General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

CR151,911

TN 76-123

SIGMA CORPORATION
TECHNICAL NOTE

(NASA-CR-151911) SUTIL: SYSTEM UTILITIES ROUTINES PROGRAMMER'S REFERENCE MANUAL (Sigma Corp., Houston, Tex.) 21 p HC A02/MF A01 CSCL 09B

N79-17578

Unclas
G3/61 13940

SUTIL; System Utilities Routines

Programmer's Reference Manual

Revision C

by: Dale Harper, Sigma Corporation

FEB 1979
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

Prepared for:

NASA Johnson Space Center
Houston, Texas 77058

December, 1976

ORIGINAL PAGE IS
OF POOR QUALITY

PREFACE

This report was prepared under Contract NAS9-14520, "Development of the Engineering Design Intergration System". The study was preformed during a period from January 1975 through December 1976. The study was funded by the National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, Houston Texas. The prime contractor for the study was Sigma Corporation and the study was monitored by Mr. Robert Abel of the NASA Engineering Analysis Division.

TABLE OF CONTENTS

PAGE

INTRODUCTION..... 2

SUBROUTINE DESCRIPTIONS

Get value from operator
IGET..... 3
Calling Sequence..... 4
Arguments..... 4
Error Conditions..... 4
RGET..... 5
Calling Sequence..... 6
Arguments..... 6
Error Conditions..... 6
OGET..... 7
Calling Sequence..... 8
Arguments..... 8
Error Conditions..... 8
Get AMOS string from operator
SGET..... 9
Calling Sequence..... 9
Arguments..... 9
Error Conditions..... 9
Get yes or no answer from operator
YESNO..... 10
Calling Sequence..... 10
Arguments..... 13
Monitor listing device
MLIST..... 12
Calling Sequence..... 12
Error Conditions..... 14
Time Delay
IDELAY..... 16
Calling Sequence..... 16
Error Conditions..... 16
Ring TTY bell
RBELL..... 16
Calling Sequence..... 16
Error Conditions..... 16
Free format ANK/TTY input
RDLIN..... 17
Calling Sequence..... 17
Error Conditions..... 17
EXAMPLES
Test..... 11
Test1..... 15

Language 18

Subroutine Length 18

Equipment 18

Authorship 18

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

INTRODUCTION

SUTIL is a package of FORTRAN callable subroutines which allow efficient communication of data between users and programs.

Calls to SUTIL routines which output text to the screen will unconditionally load monitor overlay buffer 1 with the LABEL segment (if not allready loaded).
i.e. Calls into the SUTIL package must not be concurrent with FORTRAN use of monitor overlay buffer 1.

Text and values output to the screen by SUTIL routines IGET, RGET, OGET, SGET, and YESNO cannot be printed on the PPL, as PPL driver uses monitor overlay buffer 1.

Proper utilization of the SUTIL package will significantly reduce program core requirements and expedite program developement.

IGET

IGET is a FORTRAN callable routine designed to display a message for the operator on the AGT screen.

Routine will display the minimum and maximum bounds and current value for an integer variable, accept the current value with a C/R, or a new typein value from the operator.

If an image is being displayed prior to calling IGET a subimage link is formed to enable the image to remain on the screen with message, bounds and current value display, this link is removed on return from subroutine.

If operator types in a new value, it is compared to the upper and lower bound. If within bounds, new value is returned to the calling program. If out of bounds, variable is set to the value of the crossed boundary, the TTY bell is rung, and the following error message is displayed for the operator:
LIMIT EXCEEDED, DEFAULT VALUE ASSUMED
(MIN=XXX,MAX=XXX) CURRENT=XXX

Simple math may be included in typein, however it must be in polish string format.
i.e. no ( ) and solution is from left to right.

Characters which may be included in type string are:

Symbol	Function
+	Add
-	Subtract
*	Multiply
/	Divide
=	Sum
X	Change sign

Other characters cause the TTY bell to ring and the following error message is displayed for the operator:

ERROR: ILLEGAL CHARACTER

i.e. 100/25*25-50+50 equals 100
100/2.5+60 equals 100
100 + 50X + 50 equals 100 (X = change sign)

Typein followed by an '=' sign will cause the sum to replace typein. This sum may be accepted with a C/R or a new VALUE typed.

