```
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBB         BBB  RR          RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR      RRR
LLL              BBB         BBB  RRR      RRR
LLL              BBB         BBB  RRR      RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
```

```
_$2

Val
---
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
00F
00F
00F
00F
00F
00F
00F
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

```
**FILE**ID**LBRUSR                                                    L 11
```

```
LL          BBBBBBBB    RRRRRRRR    UU      UU    SSSSSSSS  RRRRRRRR
LL          BBBBBBBB    RRRRRRRR    UU      UU    SSSSSSSS  RRRRRRRR
LL          BB      BB  RR      RR  UU      UU  SS          RR      RR
LL          BB      BB  RR      RR  UU      UU  SS          RR      RR
LL          BB      BB  RR      RR  UU      UU  SS          RR      RR
LL          BB      BB  RR      RR  UU      UU  SS          RR      RR
LL          BBBBBBBB    RRRRRRRR    UU      UU    SSSSSS    RRRRRRRR
LL          BBBBBBBB    RRRRRRRR    UU      UU    SSSSSS    RRRRRRRR
LL          BB      BB  RR  RR      UU      UU          SS  RR  RR
LL          BB      BB  RR  RR      UU      UU          SS  RR  RR
LL          BB      BB  RR    RR    UU      UU          SS  RR    RR              ....
LL          BB      BB  RR    RR    UU      UU          SS  RR    RR              ....
LLLLLLLLLL  BBBBBBBB    RR      RR  UUUUUUUUUU  SSSSSSSS    RR      RR      ....
LLLLLLLLLL  BBBBBBBB    RR      RR  UUUUUUUUUU  SSSSSSSS    RR      RR      ....


  SSSSSSSS  DDDDDDDD    LL
  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
```

```
{  LBRUSR.MDL
{       Version V02-014
{
{*************************************************************************
{*                                                                       *
{*  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   UCH  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.              *
{*                                                                       *
{*                                                                       *
{*************************************************************************
{
{ MODIFIED BY:
{
{       V02-014         PCG0004         Peter George    07-Jan-1982
{               Add HELP flag to HLP.
{
{       V02-013         PCG0003         Peter George    09-Dec-1981
{               Add LIBLIST and NOTTERM flag to HLP.
{
{       V02-012         RPG0112         Bob Grosso      11-Aug-1981
{               Support lower cased keywords
{
{       V02-011         RPG0037         Bob Grosso      20-Jul-1981
{               Add cre$c_vmsv2.
{
{       V02-010         RPG0036         Bob Grosso      15-Jul-1981
{               Add lbr$l_oldhdrptr.
{
{       V02-009         RPG0035         Bob Grosso      1-Jul-1981
{               Change lhi$w_*luhrec to lhi$l_*luhrec.
{
{       V02-008         RPG0034         Bob Grosso      18-Jun-1981
{               Change lhi$l_*luhrec to lhi$w_*luhrec.
{               Change lbr$c_maxluhlen to lbr$c_maxluhrec = 32768.
{
{       V02-007         PCG0002         Peter George    19-May-1981
{               Add PMPTDEF flag to HLP.
{
{       V02-006         PCG0001         Peter George    08-May-1981
{               Add lbr$output_help stuff.
{
```

```
{       V02-005         RPG0033         Bob Grosso      10-Apr-1981
{               Add lbr$c_maxkeylen, lhi$l_maxluhrec, lhi$l_numluhrec,
{               lhi$l_libstatus and cre$l_luhmax.
{
{       V02-004         RPG0025         Bob Grosso      20-Mar-1981
{               Add lbr$c_maxidxrd
{
{       V02-003         RPG0016         Bob Grosso      25-Feb-1981
{               Rename lbr$c_mtcbufsiz to lbr$c_putbufsiz
{
{       V02-002         RPG0012         Bob Grosso      19-Jan-1981
{               Add lbr$flush operation codes, and sizes for the buffers to
{               empty the cache.
{
{       V02-001         BLS0029         Benn Schreiber  22-Dec-1980
{                       Add shareable image symbol table type.
{
{ Librarian control table
{

