

CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCC	DDD	UUU	UUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU
CCCCCCCCCCCC	DDDDDDDDDDDD	UUUUUUUUUUUUUUUU	UUUUUUUUUUUUUUUU

```
RRRRRRRR 000000 UU UU TTTTTTTTTT IIIIII NN NN EEEEEEEEE SSSSSSSS
RRRRRRRR 000000 UU UU TTTTTTTTTT IIIIII NN NN EEEEEEEEE SSSSSSSS
RR RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR RR 00 00 UU UU TT TT NN NN EE SS
RRRRRRRR 00 00 UU UU TT TT NN NN EE SS
RRRRRRRR 00 00 UU UU TT TT NN NN EE SS
RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR 00 00 UU UU TT TT NN NN EE SS
RR RR 000000 UUUUUUUUUU TT IIIIII NN NN EEEEEEEEE SSSSSSSS
RR RR 000000 UUUUUUUUUU TT IIIIII NN NN EEEEEEEEE SSSSSSSS

LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS
```

ROUTINES
Table of contents

Lookup DCL internal routine names ^{H 13}

15-SEP-1984 23:55:16 VAX/VMS Macro V04-00

Page 0

(2)	59	Definitions
(2)	68	Internal routine table
(3)	111	Verb type code table
(4)	142	LOOKUP_ROUTINE, lookup routine name
(5)	170	LOOKUP_VERB_TYPE, lookup verb type code


```
0000 1      .title ROUTINES,      Lookup DCL internal routine names
0000 2      .ident 'V04-000'
0000 3
0000 4      *****
0000 5      *
0000 6      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 7      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 8      *  ALL RIGHTS RESERVED.
0000 9      *
0000 10     *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 11     *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 12     *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 13     *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 14     *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 15     *  TRANSFERRED.
0000 16     *
0000 17     *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 18     *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 19     *  CORPORATION.
0000 20     *
0000 21     *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 22     *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 23     *
0000 24     *
0000 25     *****
0000 26
0000 27     Abstract:
0000 28
0000 29     This module contains a list of internal dcl routine names
0000 30     and a routine to lookup the routine name and return the
0000 31     internal case index for that routine. It is coded in MACRO
0000 32     in order to use the "INTIMAGES" macro.
0000 33
0000 34     Author: Tim Halvorsen, Sept 1980
0000 35
0000 36     MODIFIED BY:
0000 37
0000 38     V03-003 PCA0000      Paul C. Anagnostopoulos 22-Mar-1983
0000 39     This module will hopefully become obsolete before V4 ships,
0000 40     and will then be removed.
0000 41
0000 42     V03-002 CWH0002      CW Hobbs      5-Aug-1982
0000 43     Add correct number of dummy parameters for previous fix.
0000 44
0000 45     V03-001 CWH0001      CW Hobbs      3-Aug-1982
0000 46     Add dummy parameters when $DEFINI and $DEFEND are
0000 47     redefined to null. A change in the MACRO-32 assembler
0000 48     produced error messages when a macro defined with
0000 49     no formal parameters is called with actuals.
0000 50
0000 51     V002      DWT0019      David W. Thiel      7-Jan-1982
0000 52     Use MCRINTMG for MCR routine names and addresses.
0000 53
0000 54     V001      DWT0005      David W. Thiel      09-Dec-1981
0000 55     Support separate tables for MCR and DCL.
0000 56
0000 57     :---
```

```
0000 59      .sbtcl Definitions
0000 60
0000 61      ;
0000 62      ; Define data structures
0000 63      ;
0000 64
0000 65      $$clitabdef
0000 66
0000 67
0000 68      .sbtcl Internal routine table
0000 69      ;
0000 70      ; Define internal image name
0000 71      ;
0000 72
0000 73      .macro intimage image,?l1
0000 74      .save
0000 75      .psect $split$,nowrt,noexe,word
0000 76 l1:    .ascic /image/
0000 77      .restore
0000 78      .long l1                      ; address of keyword string
0000 79      .long $intimage$            ; associated value
0000 80      $intimage$=$intimage$+1
0000 81      .endm intimage
0000 82
0000 83      ;
0000 84      ; Generate internal routine table
0000 85      ;
0000 86
00000000 87      .psect $code$,nowrt,exe,word
0000 88
0000 89      .enabl lsb
0000 90 dclinttbl:
0000006c' 91      .long <90$-10$>/4      ; number of entries in table
0004 92 10$:    intimages                ; generate table entries (8 bytes each)
01B4 93 90$:
01B4 94      .dsabl lsb
01B4 95
01B4 96      .enabl lsb
0000003c' 97 mcrinttbl:
01B4 98      .long <90$-10$>/4      ; number of entries in table
01B8 99 10$:    mcrintimg                ; generate table entries (8 bytes each)
02A8 100 90$:
02A8 101      .dsabl lsb
02A8 102
02A8 103 inttbls:
00000000' 02A8 104      .address 0
02AC 105      assume vec_k_dcl,equal,<<.-inttbls>/4>
00000000' 02AC 106      .address dclinttbl      ; DCL table pointer
02B0 107      assume vec_k_mcr,equal,<<.-inttbls>/4>
000001B4' 02B0 108      .address mcrinttbl      ; MCR table pointer
02B4 109
```



```
02B4 111 .sbtll Verb type code table
02B4 112 ;
02B4 113 ; Define verb type code table
02B4 114 ;
02B4 115 .macro $defini dum1,dum2,dum3 ; Redefine $DEFINI macro to null
02B4 116 .endm
02B4 117 .macro $defend dum1,dum2,dum3 ; Redefine $DEFEND macro to null
02B4 118 .endm
02B4 119 .macro $sequ symbol,value
02B4 120 s = %locate(<VERB>,symbol)+5 ; locate name portion of symbol
02B4 121 l = %length(symbol)-s
02B4 122 .iif gt l-4, l = 4 ; l = max(l,4)
02B4 123 .ascii /%extract(s,l,symbol)/ ; store first 4 chars of verb
02B4 124 .blkb 4-l ; pad to longword
02B4 125 .long value ; associated verb type code
02B4 126 .endm $sequ
02B4 127
02B4 128 dclverbtypes:
02B4 129 $clverbdef ; generate table entries (8 bytes each)
00000000 0494 130 .long 0 ; end of table
0498 131
00000003 0498 132 .mdelete $sequ,$defini,$defend
0498 133
0498 134 verbtypes:
00000000 0498 135 .address 0
049C 136 assume vec_k_dcl,equal,<<.-verbtypes>/4>
000002B4 049C 137 .address dclverbtypes ; DCL table pointer
04A0 138 assume vec_k_mcr,equal,<<.-verbtypes>/4>
000002B4 04A0 139 .address dclverbtypes ; ** temp MCR table pointer
04A4 140
```

```
04A4 142 .sbtll LOOKUP_ROUTINE, lookup routine name
04A4 143 :---
04A4 144 :
04A4 145 : Lookup a routine name in the table of internal DCL routines
04A4 146 : and return the associated internal case index.
04A4 147 :
04A4 148 : Inputs:
04A4 149 :
04A4 150 : 4(ap) = Address of descriptor of routine name
04A4 151 :
04A4 152 : Outputs:
04A4 153 :
04A4 154 : r0 = Case index if found, else 0
04A4 155 :---
04A4 156 :
0000 04A4 157 .entry lookup_routine,0
04A6 158
04A8 159 pushal -(sp) ; Address of longword to receive value
50 0000'CF 9A 04A8 160 movzbl w^clitype,r0 ; Index in table of tables
FDF6 CF40 DD 04AD 161 pushl inttbls[r0] ; Address of keyword table
04 AC DD 04B2 162 pushl 4(ap) ; Address of search string descriptor
00000000'GF 03 FB 04B5 163 calls #3,g^lib$lookup_key ; Lookup routine name in table
04 50 E9 04BC 164 blbc r0,80$ ; Branch if not found
50 8ED0 04BF 165 popl r0 ; Return value in R0
04 04C2 166 ret
50 D4 04C3 167 80$: clrl r0 ; Return failure
04 04C5 168 ret
```



```
04C6 170 .sbtll LOOKUP_VERB_TYPE, lookup verb type code
04C6 171 :---
04C6 172 :
04C6 173 Lookup a verb in the table of verb type codes
04C6 174 defined by the macro $CLIVERBDEF and return
04C6 175 the associated verb type code.
04C6 176
04C6 177 Inputs:
04C6 178
04C6 179 4(ap) = Address of descriptor of verb name
04C6 180
04C6 181 Outputs:
04C6 182
04C6 183 r0 = Verb type code if found, else 0
04C6 184 :---
04C6 185
01FC 04C6 186 .entry lookup_verb_type, ^m<r2,r3,r4,r5,r6,r7,r8>
04C8 187
50 0000'CF 9A 04C8 188 movzbl w^clitype,r0 ; Index in table of tables
58 C7 AF40 D0 04CD 189 movl verbtypes[r0],r8 ; Address of keyword table
56 04 BC 7D 04D2 190 movq @4(ap),r6 ; Get descriptor of verb name
56 56 3C 04D6 191 movzwl r6,r6 ; Expand length to longword
04 56 D1 04D9 192 cmpl r6,#4 ; More than 4 characters?
03 15 04DC 193 bleq 10$ ; Branch if not
56 04 D0 04DE 194 movl #4,r6 ; Only compare first 4 characters
68 67 56 29 04E5 197 tstl (r8) ; End of table?
05 13 04E3 196 beql 80$ ; If so, return with failure
58 08 C0 04EB 199 cmpc3 r6,(r7),(r8) ; Is verb listed in table?
F1 11 04EE 200 beql 90$ ; Branch if so
50 04 A8 D0 04F0 201 addl #8,r8 ; Skip to next entry
04 04F4 202 brb 10$ ; and continue searching
50 04 F5 203 movl 4(r8),r0 ; Return with assoc. verb type code
04 04F7 204 ret ; Return failure
04F8 205
04F8 206 .end
```


ROUTINES
Symbol table

Lookup DCL internal routine names N 13

15-SEP-1984 23:55:16 VAX/VMS Macro V04-00
4-SEP-1984 23:09:32 [CDU.SRC]ROUTINES.MAR;1

Page 6
(5)

```
$INTIMAGES      = 0000009E
CLITYPE         ***** X 02
DCLINTTBL       00000000 R 02
DCLVERBTYPES    000002B4 R 02
INTTBL          000002A8 R 02
L               = 00000004
LIB$LOOKUP_KEY  ***** X 02
LOOKUP_ROUTINE  000004A4 RG 02
LOOKUP_VERB_TYPE 000004C6 RG 02
MCRINTTBL       000001B4 R 02
S               = 0000000B
VEC_K_DCL       = 00000001
VEC_K_MCR       = 00000002
VERBTYPES       00000498 R 02
```

+-----+
! Psect synopsis !
+-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$CODE\$	000004F8 (1272.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC WORD
\$SPLIT\$	0000026E (622.)	03 (3.)	NOPIC USR CON REL LCL NOSHR NOEXE RD NOWRT NOVEC WORD

+-----+
! Performance indicators !
+-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	32	00:00:00.13	00:00:00.96
Command processing	121	00:00:00.80	00:00:04.93
Pass 1	208	00:00:06.06	00:00:15.10
Symbol table sort	0	00:00:00.36	00:00:00.59
Pass 2	59	00:00:01.31	00:00:03.49
Symbol table output	4	00:00:00.03	00:00:00.08
Psect synopsis output	1	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	427	00:00:08.72	00:00:25.19

The working set limit was 1350 pages.
32222 bytes (63 pages) of virtual memory were used to buffer the intermediate code.
There were 20 pages of symbol table space allocated to hold 185 non-local and 92 local symbols.
206 source lines were read in Pass 1, producing 30 object records in Pass 2.
15 pages of virtual memory were used to define 9 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

\$255\$DUA28:[SHRLIB]MCR.MLB;1
\$255\$DUA28:[SHRLIB]DCL.MLB;1
\$255\$DUA28:[SYSLIB]STARLET.MLB;2
TOTALS (all libraries)

2
1
5
8

394 GETS were required to define 8 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:ROUTINES/OBJ=OBJ\$:ROUTINES MSRC\$:ROUTINES/UPDATE=(ENH\$:ROUTINES)+SHRLIB\$:DCL/LIB+SHRLIB\$:MCR/LIB

0044 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

