M23 RESULT PARAMETER 5=  .35000  .
C  8 CAPTXT IFLSID=      1      OFFCAP=  1.3250      IFLJST=      2
+      ITPTXT=      6      ICHARR= *ARRAY*      ICHCNT=      5
+      M21 STRING PARAMETER 5=SMILE
C 103 SETRPL IFLRPL=      3      IUNRPL=      6

```

Do you want another plot? (y or n): n

C 104 ENDPLT

The SDL independent plot file is finished

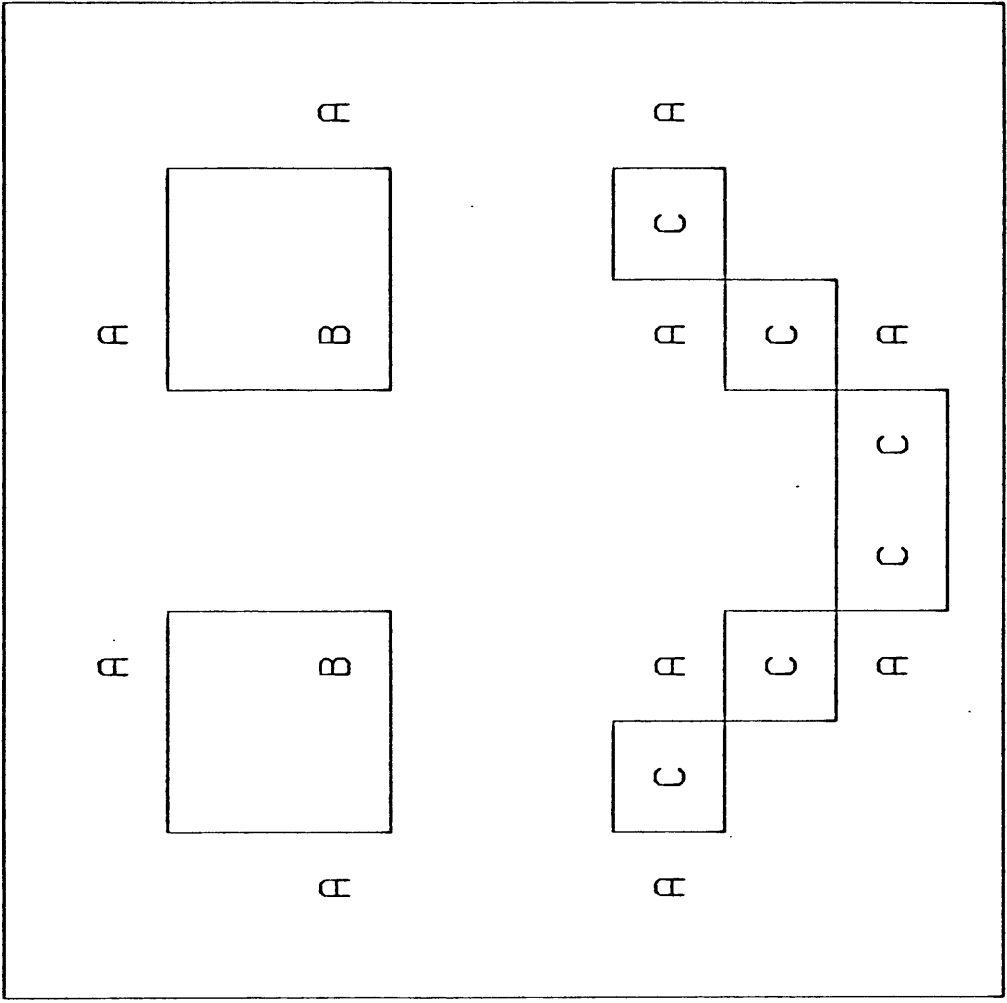
```

S
S      .
S      SUMMARY OF EXECUTION
S      SDL TOTAL CALLS=      104
S      CALLS WITH ERRORS=      0
S      SDL TOTAL FRAMES=      1
S      FRAMES WITH ERRORS=      0
S      TOTAL TIME IN SDL CALLS=  .76600      SECONDS.
S      (RESOLUTION OF TIMER IS SECS/1000.)
S      END OF SUMMARY

```

STOP

r 07:53 3.054 471



SMILE

dat sph:

(data sphere radii)

distance from the center of the surface to the eyepoint (controls the amount of perspective distortion introduced), measured in data sphere radii. a data sphere radius is the distance from the center of the surface to an imaginary sphere surrounding the surface.

permitted values: (real) 1.0 to 1,000.0

used in routines: vs/1-setvue

iaxorn:

(axis orientation)

corner of the data grid in which both the minimum x and minimum y values occur: 1 = lower left (standard orientation); 2 = upper left; 3 = upper right; 4 = lower right. the default is lower left (1).

permitted values: (integer) 1 to 4

used in routines: ed/3-setorn

idcplc:

(decimal places)

number of decimal places in the numeric value to be drawn. if idcplc=0, the decimal point will be the last character in the plotted number. if idcplc=-1, the decimal point will not be plotted.

permitted values: (integer) -1 to 20

used in routines: cs/1-setcon

cs/4-setlhl

ma/1-mapnt

ma/4-maplab

va/1-vuepnt

va/4-vuelab

inmxcl:

(number x-columns)

number of x-columns in the elevation grid.

permitted values: (integer) 2 to 10,000 (when used with ed/4-setnsp the maximum is reduced to 200)

used in routines: ed/1-setxyg

ed/4-setnsp

inmyrw:

(number y-rows)

number of y-rows in the elevation grid.

permitted values: (integer) 2 to 10,000 (when used with ed/4-setnsp, the maximum is reduced to 200)

used in routines: ed/1-setxyg

ed/4-setnsp

ithbld:

(i-th bold)

bold contour level increment. every ithbld contour line, counting from a reference level (zvlref), will be drawn as a bold line. if ithbld=0, then no contour lines are drawn bold.

permitted values: (integer) 0 to 10,000

used in routines: cs/1-setcon

ithlab:

(i-th labeled)

contour level labeling increment. every ithlab contour line, counting from a reference level (zvlref), will be labeled with the elevation it represents. if ithlab=0, then no contour levels are labeled.

permitted values: (integer) 0 to 10,000

used in routines: cs/1-setcon

itptxt:

(type text)

text type as shown in the following table:

itptxt	character size (inches)	
	length	height
1	0.04	0.07
2	0.057	0.10
3	0.08	0.14
4	0.12	0.21
5	0.16	0.28
6	0.20	0.35

permitted values: (integer) 1 to 6

used in routines: cs/1-setcon

cs/4-setlhl

ma/1-mappnt

ma/4-maplab

ma/5-mapcap

va/1-vuepnt

va/4-vuelab

va/5-vuecap

pa/2-pagtxt

pa/3-captxt

pa/4-clctsz

phidxy:

(phi degrees x-y)

angle, in degrees, of the line of sight as measured in the x-y plane. it is measured in a counterclockwise direction, where zero degrees is looking from the bottom of the grid.

permitted values: (real) 0.0 to 360.0

used in routines: vs/1-setvue

thdinc:

(theta degrees inclination)

angle of inclination, measured in degrees, between the horizontal plane and the line of sight.

permitted values: (real) -90.0 to 90.0

used in routines: vs/3-setvue

xgdmax:

(x-grid maximum)

maximum x-column location for the elevation grid, specified in data scale units.

permitted values: any valid real number

used in routines: ed/1-setxyg

xgdmin:

(x-grid minimum)

minimum x-column location for the elevation grid, specified in data scale units.

permitted values: any valid real number

used in routines: ed/1-setxyg

xpglen:

(x-page length)

length of plot page, specified in plot measurement units.

permitted values: (real) greater than 0.0

used in routines: pl/1-setpag

xwnlen:

(x-window length)

length of the display window, specified in plot measurement units.

permitted values: (real) greater than 0.0

used in routines: pl/2-setwin

ygdmax:

(y-grid maximum)

maximum y-row value for the elevation grid, specified in data scale units.

permitted values: any valid real number

used in routines: ed/1-setxyg

yghmin:

(y-grid minimum)

minimum y-row value for the elevation grid, specified in data scale units.

permitted values: any valid real number

used in routines: ed/1-setxyg

ypglen:
(y-page length)
height of the plot page, specified in plot measurement units.
permitted values: (real) greater than 0.0
used in routines: pl/1-setpag

ywnlen:
(y-window length)
height of the display window, specified in plot measurement units.
permitted values: (real) greater than 0.0
used in routines: pl/2-setwin

zgdnul:
(z-grid null)
value to be recognized in the elevation data grid (within a tolerance of epsnul) as the "null data value" which indicates parts of the surface to be left blank. the default is not to recognize any null data value.
permitted values: any valid real number
used in routines: ed/2-setnul

zprfac:
(z-proportion factor)
z-axis proportion factor.
permitted values: (real) greater than 0.0
used in routines: ed/5-setprp

zvlint:
(z-value interval)
value for interval between adjacent contour levels, specified in data scale units.
permitted values: (real) greater than 0.0
used in routines: cs/1-setcon

zvlref:
(z-value reference)
reference elevation value for contouring, specified in data scale units. a contour line will be drawn at this elevation and at intervals of zvlint throughout the full range of the data.
permitted values: any valid real number.
used in routines: cs/1-setcon

COMPILATION LISTING OF sdl_gridder

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 13:41.2 pst Thu
 Options: ans166 table map

Main Program