CALLING SEQUENCE

```
CALL IGET (IVALUE,MIN,MAX,"MESSAGE",IFLAG)
```

ARGUMENTS

1	IVALUE	Variable (I/O integer).
2	MIN	Minimum bound for IVALUE (input integer).
3	MAX	Maximum bound for IVALUE (input integer).
4	MESSAGE	ASCII string (input). An ASCII character string to be displayed for the user. The MESSAGE is small characters at left side of screen unless ASCII control characters are in MESSAGE string. i.e., @X@000,@Y000,@C,@F,@E,@S characters will change MESSAGE size and position.
5	IFLAG	(input integer). If positive, clear prior MESSAGE(s) before displaying MESSAGE. Else add this MESSAGE to any previous MESSAGE(s).

ERROR CONDITIONS

All errors are properly handled internally. If numbers are out of range as input, TTY bell rings and a message to the user will appear requesting a new input.

RGET

RGET is a FORTRAN callable routine designed to display a message for the operator on the AGT screen.

Routine will display the minimum and maximum bounds and current value for a real variable, accept the current value with a C/R, or a new typein value from the operator.

If an image is being displayed prior to calling RGET a subimage link is formed to enable the image to remain on the screen with message, bounds and current value display, this link is removed on return from subroutine.

If operator types in a new value, it is compared to the upper and lower bound. If within bounds, new value is returned to the calling program. If out of bounds, variable is set to the value of the crossed boundary, the TTY bell is rung, and the following error message is displayed for the operator:
LIMIT EXCEEDED, DEFAULT VALUE ASSUMED
(MIN=XXX,MAX=XXX) CURRENT=XXX

Simple math may be included in typein, however it must be in polish string format.
i.e. no ( ) and solution is from left to right.

Characters which may be included in type string are:

Symbol	Function
+	Add
-	Subtract
*	Multiply
/	Divide
=	Sum
X	Change sign

Other characters cause the TTY bell to ring and the following error message is displayed for the operator:

ERROR: ILLEGAL CHARACTER

i.e.	100/25*25-50+50	equals 100
	100/2.5+60	equals 100
	100 + 50X + 50	equals 100 (X = change sign)

Typein followed by an '=' sign will cause the sum to replace typein. This sum may be accepted with a C/R or a new VALUE typed.

CALLING SEQUENCE

```
CALL RGET (VALUE,RMIN,RMAX,"MESSAGE",IFLAG)
```

ARGUMENTS

ORIGINAL PAGE IS
OF POOR QUALITY

1 VALUE — Variable (I/O real).
2 RMIN — Minimum bound for VALUE (input real).
3 RMAX — Maximum bound for VALUE (input real).
4 MESSAGE — ASCII string (input).
An ASCII character string to be displayed for the user.
The MESSAGE is small characters at left side of screen unless ASCII control characters are in MESSAGE string.
i.e., 0X0000,0Y000,0C,0F,0E characters will change MESSAGE size and position.

5 IFLAG — (input integer).
If positive, clear prior MESSAGE(s) before displaying MESSAGE.
Else add this MESSAGE to any previous MESSAGE(s).

ERROR CONDITIONS

All errors are properly handled internally. If numbers are out of range as input, TTY bell rings and a message to the user will appear requesting a new input.

OGET

ORIGINAL PAGE IS
OF POOR QUALITY

OGET is a FORTRAN callable routine designed to display a message for the operator on the AGT screen.

Routine will display the minimum and maximum bounds and current value for an octal variable. Accept the current value with a C/R, or a new typein value from the operator.

If an image is being displayed prior to calling OGET a subimage link is formed to enable the image to remain on the screen with message, bounds and current value display, this link is removed on return from subroutine.

If operator types in a new value, it is compared to the upper and lower bound. If within bounds, new value is returned to the calling program. If out of bounds, variable is set to the value of the crossed boundary, the TTY bell is rung, and the following error message is displayed for the operator:
LIMIT EXCEEDED, DEFAULT VALUE ASSUMED
(MIN=XXX,MAX=XXX) CURRENT=XXX

Simple math may be included in typein, however it must be in polish string format.
i.e. no ( ) and solution is from left to right.

Characters which may be included in type string are:

Symbol	Function
+	Add
-	Subtract
*	Multiply
/	Divide
=	Sum
X	Change sign

Other characters cause the TTY bell to ring and the following error message is displayed for the operator:

ERROR: ILLEGAL CHARACTER

i.e. $40_8 + 40_8$ equals 100_8

Typein followed by an '=' sign will cause the sum to replace typein. This sum may be accepted with a C/R or a new VALUE typed.

