


```

SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ
SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ
SS M M M M BB BB RR RR EE QQ QQ
SS M M M M BB BB RR RR EE QQ QQ
SS M M M M BB BB RR RR EE QQ QQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ
SSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ
SS MM MM BB BB RR RR EE QQ QQ
SS MM MM BB BB RR RR EE QQ QQ
SS MM MM BB BB RR RR EE QQ QQ
SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ
SSSSSSSS MM MM BBBB BBBB RRRRRRRR EEEEEEEEEEE QQQQQQ

```

```

RRRRRRRR EEEEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RRRRRRRR EEEEEEEEEEE QQQQQQ
RRRRRRRR EEEEEEEEEEE QQQQQQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EE QQ QQ
RR RR EEEEEEEEEEE QQQQ QQ
RR RR EEEEEEEEEEE QQQQ QQ

```

Require file for print symbiont facility

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:
 Symbiont Services.

ABSTRACT:
 Macro and literal definitions for symbionts.

AUTHOR: G. Robert, **CREATION DATE:** 01-May-1983

MODIFIED BY:

3B-007	RRB3007	Rowland R. Bradley	20-Jul-1984
	Change PSMS_ messages NOMOREITEMS, INVSTRLEV, and INVSTMNBR to SMBS_NOMOREITEMS, SMBS_INVSTRLEV, and SMBS_INSTRMNR, respectively. Delete messages PSMS_INVSCB, PSMS_REQNOTSUP, PSMS_INVREQCOD.		
3B-006	GRR3006	Gregory R. Robert	29-Apr-1984
	Added PSMS_FLUSH		
3B-005	RRB0014	Rowland R. Bradley	27-Apr-1984
	Remove the Task_flag macro.		
3B-004	RRB0013	Rowland R. Bradley	21-Feb-1984
	Add new STRUCTURE definition for PAGE.		

```

3B-003 GRR3004 Gregory R. Robert 23-Aug-1983
Bugfixes, page_setup_modules, form_setup_modules,
sheet_feed, symbiont_initiated_pause_task and stop_stream,
hangup code, read and write item services

3B-003 GRR3003 Gregory R. Robert 03-Aug-1983
Fixes for new design.

3B-002 GRR3002 Gregory R. Robert 29-Jul-1983
Added several macros to access symbiont tables.

3B-001 GRR3001 Gregory R. Robert 23-Jun-1983
Fixed offset_table macro, added item_present macro,
fixed some bugs, added several literals.

3B-000 GRR3000 Gregory R. Robert 27-May-1983
Original version.

```

```

**
: Define program section standard names and attributes

```

```

PSECT

```

```

CODE = CODE,
PLIT = CODE,
OWN = DATA,
GLOBAL = DATA
:

```

```

: Check that structure id's have a common byteoffset

```

```

$ASSUME ($BYTEOFFSET (IOB_L_STRUCTURE), EQL, $BYTEOFFSET (PSM$L_STRUCTURE))

```

```

: Check that a quadword of item code flags is adequate

```

```

$ASSUME (SMBMSG$K_MAX_ITEM_CODE - 1, LSS, 64)

```

```

: Delcare useful builtin Bliss functions

```

```

BUILTIN

```

```

CALLG,
FFS,
INSQUE,
LOCC,
MOVCS,
MOVCS,
MOVTUC,
REMOUE,
TESTBITCC,
TESTBITCS,
TESTBITSC,
TESTBITSS
:

