


```

000000    TTTTTTTTTT    SSSSSSSS    LL    UU    UU    NN    NN
000000    TTTTTTTTTT    SSSSSSSS    LL    UU    UU    NN    NN
00        TT        SS        LL    UU    UU    NN    NN
00        TT        SS        LL    UU    UU    NN    NN
00        TT        SS        LL    UU    UU    NNNN    NN
00        TT        SS        LL    UU    UU    NNNN    NN
00        TT        SSSSSS    LL    UU    UU    NN    NN
00        TT        SSSSSS    LL    UU    UU    NN    NN
00        TT        SSSSSS    LL    UU    UU    NN    NN
00        TT        SS        LL    UU    UU    NN    NNNN
00        TT        SS        LL    UU    UU    NN    NNNN
00        TT        SS        LL    UU    UU    NN    NN
00        TT        SS        LL    UU    UU    NN    NN
000000    TT        SSSSSSSS    LLLLLLLLLL    UUUUUUUUUU    NN    NN
000000    TT        SSSSSSSS    LLLLLLLLLL    UUUUUUUUUU    NN    NN
                                SSSSSSSS    SSSSSSSS
                                SSSSSSSS    SSSSSSSS
                                SS        SS        SS        SS
                                SS        SS        SS        SS
                                SS        SS        SS        SS
                                SSSSSS    SSSSSS    SS
                                SSSSSS    SSSSSS    SS
                                SS        SS        SS        SS
                                SSSSSSSS    SSSSSSSS
                                SSSSSSSS    SSSSSSSS

```


(2) 50
(3) 84
(4) 168
(5) 252

DECLARATIONS
OTSS\$GET_LUN - Allocate one logical unit number
OTSS\$FREE_LUN - Deallocate one logical unit number
OTSS\$TAKE_LUN - Take a particular LUN

```

0000 1      .TITLE OTSS$SLUN - Resource allocator for logical unit numbers
0000 2      .IDENT /1-003/ ; File: OTSLUN.MAR
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :* ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :* TRANSFERRED.
0000 17 :*
0000 18 :* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :* CORPORATION.
0000 21 :*
0000 22 :* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27 :
0000 28
0000 29 :++
0000 30 : FACILITY: Language-independent language support
0000 31 :
0000 32 : ABSTRACT:
0000 33 :
0000 34 :     Three routines for allocating and deallocating logical unit
0000 35 :     numbers. Using these routines allows use of logical unit
0000 36 :     numbers by multiple procedures without conflicts.
0000 37 :
0000 38 : ENVIRONMENT: User Mode, AST Reentrant
0000 39 :
0000 40 : --
0000 41 : AUTHOR: John Sauter, CREATION DATE: 26-JAN-1979
0000 42 :
0000 43 : MODIFIED BY:
0000 44 :
0000 45 :
0000 46 : 1-001 - Original from LIB$EF. JBS 26-JAN-1979
0000 47 : 1-002 - Don't try to use FFC with a 100-bit size. JBS 26-JAN-1979
0000 48 : 1-003 - Allocate 100 bits, not 99. JBS 27-JAN-1979