ORIGINAL PAGE IS
OF POOR QUALITY

CALLING SEQUENCE

```
CALL OGET (OVALUE,OMIN,OMAX,"MESSAGE",IFLAG)
```

ARGUMENTS

1	OVALUE	Variable (input,output octal).
2	OMIN	Minimum bound for OVALUE (input octal).
3	OMAX	Maximum bound for OVALUE (input octal).
4	MESSAGE	ASCII string (input). An ASCII character string to be displayed for the user. The MESSAGE is small characters at left side of screen unless ASCII control characters are in MESSAGE string. i.e., @X0000,@Y000,@C,@F,@E characters will change MESSAGE size and position.
5	IFLAG	(input integer). If positive, clear prior MESSAGE(s) before displaying MESSAGE. Else add this MESSAGE to any previous MESSAGE(s).

ERROR CONDITIONS

All errors are properly handled internally. If numbers are out of range as input, TTY bell rings and a message to the user will appear requesting a new input.

SGET

ORIGINAL PAGE IS
OF POOR QUALITY

SGET is a FORTRAN callable routine designed to display a message for the operator on the AGT screen, and accept a typed string from the operator.

If an image is being displayed prior to calling SGET a subimage link is formed to enable the image to remain on the screen with message, bounds and current value display, this link is removed on return from subroutine.

CALL SEQUENCE

```
CALL SGET (ARRAY,MESSAGE,KFLAG)
```

ARGUMENTS

1 ARRAY — AMOS character array (output).
Minimum Length of array should be determined as follows:
(use a multiple of 5 for maximum characters)
If (maximum characters .EQ. 5) length = 1
else
Length = maximum characters / 5 + 1.
If typed string is .GT. 5 first word in array is pointer to characters in form: -number words !H address of string.

2 MESSAGE — ASCII string (input).
An ASCII character string to be displayed for the user. The MESSAGE is small characters at left side of screen unless ASCII control characters are in MESSAGE string. i.e., 0X0000,0Y000,0C,0F,0E characters will change MESSAGE size and position.

3 KFLAG — Optional (input integer).
Maximum number of characters to accept from operator. (range = ± 1-132)
If KFLAG argument is neg. prior MESSAGE(s) is not cleared.
If KFLAG argument is omitted, prior MESSAGE(s) is cleared, and 10 set as maximum output characters.

ERROR CONDITIONS

All errors are properly handled internally.

YESNO

YESNO is a FORTRAN function designed to display a message for the operator on the AGT screen and accept the first character of operator type in string as a YES/NO answer.
i.e. Yes...."Y" "C/R", or "C/R".
No...anything else.

Calling program can branch on the returned function value.

If an image is being displayed prior to calling YESNO a subimage link is formed to enable the image to remain on the screen with message display, this link is removed on return from subroutine.

CALLING SEQUENCE

```
IF( YESNO("MESSAGE",NFLAG)) 10,20    or    LLL=YESNO("MESSAGE",NFLAG))
```

ARGUMENTS

1. MESSAGE — ASCII string (input).
An ASCII character string to be displayed for the user. The MESSAGE is small characters at left side of screen unless ASCII control characters are in MESSAGE string.
i.e., 0X0000,0Y000,0C,0F,0E characters will change MESSAGE size and position.

2. NFLAG — (input integer).
If positive, clear prior MESSAGE(s) before displaying MESSAGE. Else add this MESSAGE to any previous MESSAGE(s).

```
EXAMPLE
LOGICAL YESNO
EXTERNAL YESNO

     IF ( YESNO("IS EVERYTHING OK",NFLAG) ) 10,20
```

If the typed responce is "Y C/R" or "C/R", the function value will be true, and statement 10 will be executed. Otherwise statement 20 will result from the false function value.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM TEST
       LOGICAL YESNO
      IVAL = RVAL = 0.0
      IMIN = RMIN = -9999.0
      IMAX = RMAX =  9999.0
      IOCTAL = 101B; MINOCTAL = 101B; MAXOCTAL = 2337B
C   PUT UP MESSAGE ON SCREEN AND ACCEPT TYPEIN IN RANGE RMIN TO RMAX

10    CALL RGET(RVAL ,RMIN, RMAX," @CTHIS MESSAGE WILL APPEAR AS @
     1SMALL CHARACTERS, AT@C@FUPPER LEFT OF SCREEN.@
     1@E@C@FTYPE IN A REAL VALUE : @S", 1)

C   ADD ANOTHER MESSAGE, ACCEPT INTEGER TYPE IN
      CALL IGET (IVAL,IMIN,IMAX," @S@C@FTHIS MESSAGE IS ADDED ALSO,@
     1THIS TIME WE ARE CALLING THE INTEGER ROUTINE, IF FRACTIONS @C@F@
     1ARE TYPED IN, ONLY THE INTEGER PORTION IS RETURNED.@E@C@F@
     1INTEGER VALUE = : @S", -1)

