		BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	RRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRRR		
LLL	iii	BBB BBB	RRR RRR	TTT	
	!!!		RRR RRR	III	LLL
IIIIIIIIIIIIIII	111111111	BBBBBBBBBBBB	RRR RRR	III	rrrrrrrrrrr
LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL	111111111	888888888888	RRR RRR	III	LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL	111111111	BBBBBBBBBBBB	RRR RRR	TTT	IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII

LI

000000 000000 00	\$	MM	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA	2222222 2222222 2222222 2222222 2222222
RRRRRRRR RR RR RR RR RR RR RR RR RR RRRR	GQQQQQ QQ QQ QQ QQ			

This file, OTSMAC.REQ, defines OTS macros. Edit: SBL1039

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.

## ! Author: T. Hastings

1-25 - REQUIRE LPSECT. TNH 19-Dec-77
1-26 - Remove SET\_CB\_BASE(). JMT 12-Apr-78
1-27 - Use RTLIN: logical name in REQUIRE. TNH 28-Apr-78
1-28 - Define ADR\_VECTOR. TNH 7-June-78
1-30 - Change name to FORMAC.REQ (with apologies to Dick Gruen) and change name of LPSECT to RTLPSECT JBS 14-NOV-78

1-031 - Add a copyright notice JBS 16-NOV-78
1-032 - Change file name to OTSMAC.REQ and remove REQUIRE of RTLPSECT.
(Let users of OTSMAC.REQ also REQUIRE RTLPSECT.) JBS 06-DEC-78 1-033 - Add offsets and lengths of the dispatch tables. JBS 25-JUN-1979

1-034 - Make them weak globals so they can be used by macro routines.

JBS 26-JUN-1979

1-035 - Remove FORTRAN offsets and lengths (moved to ISB). JBS for SBL 12-JUL-1979

1-036 - Remove BASIC offsets and lengths (moved to ISB). JBS 12-JUL-1979
1-037 - Remove PRINT statement, for new BLISS compiler. JBS 02-OCT-1979
1-038 - Add COPY\_BYTE\_A, COPY\_WORD\_A, COPY\_LONG\_A, COPY\_QUAD\_A
macros. SBL T8-Dec-1979
1-039 - Add ONE\_OF macro. SBL 18-Dec-1981

Macro for writing a character string and then advancing pointer Designed so that it is placed on the left side of a substitution ! statement. Anticpates feature being added to BLISS as a form

```
16-SEP-1984 16:51:26.35 Page 2
OTSMAC.REQ: 1
! for CHSWCHAR_A (DESPA) if looks good.
! Call: CH_WCHAR_A (CS_POINTER_ADR.ma.r) = ...;
MACRO
      CH_WCHAR_A (CS_POINTER_ADR) =
      (LOCAL T;

T = .CS_POINTER_ADR;

CS_POINTER_ADR = CH$PLUS (.CS_POINTER_ADR, 1);

.T)<0,8> %;
  Macro for writing a character without advancing the pointer. Desinged so that is placed on the left of a substitution statement. Anticipates feature being added to BLISS as a form
   for CHSWCHAR (DSTPV) if Tooks good.
! Call: CH_WCHAR (CS_POINTER.ra.v) = ...;
MACRO
      CH_WCHAR (CS_POINTER_VAL) =
      (CS_POINTER_VAL)<0.8> %;
! Macros for processing the compiled format text byte strings.
MACRO
      RBYTE_A(P) = (P = .P+1; .(.P-1)<0, 8>) %,
RWORD_A(P) = (P = .P+2; .(.P-2)<0,16>) %,
RLONG_A(P) = (P = .P+4; .(.P-4)<0,32>) %,
      CALL_VFE(P)=
( (LOCAL T; T = .(.P)<0,32>; P = .P+4; .T+.P) () ) %;
! Macros for copying values referenced by pointers.
MACRO
      COPY_BYTE_A (S,D) = (D=.D+1; (.D-1)<0,8>=RBYTE_A(S)) %,

COPY_WORD_A (S,D) = (D=.D+2; (.D-2)<0,16>=RWORD_A(S)) %,

COPY_LONG_A (S,D) = (D=.D+4; (.D-4)<0,32>=RLONG_A(S)) %,

COPY_QUAD_A (S,D) = ((.D)<0,32>=.(.S)<0,32>; (.D+4)<0,32>=.(.S+4)<0,32>; D=.D+8; S=.S+8) %;
  Macro to complete the transportable character pointer notion. Everywhere that an address (A) can be specified in BLISS,
 ! allow a character pointer with mnemonic P (rather than CP to keep one letter)
```

```
16-SEP-1984 16:51:26.35 Page 3
OTSMAC.REQ; 1
%BLISS32 (
           RO
LSSP = LSSA %,
LEQP = LEQA %,
EQLP = EQLA %,
NEQP = NEQA %,
GEQP = GEQA %,
GTRP = GTRA %,
MAXP = MAXA %,
MINP = MINA %;);
Clear a vector of BLISS values (transportable)
MACRO
     FILL_VAL (VALUE, LENGTH, ADDRESS) = % % SUPVAL, ADDRESS)) %;
  Allocate string descriptor
Rest of descriptor symbols are defined in SRMDEF.MDL
But currently no way in MDL to define a macro
   To declare and allocate a descriptor:
           LOCAL
                 name: DSC$DESCRIPTOR;
MACRO
           DSC$DESCRIPTOR = BLOCK[8, BYTE] %; ! MDL requires BYTE
```

MACRO

THE "ONEOF" MACRO

Macros to determine if the value of an expression is one of a set of specified small-integer values. These macros can be used only if the following conditions are met:

The value to be tested is in the range 0 through 127.

The values to be tested for are all in the range 0 through 31.

Example:

IF ONE\_OF (.X, 1,3,5) ...

The code generated is much more efficient than a series of comparisons (provided that the values being tested are all compile-time constants).

XBMSK [A]=

XIF NOT %CTCE(A) %THEN %ERRORMACRO('ONE\_OF argument not a CTCE') %FI

XIF (A GTRU 31) %THEN %ERRORMACRO('ONE\_OF constant greater than 31') %FI

(1 ^ (31 - (A))) %,

BMSK\_[]=
(0 OR XBMSK\_(%REMAINING)) %,

XCMP [A,B,C]=
XIF XLENGTH EQL 3
XTHEN

((A EQLU B) OR (A EQLU C))
XELSE

(A EQLU B)
XFI X,

ONE\_OF(A) =

%IF %LENGTH LEQ 1 %THEN %ERRORMACRO('Too few arguments to ONE\_OF') %FI
%IF %LENGTH LEQ 3

%THEN

XCMP\_(A, %REMAINING)

%ELSE

(( BMSK\_(%REMAINING) ) ^ (A)) LSS 0)

%FI %;

End of file OTSMAC.REQ

0202 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

