

[illegible]

```
IIIIII  SSSSSSSS  GGGGGGGG  EEEEEEEEE  NN  NN  CCCCCCCC
IIIIII  SSSSSSSS  GGGGGGGG  EEEEEEEEE  NN  NN  CCCCCCCC
II      SS      GG      EE      NN  NN  CC
II      SS      GG      EE      NN  NN  CC
II      SS      GG      EE      NNNN NN  CC
II      SS      GG      EE      NNNN NN  CC
II      SSSSSS  GG      EEEEEEE  NN  NN  CC
II      SSSSSS  GG      EEEEEEE  NN  NN  CC
II      SS      GG      EE      NN  NN  CC
II      SS      GG      EE      NN  NN  CC
II      SS      GG      EE      NN  NN  CC
IIIIII  SSSSSSSS  GGGGGG  EEEEEEEEE  NN  NN  CCCCCCCC
IIIIII  SSSSSSSS  GGGGGG  EEEEEEEEE  NN  NN  CCCCCCCC
```

```
RRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRR  EEEEEEEEE  QQQQQQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RRRRRRR  EEEEEEEEE  QQ      QQ
RRRRRRR  EEEEEEEEE  QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
```

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.
*
*****
```

++

MODULE: ISGENC.REQ

FACILITY: LINKER

ABSTRACT: REQUIRE FILE FOR I-SECTION GENERATION CONTROL

HISTORY:

AUTHOR: T.J. PORTER 01-APR-77

MODIFICATIONS:

V03-001	JWT0152	Jim Teague	10-Feb-1984
	Change ISD\$C SIZE to reflect maximum size of new variable length global section names.		
V02-002	BLS0026	Benn Schreiber	28-Nov-1980
	Remove ascii name string from isect generation tables.		
V02-001	BLS0007	Benn Schreiber	15-Aug-1980
	Convert to MDL data structures.		

--

++

functional description:

this is a require file which sets up all necessary data for generation of image sections. generation of image sections is driven by a table chosen depending upon the kind of link being performed.

exe_isgentbl is used for normal user executable images

shr_isgentbl is used for shareable images

sys_isgentbl is used for system space images

the order of entries in these tables is the order in which the image sections are generated. there is one entry for each image section to be generated. each entry provides the following information:

name of isection build routine

p-section attribute mask (the and mask)

p-section match pattern

i-section code which appears in the isd of header

size of the isd written to header

page fault cluster size

isd flags.

this file requires that sysdef be previously declared to define the isd's and the image header layout.

COMPILETIME

NUM_ISECTS=0;

! VARIABLE FOR TABLE GENERATION

DEFINE GENERIC I-SECTION TYPE PARAMETERS:-

SIZE = NUMBER OF BYTES IN ISD OF HEADER

CODE = IMAGE SECTION TYPE CODE IN ISD

PFC = PAGE FAULT CLUSTER SIZE

GBL = INDICATOR FOR WHETHER POTENTIALLY GLOBAL
TRUE OR FALSE

DZRO = DEMAND ALLOCATE ZERO FILL IMAGE SECTION
TYPE INDICATOR - TRUE OR FALSE

LITERAL

SUFFIX_SIZE = 4,

! Size of "_nnn" global isd suffix

NULL_SIZE = 0,

! NULL ISD TYPE

NULL_CODE = 0,

NULL_PFC = 0,

NULL_GBL = FALSE,

NULL_DZRO = FALSE,

NORMAL_GBL = FALSE,

NORMAL_DZRO = FALSE,

NORMAL_CODE = ISD\$K_NORMAL,

NORMAL_SIZE = ISD\$K_LENPRIV,

NORMAL_PFC = NULL_PFC,

! NORMAL ISD'S TYPE

! NORMAL ISD SIZE

! NORMALLY NULL

```

!
STACK_CODE = ISD$K_USRSTACK,
STACK_SIZE = ISD$K_LENDRZRO,
STACK_PFC = NULL_PFC,
STACK_GBL = FALSE,
STACK_DZRO = TRUE,
! STACK ISD TYPE CODE
! DEMAND ZERO ISD
! DEFAULT IS NULL

!
SHRFXD_CODE = ISL$C_SHRFXD,
SHRFXD_SIZE = ISD$K_LENPRIV,
SHRFXD_PFC = NULL_PFC,
SHRFXD_GBL = FALSE,
SHRFXD_DZRO = FALSE,
! SHARABLE AT FIXED PLACE
! IN THE ADDRESS SPACE

!
PRVFXD_CODE = ISL$C_PRVFXD,
PRVFXD_SIZE = ISD$K_LENPRIV,
PRVFXD_PFC = NULL_PFC,
PRVFXD_GBL = FALSE,
PRVFXD_DZRO = FALSE,
! PRIVATE IN FIXED PLACE OF ADDRESS

!
SHRPIC_CODE = ISL$C_SHRPIC,
SHRPIC_SIZE = ISD$K_LENPRIV,
SHRPIC_PFC = NULL_PFC,
SHRPIC_GBL = FALSE,
SHRPIC_DZRO = FALSE,
! SHAREABLE POSITION INDEPENDENT

!
PRVPIC_CODE = ISL$C_PRVPIC,
PRVPIC_SIZE = ISD$K_LENPRIV,
PRVPIC_PFC = NULL_PFC,
PRVPIC_GBL = FALSE,
PRVPIC_DZRO = FALSE,
! PRIVATE PIC SECTIONS

RDWRTEXEREL = GPSS$M_WRT OR GPSS$M_EXE OR
               GPSS$M_REC,
SHRPICMASK = RDWRTEXEREL OR GPSS$M_SHR OR
              GPSS$M_PIC,
! MASK FOR READ,
! WRITE EXECUTE AND RELOCATABLE
! SEARCH MASK FOR PIC AND SHAREABLE P-SECTIONS
! WHICH ARE READ, WRITE EXECUTABLE
! AND RELOCATABLE
! MASK FOR VECTOR IMAGE SECTIONS
! MASK FOR SHAREABLE VECTOR IMAGE SECTIONS

RDWRTEXERELVEC = RDWRTEXEREL OR GPSS$M_VEC,
SHRPICVEC = SHRPICMASK OR GPSS$M_VEC;

DEFINE THE HEADER FILLING

GLOBAL LITERAL
ISD$C_SIZE = ISD$K_MAXLENGLBL + ISL$C_SIZE : WEAK,
HDR$K_FILLCHR = 255 : BYTLIT WEAK,
IHDR$K_SHR = IHDR$K_LIM : WEAK,
IHDR$K_ACTIVOFF = IHDR$K_LENGTH : WEAK,
IHDR$K_SYMDBGOFF = IHDR$K_ACTIVOFF + IHDR$K_LENGTH : WEAK,
IHDR$K_IMGIDOFF = IHDR$K_SYMDBGOFF + IHDR$K_LENGTH : WEAK,
IHDR$K_PATCHOFF = IHDR$K_IMGIDOFF + IHDR$K_LENGTH : WEAK,
IHDR$K_MAXLENGTH = IHDR$K_PATCHOFF : WEAK,
HDR$K_MINFILL = 2 : WEAK;
! SIZE OF FULL IMAGE SECTION DESCRIPTOR INCLUDING LINKER PART
! FILL CHARACTER
! JUST A DIFFERENT NAME
! OFFSET TO ACTIVATION AREA
! OFFSET TO SYMBOL TABLE DATA
! OFFSET TO IMAGE IDENTIFICATION AREA
! OFFSET TO PATCH CONTROL AREA
! MAXIMUM HEADER LENGTH WITHOUT PATCH AREA
! MINIMUM FILL MUST EQUAL WIDTH
! OF ISD SIZE FIELD

THE FOLLOWING MACROS GENERATE ACCESSING MACROS FOR BLOCK

```


STRUCTURES WHICH HAVE FIELDS OF BYTE, WORD OR LONG WIDTH.

COMPILETIME

YYYYYYYSIZE = 0;

! VARIABLE WHICH IS RESULTANT BLOCK LENGTH

MACRO

GENBLOCKFIELDS(BLKNAME)[] =

CALLING SEQUENCE:

GENBLOCKFIELDS(BLKNAME,
FLDNAME1,FLDWIDTH1,
FLDNAME2,FLDWIDTH2,

.....

ETC,

FLDNAMEN,FLDWIDTHN);

WHERE:

FLDNAMEI = THE NAME OF FIELD I IN THE BLOCK
FLDWIDTHI = WIDTH IN BYTES OF FIELD I

THE MACRO GENERATES FIELD ACCESS MACROS OF THE FORM
BLKNAME\$X FLDNAME WHERE X IS B, W, L FOR FIELD WIDTHS OF
1, 2, 4 BYTES RESPECTIVELY.

IN ADDITION A MACRO OF FORM INIT_BLKNAME IS
GENERATED FOR STATIC INITIALIZATION OF BLOCK. FOR EXAMPLE
INITIAL (INIT_BLKNAME(VALUE1,VALUE2,....VALUEN))

GENERATE A MACRO TO REFERENCE THE START OF THE BLOCK

MACRO %NAME(%STRING(BLKNAME),'\$C_START') =
0,0,0,0%QUOTE%;

GENERATE THE ACCESSING MACROS FOR ALL THE FIELDS OF THE BLOCK

YYYYYYYBBF(BLKNAME,0,%REMAINING);

DEFINE A GLOBAL LITERAL (FOR LINK TIME ERROR DETECTION) WHOSE
VALUE IS NUMBER OF BYTES IN BLOCK

GLOBAL LITERAL %NAME(%STRING(BLKNAME),'\$C_SIZE') = YYYYYYYSIZE : WEAK;

NOW DEFINE A MACRO THAT CAN BE USED TO STATICALLY INITIALIZE
ALL THE FIELDS OF THIS BLOCK

MACRO NAME IS INIT_BLKNAME

YYCRE8INIMACRO(%NAME('INIT_',%STRING(BLKNAME)), (ZZZBBF(%REMAINING)), (XXXBBF(%REMAINING)))%,

AUXILIARY MACROS:-

YYCRE8INIMACRO(NAME, FORMAL, BODY) =
MACRO NAME FORMAL =%REMOVE BODY%QUOTE%.

ZZZBBF[FLDNAME, FLDWIDTH] = %NAME(%STRING(FLDNAME),'_FML')%,

XXXBBF[FLDNAME, FLDWIDTH] = %IF FLDWIDTH EQL 1 %THEN BYTE

```

        %ELSE %IF FLDWIDTH EQL 2 %THEN WORD
        %ELSE LONG
        %FI
    %FI
    (%NAME(%STRING(FLDNAM),'_FHL'))%,
!
YYYFLDPREFIX(WIDTH) = %IF WIDTH EQL 1
    %THEN B
    %ELSE %IF WIDTH EQL 2
        %THEN W
        %ELSE %IF WIDTH EQL 4
            %THEN L
            %ELSE T
            %FI
        %FI
    %FI%,
!
YYYYYYYBBF( BLKNAM, OFF, NAM, BYTS)[ ] =
    MACRO %NAME( %STRING(BLKNAM), '$', %STRING(YYYFLDPREFIX(BYTS)), '-', %STRING(NAM)) =
        OFF, 0, MIN(BYTS*8, 32), 0 %QUOTE%;
    %ASSIGN(YYYYYYYBSIZE,OFF+BYTS)
    YYYYYYYYBBF( BLKNAM, OFF+BYTS, %REMAINING)%;

```

```

DEFINE ACCESS MACROS FOR I-SECTION GENERATION CONTROL
TABLE ENTRIES

```

GENBLOCKFIELDS(ISC,	! NAMES ARE ISCSX YY...
MASK,2,	! P-SECTION AND MASK
MATCH,2,	! P-SECTION ATTRIBUTE MATCH
CODE,1,	! I-SECT TYPE
SIZE,1,	! ISD SIZE
PFC,1,	! PAGE FAULT CLUSTER
FLAGS,1,	! ISD FLAGS
MATCTL,1);	! MATCH CONTROL FIELD OF ISD

MACRO

```
ISECTDEF(NAME,TYPE,MATCH)=
```

```
THIS MACRO DEFINES AN IMAGE SECTION NAME AND THE PARAMETERS
OF SUCH AN IMAGE SECTION.
```

```

NAME IS A NAME TO BE ASSOCIATED WITH THE IMAGE SECTION
TYPE IS ONE OF THE GENERIC TYPES LISTED ABOVE.
MATCH IS THE PATTERN OF P-SECTION ATTRIBUTES TO BE
MATCHED IN THE SEARCH

```

LITERAL

%NAME(%STRING(NAME),'_MATCH') = MATCH OR GPSSM_REL,	! SET MATCH MASK ALWAYS RELOCATABLE
%NAME(%STRING(NAME),'_ISDCODE') = %NAME(%STRING(TYPE),'_CODE'),	! ISD TYPE CODE
%NAME(%STRING(NAME),'_ISDSIZ') = %NAME(%STRING(TYPE),'_SIZE'),	! SIZE OF ISD
%NAME(%STRING(NAME),'_PFC') = %NAME(%STRING(TYPE),'_PFC'),	! AND CLUSTER SIZE
%NAME(%STRING(NAME),'_GBL') = %NAME(%STRING(TYPE),'_GBL'),	! GLOBAL FLAG
%NAME(%STRING(NAME),'_CRF') = %IF(%NAME(%STRING(NAME),'_MATCH')	! SET THE COPY ON REFERENCE
AND (GPSSM_WRT OR	! FLAG IFF P-SECTION
GPSSM_SHR) EQL GPSSM_WRT	! WRITABLE AND NOT SHAREABLE
%THEN TRUE	! IS WRITABLE


```

                                %ELSE FALSE
                                %FI,
%NAME(%STRING(NAME),'_DZRO') = %NAME(%STRING(TYPE),'_DZRO'), ! DEMAND ZERO FLAG
%NAME(%STRING(NAME),'_WRT') = %IF(%NAME(%STRING(NAME),'_MATCH') ! WRITABLE FLAG IF P-SECTION
                                AND GPSSM_WRT) NEQ 0 ! IS WRITABLE
                                %THEN TRUE
                                %ELSE %IF %NAME(NAME,'_DZRO') ! OR IF THE IMAGE SECTION
                                %THEN TRUE ! IS A DEMAND ZERO
                                %ELSE FALSE
                                %FI
                                %FI,
%NAME(%STRING(NAME),'_MTCLFLD') = ISD$K_MATALL;% ! MATCH CONTROL FIELD OF ISD'S

```

```
ISGENTBLENT[ISNAME,SRCHMASK] = INIT_ISC(
```

THIS MACRO GENERATES ISECTION GENERATION CONTROL TABLE ENTRIES,
INITIALIZED WITH PARAMETERS ASSOCIATED WITH THE PARTICULAR
IMAGE SECTIONS.

ISNAME IS THE NAME OF THE IMAGE SECTION TYPE
SRCHMASK IS THE MASK OF P-SECTION ATTRIBUTES TO BE
CONSIDERED IN SEARCH FOR PATTERNS

```

SRCHMASK,
%NAME(%STRING(ISNAME),'_MATCH') ! STORE AND MASK
%NAME(%STRING(ISNAME),'_ISDCODE'), ! SET MATCH PATTERN
%NAME(%STRING(ISNAME),'_ISDSIZ'), ! CODE FOR ISD
%NAME(%STRING(ISNAME),'_PFC'), ! SIZE OF ISD
(ISD$M_GBL*%NAME(%STRING(ISNAME),'_GBL')) OR ! AND PAGE FAULT CLUSTER
(ISD$M_WRT*%NAME(%STRING(ISNAME),'_WRT')) OR ! SET ISECT
(ISD$M_CRF*%NAME(%STRING(ISNAME),'_CRF')) OR ! DESCRIPTOR
(ISD$M_DZRO*%NAME(%STRING(ISNAME),'_DZRO')), ! FLAGS
%NAME(%STRING(ISNAME),'_MTCLFLD')
)% ,

```

THIS MACRO GENERATES AN IMAGE SECTION GENERATION CONTROL
TABLE.

```

ISGENTBL(TBLNAM)[]=
%ASSIGN(NUM_ISECTS,(%LENGTH-1)/2) ! GET NUMBER IN
OWN ! TABLE AND SET
%NAME(%STRING(TBLNAM),'_ISECTS') : ! IN VARIABLE
    BYTE INITIAL (NUM_ISECTS),
%NAME(%STRING(TBLNAM),'_ISGENTBL') : ! GENERATE AND
    BLOCKVECTOR(NUM_ISECTS,ISC$C_SIZE,BYTE) ! INITIALIZE ALL
    INITIAL(ISGENTBLENT(%REMAINING));% ! ENTRIES OF TABLE

```

DEFINE INTERESTING IMAGE SECTION TYPES

```

ISECTDEF(_U_R,NORMAL,GPSSM_REL); ! USER RO DATA
ISECTDEF(_U_RW,NORMAL,GPSSM_WRT OR GPSSM_REL); ! USER R/W DATA
ISECTDEF(_U_X,NORMAL,GPSSM_EXE OR GPSSM_REL); ! USER EXECUTE ONLY
ISECTDEF(_U_XW,NORMAL,GPSSM_WRT OR ! USER MODIFIABLE
    GPSSM_EXE OR GPSSM_REL); ! EXECUTABLE CODE
ISECTDEF(_U_RV,NORMAL,GPSSM_REL OR GPSSM_VEC); ! RO VECTOR
ISECTDEF(_U_RVW,NORMAL,GPSSM_WRT OR GPSSM_VEC ! RW VECTOR
    OR GPSSM_REC);

```



```
ISECTDEF(_U_X_V, NORMAL, GPSSM_EXE OR GPSSM_REL      ! EXE ONLY VECTOR
          OR GPSSM_VEC);
ISECTDEF(_U_XWV, NORMAL, GPSSM_WRT OR GPSSM_EXE      ! MODIFIABLE
          OR GPSSM_VEC OR GPSSM_REL);               ! EXECUTABLE VECTOR
ISECTDEF(_STACK, STACK, 0);                          ! USER STACK
ISECTDEF(_S_ALONE, NORMAL, GPSSM_REL);               ! STAND ALONE IMAGE

      THE SET FOR SHAREABLE IMAGES

ISECTDEF(_S_R, SHRF XD, _U_R_MATCH OR GPSSM_SHR);    ! SHAREABLE NON-PIC IMAGE SECTIONS
ISECTDEF(_S_RQ, SHRF XD, _U_RQ_MATCH OR GPSSM_SHR);
ISECTDEF(_S_X, SHRF XD, _U_X_MATCH OR GPSSM_SHR);
ISECTDEF(_S_XQ, SHRF XD, _U_XQ_MATCH OR GPSSM_SHR);

ISECTDEF(_R, PRVFXD, _U_R_MATCH);                   ! PRIVATE NON-PIC SET
ISECTDEF(_RQ, PRVFXD, _U_RQ_MATCH);
ISECTDEF(_X, PRVFXD, _U_X_MATCH);
ISECTDEF(_XQ, PRVFXD, _U_XQ_MATCH);

ISECTDEF(_SPR, SHRPIC, _U_R_MATCH OR GPSSM_SHR OR GPSSM_PIC); ! SHAREABLE PIC SET
ISECTDEF(_SPRQ, SHRPIC, _U_RQ_MATCH OR GPSSM_SHR OR GPSSM_PIC);
ISECTDEF(_SPX, SHRPIC, _U_X_MATCH OR GPSSM_SHR OR GPSSM_PIC);
ISECTDEF(_SPXQ, SHRPIC, _U_XQ_MATCH OR GPSSM_SHR OR GPSSM_PIC);

ISECTDEF(_PR, PRVPIC, _U_R_MATCH OR GPSSM_PIC);      ! PRIVATE PIC SET
ISECTDEF(_PRQ, PRVPIC, _U_RQ_MATCH OR GPSSM_PIC);
ISECTDEF(_PX, PRVPIC, _U_X_MATCH OR GPSSM_PIC);
ISECTDEF(_PXQ, PRVPIC, _U_XQ_MATCH OR GPSSM_PIC);

ISECTDEF(_S_R_V, SHRF XD, _U_R_V_MATCH OR GPSSM_SHR); ! SHAREABLE NON-PIC VECTOR SET
ISECTDEF(_S_RQV, SHRF XD, _U_RQV_MATCH OR GPSSM_SHR);
ISECTDEF(_S_X_V, SHRF XD, _U_X_V_MATCH OR GPSSM_SHR);
ISECTDEF(_S_XQV, SHRF XD, _U_XQV_MATCH OR GPSSM_SHR);

ISECTDEF(_R_V, PRVFXD, _U_R_V_MATCH);                ! PRIVATE NON-PIC VECTOR SET
ISECTDEF(_RQV, PRVFXD, _U_RQV_MATCH);
ISECTDEF(_X_V, PRVFXD, _U_X_V_MATCH);
ISECTDEF(_XQV, PRVFXD, _U_XQV_MATCH);

ISECTDEF(_SPR_V, SHRPIC, _U_R_V_MATCH OR GPSSM_SHR OR GPSSM_PIC); ! SHAREABLE PIC VECTOR SET
ISECTDEF(_SPRQV, SHRPIC, _U_RQV_MATCH OR GPSSM_SHR OR GPSSM_PIC);
ISECTDEF(_SPX_V, SHRPIC, _U_X_V_MATCH OR GPSSM_SHR OR GPSSM_PIC);
ISECTDEF(_SPXQV, SHRPIC, _U_XQV_MATCH OR GPSSM_SHR OR GPSSM_PIC);

ISECTDEF(_PR_V, PRVPIC, _U_R_V_MATCH OR GPSSM_PIC);  ! PRIVATE PIC VECTOR SET
ISECTDEF(_PRQV, PRVPIC, _U_RQV_MATCH OR GPSSM_PIC);
ISECTDEF(_PX_V, PRVPIC, _U_X_V_MATCH OR GPSSM_PIC);
ISECTDEF(_PXQV, PRVPIC, _U_XQV_MATCH OR GPSSM_PIC);
```


0215 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