module $LBRCTLTBL;


aggregate LBRCTLTBL structure prefix LBR$;
    ID byte unsigned;                                   /* Control table ID
    constant CTLTBLID   equals 203  prefix LBR tag $C;  /* Ident for control table
    TBLSIZ byte unsigned;                               /* Control table size
    TYPE byte unsigned;                                 /* Type of library opened
    FUNC byte unsigned;                                 /* Operation (function) requested
    FILL_1 byte dimension 2 fill prefix LBRCTLTBL tag $$;/* Reserved extra bytes
    USRFLG_OVERLAY union;
        USRFLG longword unsigned;                       /* Flags longword
        USRFLG_BITS structure;
            LOCATE bitfield mask;                       /* Use "locate" rather than "move" mode
            OPEN bitfield mask;                         /* Library open
        end USRFLG_BITS;
    end USRFLG_OVERLAY;
    HDRPTR longword unsigned;                           /* Pointer to in-core header
    CTXPTR longword unsigned;                           /* Pointer to context control block
    CURIDX longword unsigned;                           /* Number of current index
    USRNAM longword unsigned;                           /* Pointer to user NAM block
    OLDHDRPTR longword unsigned;                        /* Pointer to unmodified in-core header block
    constant "LENGTH" equals . prefix LBR$ tag K;
    constant "LENGTH" equals . prefix LBR$ tag C;

end LBRCTLTBL;

end_module $LBRCTLTBL;

module $MHDDEF;

/*
/* Module header
/*
```

```
aggregate MHDDEF structure prefix MHD$;
    LBRFLAG byte unsigned;                              /* Librarian-controlled flag byte
    ID byte unsigned;                                   /* Ident
    constant MHDID      equals 173  prefix MHD tag $C;  /* Value that must be in the ident
    FILL_1 word fill prefix MHDDEF tag $$;              /* Reserved word
    REFCNT longword unsigned;                           /* Reference count
    constant REFLNG equals . prefix MHD$ tag K;         /* Length of record to end of ref count
    constant REFLNG equals . prefix MHD$ tag C;         /* Length of record to end of ref count
    constant INSTIME equals . prefix MHD$ tag K;        /* Label for start of insert time
    constant INSTIME equals . prefix MHD$ tag C;        /* Label for start of insert time
    DATIM longword unsigned;                            /* Date/time inserted
    FILL_2_OVERLAY union;
        FILL_2 longword fill prefix MHDDEF tag $$;      /* ...
        constant USRDAT equals . prefix MHD$ tag K;     /* Start of user additional header data
        constant USRDAT equals . prefix MHD$ tag C;     /* Start of user additional header data
        FILL_2_FIELDS structure;
            FILL_3 byte dimension 4 fill prefix MHDDEF tag $$;
            USRDAT character length 0 tag B;            /* Start of user additional header data
            constant MHDLEN equals . prefix MHD$ tag K; /* Length of fixed part of MHD
            constant MHDLEN equals . prefix MHD$ tag C; /* Length of fixed part of MHD
        end FILL_2_FIELDS;
    end FILL_2_OVERLAY;
    OBJSTAT_OVERLAY union;
        OBJSTAT byte unsigned;                          /* Status of object module
        OBJSTAT_BITS structure;
            SELSRC bitfield mask;                       /* Selective search
            OBJTIR bitfield mask;                       /* Module contains TIR records
        end OBJSTAT_BITS;
    end OBJSTAT_OVERLAY;
    OBJIDLNG_OVERLAY union;
        OBJIDLNG byte unsigned;                         /* Length of ident
        OBJIDLNG_FIELDS structure;
            FILL_4 byte fill prefix MHDDEF tag $$;
            OBJIDENT character length 0 tag T;          /* Object module ident
            constant OBJIDENT equals . prefix MHD$ tag K;/*
            constant OBJIDENT equals . prefix MHD$ tag C;/*
        end OBJIDLNG_FIELDS;
    end OBJIDLNG_OVERLAY;
end MHDDEF;

end_module $MHDDEF;

module $HLPDEF;