```


[illegible]

```

0000 84 .SBTTL OTSS$GET_LUN - Allocate one logical unit number
0000 85 :++
0000 86 : FUNCTIONAL DESCRIPTION:
0000 87 :
0000 88 : OTSS$GET_LUN allocates one logical unit number from a process-wide
0000 89 : pool. If a LUN is available for use, its number is returned
0000 90 : to the caller. If no LUNs are available, an error is returned
0000 91 : as the function value.
0000 92 :
0000 93 : CALLING SEQUENCE:
0000 94 :
0000 95 : status.wlc.v = OTSS$GET_LUN (LUN.wl.r)
0000 96 :
0000 97 : INPUT PARAMETERS:
0000 98 :
0000 99 : NONE
0000 100 :
0000 101 : IMPLICIT INPUTS:
0000 102 :
0000 103 : LUN_POOL, a table of available logical unit numbers in
0000 104 : OWN storage
0000 105 :
0000 106 : OUTPUT PARAMETERS:
0000 107 :
0000 108 : LUN.wl.r - The LUN allocated, or -1 if none were available.
0000 109 :
0000 110 : IMPLICIT OUTPUTS:
0000 111 :
0000 112 : If successful, an entry is made into LUN_POOL indicating that
0000 113 : a logical unit number has been reserved.
0000 114 :
0000 115 : FUNCTION VALUE:
0000 116 : COMPLETION CODES:
0000 117 :
0000 118 : 1 = success, 0 = failure
0000 119 : SIDE EFFECTS:
0000 120 :
0000 121 : NONE
0000 122 :
0000 123 : --
0000 124 :
4000 0000 125 .ENTRY OTSS$GET_LUN, ^M<IV> ; Save nothing
0002 126 :
0002 127 :+
0002 128 : Scan LUN_POOL for first available LUN
0002 129 :-
0002 130 :
0002 131 SCAN:
0002 132 SKPC #255, #<LUB$K_LUN_MAX+8>/8, LUN_POOL
000B 133 BEQL ALL_OUT ; None available
000D 134 R0, #<LUB$K_LUN_MAX+8>/8, R0
0011 135 ASHL #3, R0, R0 ; Approximate LUN
0015 136 FFC #0, #8, (R1), R1 ; Find exact LUN
001A 137 BEQL SCAN ; An AST took all in this byte
001C 138 ADDL2 R1, R0 ; Exact LUN
001F 139 CMPL R0, #LUB$K_LUN_MAX ; Is it too big?
0026 140 BGTR ALL_OUT ; Yes, all LUNs are reserved

```

```

00000000'EF  OF  FF  8F  3B
          50  OF  50  C3
          50  50  03  78
51  61  08  00  EB
          E6  13
          50  C0
00000077 8F  50  D1
          10  14

```


			0028	141		
			0028	142	:+	
			0028	143	:	
			0028	144	:	Now recheck and set the bit under interlock, in case someone
			0028	145	:-	has set it at AST level in the meantime.
			0028	146	:	
D2	00000000'EF	50	E2	0028	147	FOUND:
				0028	148	BBSS R0, LUN_POOL, SCAN ; Repeat scan if already set
				0030	149	
				0030	150	:+
				0030	151	:
				0030	152	:-
				0030	153	:
04	BC	50	D0	0030	154	MOVL R0, @lun(AP) ; Store LUN
	50	01	D0	0034	155	MOVL #1, R0 ; SS\$ NORMAL
			04	0037	156	RET ; Exit
				0038	157	
				0038	158	:+
				0038	159	:
				0038	160	:-
				0038	161	:
04	BC	01	CE	0038	162	ALL_OUT:
				0038	163	MNEGL #1, @lun(AP) ; Set LUN to -1
			D4	003C	164	CLRL R0 ; Insufficient LUNs
			04	003E	165	RET ; Exit
				003F	166	

```

003F 168 .SBTTL OTSS$FREE_LUN - Deallocate one logical unit number
003F 169 :++
003F 170 : FUNCTIONAL DESCRIPTION:
003F 171 :
003F 172 : OTSS$FREE_LUN is the complement of OTSS$GET_LUN. When a routine
003F 173 : called OTSS$GET_LUN to allocate a LUN, and no
003F 174 : longer needs it, OTSS$FREE_LUN should be called to free the
003F 175 : LUN for use by other routines.
003F 176 :
003F 177 : CALLING SEQUENCE:
003F 178 :
003F 179 : status.wlc.v = OTSS$FREE_LUN (LUN.rl.r)
003F 180 :
003F 181 : INPUT PARAMETERS:
003F 182 :
003F 183 : LUN.rl.r - The logical unit number to be
003F 184 : deallocated. This is the value returned
003F 185 : to the user by OTSS$GET_LUN.
003F 186 :
003F 187 : IMPLICIT INPUTS:
003F 188 :
003F 189 : LUN_POOL, a table of available logical unit numbers in OWN
003F 190 : storage.
003F 191 :
003F 192 : OUTPUT PARAMETERS:
003F 193 :
003F 194 : NONE
003F 195 :
003F 196 : IMPLICIT OUTPUTS:
003F 197 :
003F 198 : An entry is made in LUN_POOL indicating that the LUN
003F 199 : is free for use.
003F 200 :
003F 201 : FUNCTION VALUE:
003F 202 : COMPLETION CODES:
003F 203 :
003F 204 : 1 = success, 0 = failure
003F 205 : SIDE EFFECTS:
003F 206 :
003F 207 : NONE
003F 208 :
003F 209 : --
003F 210 :
4000 003F 211 .ENTRY OTSS$FREE_LUN, ^M<IV> ; Save nothing
0041 212
0041 213 :+
0041 214 : Check to see if LUN is in the proper range.
0041 215 :-
0041 216 :
00000077 8F 04 BC D1 0041 217 CMPL @lun(AP), #LUB$K_LUN_MAX ; Bigger than 99?
0049 218 BGTR RES_SYS_1 ; Yes, error
004B 219 TSTL @lun(AP) ; Less than 0?
004E 220 BLSS RES_SYS_1 ; Yes, error
0050 221 :+
0050 222 : LUN is in range. Now, unset the bit.
0050 223 :-
0050 224