```

1  external sdl_grid_ (descriptors)
2  external asr (descriptors)
3  external query_ (descriptors)
4  external close_file (descriptors)
5  external ioa_ (descriptors)
6  external read_list_$prompt (descriptors)
7  parameter (idmxcl=400, idmyrw=400, idmwrk=160000, idmxyz=1000)
8  dimension elvarr(idmxcl, idmyrw), iwkarr(idmwrk)
9  dimension xptarr(idmxyz), yptarr(idmxyz), zptarr(idmxyz)
10 character infile*168, outfile*168, ans*(10)
11 common /sgl_com1/ elvarr, xptarr, yptarr, zptarr
12 common /sgl_com2/ iwkarr
13 save icall_sdl
14 data icall_sdl/0/
15 c
16 c
17 if (icall_sdl .eq. 1) goto 5
18 call asr('>iml>SDL', '-after', 'working_dir')
19 call ioa_ ('See: sdl_names.info or the Surface Gridding Library manual for definitions of variable names.')
20 call ioa_ ('help sdl_names -section <name>')
21 icall_sdl=1
22 c
23 5 call close_file('-all')
24 inxyz=0
25 iaxorn=1
26 zgdnu1=-9999.
27 c
28 call read_list_$prompt('input data file (x,y,z): ', infile)
29 open(10, form='formatted', mode='in', file=infile, access='sequential', err=800)
30 call read_list_$prompt('output grid file: ', outfile)
31 c test opening of outfile
32 open(20, form='formatted', mode='out', file=outfile, access='sequential', err=805)
33 close(20)
34 c
35 c
36 do 10 i=1, idmxyz
37 read(10, 1000, end=15, err=810) xptarr(i), yptarr(i), zptarr(i)
38 inxyz=i
39 10 continue
40 15 call ioa_ ('^1 data points read.', inxyz)
41 close(10)
42 c
43 call ioa_ ('enter grid parameters')
44 call read_list_$prompt('inmxcl: ', inmxcl, inmyrw: ', inmyrw)
45 call read_list_$prompt('xgdmn: ', xgdmn, 'xgdmax: ', xgdmax)
46 call read_list_$prompt('ygdmin: ', ygdmin, 'ygdmax: ', ygdmax)

```



```

47 c
48 call query_("grid report to terminal? ",ans(1))
49 if(ans(1).eq. "y") goto 20
50 call ioa_("grid report in file grid_report")
51 open(6,form="formatted",mode="out",file="grid_report",access="sequential",err=820)
52 continue
53 c
54 c begin gridding
55 c
56 call bgngrd
57 call setgrd(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
58 call clegrd(xptarr,yptarr,inmxyz,elvarr,ldmxcl,ldmyrw,lvkarr,ldmwrk)
59 call endgrd
60 c
61 if(ans(1).eq. "n") close(6)
62 icode=1
63 call sdl_grid_(outfile,inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnu1,elvarr,icode)
64 if(icode.ne. 0) call ioa_("unable to write grid file")
65 goto 999
66 c
67 c error messages
68 c
69 800 call ioa_("unable to open file `a",infile)
70 goto 999
71 805 call ioa_("unable to open file `a",outfile)
72 goto 999
73 810 call ioa_("error reading record `i of `a",inmxyz+1,infile)
74 goto 999
75 820 call ioa_("unable to open file grid_report")
76 goto 999
77 999 stop
78 c
79 1000 format(v)
80 end

```

COMPILATION LISTING OF sdl_conmap

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1341.3 pst Thu
 Options: ansi66 table map

Main Program

```

1 external sdl_grid_ (descriptors)
2 external asr (descriptors)
3 external query_ (descriptors)
4 external close_file (descriptors)
5 external ioa_ (descriptors)
6 external read_list_$prompt (descriptors)
7 external clcxt_ (descriptors)
8 external upper_case_ (descriptors)
9 parameter(idmxc1=400,idmyrw=400,idmwrk=11000)
10 dimension elvarr(idmxc1,idmyrw),iwkarr(idmwrk),icharr(200)
11 dimension xwnloc(1000),ywnloc(1000)
12 character file#168,ans#1(10),string#200
13 character dvce#1
14 logical status
15 common /dvcecom/dvce
16 common /sdl_com/ elvarr,iwkarr
17 save icall_sdl
18 data icall_sdl/0/
19 c
20 c
21 string = " "
22 status=.false.
23 ans(3)="y"
24 taxorn = 1
25 c
26 c
27 if (icall_sdl .eq. 1) goto 10
28 call asr(">iml>SDL","-after","working_dir")
29 call ioa_"If using a Tektronix, the execution log is in sdl_log."
30 call ioa_"See: sdl_names.info or the SDL manual for definitions of variable names."
31 call ioa_"help sdl_names -section <name>"
32 call ioa_"CAUTION: this program overwrites existing default plot files"
33 call ioa_"To save plot files, rename them before execution.~/")
34 icall_sdl=1
35 c
36 10 call sdl_reset
37 call close_file("-all")
38 goto 30
39 c
40 c get grid and grid parameters
41 c
42 20 call query_("New grid data? ",ans(6))
43 if(ans(6) .eq. "n") goto 41
44 30 call read_list_$prompt("Data matrix file name: ",file)
45 icode=0
46 c optional subroutine acquisition of grid and grid parameters from standard grid file

```

```

47 call sdi_grid_(file,inmxcl,inmyrw,xgdmn,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul,elvarr,icode)
48 if(icode .gt. 1) goto 999
49 if(icode .eq. 0) goto 43
50 c
51 c grid not in standard form; read grid, prompt for parameters
52 open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
53 call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
54 if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
55 do 40 i=1,inmyrw
56 read(10,1000) (elvarr(j,i),j=1,inmxcl)
57 40 continue
58 close(10)
59 c
60 c 41 call read_list_$prompt("iaxorn (1-4): ",iaxorn)
61 c
62 c
63 c 43 call query_("null data? ",ans(1))
64 if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ",zgdnul)
65 c
66 c plot data points option
67 c
68 call query_("Do you want to plot data points? ",ans(5))
69 if (ans(5) .eq. "n") go to 50
70 call read_list_$prompt("data points (x,y) file name: ",file)
71 open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
72 do 45 i=1,1000
73 read(10,1000,ends=48)xwnloc(i),ywnloc(i)
74 nloc=i
75 continue
76 close(10)
77 call loa_("^i data points read.",nloc)
78 c
79 c
80 call read_list_$prompt("xpflen: ",xpflen,"ypflen: ",ypflen)
81 call read_list_$prompt("xwnlen: ",xwnlen,"ywnlen: ",ywnlen)
82 xwnoff=(xpflen-xwnlen)/2.
83 ywnoff=(ypflen-ywnlen)/2.
84 call read_list_$prompt("caption (in quotes): ",string)
85 call upper_case_(string)
86 c
87 if(icode .eq. 0) goto 55
88 if(ans(5) .eq. "n") xgdmn=0.
89 if(ans(5) .eq. "n") ygdmin=0.
90 if(ans(5) .eq. "n") xgdmax=xwnlen
91 if(ans(5) .eq. "n") ygdmax=ywnlen
92 c
93 if(ans(5) .eq. "y") call read_list_$prompt("xgdmn: ",xgdmn,"xgdmax: ",xgdmax)
94 if(ans(5) .eq. "y") call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
95 55 continue
96 c
97 call loa_("contour control variables")
98 call read_list_$prompt("zvlref: ",zvlref,"zvlint: ",zvlint,"ithbld: ",ithbld)
99 call read_list_$prompt("ithlab: ",ithlab,"itptxt (1-6): ",itptxt,"idoplc: ",idoplc)
100 call query_("depression ticks? ",ans(2))

```