```

```

: Declare special linkages

```

```
LINKAGE  
ANALYZE_LINKAGE = JSB (REGISTER=0; REGISTER=0, REGISTER=1),
```

```
!! CAN'T USE THIS UNTIL UNDERSTAND HOW TO PRESERVE R2-R4 INCLUSIVE
```

```
!! FREE_LINKAGE = JSB (REGISTER=0)
```

```
! Declare common external routines
```

```
EXTERNAL ROUTINE  
BASSEDIT,
```

```
LBR$CLOSE,  
LBR$GET_RECORD,  
LBR$INI_CTRLCL,  
LBR$LOOKUP_KEY,  
LBR$OPEN,  
LBR$RET_RMSSTV,  
LBR$SET_LOCATE,
```

```
LIB$TRIM_FILESPEC,  
LIB$GET_VM,  
LIB$FREE_VM,
```

```
STR$ANALYZE_SDESC,  
STR$ANALYZE_SDESC_R1 : ANALYZE_LINKAGE,  
STR$APPEND,  
STR$CONCAT,  
STR$COPY_DX,  
STR$COPY_R,  
STR$FREE_DX,  
STR$FREE1_DX_R4 : FREE_LINKAGE,  
STR$GET1_DX,  
STR$LEFT,  
STR$PREFIX,  
STR$RIGHT  
;
```

```
EXTERNAL LITERAL  
PSM$S_HANGUP_DISPATCH_ENTRY  
;
```

```
! Private messages
```

```
EXTERNAL LITERAL  
PSM$_BUFFEROVF,  
PSM$_EOF,  
PSM$_ESCAPE,  
PSM$_FLUSH,  
PSM$_FUNNOTSUP,  
PSM$_INVITMCOU,  
PSM$_INVVMSOSC,
```

```

PSMS_MODNOTFND,
PSMS_NEWPAGE,
PSMS_NOFILEID,
PSMS_OSCTOOLON,
PSMS_PENDING,
PSMS_SUSPEND,
PSMS_TOOMANYLEV,
SMBS_INVSTMNBR,
SMBS_INVSTRLEV,
SMBS_NOMOREITEMS
;

```

```

! Shared messages
;

```

```

$SHR_MSGDEF (PSM, PSMSK FACILITY, LOCAL,
(BA)LOGI(PC, SEVERE),      - logic error with PC value
(CLOSEIN, ERROR),        - unable to close input
(OPENIN, ERROR),         - unable to open or connect to input
(READERR, ERROR),        - error reading
(WRITEERR, ERROR)        - error writing
);

```

```

! Define structures useful for accessing parameters passed by reference
;

```

```

STRUCTURE
$BYTE          [] = $BYTE <0,08,0>,
$SIGNED_BYTE  [] = $SIGNED_BYTE <0,08,1>,

$WORD         [] = $WORD <0,16,0>,
$SIGNED_WORD  [] = $SIGNED_WORD <0,16,1>,

$LONGWORD     [] = $LONGWORD <0,32,0>,
$SIGNED_LONGWORD [] = $SIGNED_LONGWORD <0,32,1>
;

```

```

! Define structures useful for referencing the 'page' of information
;

```

```

STRUCTURE
PAGE_ARRAY[I, J, K; N, M, UNITS=1] =      ! default is byte referencing
[M * N * UNITS]
(PAGE_ARRAY + (J * K + I) * UNITS)<0,8,0>
;

```

```

! Message Item Table (MIT) and Service Routine (SRV) table building macros
;

```

```

MACRO
MIT_PRESET [TAG, RESET, TYPE, ITEM] =
  [%NAME ('SMBMSGSK-', ITEM), MIT_B_TYPE] = %NAME ('MIT_V-', TYPE),
  [%NAME ('SMBMSGSK-', ITEM), MIT_V_RESET] = %IF %NULL (RESET) %THEN 1 %ELSE RESET %FI,
  [%NAME ('SMBMSGSK-', ITEM), MIT_W_OFFSET] =
    $BYTEOFFSET (%NAME ('PSMS', TAG, '-', ITEM))
% ;

```

```

MACRO

```

```

SRV_PRESET [SERVICE, USER, TYPE] =
  [%NAME ('PSM$K', SERVICE), SRV_A_SERVICE] =
    %NAME ('PSM$', SERVICE),
  [%NAME ('PSM$K', SERVICE), SRV_V_USER_ALLOWED] =
    %IF %NULL (USER) %THEN NO_USER
    %ELSE USER %FI,
  [%NAME ('PSM$K', SERVICE), SRV_B_SERVICE_TYPE] =
    %IF %NULL (TYPE) %THEN SRV_R_GENERAL_SERVICE
    %ELSE %NAME ('SRV_K_', TYPE, '_SERVICE') %FI
% ;

```

```

! General purpose macros
!

```

```

MACRO

```

```

ACC_DATA (ITEM) =
  %$BLOCK [SCB[PSM$T_ACCOUNTING_AREA], %NAME ('SMBMSG$L_', ITEM)]
% ,

```

```

BLINK (QUEUE HEADER) =
  VECTOR [QUEUE_HEADER, 1]
% ,

```

```

CLEAR_QUAD_ (QUAD) =
  BEGIN
  VECTOR [QUAD, 0] = 0;
  VECTOR [QUAD, 1] = 0;
  END
% ,

```

```

CLEAR_STRING_ (DESC) =
  BEGIN
  IF .DESC_CLASS_ (DESC) LEQU DSC$K_CLASS_S ! 0 or 1
  THEN
    INIT_DYN_DESC_ (DESC)
  ELSE
    IF .DESC_SIZE_ (DESC) NEQ 0
    THEN
      STR$FREE1_DX (DESC)
    END
  END
% ,

