PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTTTTTTTTTTTTTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMMMMM MMMMMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPP PPP RRR RRR TTT SSS MMM MMM MMM BBB BBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPPPPPPPPPPP RRRRRRRRRRR TTT SSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSS MMM MMM BBB BBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB
PPP RRR RRR TTT SSSSSSSSSSS MMM MMM BBBBBBBBBBB

FILEID**SMBDEF

```
  SSSSSSSS  MM       MM  BBBBBBBB    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
  SSSSSSSS  MM       MM  BBBBBBBB    DDDDDDDD    EEEEEEEEEE  FFFFFFFFFF
SS          MMMM   MMMM  BB      BB  DD      DD  EE          FF
SS          MMMM   MMMM  BE      BB  DD      DD  EE          FF
SS          MM  MM  MM  BB      BB  DD      DD  EE          FF
SS          MM  MM  MM  BB      BB  DD      DD  EE          FF
  SSSSSS    MM      MM  BBBBBBBB    DD      DD  EEEEEEEE    FFFFFFFF
  SSSSSS    MM      MM  BBBBBBBB    DD      DD  EEEEEEEE    FFFFFFFF
        SS  MM      MM  BB      BB  DD      DD  EE          FF
        SS  MM      MM  BB      BB  DD      DD  EE          FF
        SS  MM      MM  BB      BB  DD      DD  EE          FF              ....
        SS  MM      MM  BB      BB  DD      DD  EE          FF              ....
SSSSSSSS    MM      MM  BBBBBBBB    DDDDDDDD    EEEEEEEEEE  FF              ....
SSSSSSSS    MM      MM  BBBBBBBB    DDDDDDDD    EEEEEEEEEE  FF              ....


  SSSSSSSS  DDDDDDDD    LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
SS          DD      DD  LL
  SSSSSS    DD      DD  LL
  SSSSSS    DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
        SS  DD      DD  LL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
SSSSSSSS    DDDDDDDD    LLLLLLLLLL
```