```

101 c
102 c
103 c
104 call query_ ("~/ Do you want to change anything before plotting? ",ans(6))
105 if (ans(6) .eq. "n") go to 60
106 call ica_ ("~/Type a comma for values that remain the same.")
107 go to 20
108 c
109 c
110 c
111 begin plot
112
113 60 continue
114 if(ans(3) .eq. "y") call sdl_device_
115 if(ans(3) .eq. "n") call sdl_device_ $preset
116 if(.not. status) call sdl_reset
117 if(.not. status) call bgnpit
118 status=.true.
119 close(20)
120 if (dvce .eq. "t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
121 if(dvce .eq. "t") call setrpl(3,20)
122 call setxg(inmxcl,xgdmin,xgdmax,inmyrw,ygdmin,ygdmax)
123 if (ans(1) .eq. "y") call setnul(1,zgdnul,1.0e-5)
124 if (ans(1) .eq. "n") call setnul(0,-9999.,1.0e-5)
125 call setorn(1,iaxorn)
126 call setpag(xpglen,ypglen,1.0,1.0)
127 call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
128 call setcon(zviref,zvrint,ithblid,ithlab,iptxt,idepic)
129 if (ans(2) .eq. "y") call settik(1,1.0,0.4,0.1)
130 if (ans(2) .eq. "n") call settik(0,1.0,0.4,0.1)
131 call conmap(rlvarr,idmxcl,idmyrw,iwkarr,idmvrk)
132 if (ans(5) .eq. "n") go to 65
133 if(dvce .ne. "t") call setrpl(2,6)
134 do 63 i=1,nloc
135 call mappnt(xwnloc(1),ywnloc(1),4,1,2,0.0)
136 continue
137 if(dvce .ne. "t") call setrpl(3,6)
138 c
139 c
140 c
141 setup caption
142 65 call clctxt_(string,ichent,icharr)
143 if (ichent .eq. 0) goto 90
144 check and set size of letters (default size = 6)
145 ntptxt = 6
146 70 call clctsz(ntptxt,icharr,ichent,txhlen,txthgt)
147 if (txhlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
148 ntptxt = ntptxt-1
149 if (ntptxt .le. 0) go to 80
150 go to 70
151 80 if (ntptxt .le. 0) call ica_("caption will not fit in space provided")
152 offcap=(ywnoff-txthgt)/2.
153 if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichent)
154 90 continue
155 if(dvce .eq. "t") call endplt
156 if(dvce .eq. "t") status=.false.
157 c
158 c
159 c
160 c
161 c
162 c
163 c
164 c
165 c
166 c
167 c
168 c
169 c
170 c
171 c
172 c
173 c
174 c
175 c
176 c
177 c
178 c
179 c
180 c
181 c
182 c
183 c
184 c
185 c
186 c
187 c
188 c
189 c
190 c
191 c
192 c
193 c
194 c
195 c
196 c
197 c
198 c
199 c
200 c
201 c
202 c
203 c
204 c
205 c
206 c
207 c
208 c
209 c
210 c
211 c
212 c
213 c
214 c
215 c
216 c
217 c
218 c
219 c
220 c
221 c
222 c
223 c
224 c
225 c
226 c
227 c
228 c
229 c
230 c
231 c
232 c
233 c
234 c
235 c
236 c
237 c
238 c
239 c
240 c
241 c
242 c
243 c
244 c
245 c
246 c
247 c
248 c
249 c
250 c
251 c
252 c
253 c
254 c
255 c
256 c
257 c
258 c
259 c
260 c
261 c
262 c
263 c
264 c
265 c
266 c
267 c
268 c
269 c
270 c
271 c
272 c
273 c
274 c
275 c
276 c
277 c
278 c
279 c
280 c
281 c
282 c
283 c
284 c
285 c
286 c
287 c
288 c
289 c
290 c
291 c
292 c
293 c
294 c
295 c
296 c
297 c
298 c
299 c
300 c
301 c
302 c
303 c
304 c
305 c
306 c
307 c
308 c
309 c
310 c
311 c
312 c
313 c
314 c
315 c
316 c
317 c
318 c
319 c
320 c
321 c
322 c
323 c
324 c
325 c
326 c
327 c
328 c
329 c
330 c
331 c
332 c
333 c
334 c
335 c
336 c
337 c
338 c
339 c
340 c
341 c
342 c
343 c
344 c
345 c
346 c
347 c
348 c
349 c
350 c
351 c
352 c
353 c
354 c
355 c
356 c
357 c
358 c
359 c
360 c
361 c
362 c
363 c
364 c
365 c
366 c
367 c
368 c
369 c
370 c
371 c
372 c
373 c
374 c
375 c
376 c
377 c
378 c
379 c
380 c
381 c
382 c
383 c
384 c
385 c
386 c
387 c
388 c
389 c
390 c
391 c
392 c
393 c
394 c
395 c
396 c
397 c
398 c
399 c
400 c
401 c
402 c
403 c
404 c
405 c
406 c
407 c
408 c
409 c
410 c
411 c
412 c
413 c
414 c
415 c
416 c
417 c
418 c
419 c
420 c
421 c
422 c
423 c
424 c
425 c
426 c
427 c
428 c
429 c
430 c
431 c
432 c
433 c
434 c
435 c
436 c
437 c
438 c
439 c
440 c
441 c
442 c
443 c
444 c
445 c
446 c
447 c
448 c
449 c
450 c
451 c
452 c
453 c
454 c
455 c
456 c
457 c
458 c
459 c
460 c
461 c
462 c
463 c
464 c
465 c
466 c
467 c
468 c
469 c
470 c
471 c
472 c
473 c
474 c
475 c
476 c
477 c
478 c
479 c
480 c
481 c
482 c
483 c
484 c
485 c
486 c
487 c
488 c
489 c
490 c
491 c
492 c
493 c
494 c
495 c
496 c
497 c
498 c
499 c
500 c
501 c
502 c
503 c
504 c
505 c
506 c
507 c
508 c
509 c
510 c
511 c
512 c
513 c
514 c
515 c
516 c
517 c
518 c
519 c
520 c
521 c
522 c
523 c
524 c
525 c
526 c
527 c
528 c
529 c
530 c
531 c
532 c
533 c
534 c
535 c
536 c
537 c
538 c
539 c
540 c
541 c
542 c
543 c
544 c
545 c
546 c
547 c
548 c
549 c
550 c
551 c
552 c
553 c
554 c
555 c
556 c
557 c
558 c
559 c
560 c
561 c
562 c
563 c
564 c
565 c
566 c
567 c
568 c
569 c
570 c
571 c
572 c
573 c
574 c
575 c
576 c
577 c
578 c
579 c
580 c
581 c
582 c
583 c
584 c
585 c
586 c
587 c
588 c
589 c
590 c
591 c
592 c
593 c
594 c
595 c
596 c
597 c
598 c
599 c
600 c
601 c
602 c
603 c
604 c
605 c
606 c
607 c
608 c
609 c
610 c
611 c
612 c
613 c
614 c
615 c
616 c
617 c
618 c
619 c
620 c
621 c
622 c
623 c
624 c
625 c
626 c
627 c
628 c
629 c
630 c
631 c
632 c
633 c
634 c
635 c
636 c
637 c
638 c
639 c
640 c
641 c
642 c
643 c
644 c
645 c
646 c
647 c
648 c
649 c
650 c
651 c
652 c
653 c
654 c
655 c
656 c
657 c
658 c
659 c
660 c
661 c
662 c
663 c
664 c
665 c
666 c
667 c
668 c
669 c
670 c
671 c
672 c
673 c
674 c
675 c
676 c
677 c
678 c
679 c
680 c
681 c
682 c
683 c
684 c
685 c
686 c
687 c
688 c
689 c
690 c
691 c
692 c
693 c
694 c
695 c
696 c
697 c
698 c
699 c
700 c
701 c
702 c
703 c
704 c
705 c
706 c
707 c
708 c
709 c
710 c
711 c
712 c
713 c
714 c
715 c
716 c
717 c
718 c
719 c
720 c
721 c
722 c
723 c
724 c
725 c
726 c
727 c
728 c
729 c
730 c
731 c
732 c
733 c
734 c
735 c
736 c
737 c
738 c
739 c
740 c
741 c
742 c
743 c
744 c
745 c
746 c
747 c
748 c
749 c
750 c
751 c
752 c
753 c
754 c
755 c
756 c
757 c
758 c
759 c
760 c
761 c
762 c
763 c
764 c
765 c
766 c
767 c
768 c
769 c
770 c
771 c
772 c
773 c
774 c
775 c
776 c
777 c
778 c
779 c
780 c
781 c
782 c
783 c
784 c
785 c
786 c
787 c
788 c
789 c
790 c
791 c
792 c
793 c
794 c
795 c
796 c
797 c
798 c
799 c
800 c
801 c
802 c
803 c
804 c
805 c
806 c
807 c
808 c
809 c
810 c
811 c
812 c
813 c
814 c
815 c
816 c
817 c
818 c
819 c
820 c
821 c
822 c
823 c
824 c
825 c
826 c
827 c
828 c
829 c
830 c
831 c
832 c
833 c
834 c
835 c
836 c
837 c
838 c
839 c
840 c
841 c
842 c
843 c
844 c
845 c
846 c
847 c
848 c
849 c
850 c
851 c
852 c
853 c
854 c
855 c
856 c
857 c
858 c
859 c
860 c
861 c
862 c
863 c
864 c
865 c
866 c
867 c
868 c
869 c
870 c
871 c
872 c
873 c
874 c
875 c
876 c
877 c
878 c
879 c
880 c
881 c
882 c
883 c
884 c
885 c
886 c
887 c
888 c
889 c
890 c
891 c
892 c
893 c
894 c
895 c
896 c
897 c
898 c
899 c
900 c
901 c
902 c
903 c
904 c
905 c
906 c
907 c
908 c
909 c
910 c
911 c
912 c
913 c
914 c
915 c
916 c
917 c
918 c
919 c
920 c
921 c
922 c
923 c
924 c
925 c
926 c
927 c
928 c
929 c
930 c
931 c
932 c
933 c
934 c
935 c
936 c
937 c
938 c
939 c
940 c
941 c
942 c
943 c
944 c
945 c
946 c
947 c
948 c
949 c
950 c
951 c
952 c
953 c
954 c
955 c
956 c
957 c
958 c
959 c
960 c
961 c
962 c
963 c
964 c
965 c
966 c
967 c
968 c
969 c
970 c
971 c
972 c
973 c
974 c
975 c
976 c
977 c
978 c
979 c
980 c
981 c
982 c
983 c
984 c
985 c
986 c
987 c
988 c
989 c
990 c
991 c
992 c
993 c
994 c
995 c
996 c
997 c
998 c
999 c
1000 c

```