```

```

CODEERR =
  SIGNAL_STOP (PSM$BADLOGICPC, 1) ! PC implied 3rd arg
% ,

```

```

COPY_DX_DX_ (FROM_DESC, TO_DESC) =
  SIGNAL_IF_ERROR_ (STR$COPY_DX (TO_DESC, FROM_DESC))
% ,

```

```

COPY_R_DX_ (FROM_SIZE, FROM_ADDRESS, TO_DESC) =
  SIGNAL_IF_ERROR_ (STR$COPY_R (TO_DESC, FROM_SIZE, FROM_ADDRESS))
% ,

```

```

COPY_QUAD_ (FROM_QUAD, TO_QUAD) =
  BEGIN

```

BAN
Syn

AD
ADF
ALF
ARC
BLA
CHA
COL
COL
DEC
DOL
DOT
HEI
LIN
NUM
PSM
RE1
SCE
SPA
SPA
SS1
UNC
WIC

PSE

: E
\$AE

Pha

In
Com
Pas
Syn
Pas
Syn
Pse
Cro
As

The
19
The
27
9

```

VECTOR [TO_QUAD, 0] = .VECTOR [FROM_QUAD, 0];
VECTOR [TO_QUAD, 1] = .VECTOR [FROM_QUAD, 1];
END

```

*,

```

DECREMENT_ (VALUE) =
  BEGIN
  VALUE = .VALUE - 1;
  END

```

*,

```

DESC_ADDR_ (DESC) =
  $BLOCK [DESC, DSC$A_POINTER]

```

*,

```

DESC_CLASS_ (DESC) =
  $BLOCK [DESC, DSC$B_CLASS]

```

*,

```

DESC_SIZE_ (DESC) =
  $BLOCK [DESC, DSC$W_LENGTH]

```

*,

```

$DYNAMIC DESC =
  $BLOCK [8] PRESET (
    [DSC$W_LENGTH]      = 0,
    [DSC$B_DTYPE]       = DSC$K_DTYPE_T,
    [DSC$B_CLASS]       = DSC$K_CLASS_D,
    [DSC$A_POINTER]     = 0
  )

```

*,

```

FLINK_ (QUEUE_HEADER) =
  VECTOR [QUEUE_HEADER, 0]

```

*,

```

INCREMENT_ (VALUE) =
  BEGIN
  VALUE = .VALUE + 1
  END

```

*,

```

INIT_DYN_DESC_ (DESC) =
  BEGIN
  BIND X_DESC = DESC: $BLOCK;
  X_DESC [DSC$W_LENGTH] = 0;
  X_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
  X_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
  X_DESC [DSC$A_POINTER] = 0;
  END

```

*,

```

INIT_QUEUE_HEADER_ (QUEUE_HEADER) =
  BEGIN
  FLINK_ (QUEUE_HEADER) = QUEUE_HEADER;
  BLINK_ (QUEUE_HEADER) = QUEUE_HEADER;