SMB

LIN

!*
*
!*

D

EXT

EXT

F

EXT

```
module PRTSMB;                  /* Private symbiont definitions

/*
/* Symbolic definitions for symbionts
/*
/*      Version 'V03-016'
/*
/*****************************************************************************
/*                                                                           *
/*  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:  Print symbiont
/*
/* ABSTRACT:  Data Structure Definitions
/*
/* ENVIRONMENT: Non-executable.
/*
/* AUTHOR:  Gregory R. Robert, August, 1982
/*
/* MODIFIED BY:
/*
/*      V03-016 GRR0016         Gregory R. Robert       09-May-1984
/*              Move hangup flag into stream flags so that it is not
/*              cleared at start task.
/*
/*      V03-015 GRR0015         Gregory R. Robert       29-Apr-1984
/*              FT2 bugfixes.  Added header_built flag, and resorted
/*              various fields.
/*
/*      V03-014 RRB0014         Rowland R. Bradley      27-Apr-1984
/*              Add dummy longwords for future work during Version V04-000
/*              Reorganize the SCB data structure for clarity.
/*
/*      V03-013 RRB0013         Rowland R. Bradley      23-Feb-1984
/*              Add definitions need for enhancement of print symbiont.
```

```
/*              Add constants for accessing and describing these definitions.
/*
/*      V03-012 GRR0009         Gregory R. Robert       29-Sep-1983
/*              Remove carriage control definitions -- now in lib.l32.
/*
/*      V03-011 GRR0008         Gregory R. Robert       28-Sep-1983
/*              Remove use of S_ACCOUNTING_DATA and S_CHECKPOINT_DATA
/*
/*      V03-010 GRR0005         Gregory R. Robert       26-Sep-1983
/*              Move public definitions to SYSDEF.SDL (lib).
/*
/*      3B-009  GRR3009         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-008  GRR3008         Gregory R. Robert       03-Aug-1983
/*              Rewrite for new design.
/*
/*      3B-007  GRR3007         Gregory R. Robert       29-Jul-1983
/*              Converted previx from SMB_ to PSM$ and many other revisions.
/*
/*      3B-006  GRR3006         Gregory R. Robert       23-Jun-1983
/*              Basic aligning, positioning, searching, checkpointing,
/*              and module handling.
/*
/*      3B-005  GRR3005         Gregory R. Robert       27-May-1983
/*              Substantially revised for FT1.
/*
/*      3B-004  GRR3004         Gregory R. Robert       24-Feb-1983
/*              Significant revision to accomodate new 3B features.
/*
/*      3B-003  GRR3003         Gregory R. Robert       28-Feb-1983
/*              Specify user parameter items and extend print flags
/*              to include alignment and rewind features.  Removed
/*              extra size definitions now generated by SDL.
/*
/*      3B-002  GRR3002         Gregory R. Robert       19-JAN-1983
/*              Give explicit length to structures declared 'character'
/*
/*
```

```
/* ************************************************************************
/* *            REMAINING DEFINTIONS ARE VMS-PRTSMB PRIVATE               *
/* ************************************************************************

/*
/* Miscellaneous constants
/*
constant PSM$K_CHECKPOINT_LIMIT equals 20;
constant PSM$K_DEFBUFSIZ        equals 512;
constant PSM$K_FACILITY         equals 262;
constant PSM$K_FIRST_CHAR_USED  equals 3;
constant PSM$K_MAXSTREAMS       equals 32;
constant PSM$K_NUMOUTBUF        equals 3;
/*
constant PSM$K_MAX_PAGE_SIZE    equals 8712;    /* Largest possible page
constant PSM$K_FORM_132x66      equals 1;       /*
constant PSM$K_FORM_132x51      equals 2;
constant PSM$K_FORM_80x66       equals 3;
constant PSM$K_FORM_80x51       equals 4;
constant PSM$K_FORM_40xANY      equals 5;
constant PSM$K_FORM_NON_STD     equals 0;
/*
/*      Standard carriage controls --
/*          leading line feed, trailing carriage return
/*          leading form feed, trailing carriage return
/*          leading form feed
/*

constant PSM$K_LF_CR    equals %X0D010A01;
constant PSM$K_FF_CR    equals %X0D010C01;
constant PSM$K_FF       equals %X00000C01;


/*
/*      Format effector codes
/*
constant (

    BS,         /* Backspace
    HT,         /* Horizontal tab
    LF,         /* Line feed
    VT,         /* Vertical tab
    FF,         /* Form feed
    CR          /* Carriage return

) equals 8 increment 1 prefix PSM$ tag K_CHAR;

constant PSM$K_CHAR_APC equals 095;     /* "_" Application Program Command
constant PSM$K_CHAR_CSI equals 091;     /* "[" Control String Introducer
constant PSM$K_CHAR_DCS equals 080;     /* "P" Device Control String
constant PSM$K_CHAR_ESC equals 027;     /* "" ESCape
constant PSM$K_CHAR_OSC equals 093;     /* "]" Operating System Command
constant PSM$K_CHAR_PM  equals 094;     /* "^" Privacy Message
constant PSM$K_CHAR_ST  equals 092;     /* "\" String Terminator
```

```
/*
/*      Structure types
/*
constant (

    DSB,        /* Dynamic string block
    IOB,        /* IO control block
    SCB         /* Stream control block

) equals 1 increment 1 prefix PSM$ tag K_STRUCTURE;


/*
/*      Dynamic String Block
/*
aggregate DSB structure prefix DSB_;