```

155 o      call query_ ("~/ Do you want another plot?",ans(3))
156      if (ans(3) .eq. "n") goto 999
157      call query_ ("~/ Do you want to change device? ",ans(3))
158      if (ans(3) .eq. "y" .and. status) call endplt
159      if (ans(3) .eq. "y" .and. status) status=.false.
160      if (ans(3) .eq. "n" .and. dvoe .ne. "t") call nrtfrm
161      call query_ ("~/ Do you want to change anything else?",ans(6))
162      if (ans(6) .eq. "n") go to 60
163      call ioa_ ("~/Type a comma for values that remain the same.")
164      go to 20
165 o
166 o
167 o error messages
168 o
169      810 call ioa_ ("Data matrix too large. Maximum size is 400 by 400")
170      go to 999
171      820 call ioa_ ("sdl_conmap: unable to open file sdl_log.")
172      go to 999
173      830 call ioa_ ("sdl_conmap: unable to open file ^a",file)
174      999 if (status) call endplt
175      status=.false.
176      call sdl_reset
177      stop
178
179      1000 format(v)
180      end

```

COMPILATION LISTING OF sdl_convue

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1341.1 pst Thu
 Options: ansi66 table map

Main Program

```

1  external sdl_grid_(descriptors)
2  external asr_(descriptors)
3  external query_(descriptors)
4  external close_file_(descriptors)
5  external ioa_(descriptors)
6  external read_list_$prompt_(descriptors)
7  external clcxtxt_(descriptors)
8  external upper_case_(descriptors)
9  parameter(idmxcl=400,idmyrw=400,idmwrk=100000)
10 parameter(xpglen=10.,ypglen=10.,xwnlen=8.,ywnlen=8.,xwnoff=1.,ywnoff=1.)
11 logical status
12 common /dvcecom/dvce
13 common /sdl_vue/ elvarr, iwkarr
14 dimension xvloc(idmxcl,idmyrw), iwkarr(idmwrk), icharr(200)
15 dimension yvloc(1000), yvuloc(1000), zvuloc(1000)
16 character file*168, ans#1(10), string*200, dvce#1
17 save icall_sdl
18 data icall_sdl/0/
19 c
20 c
21 status=.false.
22 string = " "
23 ans(3)="y"
24 datsph=10.0
25 thdinc=20.0
26 phidxy=325.0
27 faxorn = 1
28 c
29 c
30
31 if (icall_sdl.eq. 1) goto 10
32 call asr(">iml>SDL", "-after", "working_dir")
33 call ioa_("If using a Tektronix, the execution log is in sdl_log.")
34 call ioa_("See: sdl_names.info or the SDL manual for definitions of variable names.")
35 call ioa_("help sdl_names-section <name>")
36 call ioa_("/CAUTION: this program overwrites existing default plot files")
37 icall_sdl=1
38 c
39
40 10 call sdl_reset
41 call close_file("all")
42 goto 30
43 c
44 c
45 get grid and grid parameters
46
47 20 call query_("New grid data? ", ans(6))
48 if(ans(6) .eq. "n") goto 41

```

```

47 30 call read_list_$prompt("Data matrix file name: ",file)
48 icode=0
49 c optional subroutine acquisition of grid and grid parameters from standard grid file
50 call sdi_grid_(file,inmxcl,inmyrw,xgdmn,ygdmax,ygdmin,ixorn,zgdnul,elvarr,icode)
51 if(icode .gt. 1) goto 999
52 if(icode .eq. 0) goto 43
53 c
54 c grid not in standard form; read grid, prompt for parameters
55 open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
56 call read_list_$prompt("inmxcl: ",inmxcl,"inmyrw: ",inmyrw)
57 if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
58 do 40 i=1,inmyrw
59 read(10,1000) (elvarr(j,i),j=1,inmxcl)
60 continue
61 close(10)
62 c
63 41 call read_list_$prompt("xgdmn: ",xgdmn,"ygdmax: ",ygdmax)
64 call read_list_$prompt("ygdmin: ",ygdmin,"ygdmax: ",ygdmax)
65 call read_list_$prompt("ixorn (1-4): ",ixorn)
66 c
67 c
68 43 call query_("(null data? ",ans(1))
69 if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ",zgdnul)
70 c
71 c plot data points option
72 c
73 call query_("(Do you want to plot data points? ",ans(5))
74 if (ans(5) .eq. "n") go to 50
75 call read_list_$prompt("data points (x,y,x) file name: ",file)
76 open(10,form="formatted",mode="in",file=file,access="sequential",err=830)
77 do 45 i=1,1000
78 read(10,1000,end=48)xvuloc(i),yvuloc(i),zvuloc(i)
79 nloc=i
80 continue
81 close(10)
82 call loa_("i data points read.",nloc)
83 c
84 c
85 50 continue
86 call read_list_$prompt("caption (in quotes): ",string)
87 call upper_case_(string)
88 c
89 call loa_("contour control variables")
90 call read_list_$prompt("zvlref: ",zvlref,"zvlint: ",zvlint,"ithblid: ",ithblid)
91 call read_list_$prompt("ithlab: ",ithlab,"itptxt (1-6): ",itptxt,"idople: ",idople)
92 call query_("(Scale elvarr? ",ans(2))
93 if (ans(2) .eq. "y") call read_list_$prompt("zprfac: ",zprfac)
94 call loa_("default perspective is: datsph=f, thdinc=f, phidxy=f", datsph,thdinc,phidxy)
95 call query_("(changes? ",ans(7))
96 if (ans(7) .eq. "y") call read_list_$prompt("datsph: ",datsph,"thdinc: ",thdinc,"phidxy: ",phidxy)
97 call query_("(depression ticks? ",ans(4))
98 c
99 call query_ ("?/ Do you want to change anything before plotting? ",ans(6))
100 if (ans(6) .eq. "n") go to 60

```