C   ADD ANOTHER MESSAGE
      CALL IGET (IVAL,IMIN,IMAX," @S@C@FSIMPLE MATH MAYBE INCLUDED @
     1IN TYPEIN,@C@FHOWEVER IT MUST BE IN POLISH STRING FORMAT.@
     1I.E. NO ( ) AND SOLUTION IS FROM LEFT TO RIGHT.@
     1@C@F       ONLY + - * / X = ARE ALLOWED.@C@F@
     1ANY OTHER CHARACTERS CAUSE THE TTY BELL TO RING AND A NEW @
     1INPUT REQUESTED.@C@F@
     1I.E.     100/25+25-50+50    EQUALS 100@C@F        @
     1OR@C@F@
     1         100/2.5+60         IS ALSO 100@C@F        @
     1OR@C@F@
     1         100 + 50X + 50     EQUALS 100 (X = CHANGE SIGN)@
     1@C@FTYPEIN FOLLOWED BY AN '=' SIGN WILL CAUSE THE SUM TO@
     1@C@FREPLACE TYPEIN, THIS SUM MAY BE ACCEPTED WITH A C/R @
     1@C@FOR A NEW VALUE MAY BE TYPED IN.@
     1@E@C@FINTEGER VALUE = : @S ", -1)
C   PLACE ANSWER IN RVAL AS CURRENT VALUE
      RVAL = IVAL
C   ASK USER FOR A DATA PACK-VOLUME
     CALL OGET (IOCTAL,MINOCTAL,MAXOCTAL,"@S@C@F@
    1TYPE IN PACK-VOLUME FOR DATA.@
    1@E@C@FPVV = :",-1)
C   DO NOT CLEAR SCREEN, ASK IF OPERATOR WANTS TO REPEAT.
     IF (YESNO (" @E@C@FWANT TO GO THRU AGAIN ? :", -1) ) 10,20
C C/R OR Y  C/R RETURN TO 10, ANYTHING ELSE RETURNS FROM PROGRAM

20    CALL EXIT
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

MLIST

MLIST is a collection of FORTRAN callable subroutines which allow convenient access to the Adage monitor listing device routines.

CALLING SEQUENCE

Select and initialize listing device.

```
   CALL INLIST (ICODE)
            or
   CALL INLIST (ICODE, 'NAME', IPVV)
```

Output an AMOS string with 0 to NCR carriage returns prior to string.

```
   CALL FAMOS ('AA',NCR)
```

Output date.

```
   CALL IDATE (date)
```

Output an ASCII string (listing device 1 only (ICODE =1)).

```
   CALL FASCII("BB")
```

Tab to specified column of listing device.

```
   CALL TABTO (ICOL)
```

Output a carriage return and line feed.

```
   CALL OSTCR
```

Eject current page.

```
   CALL EJECT
```

Unselect listing device.

```
   CALL UNSELECT
```

Output decimal integer.

```
   CALL I2DEC (IVALUE,IDIGITS)
```

Output floating point number.

```
   CALL R2DEC (VALUE,ITOTAL,IFRAC)
```

After initial call to INLIST, the following paramenters, may be accessed at : (use FORTRAN equivalence statement).

70007_8 = current character position in line.
70010_8 = current line count.
70011_8 = maximum lines/page for selected device.
70012_8 = busy flag (negative if busy).

ARGUMENTS

ARGUMENT	TYPE	FUNCTION
ICODE	input integer	Desired listing device. 0 = TTY 1 = SCREEN 2 = PPL 3 = ATEXT disk file NOTE If ICODE = 3 , and arguments NAME and PVV are omitted, file is output to PVV 101_8 with name DBUGL. If more than 1 argument to INLIST all are required.
NAME	input AMOS string	If ICODE = 3, the output disk file name is set = NAME
PVV	input octal	Output pack-volume for disk file.
AA	input AMOS string	String for output.
NCR	input integer	Number of carriage returns to output prior to AA.
DATE	input date	Address of a cell containing a date in AMOS format, usually $DATE or $FILEDATE
BB	input ASCII string	String for output.
ICOL	input integer	Desired character column.
IVALUE	input integer	Integer value to be output.
IDIGITS	integer input	Total digits, range = ± 10. If positive ...right justify. If negative ...left justify. Round off will occur if value exceeds number of digits specified.

ARGUMENT	TYPE	FUNCTION
VALUE	(input real)	Real value to be output. If VALUE exceeds 536,870,656, 'E' format is used.
ITOTAL	(integer input)	Total digits, range = ±10. If positive, right justify (integer part). If negitive, left justify (integer part). Round off will occur if value exceeds number of digits specified.
IFRAC	(input integer)	Fractional digits. Round off will occur if value exceeds number of digits specified.