    QLINKS      quadword unsigned;              /* Queue links
    DESC        quadword unsigned;              /* Descriptor
end;


/*
/*      Input Output control Block
/*
aggregate IOB structure prefix IOB_;
    QLINKS      quadword unsigned;              /* Output queue buffer links
    STRUCTURE   structure longword unsigned;    /* Structure ID
        TYPE        byte unsigned;              /*  - type
        LEVEL       byte unsigned;              /*  - level
        SIZE        word unsigned;              /*  - size
        end;
    IOSB        structure quadword unsigned;    /* IOSB
        IO_STATUS   word unsigned;              /*  - word 1
        IO_LENGTH   word unsigned;              /*  - word 2
        IO_3        word unsigned;              /*  - word 3
        IO_4        word unsigned;              /*  - word 4
        end;
    CONTEXT     address;                        /* User context
    ROUTINE     address;                        /* Routine address
    BUFFER      quadword unsigned;              /* Buffer descriptor
    RECORD      quadword unsigned;              /* Record descriptor
    FLAGS       structure longword unsigned;    /* IOB flags
      CHECKPOINT_PENDING bitfield mask;         /*  - checkpoint pending
      FLUSH_PENDING      bitfield mask;         /*  - flushing output stream
      PASSALL            bitfield mask;         /*  - write noformat
      PAUSE_PENDING      bitfield mask;         /*  - respond when IO complete
      FILLER             bitfield length 32-^ fill;
      end;
    CHECKPOINT_DATA character length 24;        /* Checkpoint
end;


/*
/*      Message Item Table entry structure
/*
aggregate MIT structure prefix MIT_;
```

```
    TYPE        byte;                           /* Item type (SIGNED)
        constant BYTE       equals 1;           /*  - byte
        constant WORD       equals 2;           /*  - word
        constant LONGWORD   equals 4;           /*  - longword
        constant QUADWORD   equals 8;           /*  - quadword

        constant DYNAMIC    equals -1;          /*  - dynamic descriptor
        constant STATIC     equals -2;          /*  - static descriptor

    FLAGS               ture byte unsigned;     /* IO flags
        RESET           ield mask;              /*  - reset at start of task
        FILLER      field length 08-^ fill;
        end;

    OFFSET          rd unsigned;                /* Offset to item in structure
end;

/*
/*      Servi    outine Table entry structure
/*
aggregate SRV structure prefix SRV_;
    SERVICE     address;                        /* Address of service routine
    USER_ARGUEMENT longword unsigned;           /* User supplied arguement

    FLAGS       structure longword unsigned;    /* Service flags
        USER_ALLOWED    bitfield mask;          /*  - user may provide service
        USER_SUPPLIED   bitfield mask;          /*  - user has provided service
        FILLER  bitfield length 08-^ fill;
        END;
    SERVICE_TYPE byte unsigned;                 /* Type of service routine
        constant (                              /* Define service types
            FILE_SERVICE,                       /*  - file processor
            FORMAT_SERVICE,                     /*  - formatting routine
            GENERAL_SERVICE,                    /*  - other input routines
            LIBRARY_SERVICE,                    /*  - library routines
            OUTPUT_SERVICE                      /*  - output routine
        ) equals 1 increment 1;

    RESERVED    character length 3 tag Z;       /* (fill out the longword)
end;
```

```
/*
/*      Stream Control Block
/*
aggregate SCB structure prefix PSM$;

    /*
    /* Standard structure header
    /*
    QLINKS      quadword unsigned;              /* Structure queue links
    STRUCTURE           structure longword unsigned;    /* Structure ID
        TYPE                byte unsigned;              /*  - type
        LEVEL               byte unsigned;              /*  - level
        SIZE                word unsigned;              /*  - size
        end;

    /*
    /*  Symbiont state flags
    /*
    STREAM_FLAGS        structure longword unsigned;    /* Stream state flags
        ACTIVE              bitfield mask;      /*  - Queue is started
        EXPAND_TABS         bitfield mask;      /*  - Expand tabs to spaces
        HANGUP              bitfield mask;      /*  - data set hangup
        JOB_INITIALIZE      bitfield mask;      /*  - Job reset pending
        FILLER              bitfield length 32-^ fill;
        end;

    TASK_FLAGS          structure longword unsigned;    /* Job state flags
        ALIGN               bitfield mask;      /*  - alignment in progress
        BUSY                bitfield mask;      /*  - task in progress
        EOF                 bitfield mask;      /*  - force end of file
        ESCAPE_IN_PROGRESS  bitfield mask;      /*  - escape sequence in progress
        FAB_VALID           bitfield mask;      /*  - FAB connect succeeded
        FIRST_RECORD        bitfield mask;      /*  - first record of input service
        IMPLICIT_FORMFEED   bitfield mask;      /*  - implicit formfeed detected