```

101 call loa_ ("~/Type a comma for values that remain the same.")
102 go to 20
103 c
104 c begin plot
105 c
106 continue
107 if(ans(3).eq. "y") call sdl_device_
108 if(ans(3).eq. "n") call sdl_device_$preset
109 if(.not. status) call sdl_reset
110 if(.not. status) call bgndplt
111 status=.true.
112 close(20)
113 if(dvce.eq. "t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
114 if(dvce.eq. "t") call setrpl(3,20)
115 call setxyg(inmxcl,xgdmn,ygdmx,imyrw,ygdmn,ygdmx)
116 if (ans(1).eq. "y") call setnul(1,zgdnul,1.0e-5)
117 if (ans(1).eq. "n") call setnul(0,-9999.,1.0e-5)
118 if (ans(2).eq. "y") call setrpl(1,1.0,1.0,zprfac)
119 call setorn(1,iaorn)
120 call setpag(xpglen,ypglen,1.0,1.0)
121 call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
122 call setcon(zviref,zvlint,ithbid,ithlab,ltptxt,ideple)
123 if (ans(4).eq. "y") call settik(1,1.0,0.1)
124 if (ans(4).eq. "n") call settik(0,1.0,0.1)
125 call setvue(datsph,thdinc,phidxy)
126 call convue(elvarr,idxcl,imyrw,ikarr,ldmwrk)
127 if (ans(5).eq. "n") go to 65
128 if(dvce .ne. "t") call setrpl(2,6)
129 do 63 i=1,nloc
130 call vuepnt(xvuloc(i),yvuloc(i),zvuloc(i),4,1,2,0.0)
131 continue
132 if(dvce .ne. "t") call setrpl(3,6)
133 c
134 c setup caption
135 call clctxt_(string,ichent,icharr)
136 if (ichent .eq. 0) goto 90
137 c check and set size of letters (default size = 6)
138 ntptxt = 6
139 call clctsz(ntptxt,icharr,ichent,txtlen,txthgt)
140 if (txtlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
141 ntptxt = ntptxt-1
142 if (ntptxt .le. 0) go to 80
143 go to 70
144 if (ntptxt .le. 0) call loa_("caption will not fit in space provided")
145 offcap=(ywnoff-txthgt)/2
146 if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichent)
147 continue
148 if(dvce .eq. "t") call endplt
149 if(dvce .eq. "t") status=.false.
150 c
151 c setup for next plot or stop
152 c
153 call query_ ("~/ Do you want another plot?",ans(3))
154 if (ans(3) .eq. "n") goto 999

```



```

155 call query_("&"/ Do you want to change device? ",ans(3))
156 if(ans(3).eq. "y" .and. status) call endplt
157 if(ans(3).eq. "y" .and. status) status=.false.
158 if(ans(3).eq. "n" .and. dvce .ne. "t") call nxtfrm
159 call query_("&"/ Do you want to change anything else?",ans(6))
160 if (ans(6) .eq. "n") go to 60
161 call ioa_ ("&"/Type a comma for values that remain the same.")
162 go to 20
163 c
164 c error messages
165 c
166 810 call ioa_("Data matrix too large. Maximum size is 400 by 400")
167 go to 999
168 820 call ioa_("sdl_convue: unable to open file sdl_log.")
169 go to 999
170 830 call ioa_("sdl_convue: unable to open file ^a",file)
171 999 if(status) call endplt
172 status=.false.
173 call sdl_reset
174 stop
175
176 1000 format(v)
177 end

```

COMPILATION LISTING OF sdl_mshvue

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1341.2 pst Thu
 Options: ansi66 table map

Main Program

```

1  external sdl_grid_ (descriptors)
2  external asr (descriptors)
3  external query_ (descriptors)
4  external close_file (descriptors)
5  external ioa_ (descriptors)
6  external read_list_$prompt (descriptors)
7  external clctxt_ (descriptors)
8  external upper_case_ (descriptors)
9  parameter(idmxcl=400,idmyrw=400,idmwrk=100000)
10 parameter(xpglen=10.,ypglen=10.,xwnlen=8.,ywnlen=8.,xwnoff=1.,ywnoff=1.)
11 logical status
12 common /dvcecom/dvce
13 common /sdl_msh/ elvarr, iwkar
14 dimension elvarr(idmxcl,idmyrw), iwkar(idmwrk), icharr(200)
15 dimension xvuloc(1000), yvuloc(1000), zvuloc(1000)
16 character file*168, ans*1(10), string*200, dvce*1
17 save icall_sdl
18 data icall_sdl/0/
19 c
20 c
21 status=.false.
22 string = " "
23 ans(3)="y"
24 datsph=10.0
25 thdinc=20.0
26 phidxy=325.0
27 iaxorn = 1
28 c
29 c
30 if (icall_sdl .eq. 1) goto 10
31 call asr(">iml>SDL", "-after", "working_dir")
32 call ioa_(">If using a Tektronix, the execution log is in sdl_log.*")
33 call ioa_(">See: sdl_names.info or the SDL manual for definitions of variable names.*")
34 call ioa_(">help sdl_names -section <name>")
35 call ioa_(">CAUTION: this program overwrites existing default plot files")
36 call ioa_(">to save plot files, rename them before execution.*")
37 icall_sdl=1
38 c
39 10 call sdl_reset
40 call close_file("#-all")
41 goto 30
42 c
43 c get grid and grid parameters
44 c
45 20 call query_(">New grid data? ", ans(6))
46 if(ans(6) .eq. "n") goto 41

```

```

47 30 call read_list_$prompt("Data matrix file name: ", file)
48   icode=0
49 optional subroutine acquisition of grid and grid parameters from standard grid file
50 call sdl_grid_(file, inmxcl, inmyrw, xgdmn, xgdmax, ygdmin, ygdmax, iaxorn, zgdnul, elvarr, icode)
51 if(icode .gt. 1) goto 999
52 if(icode .eq. 0) goto 43
53 c
54 c grid not in standard form; read grid, prompt for parameters
55 open(10, form="formatted", mode="in", file=file, access="sequential", err=830)
56 call read_list_$prompt("inmxcl: ", inmxcl, inmyrw: ", inmyrw)
57 if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
58 do 40 i=1, inmyrw
59 read(10, 1000) (elvarr(j,i), j=1, inmxcl)
60 continue
61 close(10)
62 c
63 c 41 call read_list_$prompt("xgdmn: ", xgdmn, "xgdmax: ", xgdmax)
64 call read_list_$prompt("ygdmin: ", ygdmin, "ygdmax: ", ygdmax)
65 call read_list_$prompt("iaxorn (1-4): ", iaxorn)
66 c
67 c
68 c 43 call query_("(null data? ", ans(1))
69 if (ans(1) .eq. "y" .and. icode .eq. 1) call read_list_$prompt("zgdnul: ", zgdnul)
70 c
71 c plot data points option
72 c
73 call query_("(Do you want to plot data points? ", ans(5))
74 if (ans(5) .eq. "n") go to 50
75 call read_list_$prompt("data points (x,y,x) file name: ", file)
76 open(10, form="formatted", mode="in", file=file, access="sequential", err=830)
77 do 45 i=1, 1000
78 read(10, 1000, end=48) xvuloc(i), yvuloc(i), zvuloc(i)
79 nloc=i
80 continue
81 close(10)
82 call loa_( "i data points read.", nloc)
83 c
84 c
85 50 continue
86 call read_list_$prompt("caption (in quotes): ", string)
87 call upper_case_(string)
88 call query_("scale elvarr? ", ans(2))
89 if (ans(2) .eq. "y") call read_list_$prompt("zprfac: ", zprfac)
90 call loa_( "default perspective is: datsph=^f, thdinc=^f, phidxy=^f", datsph, thdinc, phidxy)
91 call query_("changes? ", ans(7))
92 if (ans(7) .eq. "y") call read_list_$prompt("datsph: ", datsph, "thdinc: ", thdinc, "phidxy: ", phidxy)
93 c
94 call query_ ("~/ Do you want to change anything before plotting? ", ans(6))
95 if (ans(6) .eq. "n") go to 60
96 call loa_ ("~/Type a comma for values that remain the same.")
97 go to 20
98 c
99 c begin plot
100 c

```