/*
/* Data structures for help processing
/*



aggregate HLPDEF  union prefix HLP$;
    HLPDEF_BITS structure;
        PROMPT bitfield mask;                           /* Prompting enabled
        PROCESS bitfield mask;                          /* Process logical name table searches enabled
        GROUP bitfield mask;                              /* Group logical name table searches enabled
        SYSTEM bitfield mask;                             /* System logical name table searched enabled
        LIBLIST bitfield mask;                            /* Output list of default libraries
        HELP bitfield mask;                               /* Display help on help before list of topics
        SPARE1 bitfield mask;
        SPARE2 bitfield mask;

        PAGE bitfield mask;                               /* Page breaks enabled
        OUTPUT bitfield mask;                             /* Output listing file enabled
        LIBRARY bitfield mask;                            /* Main library specified by user
        ALL bitfield mask;                                /* Some logical name table searches enabled
        PAGEDEF bitfield mask;                            /* Page flag defaulting disabled
        PMPTDEF bitfield mask;                            /* Prompt flag defaulting disabled
        NOTTERM bitfield mask;                            /* Input device is not a terminal
    end HLPDEF_BITS;
end HLPDEF;

    aggregate HLPDEF1 structure prefix HLP$;
    DESC longword unsigned;                               /* Address of string descriptor for line
    FLAGS_OVERLAY union;
        FLAGS longword unsigned;                          /* Flags
        FLAGS_BITS structure;
            NOHLPTXT bitfield mask;                       /* Line is part of text due to no help found
            KEYNAMLIN bitfield mask;                      /* Line contains keynames to be printed
            OTHERINFO bitfield mask;                      /* Line is part of "other info available"
        end FLAGS_BITS;
    end FLAGS_OVERLAY;
    DATA longword unsigned;                               /* Address of user data passed to GET_HELP
    LEVEL longword unsigned;                              /* Level of this help
    end HLPDEF1;

end_module $HLPDEF;

module $LBRDEF;

/*
/* Types of libraries
/*


                                                          /* Define the library types
constant TYP_UNK        equals 0  prefix LBR tag $C;      /* Unknown/unspecified library type
constant TYP_OBJ        equals 1  prefix LBR tag $C;      /* Object/shareable image library
constant TYP_MLB        equals 2  prefix LBR tag $C;      /* Macro library
constant TYP_HLP        equals 3  prefix LBR tag $C;      /* Help file library
constant TYP_TXT        equals 4  prefix LBR tag $C;      /* TEXT library
constant TYP_SHSTB      equals 5  prefix LBR tag $C;      /* Shareable image symbol library
constant TYP_DECMX      equals 5  prefix LBR tag $C;      /* Maximum Digital library type defined
constant TYP_RDEC       equals 127  prefix LBR tag $C;    /* Types between DECMX and RDEC are
                                                          /*   reserved to Digital
constant TYP_USRLW      equals 128  prefix LBR tag $C;    /* User library types range from 128
constant TYP_USRHI      equals 255  prefix LBR tag $C;    /* to 255.