        NEW_RECORD          bitfield mask;      /*  - new input record
        PAGE_HEADER_BUILT   bitfield mask;      /*  - Page header template built
        READ_OFFSET         bitfield mask;      /*  - offset next read
        RESET               bitfield mask;      /*  - STOP/NEXT or STOP/RESET in progress
        RESUME_WAIT         bitfield mask;      /*  - waiting for resume command
        SEARCH_FOR_PAGE     bitfield mask;      /*  - page search in progress
        SEARCH_FOR_STRING   bitfield mask;      /*  - string search in progress
        SUPPRESS_OUTPUT     bitfield mask;      /*  - output suppressed
        FILLER              bitfield length 32 - ^ fill;
        end;

/*
/* Message items
/*
ACCOUNTING_DATA     quadword unsigned;      /* Accounting information
ACCOUNT_NAME        quadword unsigned;      /* Account name
AFTER_TIME          quadword unsigned;      /*   /AFTER
ALIGNMENT_PAGES     longword unsigned;      /*   /ALIGN
BOTTOM_MARGIN       longword unsigned;      /* Trailing blank lines
CHARACTERISTICS     quadword unsigned;      /*   /CHARACTERISTICS
CHECKPOINT_DATA     quadword unsigned;      /* Checkpoint information
CONDITION_VECTOR    quadword unsigned;      /* Error messages
```

```
    DEVICE_NAME             quadword unsigned;      /*   /ON
    DEVICE_STATUS           longword unsigned;      /* Device status
    ENTRY_NUMBER            longword unsigned;      /* Job entry number
    EXECUTOR_QUEUE          quadword unsigned;      /*   /QUEUE
    FILE_COPIES             longword unsigned;      /*   /COPIES
    FILE_COUNT              longword unsigned;      /* Current file copy number
    FILE_SETUP_MODULES      quadword unsigned;      /*   /SETUP (print)
    FIRST_PAGE              longword unsigned;      /*   /PAGES
    FORM_LENGTH             longword unsigned;      /*   /LENGTH
    FORM_SETUP_MODULES      quadword unsigned;      /*   /SETUP (define/form)
    FORM_NAME               quadword unsigned;      /* Name of physical form
    FORM_WIDTH              longword unsigned;      /*   /WIDTH
    FILE_IDENTIFICATION     quadword unsigned;      /* Device, fid, and did
    FILE_SPECIFICATION      quadword unsigned;      /* File name
    JOB_COPIES              longword unsigned;      /*   /JOB_COUNT
    JOB_COUNT               longword unsigned;      /* Current job copy number
    JOB_NAME                quadword unsigned;      /*   /NAME
    JOB_RESET_MODULES       quadword unsigned;      /*   /SEPARATION
    LAST_PAGE               longword unsigned;      /*   /PAGES
    LEFT_MARGIN             longword unsigned;      /*   /LEFT_MARGIN
    LIBRARY_SPECIFICATION quadword unsigned;        /*   /LIBRARY
    MAXIMUM_STREAMS         longword unsigned;      /* Maximum streams supported
    MESSAGE_VECTOR          quadword unsigned;      /* Error messages to print
    NOTE                    quadword unsigned;      /*   /NOTE
    PAGE_SETUP_MODULES      quadword unsigned;      /* To be implemented
    PARAMETER_1             quadword unsigned;      /*   /PARAMETER
    PARAMETER_2             quadword unsigned;      /*   /PARAMETER
    PARAMETER_3             quadword unsigned;      /*   /PARAMETER
    PARAMETER_4             quadword unsigned;      /*   /PARAMETER
    PARAMETER_5             quadword unsigned;      /*   /PARAMETER
    PARAMETER_6             quadword unsigned;      /*   /PARAMETER
    PARAMETER_7             quadword unsigned;      /*   /PARAMETER
    PARAMETER_8             quadword unsigned;      /*   /PARAMETER
    PRINT_CONTROL           longword unsigned;      /* Printing control
    PRIORITY                longword unsigned;      /* Queue priority
    QUEUE                   quadword unsigned;      /*   /QUEUE
    REFUSE_REASON           quadword unsigned;      /* Reason task refused
    RELATIVE_PAGE           longword;               /*   /BACKWARD, /FORWARD values
    REQUEST_CONTROL         longword unsigned;      /* Request control
    REQUEST_RESPONSE        longword unsigned;      /* Request being responded to
    RIGHT_MARGIN            longword unsigned;      /*   /RIGHT_MARGIN
    SEARCH_STRING           quadword unsigned;      /*   /SEARCH value
    SEPARATION_CONTROL      longword unsigned;      /* Separation control
    STOP_CONDITION          longword unsigned;      /*   /ABORT or /REQUEUE
    TIME_QUEUED             quadword unsigned;      /* Time queued
    TOP_MARGIN              longword unsigned;      /*   /TOP_MARGIN
    UIC                     longword unsigned;      /* UIC of submittor
    USER_NAME               quadword unsigned;      /* Username

    /*
    /* Queue headers
    /*
    BUFFER_QUEUE            quadword unsigned;      /* Buffer queue headers
    CHECKPOINT_QUEUE        quadword unsigned;      /* Checkpoint queue headers
    INPUT_QUEUE             quadword unsigned;      /* Nested input queue
```

```
    /*
    /* Miscellaneous
    /*
    CHANNEL             longword unsigned;      /* Output channel number
    CHECKPOINT          address;                /* Address of checkpoint
    COLUMN              longword unsigned;      /* Current column number
    CONDITION_TEXT      quadword unsigned;      /* Expanded error mesages