```

101 continue
102 if(ans(3).eq."y") call sdl_device_
103 if(ans(3).eq."n") call sdl_device_$preset
104 if(.not. status) call sdl_reset
105 if(.not. status) call bgnpit
106 status=.true.
107 close(20)
108 if(dvce.eq."t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
109 if(dvce.eq."t") call setrpl(3,20)
110 call setxg(i,inxcl,xgdmn,igdmax,inxrw,ygdmin,ygdmax)
111 if (ans(1).eq."y") call setnul(1,zgdnul,1.0e-5)
112 if (ans(1).eq."n") call setnul(0,-9999.,1.0e-5)
113 if (ans(2).eq."y") call setprp(1,1.0,1.0,zprfac)
114 call setorn(1,iaxorn)
115 call setpag(xpglen,yglen,1.0,1.0)
116 call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
117 call setvue(datsph,thdinc,phidxy)
118 call mshvue(elvarr,ldmxcl,ldmyrw,ldmkarr,ldmwrk)
119 if (ans(5).eq."n") go to 65
120 if(dvce.ne."t") call setrpl(2,6)
121 do 63 i=1,nloc
122 call vuepnt(xvuloc(i),yvuloc(i),zvuloc(i),4,1,2,0.0)
123 continue
124 if(dvce.ne."t") call setrpl(3,6)
125 c
126 c setup caption
127 65 call cictxt_(string,ichent,icharr)
128 if (ichent .eq. 0) goto 90
129 c check and set size of letters (default size = 6)
130 ntptxt = 6
131 70 call clctsz(ntptxt,icharr,ichent,txhlen,txthgt)
132 if (txhlen .lt. xwnlen .and. txthgt .lt. ywnoff) go to 80
133 ntptxt = ntptxt-1
134 if (ntptxt .le. 0) go to 80
135 go to 70
136 80 if (ntptxt .le. 0) call ioa_("caption will not fit in space provided")
137 offcap=(ywnoff-txthgt)/2
138 if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichent)
139 continue
140 if(dvce .eq. "t") call endplt
141 if(dvce .eq. "t") status=.false.
142 c
143 c setup for next plot or stop
144 c
145 call query_ ("~/ Do you want another plot?",ans(3))
146 if (ans(3) .eq. "n") goto 999
147 call query_ ("~/ Do you want to change device? ",ans(3))
148 if(ans(3) .eq. "y" .and. status) call endplt
149 if(ans(3) .eq. "y" .and. status) status=.false.
150 if(ans(3) .eq. "n" .and. dvce .ne. "t") call nxtfrm
151 call query_ ("~/ Do you want to change anything else?",ans(6))
152 if (ans(6) .eq. "n") go to 60
153 call ioa_ ("~/Type a comma for values that remain the same.")
154 go to 20

```

```
155 c
156 c error messages
157 c
158 810 call ioa_("Data matrix too large. Maximum size is 400 by 400")
159 go to 999
160 call ioa_("sdl_mshvue: unable to open file sdl_log.")
161 go to 999
162 830 call ioa_("sdl_mshvue: unable to open file 'a',file)
163 999 if(status) call endplt
164 status=.false.
165 call sdl_reset
166 stop
167
168 1000 format(v)
169 end
```

COMPILATION LISTING OF sdl_cellmap

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1341.3 pst Thu
 Options: ansi66 table map

Main Program

```

1  external asr (descriptors)
2  external query_ (descriptors)
3  external close_file (descriptors)
4  external ioa_ (descriptors)
5  external read_list_$prompt (descriptors)
6  external cletxt_ (descriptors)
7  external upper_case_ (descriptors)
8  parameter (idmxcl=40, idmyrw=400, idmwrk=11000)
9  dimension iclarr(idmxcl, idmyrw), iwkarr(idmwrk), icharr(200)
10 dimension valarr(idmxcl, idmyrw)
11 dimension xwnloc(1000), ywnloc(1000)
12 character file*168, ans*(10), string*200
13 character dvce#1
14 logical status
15 common /dvcecom/dvce
16 common /cellcom1/ iclarr, iwkarr
17 common /cellcom2/ valarr
18 save icall_sdl
19 data icall_sdl/0/
20 c
21 c
22 string = " "
23 status=.false.
24 ans(3)="y"
25 iaxorn = 1
26 c
27 c
28 if (icall_sdl .eq. 1) goto 10
29 call asr(">imi>SDL", "-after", "working_dir")
30 call ioa_("If using a Tektronix, the execution log is in sdl_log.")
31 call ioa_("See: sdl_names.info and sdl_cellmap.info for definitions of variable names.")
32 call ioa_("help sdl_names -section <name>")
33 call ioa_("/CAUTION: this program overwrites existing default plot files")
34 call ioa_("To save plot files, rename them before execution.~/")
35 icall_sdl=1
36 c
37 10 call sdl_reset
38 call close_file("all")
39 goto 30
40 c
41 c
42 20 call query_("New Grid data? ", ans(6))
43 if(ans(6) .eq. "n") goto 40
44 call ioa_("Data matrix of digits and upper case characters")
45 call read_list_$prompt("Data matrix file name: ", file)
46 open(10, form="formatted", mode="in", file=file, access="sequential", err=830)

```

```

47 call read_list_$prompt("inmxcl: ", inmxcl, "inmyrw: ", inmyrw)
48 if(inmxcl.gt.400.or.inmyrw.gt.400) goto 810
49 do 40 i=1, inmyrw
50 read(10,1000) (icelarr(j,i), j=1, inmxcl)
51 40 continue
52 close(10)
53 c
54 c Plot data points option
55 c
56 call query_("Do you want to plot data points? ", ans(5))
57 if (ans(5).eq. "n") go to 50
58 call read_list_$prompt("data points (x,y) file name: ", file)
59 open(10,formatted="in",file=file,access="sequential",err=830)
60 do 45 i=1,1000
61 read(10,1000,end=48)xwnloc(i),ywnloc(i)
62 nloc=i
63 continue
64 close(10)
65 call ioa_("%i data points read.",nloc)
66 c
67 c
68 50 call read_list_$prompt("xpqlen: ", xpqlen, "ypqlen: ", ypqlen)
69 call read_list_$prompt("xwnlen: ", xwnlen, "ywnlen: ", ywnlen)
70 xwnoff=(xpqlen-xwnlen)/2.
71 ywnoff=(ypqlen-ywnlen)/2.
72 call read_list_$prompt("caption (in quotes): ", string)
73 call upper_case_(string)
74 xgdmn=0.
75 ygdmn=0.
76 xgdmax=xwnlen
77 ygdmax=ywnlen
78 if(ans(5).eq. "y") call read_list_$prompt("xgdmn: ", xgdmn, "xgdmax: ", xgdmax)
79 if(ans(5).eq. "y") call read_list_$prompt("ygdmin: ", ygdmin, "ygdmax: ", ygdmax)
80 call read_list_$prompt("iaxorn (1-4): ", iaxorn)
81 call read_list_$prompt("distance (in no. of cells) between standard labels: ", intrvl)
82 if (intrvl.lt. 2) intrvl=2
83 c
84 c compute cell size
85 xellen=xwnlen/(inmxcl-1)-0.05
86 yelhgt=ywnlen/(inmyrw-1)-0.05
87 c
88 c
89 c
90 call query_ ("~/ Do you want to change anything before plotting? ", ans(6))
91 if (ans(6).eq. "n") go to 60
92 call ioa_ ("~/Type a comma for values that remain the same.")
93 go to 20
94 c
95 60 continue
96 if(ans(3).eq. "y") call sdi_device_
97 if(ans(3).eq. "n") call sdi_device_$preset
98 if(.not. status) call sdi_reset
99 if(.not. status) call bgnplt
100 status=.true.

```