ERROR CONDITIONS

If ICODE is invalid or incorrect number of arguments default is to TTY. ASCII output is legal to device 1 only.

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM TEST1
      REAL INBUFF(100),OBUFF1(100),OBUFF2(100)
      A = B = 10.0
C GET DESIRED LISTING DEVICE FROM USER.
      IDEVICE = 1
      CALL IGET (IDEVICE,0,3," OUTPUT DEVICE OPTIONS:@C@F@
     10 = TTY@C@F@
     11 = SCREEN@C@F@
     12 = PPL@C@F@
     13 = DISK FILE@C@F@
     1 TYPE IN DESIRED OPTION : ",1)
C INITIALIZE INBUFF
      DO 10 J = 1,100
      INBUFF(J) = A
      A = A + B
      IF (A .EQ. 250.0 ) B = -10.0
      IF (A .EQ. -250.0 ) B = 10.0
10    CONTINUE
C CLEAR OBUFF 1&2
      DO 11 J = 1,100
      OBUFF1(J) = OBUFF2(J) = 0.0
     11 CONTINUE
C PLACE SAMPLES IN OBUFF1 AT 2 TIMES SAMPLE RATE OF INBUFF
      CALL NTRPOLATE (INBUFF(1),OBUFF1(1),50,0.5)
C PLACE SAMPLES IN OBUFF2 AT 1/2 SAMPLE RATE OF INBUFF
      CALL NTRPOLATE (INBUFF(1),OBUFF2(1),50, 2.0)
C INITIALIZE LISTING DEVICE
      CALL INLIST (IDEVICE)
C OUTPUT HEADER FOR LISTING
      CALL FAMOS ('BUFFER LISTING OF INTERPOLATION RESULTS',0)
      CALL FAMOS ('INBUFF', 2)
      CALL TABTO (20)
      CALL FAMOS ('OBUFF1', 0)
      CALL TABTO (40)
      CALL FAMOS ('OBUFF2',0)
      CALL OSTCR
      C LIST CONTENTS OF BUFFERS
      DO 20 J = 1,50
      CALL R2DEC (INBUFF(J),8,3)
      CALL TABTO (20)
      CALL R2DEC (OBUFF1(J),8,3)
      CALL TABTO (40)
      CALL R2DEC (OBUFF2(J),8,3)
      CALL OSTCR
20    CONTINUE
C EJECT PAGE, AND UNSELECT DEVICE
      CALL EJECT
      CALL UNSELECT
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

IDELAY

IDELAY is a FORTRAN callable Subroutine which causes a time delay.

CALLING SEQUENCE

Call IDELAY (ICOUNT)

ARGUMENTS

ICOUNT (input integer) Number of milliseconds of desired delay, range 1 to 4095.

ERROR CONDITIONS

Out of range number defaults to a one second delay.

RBELL

RBELL is a FORTRAN callable routine to ring the TTY bell.

CALLING SEQUENCE

CALL RBELL

ERROR CONDITIONS

None

ORIGINAL PAGE IS
OF POOR QUALITY

RDLIN

RDLIN is a FORTRAN callable routine designed to allow free format input from the ANK/TTY.

RDLIN will unconditionally load the LABEL segment into monitor overlay buffer 1 (unless allready loaded).

Typein is stored in Global array 'PLIST' as left justified 'PADKR' filled AMOS characters.

Typein is also stored in Global array 'ULIST', right justified, one character per word.

Global variable 'NICH' contains the number of typed characters.

Global variable 'MXKAR' controls the maximum number of typed characters allowed (range 1-132) .

Global variable 'PADKR' (normally 40_8 (space)) is used as pad character for left justified strings.
If 'PADKR' is changed by calling program it should be restored to 40_8.

Global flag 'NODSP', if negative assumes external display of monitor label buffer (#DBUFF).

Cursor appears on screen 2 spaces to the right of last output to label buffer, i.e. a write to unit 25 or SUTIL routine MLIST.

Calling Sequence

```
CALL RDLIN
```

ORIGINAL PAGE IS
OF POOR QUALITY

Error Conditions

After number of typed characters .EQ. 'MXKAR' no further characters are accepted.

LANGUAGE

ADEPT (Adage machine language)

SUBROUTINE LENGTH

Approximately 1500 words.

EQUIPMENT

AGT-10 through AGT-340

AUTHORSHIP

D. HARPER, Sigma Corp.