/*
/* Function codes for lbr$flush
/*
```

```
constant FLUSHDATA      equals 1  prefix LBR tag $C;    /* Flush data blocks from cache
constant FLUSHALL       equals 0  prefix LBR tag $C;    /* Flush data blocks, then index blocks

/*
/* Librarian fixed parameters
/*
constant MAXRECSIZ      equals 2048  prefix LBR tag $C; /* Maximum record size allowed
constant PAGESIZE       equals 512  prefix LBR tag $C;  /* Size of memory page
constant HASHSIZE       equals 512  prefix LBR tag $C;  /* Size of hash table ** Must be power of 2 **
constant TEXTPAGE       equals 508  prefix LBR tag $C;  /* No. of useable bytes on a text page
constant DEXTQ  equals 50  prefix LBR tag $C;           /* Library default extend quantity
constant MAXCTL equals 16  prefix LBR tag $C;           /* Maximum number of open libraries
constant MAXHDRSIZ      equals 128  prefix LBR tag $C;  /* Maximum length of module header
                                                        /*  (max user length is:
                                                        /*    lbr$c_maxhdrsiz-mhd$c_length)
constant DEFENTALL      equals 300  prefix LBR tag $C;  /* Number of entries to allocate by default
constant RETRYOPEN      equals 30  prefix LBR tag $C;   /* Number of times to retry open on RMS$_FLK
constant RETRYWAIT      equals 1  prefix LBR tag $C;    /* Number of seconds to wait between retries
constant MINREAD        equals 2  prefix LBR tag $C;    /* Minimum number of blocks to read
constant MAXREAD        equals 50  prefix LBR tag $C;   /* Max blocks can ever read
constant MEMXTRA        equals 50  prefix LBR tag $C;   /* Number blocks to expand region by above and beyond lbr$gl_maxread
constant PUTBUFSIZ      equals 30  prefix LBR tag $C;   /* Block size of VM empty cache buffer
constant FLSHBFSIZ      equals 1  prefix LBR tag $C;    /* Block size of stack mt cache buffer
constant MAXIDXRD       equals 20  prefix LBR tag $C;   /* Maximum blocks in one index read
constant MAXKEYLEN      equals 128  prefix LBR tag $C;  /* Maximum length of an ASCII keyword, at most N, where
                                                        /* 3*(N+7) < or = 506
constant MAXLUHREC      equals 32768  prefix LBR tag $C;/* maximum library update history record length

/*
/* LBR$INI_CONTROL argument list
/*


aggregate LBRDEF structure prefix LBR$;
    FILL_1 longword fill prefix LBRDEF tag $$;          /* Argument count
    IC_CTLTBL longword unsigned;                        /* Control index address
    IC_FUNC longword unsigned;                          /* Function
    constant CREATE     equals 0  prefix LBR tag $C;    /* Create a new library
    constant READ       equals 1  prefix LBR tag $C;    /* Read an existing library
    constant UPDATE     equals 2  prefix LBR tag $C;    /* Update an exisiting library
    constant MAXFUNC    equals 2  prefix LBR tag $C;    /* Maximum legal function
    IC_TYPE longword unsigned;                          /* Type of library expected to open

/*
/* LBR$OPEN argument list
/*

end LBRDEF;

aggregate LBRDEF1 structure prefix LBR$;
    FILL_2 longword fill prefix LBRDEF tag $$;          /* Argument count
    OP_CTLTBL longword unsigned;                        /* Control index address
    OP_FNS longword unsigned;                           /* Address of string descriptor for filename
    OP_CREOPT longword unsigned;                        /* Address of create options array
    OP_DNS longword unsigned;                           /* Address of descriptor for default name string
    OP_RLFNA longword unsigned;                         /* Address of NAM block for related file
    OP_RNS longword unsigned;                         /* Address of descriptor for related filename string
    OP_RNSLEN longword unsigned;                      /* Address of longword to store resultant filename string length

/*
/* LBR$CLOSE argument list
/*

end LBRDEF1;

aggregate LBRDEF2 structure prefix LBR$;
    FILL_3 longword fill prefix LBRDEF tag $$;        /* Argument count
    CL_CTLTBL longword unsigned;                      /* Control index address