```

101 close(20)
102 if (dvce .eq. "t") open(20,form="formatted",mode="out",file="sdl_log",access="sequential",err=820)
103 if(dvce .eq. "t") call setrpl(3,20)
104 call setxyg(ixmxcl,xgdmn,xgdmx,imyrw,ygdmn,ygdmx)
105 call setorn(1,iaorn)
106 call setpag(xpglen,ypglen,1,0,1,0)
107 call setwin(xwnlen,ywnlen,xwnoff,ywnoff)
108 c compute maximum cell label size
109 itptxt=6
110 call cletsz(itptxt,iclarr,1,txtlen,txthgt)
111 if (txtlen.lt. xellen .and. txthgt.lt. yelhgt) goto 62
112 itptxt = itptxt-1
113 if (itptxt .le. 0) goto 62
114 goto 61
115 62 if (itptxt .le. 0) call loa_("cell size too small for label")
116 if (dvce .ne. "t") call loa_("maximum label size (itptxt) that fits in one cell is '1',itptxt)
117 if (dvce .ne. "t") call read_list_$prompt("itptxt (0-6): ",itptxt)
118 c
119 call nulmap
120 if (ans(5) .eq. "n") go to 65
121 if(dvce .ne. "t") call setrpl(2,6)
122 do 63 i=1,nloc
123 call mappnt(xwnloc(i),ywnloc(i),4,1,2,0,0)
124 63 continue
125 if(dvce .ne. "t") call setrpl(3,6)
126 c
127 c setup caption
128 65 call cletxt_(string,ichent,icharr)
129 if (ichent .eq. 0) goto 90
130 c check and set size of letters (default size = 6)
131 ntpxt = 6
132 call cletsz(ntptxt,icharr,ichent,txhlen,txthgt)
133 if (txhlen.lt. xwnlen .and. txthgt.lt. ywnoff) go to 80
134 ntpxt = ntpxt-1
135 if (ntptxt .le. 0) go to 80
136 go to 70
137 80 if (ntptxt .le. 0) call loa_("caption will not fit in space provided")
138 offcap=(ywnoff-ntpht)/2.
139 if (ntptxt .gt. 0) call captxt(1,offcap,2,ntptxt,icharr,ichent)
140 continue
141 call bgncl
142 call boxplt
143 call pltcel(iclarr,idxcl,imyrw,iwkarr,idxwrk)
144 if (itptxt .gt. 0) call label(iclarr,idxcl,imyrw,itptxt,intrvl,ivalarr)
145 call endcel
146 if(dvce .eq. "t") call endplt
147 if(dvce .eq. "t") status=.false.
148 c
149 c setup for next plot or stop
150 c
151 call query_ (" / Do you want another plot?",ans(3))
152 if (ans(3) .eq. "n") goto 999
153 call query_ (" / Do you want to change device? ",ans(3))
154 if(ans(3) .eq. "y" .and. status) call endplt

```



```

155 if(ans(3) .eq. "y" .and. status) status=.false.
156 if(ans(3) .eq. "n" .and. dvce .ne. "tw") call nxtfrm
157 call query_ ("~/ Do you want to change anything else?",ans(6))
158 if (ans(6) .eq. "n") go to 60
159 call ioa_ ("~/Type a comma for values that remain the same.")
160 go to 20
161 c
162 c error messages
163 c
164 810 call ioa_("Data matrix too large. Maximum size is 400 by 400")
165 go to 999
166 820 call ioa_("sdl_cellmap: unable to open file sdl_log.")
167 go to 999
168 830 call ioa_("sdl_cellmap: unable to open file `a",file)
169 999 if(status) call endplt
170 status=.false.
171 call sdl_reset
172 stop
173
174 1000 format(400a1)
175 end

```

COMPILATION LISTING OF sdl_device_

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1345.8 pst Thu
 Options: ansi66 table map

Subroutine sdl_device_

```

1  subroutine sdl_device_
2  external initiate (descriptors)
3  external ioa_ (descriptors)
4  external read_list_$prompt (descriptors)
5  external setup_tektronix_tcs (descriptors)
6  external get_term_type_ (descriptors)
7  character dvce*1,ttp*32
8  common /dvcecom/dvce
9  save icall_tcs
10 data icall_tcs/0/
11 c
12 c
13 5 call read_list_$prompt(""/device (t, v, h, p, or ?): ",dvce)
14 entry preset
15 if(dvce.eq."t") goto 20
16 if(dvce.eq."v") goto 30
17 if(dvce .eq. "h") goto 40
18 if(dvce .eq. 'p') goto 50
19 if(dvce .eq. "?") call ioa_(%t=Tektronix, v=Versatec)
20 if(dvce .eq. "h") call ioa_(%h=Houston, p=postprocessor")
21 goto 5
22 c
23 20 if(icall_tcs .eq. 0) call setup_tektronix_tcs
24 icall_tcs = 1
25 call initiate(">iml>tcs>dgxplt", "-force")
26 call get_term_type_(ttp)
27 if(ttp .eq. "TEK4025") call tkx_4025_setup
28 return
29 c
30 30 call initiate(">iml>v_plot>dgxplt", "-force")
31 return
32 c
33 40 call initiate(">iml>houston>dgxplt", "-force")
34 return
35 c
36 50 call initiate(">iml>sdl>dgxplt_post", "dgxplt", "-force")
37 return
38 end

```

COMPILATION LISTING OF sdl_grid_

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 04/19/81 0955.7 pst Sun
 Options: ansi66 table map

Subroutine sdl_grid_

```

1  subroutine sdl_grid_(file,inxcl,inmyrw,xgdmin,ygdmax,ixorn,zgdnul,elvarr,icode)
2  c
3  parameter(idmxcl=400,idmyrw=400)
4  dimension elvarr(idmxcl,idmyrw)
5  character file*168,grid
6  c
7  c on input:
8  icode = 1 to save grid
9  icode = 0 to retrieve grid
10 c on output:
11 icode = 0 indicates no error
12 icode = 1 indicates file is not sdl grid
13 icode > 1 indicates failure
14 c
15 external foa_(descriptors)
16 c
17 if (icode .lt. 0 .or. icode .gt. 1) goto 803
18 c
19 if (icode .eq. 0) goto 100
20 c
21 c write grid file
22 c
23 if (inxcl .gt. idmxcl .or. inmyrw .gt. idmyrw) goto 803
24 close(10)
25 open(10,file=file,form="formatted",mode="out",access="sequential",err=800)
26 write(10,1000) "SDL1"
27 write(10,1001) inmxcl,inmyrw,xgdmin,xgdmax,ygdmin,ygdmax,ixorn,zgdnul
28 do 20 i=1,inmyrw
29   write(10,1001) (elvarr(j,i),j=1,inxcl)
30 continue
31 close(10)
32 call foa_("/grid written to 'a",file)
33 call foa_("#inxcl = 'i', inmyrw = 'i', ixorn = 'i', inmxcl,inmyrw,ixorn)
34 call foa_("#xgdmin = 'f', xgdmax = 'f', ygdmin = 'f', ygdmax, ygdmin,ygdmax)
35 call foa_("#zgdnul = 'f','",zgdnul)
36 icode=0
37 goto 999
38 c
39 c
40 c retrieve grid file
41 c
42 100 continue
43 close(10)
44 open(10,file=file,form="formatted",mode="in",access="sequential",err=800)
45 read(10,1000,err=801) grid
46 if(grid .ne. "SDL1") goto 802

```

