D 8

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
LL            IIIIII    BBBBBBBB      SSSSSSSS  TTTTTTTTTT  RRRRRRRR    RRRRRRRR    EEEEEEEEEE  TTTTTTTTTT
LL            IIIIII    BBBBBBBB      SSSSSSSS  TTTTTTTTTT  RRRRRRRR    RRRRRRRR    EEEEEEEEEE  TTTTTTTTTT
LL              II      BB      BB  SS              TT      RR      RR  RR      RR  EE              TT
LL              II      BB      BB  SS              TT      RR      RR  RR      RR  EE              TT
LL              II      BB      BB  SS              TT      RR      RR  RR      RR  EE              TT
LL              II      BB      BB  SS              TT      RR      RR  RR      RR  EE              TT
LL              II      BBBBBBBB      SSSSSS        TT      RRRRRRRR    RRRRRRRR    EEEEEEEE        TT
LL              II      BBBBBBBB      SSSSSS        TT      RRRRRRRR    RRRRRRRR    EEEEEEEE        TT
LL              II      BB      BB          SS      TT      RR  RR      RR  RR      EE              TT
LL              II      BB      BB          SS      TT      RR  RR      RR  RR      EE              TT
LL              II      BB      BB          SS      TT      RR    RR    RR    RR    EE              TT      ....
LL              II      BB      BB          SS      TT      RR    RR    RR    RR    EE              TT      ....
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS      TT      RR      RR  RR      RR  EEEEEEEEEE      TT      ....
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS      TT      RR      RR  RR      RR  EEEEEEEEEE      TT      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE LIB$$STRTO_RET (%TITLE'Library routine to convert a string signal to a returr'
;     2          0002  0         IDENT = 'T-002'          . File: LIBSTRRET.B32   Edit: SBL1002
;     3          0003  0                 ) =
;     4          0004  1 BEGIN
;     5          0005  1
;     6          0006  1 !
;     7          0007  1 !*****************************************************************************
;     8          0008  1 !*                                                                           *
;     9          0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    10          0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    11          0011  1 !*  ALL RIGHTS RESERVED.                                                     *
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    14          0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15          0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    16          0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    17          0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    18          0018  1 !*  TRANSFERRED.                                                             *
;    19          0019  1 !*                                                                           *
;    20          0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    21          0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    22          0022  1 !*  CORPORATION.                                                             *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    25          0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
;    26          0026  1 !*                                                                           *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*****************************************************************************
;    29          0029  1 !
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 ! FACILITY:  Utility Library
;    33          0033  1 !
;    34          0034  1 ! ABSTRACT:
;    35          0035  1 !
;    36          0036  1 !       LIB$$STRTO_RET converts any condition value signalled by
;    37          0037  1 !       STR$COPY or STR$GET1 or STR$FREE1 into the corresponding LIB status
;    38          0038  1 !       and an ordinary procedure return to the caller of the procedure
;    39          0039  1 !       which established the handler which called LIB$$STRTO_RET.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:  User mode, re-entrant, AST level or not or mixed.
;    42          0042  1 !
;    43          0043  1 ! AUTHOR:       Rebecca Will CREATION DATE:  22-Jan-1980
;    44          0044  1 !
;    45          0045  1 ! MODIFIED BY:
;    46          0046  1 !
;    47          0047  1 ! 01    -  original
;    48          0048  1 ! 1-002 - Add prologue file.  SBL 24-June-1983
;    49          0049  1 !--
```

```
;    51          0050  1 !
;    52          0051  1 ! PROLOGUE FILE:
;    53          0052  1 !
;    54          0053  1
;    55          0054  1 REQUIRE 'RTLIN:LIBPROLOG';              . LIB$ definitions
;    56          0125  1
;    57          0126  1 !
;    58          0127  1 ! TABLE OF CONTENTS:
;    59          0128  1 !
;    60          0129  1
;    61          0130  1 FORWARD ROUTINE
;    62          0131  1         LIB$$STRTO_RET;         ! Convert a signal to a return
;    63          0132  1
;    64          0133  1 !
;    65          0134  1 ! MACROS:
;    66          0135  1 !
;    67          0136  1 !       NONE
;    68          0137  1 !
;    69          0138  1 ! EQUATED SYMBOLS:
;    70          0139  1 !
;    71          0140  1 !       NONE
;    72          0141  1
;    73          0142  1 !
;    74          0143  1 ! OWN STORAGE:
;    75          0144  1 !
;    76          0145  1
;    77          0146  1 !
;    78          0147  1 ! EXTERNAL REFERENCES:
;    79          0148  1 !
;    80          0149  1
;    81          0150  1 EXTERNAL LITERAL
;    82          0151  1     LIB$_FATERRLIB,                 ! Fatal error in the library
;    83          0152  1     LIB$_STRIS_INT,                 ! String is interlocked
;    84          0153  1     LIB$_INSVIRMEM,                 ! Insufficient virtual memory
;    85          0154  1     LIB$_INVSTRDES,                 ! Invalid string descriptor
;    86          0155  1     STR$_FATINTERR,                 ! Fatal internal error in STR
;    87          0156  1     STR$_STRIS_INT,                 ! String is interlocked
;    88          0157  1     STR$_ILLSTRCLA,                 ! Illegal string class
;    89          0158  1     STR$_INSVIRMEM;                 ! Insufficient virtual memory
```

