


```

KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EE          YY      YY  PP      PP  AA      AA  DD      DD  DD      DD  EE          FF
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF
KK      KK  EEEEEEEEE  YY      YY  PPPPPPP  AAAAAA  DDDDDDD  DDDDDDD  EEEEEEEEE  FFFFFFFF

```

```

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

```

```

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

```



```

*****
*
* 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.
*
*****

```

```

+
This file, KEYPADDEF.REQ, defines symbols and structures for the keypad,
and for definable keys in general.

```

```

Revision history:

```

- 1-001 - Original. JBS 13-Aug-1982
 - 1-002 - Provide a large hash table on VAX/VMS. JBS 17-Aug-1982
- ```

<BLF/PAGE>

```

```

!+
! Define symbols for the values that correspond to the keys.
! Values 0 to 127 are ASCII characters, 128 to 255 are the
! DEC Multinational Supplementary set.
!-

```

```

LITERAL
 K_KPAD_BASE = 300, ! Base for the 22 keypad keys
 K_FUN_BASE = 400, ! Base for the 100 function keys
 K_MAX_FUN_VAL = 99, ! Maximum function key value
 K_GOLD_BASE = 500, ! Add for GOLD prefix
 K_KEY_MAX = 999; ! Maximum key value

```

```

!+
! Define the special keypad keys.
! These are the numeric values used for the definable keys.
!-

```

```

LITERAL
 K_0 = K_KPAD_BASE + 0,
 K_1 = K_KPAD_BASE + 1,
 K_2 = K_KPAD_BASE + 2,
 K_3 = K_KPAD_BASE + 3,
 K_4 = K_KPAD_BASE + 4,
 K_5 = K_KPAD_BASE + 5,
 K_6 = K_KPAD_BASE + 6,
 K_7 = K_KPAD_BASE + 7,
 K_8 = K_KPAD_BASE + 8,
 K_9 = K_KPAD_BASE + 9,
 K_PF2 = K_KPAD_BASE + 10,
 K_PF3 = K_KPAD_BASE + 11,
 K_UP = K_KPAD_BASE + 12,
 K_DOWN = K_KPAD_BASE + 13,
 K_RIGHT = K_KPAD_BASE + 14,
 K_LEFT = K_KPAD_BASE + 15,
 K_DOT = K_KPAD_BASE + 16,
 K_PF4 = K_KPAD_BASE + 17,
 K_MINUS = K_KPAD_BASE + 18,
 K_COMMA = K_KPAD_BASE + 19,
 K_PF1 = K_KPAD_BASE + 20,
 K_ENTER = K_KPAD_BASE + 21;

```

```

!+
! The following structure holds a defined key.
!-

```

```

FIELD
 KEY_DEF_FIELD =
 SET
 KEY_DEF_NEXT = [0, 0, %BPADDR, 0] ! Pointer to next in this bucket, must be first
 KEY_DEF_KEY = [%BPADDR/8, 0, 16, 0] ! code for this key, 0 to K_KEY_MAX
 KEY_DEF_LEN = [(%BPADDR/8) + 2, 0, 8, 0] ! Length of the definition
 KEY_DEF_TEXT = [(%BPADDR/8) + 3, 0, 0, 0] ! Start of definition text
 TES;

```



```
LITERAL
 KEY_DEF_OVERHEAD = %FIELDEXPAND (KEY_DEF_TEXT, 0); ! Amount to add to text size to get structure size
!+
! Define the length of the hash table that holds pointers to key definitions.
%IF %BLISS(BLISS16) %THEN
 LITERAL K_KPAD_HASHSIZ = 1; ! Very short table for PDP-11
%ELSE
 LITERAL K_KPAD_HASHSIZ = 199; ! Plenty of room on VAX-11
%FI
!
! [End of file KEYPADDEF.REQ]
```



