

Require file for the Library access routines

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 access procedures

ABSTRACT:

The VAX/VMS librarian procedures implement a standard access method to libraries through a shared, common procedure set.

ENVIRONMENT:

VAX native, user mode.

--

AUTHOR: Benn Schreiber, Tim Halvorsen

CREATION DATE: June-1979

MODIFIED BY:

--

```
Define VMS block structures
```

```
STRUCTURE
  BBLOCK [O, P, S, E; N] =
    [N]
    (BBLOCK+O)<P,S,E>;
```

```
Define routine linkages
```

```
LINKAGE
  JSB_0 = JSB,
  JSB_1 = JSB (REGISTER = 0),
  JSB_2 = JSB (REGISTER = 0, REGISTER = 1),
  JSB_3 = JSB (REGISTER = 0, REGISTER = 1, REGISTER = 2);
```

```
Useful macros
```

```
MACRO
```

```
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; ! then 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);
  %ELSE
    RETURN .status;    ! then return with error
  %FI
END%;
```

```

: 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
SYM%_MAXLNG = 31,      ! Max length of symbols in object/macro libs
SYM%_SHORTSYM = 15,   ! Max length of short symbols (help library)
TRUE = 1,              ! Boolean TRUE
FALSE = 0;            ! Boolean FALSE
```

CA

SY

AD

AL

CA

CA

CA

CA

CA

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

CT

DE

EM

FI

GE

LB

LB

LB

LO

RE

WR

PS

--

:

S

S

The grid contains 100 individual panels, each displaying a different technical visualization or code snippet. The panels are arranged in a 10x10 grid. Some panels are more prominent than others, with larger text or distinct graphical elements. The following table lists the labels for the most prominent panels:

Row	Column	Label
1	8	CACHE LIS
1	9	DATA LIS
3	5	LBR
4	5	LBRSHR MAP
6	7	OLD.LIBFMT MDL
6	8	DUMP LIS
8	5	LBR MDL
8	7	PREFIX REQ
8	8	LBRUSR MDL