

```

SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTTTTTT 33333333 22222222
SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTTTTTT 33333333 22222222
SSSSSSSSSSSS 00000000 RRRRRRRRRR TTTTTTTTTTTTTT 33333333 22222222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT          333      333  222      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT          333      333  222      222
SSSSSSSSSS 000      000  RRRRRRRRRR TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSS          000      000  RRR      RRR  TTT          333      333  222      222
SSSSSSSSSS 00000000 RRR      RRR  TTT          33333333 22222222
SSSSSSSSSS 00000000 RRR      RRR  TTT          33333333 22222222
SSSSSSSSSS 00000000 RRR      RRR  TTT          33333333 22222222

```

_S2

Pse

SOR

SOR

SOR

SOR

_L I

```

DDDDDDDD      KK      KK      SSSSSSSS
DDDDDDDD      KK      KK      SSSSSSSS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DD      DD      KKKKKK      SSSSSS
DD      DD      KKKKKK      SSSSSS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DD      DD      KK      KK      SS
DDDDDDDD      KK      KK      SSSSSSSS
DDDDDDDD      KK      KK      SSSSSSSS

```

```

....
....
....
....

```

```

RRRRRRRR      EEEEEEEEE      QQQQQQ
RRRRRRRR      EEEEEEEEE      QQQQQQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RR      RR      EE      QQ      QQ
RRRRRRRR      EEEEEEEEE      QQ      QQ
RRRRRRRR      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

```

```

*PU
*BB
*BB
*BB
*BB
*BB
*BB
*BB
*BL
*BL
*FF
*FF
*CM
*CM
*EX
*EX
*IN
*IN
*AC
*AO
*AO
*SO
*SO
*CV
*CV
*AS
*CA
*CA
*XF
*ES
*ES

```

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.
*
*****

```

```

%IF %SWITCHES(DEBUG)
%THEN
  GLOBAL D: INITIAL(%BLISS(BLISS16));
  %IF %BLISS(BLISS16) %THEN
    REQUIRE 'SYS$LIBRARY:TUTIO';
    MACRO OUTPUT (X)[] =
      (IF .D THEN (TTY_PUT_QUO(X); TTY_PUT_CRLF())) %;
  %FI
  %IF %BLISS(BLISS32) %THEN
    EXTERNAL ROUTINE SOR$$OUTPUT;
    MACRO OUTPUT (X)[] =
      (IF .D THEN SOR$$OUTPUT(UPLIT(%CHARCOUNT(X),UPLIT BYTE(X))
        %IF %LENGTH GTR 1 %THEN , %REMAINING %FI )) %;
  %FI
%ELSE
  MACRO
    OUTPUT_(X) = %;
%FI

%IF %BLISS(BLISS32) %THEN
  REQUIRE 'SRC$:COM';           ! Common definitions for VAX-11 SORT/MERGE
  LIBRARY 'SRC$:SRTSPC';       ! Common definitions needed for this module
  LIBRARY 'SRC$:SFKEYWRD';     ! Spec file keyword definitions
%ELSE
  LIBRARY 'S11V3SRC:SMCOM';    ! Common definitions for PDP-11 SORT/MERGE
  LIBRARY 'S11V3SRC:SRTSPC';   ! Common definitions needed for this module
  LIBRARY 'S11V3SRC:SFKEYWRD'; ! Spec file keyword definitions
%FI

MACRO

```

MAC
 *CV
 *CV
 *AD
 *AD
 *SU
 *SU
 *MU
 *MU
 *DI
 *DI
 *CV
 *CV
 *CV
 *CV
 *CV
 *CV
 *AC
 *MO
 *CM
 *MN
 *TS
 *EM
 *PO
 *CV
 *AD
 *AD
 *SU
 *SU
 *MU
 *MU
 *DI
 *DI
 *CV
 *CV
 *CV
 *CV
 *CV
 *CV
 *CV
 *AC
 *MO
 *CM
 *MN
 *TS
 *EM
 *PO
 *CV
 *CL
 *MO
 *MO
 *PU
 *CV
 *CV
 *CV