```


OTSS\$SLUN
1-003

H 15
- Resource allocator for logical unit nu 16-SEP-1984 00:32:30 VAX/VMS Macro V04-00
OTSS\$FREE_LUN - Deallocate one logical u 6-SEP-1984 11:14:58 [LIBRTL.SRC]OTSLUN.MAR;1

Page 6
(4)

```
07 00000000'EF 04 BC E5 0050 225 BBCC @lun(AP), LUN_POOL, ALR_FRE ; Clear but error if
                                0059 226 ; already clear.
                                0059 227
                                0059 228 :+
                                0059 229 :+ Return success
                                0059 230 :-
                                0059 231
50 01 D0 0059 232 MOVL #1, R0 ; SSS_NORMAL
04 04 005C 233 RET
                                005D 234
                                005D 235 :+
                                005D 236 :+ Error if LUN is not in the proper range
                                005D 237 :-
                                005D 238
50 D4 005D 239 RES_SYS_1:
04 04 005F 240 CLRL R0 ; LUN invalid
                                0060 241 RET
                                0060 242
                                0060 243 :+
                                0060 244 :+ Error if LUN already free.
                                0060 245 :-
                                0060 246
50 D4 0060 247 ALR_FRE:
04 04 0060 248 CLRL R0 ; LUN already free
                                0062 249 RET
                                0063 250
```

```

0063 252 .SBTTL OTSS$TAKE_LUN - Take a particular LUN
0063 253 :++
0063 254 : FUNCTIONAL DESCRIPTION:
0063 255 :
0063 256 : OTSS$TAKE_LUN is used when a routine wants to allocate
0063 257 : a particular LUN. This is different
0063 258 : from OTSS$GET_LUN which allocates an arbitrary LUN.
0063 259 :
0063 260 : To deallocate an LUN reserved with OTSS$TAKE_LUN
0063 261 : use OTSS$FREE_LUN.
0063 262 :
0063 263 : CALLING SEQUENCE:
0063 264 :
0063 265 : status.wlc.v = OTSS$TAKE_LUN (LUN.rl.r)
0063 266 :
0063 267 : INPUT PARAMETERS:
0063 268 :
0063 269 : LUN - The logical unit number desired to be
0063 270 : allocated.
0063 271 :
0063 272 : IMPLICIT INPUTS:
0063 273 :
0063 274 : LUN_POOL, a table of available LUNs located in OWN
0063 275 : storage.
0063 276 :
0063 277 : OUTPUT PARAMETERS:
0063 278 :
0063 279 : NONE
0063 280 :
0063 281 : IMPLICIT OUTPUTS:
0063 282 :
0063 283 : An entry is made in LUN_POOL indicating that the LUN
0063 284 : is allocated.
0063 285 :
0063 286 : FUNCTION VALUE:
0063 287 : COMPLETION CODES:
0063 288 :
0063 289 : 1 = success, 0 = failure
0063 290 : SIDE EFFECTS:
0063 291 :
0063 292 : NONE
0063 293 :
0063 294 :--
0063 295 :
4000 0063 296 .ENTRY OTSS$TAKE_LUN, ^M<IV> ; Save nothing
0065 297 :
0065 298 :+
0065 299 : First check to see if LUN is in range.
0065 300 :-
0065 301 :
00000077 8F 04 BC D1 0065 302 CMPL @lun(AP), #LUB$K_LUN_MAX ; Greater than 99?
0065 303 BGTR RES_SYS_2 ; Yes, error
006D 304 TSTL @lun(AP) ; Less than 0?
006F 305 BLSS RES_SYS_2 ; Yes, error
0074 306 :+
0074 307 : Now attempt to allocate that LUN
0074 308 :-