/*
/* LBR$GET_HEADER argument list
/*

end LBRDEF2;

aggregate LBRDEF3 structure prefix LBR$;
    FILL_4 longword fill prefix LBRDEF tag $$;        /* Argument count
    GH_CTLTBL longword unsigned;                      /* Address of Control index
    GH_RETARY longword unsigned;                      /* Address of 128-longword array for return info

/*
/* LBR$SET_INDEX argument list
/*

end LBRDEF3;

aggregate LBRDEF4 structure prefix LBR$;
    FILL_5 longword fill prefix LBRDEF tag $$;        /* Argument count
    SI_CTLTBL longword unsigned;                      /* Control index address
    SI_IDXNUM longword unsigned;                      /* Address of index number

/*
/* LBR$LOOKUP_KEY argument list
/*

end LBRDEF4;

aggregate LBRDEF5 structure prefix LBR$;
    FILL_6 longword fill prefix LBRDEF tag $$;        /* Argument count
    LK_CTLTBL longword unsigned;                      /* Control index address
    LK_KEYNAM longword unsigned;                      /* Address of string descriptor or binary value
    LK_TXTRFA longword unsigned;                      /* Address of quadword to return RFA if found

/*
/* LBR$INSERT_KEY argument list
/*

end LBRDEF5;

aggregate LBRDEF6 structure prefix LBR$;
    FILL_7 longword fill prefix LBRDEF tag $$;        /* Argument count
    IK_CTLTBL longword unsigned;                      /* Cont:ol index address
    IK_KEYNAM longword unsigned;                            /* Address of string descriptor or binary value
    IK_TXTRFA longword unsigned;                            /* Address of RFA of text

/*
/* LBR$REPLACE_MOD argument list
/*

/*          F         ,L,1                         /* Argument count
/*          F         RM_CTLTBL,L                 /* Control index address
/*          F         RM_KEYNAM,L                 /* Address of string descriptor or binary value
/*          F         RK_TXTRFA,L                 /* RFA of new text
/*          P         1

/*
/* LBR$REPLACE_KEY argument list
/*
end LBRDEF6;

aggregate LBRDEF7 structure prefix LBR$;
    FILL_8 longword fill prefix LBRDEF tag $$;              /* Argument count
    RK_CTLTBL longword unsigned;                            /* Control index address
    RK_KEYNAM longword unsigned;                            /* Address of string descriptor or binary value
    RK_OLDRFA longword unsigned;                            /* Address of RFA of old text
    RK_NEWRFA longword unsigned;                            /* Address of RFA of new text
/*
/* LBR$DELETE_KEY argument list
/*

end LBRDEF7;

aggregate LBRDEF8 structure prefix LBR$;
    FILL_9 longword fill prefix LBRDEF tag $$;              /* Argument count
    DK_CTLTBL longword unsigned;                            /* Control index address
    DK_KEYNAM longword unsigned;                            /* Address of string descriptor or binary value

/*
/* LBR$DELETE_DATA argument list
/*

end LBRDEF8;

aggregate LBRDEF9 structure prefix LBR$;
    FILL_10 longword fill prefix LBRDEF tag $$;             /* Argument count
    DD_CTLTBL longword unsigned;                            /* Control index address
    DD_TXTRFA longword unsigned;                            /* Address of RFA to delete from

/*
/* LBR$GET_RECORD argument list
/*

end LBRDEF9;

aggregate LBRDEF10 structure prefix LBR$;
    FILL_11 longword fill prefix LBRDEF tag $$;             /* Argument count
    GR_CTLTBL longword unsigned;                            /* Control index address
    GR_BUFDES longword unsigned;                            /* Address of descriptor of buffer
    GR_BUFLEN longword unsigned;                            /* Address of longword to return record size

/*
/* LBR$PUT_RECORD argument list
/*

end LBRDEF10;

aggregate LBRDEF11 structure prefix LBR$;
    FILL_12 longword fill prefix LBRDEF tag $$;             /* Argument count
    PR_CTLTBL longword unsigned;                            /* Control index address
    PR_BUFDES longword unsigned;                            /* Address of descriptor of buffer to output
    PR_TXTRFA longword unsigned;                            /* Address of quadword to return RFA
                                                            /*  of text record

