G 6

```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR        AAAAAAAAA     RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBB         BBB  RRR         RRR  AAA         AAA  RRR         RRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR     AAA         AAA  RRRRRRRRRRRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR   RRR        AAAAAAAAAAAAAAA  RRR   RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLL                    III        BBB         BBB  RRR      RRR     AAA         AAA  RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR  AAA         AAA  RRR         RRR
```

```
PPPPPPPP    RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF    IIIIII    XX      XX
PPPPPPPP    RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF    IIIIII    XX      XX
PP      PP  RR      RR  EE          FF              II      XX      XX
PP      PP  RR      RR  EE          FF              II      XX      XX
PP      PP  RR      RR  EE          FF              II        XX  XX
PP      PP  RR      RR  EE          FF              II        XX  XX
PPPPPPPP    RRRRRRRR    EEEEEEEE    FFFFFFFF        II          XX
PPPPPPPP    RRRRRRRR    EEEEEEEE    FFFFFFFF        II          XX
PP          RR  RR      EE          FF              II        XX  XX
PP          RR  RR      EE          FF              II        XX  XX
PP          RR    RR    EE          FF              II      XX      XX      ....
PP          RR    RR    EE          FF              II      XX      XX      ....
PP          RR      RR  EEEEEEEEEE  FF          IIIIII    XX      XX      ....
PP          RR      RR  EEEEEEEEEE  FF          IIIIII    XX      XX      ....

RRRRRRRR    EEEEEEEEEE    QQQQQQ
RRRRRRRR    EEEEEEEEEE    QQQQQQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RR      RR  EE          QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RRRRRRRR    EEEEEEEE    QQ      QQ
RR  RR      EE          QQ  QQ  QQ
RR  RR      EE          QQ  QQ  QQ
RR    RR    EE          QQ    QQ
RR    RR    EE          QQ    QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
RR      RR  EEEEEEEEEE    QQQQ  QQ
```

```
Require file for the Library command utility

Version:      'V04-000'
```



```
++

FACILITY:  Library command utility

ABSTRACT:

      The VAX/VMS library command utilizes the librarian procedures
      to perform operations on libraries.

ENVIRONMENT:

      VAX native, user mode.

--

AUTHOR:  Benn Schreiber

CREATION DATE: June-1979

MODIFIED BY:

      V03-002 GJA0070         Greg Awdziewicz         6-Mar-1984
              Add symbols to allow text and sharable image libraries
              to have key lengths as large as the new filenames (39).

      V03-001 JWT0103         Jim Teague              22-Mar-1983
```

```
!
!               Add two useful macros, "sd" and "dynamic_descriptor".
!
!--
```

```
!
! Define VMS block structures
!
STRUCTURE
        BBLOCK [O, P, S, E; N] =
                [N]
                (BBLOCK + O) <P, S, E>;
!
! Useful macros
!
MACRO
        sd [a] = bind %name('sd_',a) = $descriptor(a)%,
        dynamic_descriptor = block [dsc$c_s_bln,byte] preset ( [dsc$w_length] = 0,
                                                               [dsc$b_class] = dsc$k_class_d,
                                                               [dsc$a_pointer] = 0 )%,
!
! Macro to generate a pointer to a counted string
!
    CSTRING (STRING) = UPLIT BYTE (%CHARCOUNT (STRING), STRING)%,
!
! Macro to describe a string
!
    STRINGDESC (STRING) = %CHARCOUNT (STRING), UPLIT (%ASCII STRING)%,
!
! Macro to generate a quadword string descriptor
!
    DESCRIPTOR (STRING) = BBLOCK [DSC$C_S_BLN] INITIAL (STRINGDESC (STRING))%,
!
! Macro to generate a counted string
!
    COUNTEDSTRING(STRING) = VECTOR [%CHARCOUNT (STRING)+1, BYTE] INITIAL (BYTE(%CHARCOUNT(STRING),%ASCII STRING ))%,
!
! Macro to execute a given sequence of commands and return if any error
!
    perform (command, errorcode) =
        BEGIN
        LOCAL
            status;
        status = command;
        IF NOT .status                  ! If error detected,
        THEN
            %IF %LENGTH GTR 1           ! If errorcode arg is present
                %THEN SIGNAL (errorcode, %REMAINING, .status);
                %ELSE RETURN .status;           ! Otherwise return with error
            %FI
        END%,
!
! Macro to execute a command which may have and RMS error return
! involving both a status and STV value
!
    rms_perform (command, errorcode, rms_stv) =
        BEGIN
        LOCAL
            status;
        status = command;
        IF NOT .status                  ! If error detected,
```

```
        THEN
            %IF %LENGTH GTR 1           ! If errorcode arg is present
                %THEN SIGNAL (errorcode, %REMAINING, .status, rms_stv); ! then signal the error
                %ELSE RETURN .status;           ! Otherwise just return with error
            %FI
        END%;


!
! $SHR_MESSAGES - a macro which defines facility-specific message codes
!       which are based on the system-wide shared message codes.
!
!       $SHR_MESSAGES ( name, code, (msg,severity), ... )
!
!       where:
!           "name" is the name of the facility (e.g., LIBRAR)
!           "code" is the corresponding facility code (e.g., 134)
!           "msg" is the name of the shared message (e.g., BEGIN)
!           "severity" is the desired message severity (either STS$K_severity
!                                   or severity)
!

MACRO
    $SHR_MESSAGES ( FACILITY_NAME, FACILITY_CODE ) =
        LITERAL
        SHR$MSG_IDS ( FACILITY_NAME, FACILITY_CODE, %REMAINING ); %,

    SHR$MSG_IDS ( FACILITY_NAME, FACILITY_CODE) [ VALUE ] =
        SHR$MSG_CALC ( FACILITY_NAME, FACILITY_CODE, %REMOVE (VALUE) ) %,

    SHR$MSG_CALC ( FACILITY_NAME, FACILITY_CODE, MSG_ID, SEVERITY ) =
        %NAME (FACILITY_NAME,'$_',MSG_ID) = %NAME ('SHR$_',MSG_ID) +
                        FACILITY_CODE*65536 +
                        %IF %DECLARED (%NAME ('STS$K_',SEVERITY))
                                %THEN %NAME ('STS$K_',SEVERITY)
                                %ELSE SEVERITY %FI %;

!
! Librarian message codes from shared message definitions
!

$SHR_MESSAGES (LIB, 134,                        !
                (openin, warning),      ! error opening "x" as input
                (openout, severe),      ! error opening "x" as output
                (closein, warning),     ! error closing "x" as input
                (closeout, warning),    ! error closing "x" as output
                (readerr, error),       ! error reading "x"
                (writeerr, error),      ! error writing "x"
                (parsefail, severe),    ! error parsing "x"
                (nowild, severe),       ! no wild card allowed
                (novalue, severe),      ! no value
                (badkey, severe));      ! Bad key value

!
! Macros to ease typing
!
MACRO
```

```
    SHORT = UNSIGNED (6)%,              ! Short attribute
    BYTLIT = UNSIGNED (8)%,             ! Unsigned byte attribute
    WORDLIT = UNSIGNED (16)%;           ! Unsigned word attribute
!
! Equated symbols
!
LITERAL
    LIB$C_FILENAME_LENGTH=      39,     ! Filename length used for text
                                        ! & sharble image libraries.
    SYM$C_MAXLNG = 31,                  ! Max length of symbols in object/macro libs
    TRUE        =       1,              ! Boolean TRUE
    FALSE       =       0;              ! Boolean FALSE
```

TRANSFER
LIS

DATABASE
LIS

PUTCACHE
LIS

LIBRAR

PREFIX
REQ

LIBRARIAN
MAP

CROSS
LIS

SUBS
LIS

FILEIO
LIS

PADLBR
LIS

COMPRESS
LIS

EXTRACT
LIS

LIB
MDL

DELETE
LIS