```
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBB         BBB  RRR         RRR         TTT         LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR             TTT         LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR             TTT         LLL
CCC              000         000  BBBBBBBBBBBB     RRRRRRRRRRRR             TTT         LLL
CCC              000         000  BBB         BBB  RRR   RRR               TTT         LLL
CCC              000         000  BBB         BBB  RRR   RRR               TTT         LLL
CCC              000         000  BBB         BBB  RRR   RRR               TTT         LLL
CCC              000         000  BBB         BBB  RRR      RRR            TTT         LLL
CCC              000         000  BBB         BBB  RRR      RRR            TTT         LLL
CCC              000         000  BBB         BBB  RRR      RRR            TTT         LLL
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLL
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLL
   CCCCCCCCCCC     000000000     BBBBBBBBBBBB     RRR         RRR         TTT         LLLLLLLLLLLLLLL
```

```
IIIIII    NN      NN  TTTTTTTTTT  PPPPPPPP      AAAAAA    RRRRRRRR
IIIIII    NN      NN  TTTTTTTTTT  PPPPPPPP      AAAAAA    RRRRRRRR
  II      NN      NN      TT      PP      PP  AA      AA  RR      RR
  II      NN      NN      TT      PP      PP  AA      AA  RR      RR
  II      NNNN    NN      TT      PP      PP  AA      AA  RR      RR
  II      NNNN    NN      TT      PP      PP  AA      AA  RR      RR
  II      NN  NN  NN      TT      PPPPPPPP    AA      AA  RRRRRRRR
  II      NN  NN  NN      TT      PPPPPPPP    AA      AA  RRRRRRRR
  II      NN    NNNN      TT      PP          AAAAAAAAAA  RR  RR
  II      NN    NNNN      TT      PP          AAAAAAAAAA  RR  RR
  II      NN      NN      TT      PP          AA      AA  RR    RR        ....
  II      NN      NN      TT      PP          AA      AA  RR    RR        ....
IIIIII    NN      NN      TT      PP          AA      AA  RR      RR      ....
IIIIII    NN      NN      TT      PP          AA      AA  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
```

```
{ REQUIRE file for COBOL Intermediate data type
{ File: INTPAR.SDL  Edit: SBL2001
{*********************************************************************
{*                                                                   *
{*  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.          *
{*                                                                   *
{*                                                                   *
{*********************************************************************
{
{ Edit history:
{ 1-001 - original DGP 29-Jan-79
{ 1-002 - Revised to reflect ECO for COBOL Intermediate Data Type
{         (SRM ECO7904-17) RKR 16-Aug-79.
{ 1-003 - Add copyright notice.  SBL 11-MAR-1980
{ 2-001 - Convert to SDL.  SBL 23-Aug-1982
{
{++
{
{ The COBOL compiler allocates twelve bytes for an intermediate
{ temporary value, which are arranged as follows:
{
{       +--------------+
{       | INT$W_I_EXP  |        : Intermediate temp
{       +--------------+
{       | INT$P_I_FRACT|
{       |              |
{       |              |
{       |              |
{       +--------------+
{
{ The fraction is a packed, 18-digit number.  The fraction is normalized
{ with the high-order digit non-zero (unless the whole number is zero).
{ The exponent is kept in true form.  If the exponent is -32768, the
{ number is the reserved operand.  Otherwise the exponent must be in the
{ range from +99 to -99 decimal.
{
{ Define the symbols which are used to access and manipulate the
{ intermediate data type.
{
MODULE $INTDEF;
```

```
AGGREGATE INT$R_DEF STRUCTURE PREFIX INT$;
    I_EXP WORD UNSIGNED;                        { Exponent field
    I_FRACT DECIMAL PRECISION (18,0);   { Fraction field

    CONSTANT I_FRACT_D  EQUALS 18;      { Length of fraction field in digits
    CONSTANT I_FRACT_L  EQUALS 10;      { Length of fraction field in bytes
    CONSTANT I_EXP_HI   EQUALS 99;      { Max value of exponent
    CONSTANT I_EXP_LO   EQUALS -99;     { Min value of exponent
    CONSTANT I_EXP_RES  EQUALS -32768;  { Exponent value for "reserved"
    CONSTANT I_LEN      EQUALS INT$K_I_FRACT_L+2; { Length of number in bytes

    END INT$R_DEF;

END_MODULE $INTDEF;
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TYPEMAIN
LIS

COBPROLOG
REQ

COBACCDAT
LIS

COBACCDWK
LIS

UNLOCK
LIS

UTILSUBS
LIS

INTPAR
SDL

COBDEF
REQ

COBACCDAY
LIS

COBACCECV
LIS

COBRTL

COBLNK
REQ

COBRTL
MAP