```



```

07 00000000'EF  04 BC  E2  0074  309      BBSS  @lun(AP), LUN_POOL, ALR_RES ; Reserve it, but error
                                0074  310      ; if already reserved.
                                007D  311
                                007D  312
                                007D  313  :+
                                007D  314      Return success
                                007D  315  :-
                                007D  316
50  01  D0  007D  317      MOVL  #1, R0      ; SS$ NORMAL
04  04  0080  318      RET      ; Exit
                                0081  319
                                0081  320  :+
                                0081  321      Error if LUN out of range
                                0081  322  :-
                                0081  323
50  D4  0081  324 RES_SYS_2:
04  04  0081  325      CLRL  R0      ; Invalid logical unit number
                                0083  326      RET      ; Exit
                                0084  327
                                0084  328  :+
                                0084  329      Error if already allocated.
                                0084  330  :-
                                0084  331
50  D4  0084  332 ALR_RES:
04  04  0084  333      CLRL  R0      ; Logical unit number taken already
                                0086  334      RET      ; Exit
                                0087  335
                                0087  336      .END

```

OTSS\$SLUN
Symbol table

K 15
- Resource allocator for logical unit nu 16-SEP-1984 00:32:30 VAX/VMS Macro V04-00
6-SEP-1984 11:14:58 [LIBRTL.SRC]OTSLUN.MAR;1

Page 9
(5)

ALL_OUT	00000038	R	02
ALR_FRE	00000060	R	02
ALR_RES	00000084	R	02
FOUND	00000028	R	02
LUB\$K_LUN_MAX	= 00000077		
LUB\$K_LUN_MIN	= 00000000		
LUN	= 00000004		
LUN_POOL	00000000	R	01
OTSS\$FREE_LUN	0000003F	RG	02
OTSS\$GET_LUN	00000000	RG	02
OTSS\$TAKE_LUN	00000063	RG	02
RES_SYS_1	0000005D	R	02
RES_SYS_2	00000081	R	02
SCAN	00000002	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
OTSS\$DATA	0000000F (15.)	01 (1.)	PIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC LONG
OTSS\$CODE	00000087 (135.)	02 (2.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.05	00:00:00.59
Command processing	149	00:00:00.38	00:00:02.68
Pass 1	118	00:00:00.82	00:00:03.13
Symbol table sort	0	00:00:00.08	00:00:00.68
Pass 2	67	00:00:00.45	00:00:02.05
Symbol table output	3	00:00:00.01	00:00:00.01
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	373	00:00:01.81	00:00:09.16

The working set limit was 1050 pages.
7533 bytes (15 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 146 non-local and 0 local symbols.
336 source lines were read in Pass 1, producing 18 object records in Pass 2.
6 pages of virtual memory were used to define 1 macro.

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

Macro library name	Macros defined
-\$255\$DUA28:[LIBRTL.OBJ]LIBRTL.MLB;1	1
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	1

OTSS\$SLUN
VAX-11 Macro Run Statistics - Resource allocator for logical unit ^{L 15} nu 16-SEP-1984 00:32:30 VAX/VMS Macro V04-00 Page 10
6-SEP-1984 11:14:58 [LIBRTL.SRC]OTSLUN.MAR;1 (5)

136 GETS were required to define 1 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LISS:OTSLUN/OBJ=OBJ\$:OTSLUN MSRC\$:OTSLUN/UPDATE=(ENH\$:OTSLUN)+LIB\$:LIBRTL/LI

0212 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