```
;    91          0159  1 GLOBAL ROUTINE LIB$$STRTO_RET  (        ! Convert a signal to a return
;    92          0160  1         SIG_ARGS_ADR,    ! Adr. of signal args vector
;    93          0161  1         MCH_ARGS_ADR)    ! Adr. of mechanism args vector
;    94          0162  1                 =        ! Value is success, unless failure from SYS$UNWIND
;    95          0163  1
;    96          0164  1 !++
;    97          0165  1 ! FUNCTIONAL DESCRIPTION:
;    98          0166  1 !
;    99          0167  1 !       LIB$STRTO_RET is called with the argument list passed to a condition
;   100          0168  1 !       handler by the condition handling facility.  It converts and translates
;   101          0169  1 !       the STR signalled condition
;   102          0170  1 !       into a LIB return status to the procedure which called the
;   103          0171  1 !       procedure which established the handler handling the signal.  The
;   104          0172  1 !       stack is unwound to the caller of the establisher and the condition code
;   105          0173  1 !       is returned as the value in R0.
;   106          0174  1 !
;   107          0175  1 !       In BLISS the argument list can be passed in toto by using the
;   108          0176  1 !       BUILTIN function CALLG and AP register, thus:
;   109          0177  1 !
;   110          0178  1 !               status = CALLG (.AP, LIB$SIG_TO_RET);
;   111          0179  1 !
;   112          0180  1 !       Or more simply by: ENABLE LIB$SIG_TO_RET ();
;   113          0181  1 !       If there is no need for the handler to do any more processing
;   114          0182  1 !
;   115          0183  1 ! FORMAL PARAMETERS:
;   116          0184  1 !
;   117          0185  1 !       SIG_ARGS_ADR.rl.ra      Adr. of signal args vector
;   118          0186  1 !       MCH_ARGS_ADR.ml.ra      Adr. of mechanism args vector
;   119          0187  1 !       any other args to handler
;   120          0188  1 !
;   121          0189  1 ! IMPLICIT INPUTS:
;   122          0190  1 !
;   123          0191  1 !       NONE
;   124          0192  1 !
;   125          0193  1 ! IMPLICIT OUTPUTS:
;   126          0194  1 !
;   127          0195  1 !       NONE
;   128          0196  1 !
;   129          0197  1 ! COMPLETION CODES:
;   130          0198  1 !
;   131          0199  1 !       SS$_NORMAL if SYS$UNWIND ok, else error codes form SYS$UNWIND.
;   132          0200  1 !
;   133          0201  1 ! SIDE EFFECTS:
;   134          0202  1 !
;   135          0203  1 !       Causes the stack to marked to be unwound to the caller of the
;   136          0204  1 !       establishing procedure of the handler which was called on this signal.
;   137          0205  1 !
;   138          0206  1 !--
;   139          0207  1
;   140          0208  2     BEGIN
;   141          0209  2     MAP
;   142          0210  2         SIG_ARGS_ADR: REF BLOCK[8, BYTE],       ! Signal vector
;   143          0211  2         MCH_ARGS_ADR: REF BLOCK[20, BYTE];      ! mechanism vector
;   144          0212  2
;   145          0213  2     !+
;   146          0214  2     ! If this is unwind condition, just let unwinding continue since
;   147          0215  2     ! propably it was this handler which invoked the unwind.
```

```
;  148          0216  2      !-
;  149          0217  2
;  150          0218  2      IF .SIG_ARGS_ADR[CHF$L_SIG_NAME] EQL SS$_UNWIND THEN RETURN SS$_NORMAL;
;  151          0219  2
;  152          0220  2      !+
;  153          0221  2      ! Copy condition value to saved image of R0
;  154          0222  2      !-
;  155          0223  2
;  156          0224  2      MCH_ARGS_ADR[CHF$L_MCH_SAVR0] =
;  157          0225  3          (SELECTONE .SIG_ARGS_ADR[CHF$L_SIG_NAME] OF
;  158          0226  3              SET
;  159          0227  3              [STR$_FATINTERR] : LIB$_FATERRLIB;
;  160          0228  3              [STR$_ILLSTRCLA] : LIB$_INVSTRDES;
;  161          0229  3              [STR$_INSVIRMEM] : LIB$_INSVIRMEM;
;  162          0230  3              [STR$_STRIS_INT] : LIB$_STRIS_INT;
;  163          0231  3              [OTHERWISE]: .SIG_ARGS_ADR[CHF$L_SIG_NAME];
;  164          0232  2              TES);
;  165          0233  2
;  166          0234  2      !+
;  167          0235  2      ! Set to unwind stack using default depth and default new PC,
;  168          0236  2      ! namely return to caller of the procedure which established the handler
;  169          0237  2      !-
;  170          0238  2
;  171          0239  2      RETURN $UNWIND ();
;  172          0240  1      END;                            ! End of LIB$$STRTO_RET routine