```

47 read(10,1001,err=801) inmxcl,inmyrw,xgdmmin,xgdmax,ygdmin,ygdmax,iaxorn,zgdnul
48 if(inmxcl.gt. idmxcl) call ioa_("sdl_grid_": warning inmxcl=~i, ~i used",inmxcl,idmxcl)
49 if(inmxcl.gt. idmxcl) inmxcl=idmxcl
50 if(inmyrw.gt. idmyrw) call ioa_("sdl_grid_": warning inmyrw=~i, ~i used",inmyrw,idmyrw)
51 if(inmyrw.gt. idmyrw) inmyrw=idmyrw
52 do 120 i=1,inmyrw
53 read(10,1001,err=801) (elvarr(j,i),j=1,inmxcl)
54 continue
55 close(10)
56 call ioa_("~/grid read from ~a",file)
57 call ioa_("inmxcl = ~i, inmyrw = ~i, iaxorn = ~i", inmxcl,inmyrw,iaxorn)
58 call ioa_("xgdmmin = ~f, xgdmax = ~f, ygdmin = ~f, ygdmax = ~f", xgdmmin,xgdmax,ygdmin,ygdmax)
59 call ioa_("zgdnul = ~f~/",zgdnul)
60 icode=0
61 goto 999
62 c
63 c
64 c error messages
65 c
66 800 call ioa_("sdl_grid_: unable to open file ~a",file)
67 icode=3
68 goto 999
69 801 call ioa_("sdl_grid_: error reading file ~a",file)
70 icode=4
71 goto 999
72 802 icode=1
73 goto 999
74 803 call ioa_("bad call to sdl_grid_")
75 icode=10
76 return
77 c
78 999 close(10)
79 return
80 format(a4)
81 1001 format(v)
82 end

```

COMPILATION LISTING OF sdlpop_device_

Compiled by: Multics New Fortran Compiler, Release 8a
 Compiled on: 03/26/81 1341.3 pst Thu
 Options: ansi66 table map

Subroutine sdlpop_device_

```

1  subroutine sdlpop_device_
2  external initiate (descriptors)
3  external read_list_$prompt (descriptors)
4  external setup_tektronix_tcs (descriptors)
5  external get_term_type_ (descriptors)
6  character dvce#1,ttp#32
7  common /dvcecom/dvce
8  save icall_tcs
9  data icall_tcs/0/
10 5  call read_list_$prompt("device (t, v, or h): ",dvce)
11  entry preset
12  if(dvce.eq."t") goto 20
13  if(dvce.eq."v") goto 30
14  if(dvce.eq."h") goto 40
15  goto 5
16 20  if(icall_tcs .eq. 0) call setup_tektronix_tcs
17  icall_tcs = 1
18  call initiate(">iml>tcs>dgxplt", "-force")
19  call get_term_type_(ttp)
20  if(ttp .eq. "TER4025") call tkx_4025_setup
21  return
22 30 call initiate(">iml>v_plot>dgxplt", "-force")
23  return
24 40 call initiate(">iml>houston>dgxplt", "-force")
25  return
26  end

```

COMPILATION LISTING OF SEGMENT upper_case_
Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
Compiled at: USGS; Menlo Park, California
Compiled on: 03/26/81 1403.0 pst Thu
Options: map table

```
1 upper_case_ : proc (string);  
2   string char (*);  
3   do1 translate builtin;  
4     string = translate (string, "ABCDEFGHIJKLMNOPQRSTUVWXYZ",  
5       "abcdefghijklmnopqrstuvwxyz");  
6   end;
```

COMPILATION LISTING OF tkx_4025_setup

Compiled by: Multics New Fortran Compiler, Release 8a
Compiled on: 03/26/81 1341.4 pst Thu
Options: ansi66 table map

Subroutine tkx_4025_setup

```
1   subroutine tkx_4025_setup  
2     external ioa_ (descriptors)  
3     call ioa_ ('lwor 33 hw')  
4     call ioa_ ('wgra 1 40w')  
5     call ioa_ ('lshrink bw')  
6     return  
7   end
```

COMPILATION LISTING OF SEGMENT cletxt_
 Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
 Compiled at: USGS; Menlo Park, California
 Compiled on: 03/26/81 1402.9 pst Thu
 Options: map table

```

1 cletxt: cletxt_ = proc (string, ichcnt, icharr);
2 /*****
3 **
4 ** The SPL subroutine cletxt takes as input a string constant and
5 ** its length. It returns a integer array with one character
6 ** left-justified in each word. This version of cletxt takes as
7 ** input a character variable. It calculates the length of
8 ** left-justified text and returns both the length and the integer
9 ** array.
10 **
11 **      By Robert Mark      6/10/79
12 **
13 *****/
14 decl string char (*),
15      ichcnt fixed bin,
16      icharr (*) fixed bin (35);
17 decl (i, ichdim) fixed bin;
18 decl vstring char (length (string)) var;
19 decl ioa_ entry options (variable);
20 decl bstring (dim (icharr, 1)) char (4) based (addr (icharr));
21      vstring = rtrim (string);
22      bstring = " ";
23      ichcnt = length (vstring);
24      if ichcnt = 0 then return;
25      ichdim = dim (icharr, 1);
26      if (ichcnt > ichdim) then do;
27          call ioa_ ("cletxt: Error, string(^i) longer than icharr(^i).",
28                  ichcnt, ichdim);
29          ichcnt = 0;
30          return;
31      end;
32      do i = 1 to ichcnt;
33          bstring (i) = substr (vstring, i, 1) || " ";
34      end;
35      end;
  
```

COMPILATION LISTING OF SEGMENT get_term_type_
 Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
 Compiled at: USGS; Menlo Park, California
 Compiled on: 03/26/81 1402.9 pst Thu
 Options: map table

```

1 get_term_type_: proc (term_type);
2 del term_type char (32);
3 del iox_$find_iocb entry (char (*), ptr, fixed bin (35));
4 del iox_$control entry (ptr, char (*), ptr, fixed bin (35));
5 del com_err_ entry options (variable);
6 del (iocb_ptr, info_ptr) ptr init (null);
7 del null builtin;
8 del code fixed bin (35);
9 del 1 terminal_info aligned,
10 del 2 version fixed bin,
11 del 2 id char (4) unaligned,
12 del term_type char (32) unaligned,
13 del 2 line_type fixed bin,
14 del baud_rate fixed bin,
15 del 2 reserved (4) fixed bin;
16 info_ptr = addr (terminal_info);
17 version = 1;
18 term_type = "";
19 call iox_$find_iocb ("user_i/o", iocb_ptr, code);
20 if (iocb_ptr = null) then do;
21 call com_err_ (code, "get_term_type_");
22 return;
23 end;
24 call iox_$control (iocb_ptr, "terminal_info", info_ptr, code);
25 if (code ^= 0) then do;
26 call com_err_ (code, "get_term_type_");
27 return;
28 end;
29 term_type = terminal_info.term_type;
30 return;
31 end;
  
```


COMPILATION LISTING OF SEGMENT query_
Compiled by: Multics PL/I Compiler, Release 26a, of September 3, 1980
Compiled at: USGS: Menlo Park, California
Compiled on: 03/26/81 1402.7 pst Thu
Options: map table

```
1 query_: proc (prompt, answer);  
2 decl prompt char (*);  
3 decl answer char (1);  
4 decl read_list_$prompt entry options (variable);  
5 query: call read_list_$prompt (prompt||" (y or n): ", answer);  
6     if (answer ^= "y" & answer ^= "n") then goto query;  
7     return;  
8 end;
```

LISTING OF sdlpop.ec

```
1 &command_line off  
2 >udd>MINITAB>sdl>sdlpop_device_  
3 >iml>sdl>sdlpop  
4 &if [not [query "delete sdl_plot ? "]] &then &quit  
5 delete sdl_plot  
6 &quit
```