    DEVICE_CLASS        longword unsigned;      /* Device class
    DEVICE_TYPE         longword unsigned;      /* Device type
    FUNCTION_ARGUMENT   longword unsigned;      /* Work area for service calls
    IOB                 address;                /* I/O block address
    ITEM_FLAGS          quadword unsigned;      /* Item flags
    KEEP_ALIVE          longword unsigned;      /* Keep alive count
    L_MARGIN            longword unsigned;      /* Current left margin
    LAST_ENTRY_NUMBER   longword unsigned;      /* Entry number of previous job
    LIBRARY_INDEX       longword unsigned;      /* Library context
    LINE                longword unsigned;      /* Current line number
    MODULE_LIST         quadword unsigned;      /* Library module list
    MODULE_NAME         quadword unsigned;      /* Current library module
    NON_AST_STATUS      longword unsigned;      /* Status from non-ast routines
    OUTPUT_BUFFER       quadword unsigned;      /* Output buffer descriptor
    OUTPUT_QIOS         longword unsigned;      /* Qio's output
    PAGE                longword unsigned;      /* Page number
    PAGE_HEADER         quadword unsigned;      /* Page header descriptor
    PAGE_LENGTH         longword unsigned;      /* Current Page Length
    PAGE_POINTER        address;                /* Address of page array buffer
    PAGE_WIDTH          longword unsigned;      /* Current Page Width
    PRINT_FLAGS         longword unsigned;      /* Print format flags
    RFA                 quadword;               /* Record file address
    SEARCH_CONTEXT      quadword unsigned;      /* Search context
    SERVICE_LIST        longword unsigned;      /* Service routines to execute
    SERVICE_OPEN        longword unsigned;      /* Service routines that are open
    SERVICE_STATUS      longword unsigned;      /* Service completion status
    START_PAGE          longword unsigned;      /* Start when reached
    STOP_PAGE           longword unsigned;      /* Stop when reached
    STREAM_INDEX        longword unsigned;      /* Job controller stream
    T_MARGIN            longword unsigned;      /* Current top margin
    TIME_PRINTED        quadword unsigned;      /* Time printed
    USER_BUFFER         quadword unsigned;      /* Input buffer descriptor
    XLATE_TABLE         address;                /* Translation table

    /*
    /* RMS block structure pointers
    /*
    FAB                 address;                /* FAB block
    NAM                 address;                /* NAM block
    RAB                 address;                /* RAB block
    XABDAT              address;                /* XABDAT block
    XABFHC              address;                /* XABFHC block
    XABPRO              address;                /* XABPRO block

    /*
    /* No longword alignment past here
    /*

    /*
```

```
    /* Input service context area
    /*
    SERVICE_CONTEXT structure;
      INPUT_RECORD       quadword unsigned;      /* Input record descriptor
      RECORD_HEADER      longword unsigned;      /* Input record header
      RECORD_NUMBER      longword unsigned;      /* Current record number
      USER_RECORD        quadword unsigned;      /* User record desc
      CARCON    structure longword unsigned;    /* Carriage control
        PREFIX_COUNT         byte unsigned;     /*  - prefix byte count
        PREFIX_CHAR          byte unsigned;     /*  - prefix character
        POSTFIX_COUNT        byte unsigned;     /*  - postfix byte count
        POSTFIX_CHAR         byte unsigned;     /*  - postfix record character
        end;
      CC_TYPE            byte unsigned;        /* Carriage control type
      SERVICE_INDEX      byte unsigned;        /* Input service index
    end;

    ACCOUNTING_AREA      character length 16;  /* Accounting data
    CONDITION_AREA       character length 4 * 5; /* Condition data

    CHECKPOINT_DEPTH     byte;                 /* Number of checkpoints saved
    ESCAPE_STATE         byte unsigned;        /* Escape sequence state
    FILE_BURST_CHAR      byte unsigned;        /* Burst char for flag page
    INPUT_DEPTH          byte;                 /* Number of services "pushed"
    JOB_BURST_CHAR       byte unsigned;        /* Burst char for job page
    STATE                byte unsigned;        /* Symbiont state

    /*
    /* Patch area
    /*
    dummy_0              longword unsigned;
    dummy_1              longword unsigned;
    dummy_2              longword unsigned;
    dummy_3              longword unsigned;
    dummy_4              longword unsigned;
    dummy_5              longword unsigned;
    dummy_6              longword unsigned;
    dummy_7              longword unsigned;
    dummy_8              longword unsigned;
    dummy_9              longword unsigned;

/*
/* NB: The user context area begins at the end of the SCB and extends for
/*     a user defined amount.  No SCB definitions may be placed after it.
/*
    USER_CONTEXT_AREA   longword unsigned tag R; /* User context area

end;
end_module;
```

BANNER
LIS

PLIWRITE
LIS

SMBREQ
REQ

PLIVECTOR
LIS

SMBSRVSHR
MAP

FORMAT
LIS

PLIRODATA
LIS

SMBDEF
SDL

PRTSMB

DISPATCH
L13

PLISTRING
LIS

PRTSMB
MAP