                                                            .TITLE  LIB$$STRTO_RET Library routine to convert a str
;                                                                                  ing signal to a
                                                            .IDENT  \1-002\

                                                            .EXTRN  LIB$_FATERRLIB, LIB$_STRIS_INT
                                                            .EXTRN  LIB$_INSVIRMEM, LIB$_INVSTRDES
                                                            .EXTRN  STR$_FATINTERR, STR$_STRIS_INT
                                                            .EXTRN  STR$_ILLSTRCLA, STR$_INSVIRMEM
                                                            .EXTRN  SYS$UNWIND

                                                            .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2

                                    0000 00000              .ENTRY  LIB$$STRTO_RET, Save nothing                ; 0159
                50          04  AC  D0 00002              MOVL    SIG_ARGS_ADR, R0                            ; 0218
                50          04  A0  D0 00006              MOVL    4(R0), R0                                   ;
00000920        8F              50  D1 0000A              CMPL    R0, #2336                                   ;
                                04  12 00011              BNEQ    1$                                          ;
                50              01  D0 00013              MOVL    #1, R0                                      ;
                                    04 00016              RET                                                 ;
                51          08  AC  D0 00017 1$:          MOVL    MCH_ARGS_ADR, R1                            ; 0224
00000000G       8F              50  D1 0001B              CMPL    R0, #STR$_FATINTERR                         ; 0227
                                09  12 00022              BNEQ    2$                                          ;
                50 00000000G    8F  D0 00024              MOVL    #LIB$_FATERRLIB, R0                         ;
                                34  11 0002B              BRB     5$                                          ;
00000000G       8F              50  D1 0002D 2$:          CMPL    R0, #STR$_ILLSTRCLA                         ; 0228
                                09  12 00034              BNEQ    3$                                          ;
                50 00000000G    8F  D0 00036              MOVL    #LIB$_INVSTRDES, R0                         ;
                                22  11 0003D              BRB     5$                                          ;
00000000G       8F              50  D1 0003F 3$:          CMPL    R0, #STR$_INSVIRMEM                         ; 0229
                                09  12 00046              BNEQ    4$                                          ;
```

```
                            50 00000000G   8F  D0 00048          MOVL    #LIB$_INSVIRMEM, R0                     ;
                                           10  11 0004F          BRB     5$                                      ;
                00000000G   8F             50  D1 00051 4$:     CMPL    R0, #STR$_STRIS_INT                     ; 0230
                                           07  12 00058          BNEQ    5$                                      ;
                            50 00000000G   8F  D0 0005A          MOVL    #LIB$_STRIS_INT, R0                     ;
                      0C    A1             50  D0 00061 5$:     MOVL    R0, 12(R1)                              ; 0225
                                           7E  7C 00065          CLRQ    -(SP)                                   ; 0239
                00000000G   00             02  FB 00067          CALLS   #2, SYS$UNWIND                          ;
                                               04 0006E          RET                                             ; 0240

; Routine Size:  111 bytes,    Routine Base:  _LIB$CODE + 0000
```

```
;  174          0241  1 END                              ! End of module LIB$$STRTO_RET
;  175          0242  0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
_LIB$CODE	111	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:00.7
_255DUA28:[LIBRTL.OBJ]RTLLIB.L32;1	36	0	0	8	00:00.1

COMMAND QUALIFIERS

```
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:LIBSTRRET/OBJ=OBJ$:LIBSTRRET MSRC$:LIBSTRRET/UPDATE=(ENH$:LIBSTRRET
;       )

; Size:          111 code + 0 data bytes
; Run Time:         00:02.6
; Elapsed Time:     00:16.2
; Lines/CPU Min:     5500
; Lexemes/CPU-Min: 11840
; Memory Used:  44 pages
; Compilation Complete
```

LIBSPAWN
LIS

LIBSTATVM
LIS

LIBTRAA2E
LIS

LIBSPANC
LIS

LIBSYMBOL
LIS

LIBTRNLOG
LIS

LIBSKPC
LIS

LIBTIMER
LIS

LIBTPARSE
LIS

LIBTRIMFI
LIS

LIBSTRRET
LIS

LIBTRAE2A
LIS