/*
/* LBR$PUT_END argument list
/*

end LBRDEF11;

aggregate LBRDEF12 structure prefix LBR$;
    FILL_13 longword fill prefix LBRDEF tag $$;             /* Argument count
    PE_CTLTBL longword unsigned;                            /* Control index address

/*
/* LBR$SEARCH argument list
/*

end LBRDEF12;

aggregate LBRDEF13 structure prefix LBR$;
    FILL_14 longword fill prefix LBRDEF tag $$;             /* Argument count
    SR_CTLTBL longword unsigned;                            /* Control index address
    SR_IDXNUM longword unsigned;                            /* Address of index number
    SR_RFA longword unsigned;                               /* Address of RFA to search index for
    SR_USRTN longword unsigned;                             /* User routine to call on match

/*
/* Argument list for user routine called by LBR$SEARCH
/*

end LBRDEF13;

aggregate LBRDEF14 structure prefix LBR$;
    FILL_15 longword fill prefix LBRDEF tag $$;             /* Argument count
    SU_KEYDES longword unsigned;                            /* Address of string descriptor or binary value
    SU_TXTRFA longword unsigned;                            /* Address of array containing rfa of module

/*
/* LBR$GET_INDEX argument list
/*

end LBRDEF14;

aggregate LBRDEF15 structure prefix LBR$;
```

```
    FILL_16 longword fill prefix LBRDEF tag $$;         /* Argument count
    GI_CTLTBL longword unsigned;                        /* Control index address
    GI_IDXNUM longword unsigned;                        /* Index number
    GI_USRTN longword unsigned;                         /* User routine to call for each entry
/*
/* Argument list for user routine called by LBR$GET_INDEX
/*

end LBRDEF15;

aggregate LBRDEF16 structure prefix LBR$;
    FILL_17 longword fill prefix LBRDEF tag $$;         /* Argument count
    GU_KEYADR longword unsigned;                        /* Address of descriptor or binary value
    GU_TXTRFA longword unsigned;                        /* RFA of associated text

/*
/* LBR$ADD_UPDATE argument list
/*

end LBRDEF16;

aggregate LBRDEF17 structure prefix LBR$;
    FILL_18 longword fill prefix LBRDEF tag $$;         /* Argument count
    AU_CTLTBL longword unsigned;                        /* Control index address
    AU_FLAGS longword unsigned;                         /* Flags
    AU_KEYNAM longword unsigned;                        /* Address of string descriptor or binary value
                                                        /* Types of operations logged
    constant ADDMOD     equals 1  prefix LBR tag $C;    /* Add module
    constant DELMOD     equals 2  prefix LBR tag $C;    /* Delete module
    constant REPMOD     equals 3  prefix LBR tag $C;    /* Replace module

/*
/* LBR$GET_UPDATES argument list
/*

end LBRDEF17;

aggregate LBRDEF18 structure prefix LBR$;
    FILL_19 longword fill prefix LBRDEF tag $$;         /* Argument count
    GU_CTLTBL longword unsigned;                        /* Control index address
    GU_USRTN longword unsigned;                         /* User routine to call for each update

/*
/* Argument list for user routine called by LBR$GET_UPDATES
/*

end LBRDEF18;

aggregate LBRDEF19 structure prefix LBR$;
    FILL_20 longword fill prefix LBRDEF tag $$;         /* Argument list
    UU_UPDESC longword unsigned;                        /* String descriptor for history line

end LBRDEF19;

end_module $LBRDEF;
```

```
module $LHIDEF;