```

Mac

-\$2

-\$2

TOT

384

The

MAC


```

    END
X,
INIT_STAT_DESC_ (DESC, LENGTH, POINTER) =
    BEGIN
    BIND X_DESC = DESC: $BLOCK;
    X_DESC [DSC$W_LENGTH] = LENGTH;
    X_DESC [DSC$B_DTYPE] = 0;
    X_DESC [DSC$B_CLASS] = 0;
    X_DESC [DSC$A_POINTER] = POINTER;
    END
X,
INSERT_HEAD (ENTRY_ADDR_, QUEUE_HEADER) =
    INSQUE (ENTRY_ADDR_, .FLINK_ (QUEUE_HEADER))
X,
INSERT_TAIL (ENTRY_ADDR_, QUEUE_HEADER) =
    INSQUE (ENTRY_ADDR_, .BLINK_ (QUEUE_HEADER))
X,
ITEM_PRESENT (ITEM_CODE) =
    BITVECTOR [SCB[PSM$Q_ITEM_FLAGS], %NAME ('SMBMSG$K_', ITEM_CODE)]
X,
OFFSET_TABLE_REPEAT_ [OFFSET, FIRST_BIT, SIZE, SIGN] =
    OFFSET
X,
PARAMETER_INDEX_ [] =
    BUILTIN NUL[PARAMETER;
    LITERAL PARAMETER_INDEX_REPEAT_ (%REMAINING)
X,
PARAMETER_INDEX_REPEAT [PARAMETER] =
    %NAME ('_P_', PARAMETER) = %COUNT + 1
X,
PARAMETER_PRESENT (PARAM) =
    NOT NULLPARAMETER (%NAME ('_P_', PARAM))
X,
PRINT_FLAG (FLAG_NAME) =
    $BLOCK [SCB[PSM$L_PRINT_FLAGS], %NAME ('SMBMSG$V_', FLAG_NAME)]
X,
READ_CHAR_ =
    BEGIN
    DECREMENT (SCB_SIZE_ (INPUT_RECORD));
    CHRCHAR_A (SCB_ADDR_ (INPUT_RECORD))
    END
X,
REMOVE_HEAD (RESULT, QUEUE_HEADER) =
    REMQUE (.FLINK_ (QUEUE_HEADER), RESULT)
X,
```

```
REMOVE_TAIL (RESULT, QUEUE_HEADER) =
  REMOVE_T.BLINK_ (QUEUE_HEADER), RESULT)
%,

REQUEST_FLAG (FLAG_NAME) =
  $BBLOCK [SCB[PSM$L_REQUEST_CONTROL], %NAME ('SMBMSG$V_', FLAG_NAME)]
%,

SEPARATE_FLAG (FLAG_NAME) =
  $BBLOCK [SCB[PSM$L_SEPARATION_CONTROL], %NAME ('SMBMSG$V_', FLAG_NAME)]
%,

SERVICE_LIST (SERVICE) =
  BITVECTOR [SCB[PSM$L_SERVICE_LIST], %NAME ('PSM$K_', SERVICE)]
%,

RETURN_IF_ERROR_ (ACTION) =
  BEGIN
  LOCAL STATUS;
  STATUS = ACTION;
  IF NOT .STATUS THEN RETURN (.STATUS);
  .STATUS
  END
%,

SET_DYN_DESC_ (DESC) =
  BEGIN
  $BBLOCK [SCB[%NAME ('PSM$Q_', DESC)], DSC$B_DTYPE] = DSC$K_DTYPE_T;
  $BBLOCK [SCB[%NAME ('PSM$Q_', DESC)], DSC$B_CLASS] = DSC$K_CLASS_D;
  END
%,

SIGNAL_IF_ERROR_ (ACTION) =
  BEGIN
  LOCAL STATUS;
  STATUS = ACTION;
  IF NOT .STATUS THEN SIGNAL (.STATUS);
  .STATUS
  END
%,

PSM$L_ = 0,0,32,0
%,

SCB_ADDR_ (DESC) =
  DESC_ADDR_ (SCB [ %NAME ('PSM$Q_', DESC) ])
%,

SCB_CLASS_ (DESC) =
  DESC_CLASS_ (SCB [ %NAME ('PSM$Q_', DESC) ])
%,

SCB_SIZE_ (DESC) =
  DESC_SIZE_ (SCB [ %NAME ('PSM$Q_', DESC) ])
%,
```

```
WRITE_CHAR_ (CHAR) =  
  BEGIN  
    CH$WCHAR_A (CHAR, SCB_ADDR (OUTPUT_BUFFER));  
    DECREMENT_ (SCB_SIZE_ (OUTPUT_BUFFER));  
  END
```

⌘
:

.....

0309 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

This image displays a grid of 12x12 small thumbnail images, each representing a different software interface or output from the AH-BT13A-SE VAX/VMS V4.0 system. The thumbnails are arranged in a regular grid pattern. Several thumbnails are clearly labeled with titles and 'LIS' (likely listing) or 'REQ' (request) suffixes:

- BANNER LIS**: Located in the second row, fourth column.
- PLWRITE LIS**: Located in the third row, second column.
- SMBREQ REQ**: Located in the fourth row, fourth column.
- PLVECTOR LIS**: Located in the fifth row, second column.
- SMBRSUSHR MAP**: Located in the fifth row, third column.
- PLTIRODATA LIS**: Located in the sixth row, first column.
- SMBDEF SOL**: Located in the sixth row, fourth column.
- PLSTRNG LIS**: Located in the tenth row, first column.
- PRTSMB MAP**: Located in the tenth row, second column.
- DISPATCH LIS**: Located in the eleventh row, fourth column.
- FORMAT LIS**: Located in the seventh row, tenth column.

The other thumbnails in the grid show various graphical displays, including bar charts, data tables, and command-line text outputs. The overall appearance is that of a technical manual or a software catalog page.