```
IF_ERROR_( X ) = %IF %BLISS( BLISS16 ) %THEN IF X NEQ SSS_NORMAL
%ELSE IF NOT X %FI-%;
```

```
MACRO
WRN_(MSG) =
  BEGIN
  IF ERR_CNTL( %IF %DECLARED(KYW_LINE) %THEN .KYW_LINE %ELSE 0 %FI,
  MSG )
  NEQ SUCCESS THEN RETURN FAIL;
  END %;
ERR_(LINE, MSG) =
  BEGIN
  IF ERR_CNTL( LINE,
  %IF %BLISS( BLISS16 ) %THEN -ABS( MSG ) %ELSE MSG %FI )
  NEQ SUCCESS THEN RETURN FAIL;
  END %;
```

```
MACRO
ERR_CNTL = EC$ERR_CNTL %,
SPC_HEAP = SH$SPC_HEAP %,
SPC_ALLOC = SA$SPC_ALLOC %,
SKIP_IGNORED = SI$SKIP_IGNORED %,
SKIP_COMMA = SC$SKIP_COMMA %,
GET_KYW_TYPE = GKT$GET_KYW_TYPE %,
GET_CHAR_CLAUSE = GCC$GET_CHAR_CLAUSE %,
GET_ONE_CHAR = GOC$GET_ONE_CHAR %,
GET_NEXT_SPEC = GNS$GET_NEXT_SPEC %,
GET_SUB_SPEC = GSS$GET_SUB_SPEC %,
GET_FILE_SPEC = GFS$GET_FILE_SPEC %,
GET_STRING = GSS$GET_STRING %,
PARSE_COLL = PC$PARSE_COLL %,
PARSE_MOD = PM$PARSE_MOD %,
PARSE_IGN = PI$PARSE_IGN %,
PARSE_TEST = PT$PARSE_TEST %,
PARSE_KEY = PK$PARSE_KEY %,
CONV_CONSTANTS = CC$CONV_CONSTANTS %,
SEARCH_TABLE = ST$SEARCH_TABLE %,
INIT_CS_TAB = ICT$INIT_CS_TAB %,
DO_FOLD = DF$DO_FOLD %,
CVT_ATB = CAS$CVT_ATB %;
```

```
%IF %BLISS(BLISS16) %THEN
```

```
MACRO
SOR$$$FPRS =
  %IF VAR_IS_SORT_(%VARIANT) %THEN
  $$$FPR
  %ELSE
  $$$MSPR
  %FI
%;
```

```
%FI
```

```
!EXTERNAL ROUTINE
! ERR_CNTL : CA_LINKAGE; ! Error control routine
```

```
%IF NOT %DECLARED(SOR$WKAREA) %THEN
LITERAL SOR$WKAREA = SOR$_SRTIWA;
%FI
```

! Define the keyword literals (KW_xxx)

```
MACRO NAM_[A, B] = %NAME('KW_',A) = %COUNT %;  
LITERAL NAM_( KEYWORDS );
```

OPC

MAC

LIT

UND

```

%IF NOT %DECLARED(FAIL ) %THEN LITERAL FAIL = 0; %FI
%IF NOT %DECLARED(SUCCESS) %THEN LITERAL SUCCESS = 1; %FI
%IF NOT %DECLARED(FALSE ) %THEN LITERAL FALSE = 0; %FI
%IF NOT %DECLARED(TRUE ) %THEN LITERAL TRUE = 1; %FI

```

MACRO

```

GC_(X,O,Y) =
  %IF %IDENTICAL(X,S) %THEN 1 %ELSE
  %IF %IDENTICAL(X,D) %THEN 2 %ELSE X %FI %FI ^ 4 +
  %IF %IDENTICAL(O,LT) %THEN %B'11' %ELSE
  %IF %IDENTICAL(O,EQ) %THEN %B'00' %ELSE
  %IF %IDENTICAL(O,GT) %THEN %B'01' %ELSE 0 %FI %FI %FI ^ 2 +
  %IF %IDENTICAL(Y,S) %THEN 1 %ELSE
  %IF %IDENTICAL(Y,D) %THEN 2 %ELSE Y %FI %FI ^ 0 %,
GC_L1_(X) = X<4,2,0> %,
GC_OP_(O) = O<2,2,1> %,
GC_L2_(Y) = Y<0,2,0> %;

```

LITERAL

```

GC_SINGLE= 1, ! returns from get_char_clause
GC_DOUBLE= 2, ! single char
GC_S_TO_S= 3, ! double char
! single - single

```

LITERAL

```

MAX_CONDX = TDT_MAX, ! Max conditions in omit/incls
MAX_CONST = CFT_MAX, ! Max constants in omit/incls
MAX_FIELDS = FDT_MAX; ! Max fields definitions

```

LITERAL

```

%UPADDR = ( %BPADDR + %BPUNIT -1 ) / %BPUNIT; ! Units per address

```

```

! Definitions of fields in SYM_TAB

```

MACRO

```

SYM_NAM_ADR = 0, 0, %BPADDR, 0 %, ! Address of name in spec buffer
SYM_NAM_LEN = 1, 0, 8, 0 %, ! Length of name in spec buffer
SYM_INDEX = 1, 8, 8, 0 %, ! Index into FDT or TDT

```

STRUCTURE

```

SYM_TAB[ O,B,P,S,E; BS ] = ! Local symbol table
  [ BS*(%UPADDR+2) ]
  ( SYM_TAB + O*(%UPADDR+2) + B*%UPADDR )<P,S,E>;

```

```

! Definitions of fields in CON_SYM_TAB

```

MACRO

```

CON_NAM_ADR = 0, 0, %BPADDR, 0 %, ! Address of name in spec buffer
CON_NAM_LEN = 1, 0, 8, 0 %, ! Length of name in spec buffer
CON_INDEX = 1, 8, 8, 0 %, ! Index into appropriate table

```

```

%IF %BLISS(BLISS32) %THEN

```

MACRO

```

CON_LENGTH = 1, 16, 8, 0 %, ! Result length, for condx key/data only

```

```

%ELSE

```

MACRO

```

CON_LENGTH = 2, 0, 8, 0 %, ! Result length, for condx key/data only

```

```

%FI

```

```

STRUCTURE                                     ! Local constant symbol table
CON_SYM_TAB[ O,B,P,S,E; BS ] =
  [ BS*(%UPADDR+4) ]
  ( CON_SYM_TAB + 0*(%UPADDR+4) + B*%UPADDR )<P,S,E>;

```

```

MACRO
LOWER_(X) = ((X) OR %X'20') %;
UPPER_(X) = ((X) AND NOT %X'20') %;

```

```

LITERAL
_LEN = 0,
_PTR = 1,
_LINE = 2;

```

```

!MACRO
ALLOC (X) =
  %IF %CTCE(X) %THEN %IF X EQL 0
  %THEN .CA[CA WRK ADR]
  %ELSE SPC_ALLOC(X) %FI
  %ELSE SPC_ALLOC(X) %FI %;

```

```

MACRO
ALLOC (X) =
  %IF %CTCE(X) AND (X) EQL 0
  %THEN .CA[CA WRK ADR]
  %ELSE %IF %BLISS(BLISS32) %THEN SPC_ALLOC(X)
  %ELSE BEGIN
    LOCAL S;
    IF (S = SPC_ALLOC(X)) EQL 0 THEN RETURN FAIL;
    .S
  END
  %FI %;

```

```

MACRO
HEAP (X) =
  %IF %BLISS(BLISS32) %THEN SPC_HEAP(X)
  %ELSE BEGIN
    LOCAL S;
    IF (S = SPC_HEAP(X)) EQL 0 THEN RETURN FAIL;
    .S
  END
  %FI %;

```

```

LITERAL
TRDT_UNIT = 3;                                     ! Temporary RDT, first three bytes of RDT

```

```

STRUCTURE
TRDT_TAB[ O,B,P,S,E; BS ] =
  [ BS*TRDT_UNIT ]
  ( TRDT_TAB + 0*TRDT_UNIT + B )<P,S,E>;

```

```

MACRO
TRDT_INCLUDE      = 0, 0, 1, 0 %;                ! Include/omit, Include = 1
TRDT_CONDX        = 0, 1, 1, 0 %;                ! Conditional = 1
TRDT_TDT_IDX      = 1, 0, 8, 0 %;                ! Index into TDT

```