/*
/* Library header information array offsets
/*



aggregate LHIDEF structure prefix LHI$;
    TYPE longword unsigned;                             /* Library type
    NINDEX longword unsigned;                           /* Number of indices
    MAJORID longword unsigned;                          /* Library format major id
    MINORID longword unsigned;                          /* Library format minor id
    LBRVER character length 32;                         /* ASCIC version of librarian that created
    CREDAT longword unsigned;                           /* Creation date/time
    FILL_1 longword fill prefix LHIDEF tag $$;          /* ...
    UPDTIM longword unsigned;                           /* Date/time of last update
    FILL_2 longword fill prefix LHIDEF tag $$;          /* ...
    UPDHIS longword unsigned;                           /* VBN of start of update history
    FREEVBN longword unsigned;                          /* 1st logically deleted block
    FREEBLK longword unsigned;                          /* Number of deleted blocks
    NEXTRFA byte unsigned dimension 6;                  /* RFA of end of library
    RFAXTR word unsigned;                               /* Spare word, zeroed
    NEXTVBN longword unsigned;                          /* Next VBN to allocate at end of file
    FREIDXBLK longword unsigned;                        /* Number of free pre-allocated index blocks
    FREEIDX longword unsigned;                          /* Listhead for pre-allocated index blocks
    HIPREAL longword unsigned;                          /* VBN of highest pre-allocated block
    IDXBLKS longword unsigned;                          /* Number of index blocks in use
    IDXCNT longword unsigned;                           /* Number of index entries (total)
    MODCNT longword unsigned;                           /* Number of entries in index 1 (module names)
    MHDUSZ longword unsigned;                           /* Length of user-maintained info in module header
    MAXLUHREC longword unsigned;                        /* Maximum number of library update histories records maintained
    NUMLUHREC longword unsigned;                        /* Number of LUH records in history
    LIBSTATUS longword unsigned;                        /* False if there was an error closing lib
end LHIDEF;

end_module $LHIDEF;

module $CREDEF;

/*
/* Create options table
/*



aggregate CREDEF structure prefix CRE$;
    TYPE longword unsigned;                             /* Type of library
                                                        /*  (library types defined in $LBRDEF)

    KEYLEN longword unsigned;                           /* Length of keys in library
    ALLOC longword unsigned;                            /* Initial file allocation
    IDXMAX longword unsigned;                           /* Maximum number of indices
    UHDMAX longword unsigned;                           /* Size of additional module header data
    ENTALL longword unsigned;                           /* Number of index entries to pre-allocate
    LUHMAX longword unsigned;                           /* Number of library update history records to store
    VERTYP longword unsigned;                           /* Version type of library to create
    constant VMSV2      equals 2  prefix CRE tag $C;    /* VMS version V04-000 format
    constant VMSV3      equals 3  prefix CRE tag $C;    /* VMS version 3 format
    IDXOPT_OVERLAY union;
        IDXOPT longword unsigned;                        /* Index options
        IDXOPT_BITS structure;
            NOCASECMP bitfield mask;                     /* Do not upper case the match key
            NOCASENTR bitfield mask;                     /* Do not upper case the index key when comparing with a match key
            UPCASNTRY bitfield mask;                     /* Upper case the index key when entering it into the library
        end IDXOPT_BITS;
        constant HLPCASING      equals 6  prefix CRE tag $C;/* Treat upper casing as it is for HELP libs
        constant OBJCASING      equals 3  prefix CRE tag $C;/* Treat upper casing as it is for OBJECT libs
        constant MACTXTCAS      equals 0  prefix CRE tag $C;/* Treat upper casing as it is for MACRO and TEXT libs
    end IDXOPT_OVERLAY;
    FILL_1 longword dimension 11 fill prefix CREDEF tag $$;/* Reserved 11 longwords
    constant "LENGTH" equals . prefix CRE$ tag K;       /*
    constant "LENGTH" equals . prefix CRE$ tag C;       /*

end CREDEF;

end_module $CREDEF;
```

CA
VO

0197 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY