



```

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  EEEEEEEEEE  FFFFFFFFFF  IIIIII  XX  XX
PPPPPPPP  RRRRRRRR  EEEEEEEEEE  FFFFFFFFFF  IIIIII  XX  XX
PP  RR  RR  EE  FF  II  XX  XX
PP  RR  RR  EE  FF  II  XX  XX
PP  RR  RR  EE  FF  II  IIIIII  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  EEEEEEEEEE  QQ  QQ  QQ
RRRRRRRR  EEEEEEEEEE  QQ  QQ  QQ
RR  RR  EE  QQ  QQ  QQ
RR  RR  EE  QQ  QQ  QQ
RR  RR  EE  QQ  QQ  QQ
RR  RR  EEEEEEEEEE  QQQQ  QQ
RR  RR  EEEEEEEEEE  QQQQ  QQ

```

.....

Require file for the Library command utility

Version: 'V04-000'

```

*****
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*****

```

++

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 + 0) <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 STSSK_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 ( 'STSSK ', SEVERITY ) )
    %THEN %NAME ( 'STSSK ', 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
  (noval, severe),       ! no value
  (badkey, severe));    ! Bad key value

```

```

! Macros to ease typing

```

```

MACRO

```



SHORT = UNSIGNED (6)%  
BYTLIT = UNSIGNED (8)%  
WORDLIT = UNSIGNED (16)%;

! Short attribute  
! Unsigned byte attribute  
! Unsigned word attribute

! Equated symbols

LITERAL

LIB\$C\_FILENAME\_LENGTH= 39,

! Filename length used for text  
! & sharble image libraries.  
! Max length of symbols in object/macro libs  
! Boolean TRUE  
! Boolean FALSE

SYMSC\_MAXLNG = 31,  
TRUE = 1,  
FALSE = 0;

.....



0200 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

TRANSFER LIS

PUTCACHE LIS

LIBRAR

LIBRARIAN MAP

PREFIX REQ

CROSS LIS

SUBS LIS

PADL BR LIS

COMPRESS LIS

LIB MOL

DATABASE LIS

FILETO LIS

EXTRACT LIS

DELETE LIS