


```

CCCCCCCC LL      UU      UU      SSSSSSSS UU      UU      TTTT TTTT TTTT      IIIIII      LL
CCCCCCCC LL      UU      UU      SSSSSSSS UU      UU      TTTT TTTT TTTT      IIIIII      LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SSSSSS   UU      UU      TT       II       LL
CC        LL      UU      UU      SSSSSS   UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CC        LL      UU      UU      SS       UU      UU      TT       II       LL
CCCCCCCC LLLLLLLLLL UUUUUUUUUU SSSSSSSS UUUUUUUUUU TT       TT       IIIIII      LL
CCCCCCCC LLLLLLLLLL UUUUUUUUUU SSSSSSSS UUUUUUUUUU TT       TT       IIIIII      LL
LLLLLLLLLL LLLLLLLLLL UUUUUUUUUU SSSSSSSS UUUUUUUUUU TT       TT       IIIIII      LL
LLLLLLLLLL LLLLLLLLLL UUUUUUUUUU SSSSSSSS UUUUUUUUUU TT       TT       IIIIII      LL

```

```

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
LLLLLLLLLL IIIIII      SSSSSSSS
LLLLLLLLLL IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE CLUSUTIL (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 0
6 0006 0 *****
7 0007 0 *
8 0008 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 0 * ALL RIGHTS RESERVED. *
11 0011 0 *
12 0012 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 0 * TRANSFERRED. *
18 0018 0 *
19 0019 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 0 * CORPORATION. *
22 0022 0 *
23 0023 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 0 *
26 0026 0 *
27 0027 0 *****
28 0028 0
29 0029 0 **
30 0030 0
31 0031 0 FACILITY:
32 0032 0
33 0033 0 DISMOUNT Utility Structure Levels 1 & 2
34 0034 0
35 0035 0 ABSTRACT:
36 0036 0
37 0037 0 This module contains all the various and sundry general
38 0038 0 purpose utility routines used by cluster functions within
39 0039 0 DISMOUNT. These routines are also called by the MOUNT system
40 0040 0 service.
41 0041 0
42 0042 0
43 0043 0 Environment:
44 0044 0
45 0045 0 STARLET operating system, including privileged system services
46 0046 0 and internal exec routines.
47 0047 0
48 0048 0 Author:
49 0049 0
50 0050 0 Hai Huang
51 0051 0
52 0052 0 Creation date:
53 0053 0
54 0054 0 23-feb-1984
55 0055 0
56 0056 0 Revision history:
57 0057 0

```

CLUSUTIL
V04-000

: 58
: 59
: 60
: 61

0058 0 :
0059 0 :--
0060 0
0061 1 BEGIN

D 1
15-Sep-1984 23:41:53
14-Sep-1984 12:20:02

VAX-11 Bliss-32 V4.0-742
[DISMOU.SRC]CLUSUTIL.B32;1

Page 2
(1)

D
V

' Start of CLUSUTIL

```

: 63 0062 1 LIBRARY 'SYSSLIBRARY:LIB.L32';
: 64 0063 1 REQUIRE 'SHRLIBS:CSPDEF';
: 65 0257 1 REQUIRE 'LIBS:MOUDEF.B32';
: 66 0789 1
: 67 0790 1 FORWARD ROUTINE
: 68 0791 1     SEND_CLUSTER,           ! Routine for cluster communication
: 69 0792 1     SEND_CLUSTER_HANDLER, ! Turn kernel mode signals to retron
: 70 0793 1     IN_CLUSTER;         ! Determine if the system is in a cluster
: 71 0794 1
: 72 0795 1
: 73 0796 1
: 74 0797 1 ! Define routine linkages
: 75 0798 1
: 76 0799 1 LINKAGE
: 77 0800 1     ALLOC_CSD      = JSB (REGISTER=1; REGISTER=2)
: 78 0801 1           : NOPRESERVE (3) NOTUSED (4,5,6,7,8,9,10,11),
: 79 0802 1     CSP_CALL    = JSB (REGISTER=2)
: 80 0803 1           : NOTUSED (3,4,5,6,7,8,9,10,11),
: 81 0804 1     DALLOC_CSD  = JSB (REGISTER=0)
: 82 0805 1           : NOPRESERVE (2,3) NOTUSED (4,5,6,7,8,9,10,11);
: 83 0806 1
: 84 0807 1
: 85 0808 1
: 86 0809 1 ! Run-time library and other routines external to the facility
: 87 0810 1
: 88 0811 1 EXTERNAL ROUTINE
: 89 0812 1     LIBSIG TO RET      : ADDRESSING_MODE (GENERAL), ! Convert a signal to a return
: 90 0813 1     EX$ALLOC_CSD    : ALLOC_CSD ADDRESSING_MODE (GENERAL), ! Allocate a CSD structure
: 91 0814 1     EX$CSP BRDCST  : CSP CALL ADDRESSING_MODE (GENERAL), ! Broadcast with CSP
: 92 0815 1     EX$DEALLOC_CSD : DALLOC_CSD ADDRESSING_MODE (GENERAL); ! Release CSD structure
: 93 0816 1
: 94 0817 1
: 95 0818 1
: 96 0819 1 ! Define the CODE psect so that the generated code has PIC and SHR attributes
: 97 0820 1
: 98 0821 1 PSECT CODE = Z$DISMOUNT (PIC, SHARE);
: 99 0822 1
: 100 0823 1

```

```

102 0824 1
103 0825 1 GLOBAL ROUTINE SEND_CLUSTER ( MSG_PTR, MSG_LEN, MSG_PRM ) =
104 0826 1
105 0827 1 :++
106 0828 1
107 0829 1 : FUNCTIONAL DESCRIPTION:
108 0830 1
109 0831 1 : Kernel mode routine to send a message to every remote node
110 0832 1 : on the cluster.
111 0833 1
112 0834 1 : CALLING SEQUENCE:
113 0835 1
114 0836 1 : SEND_CLUSTER ( ARG1, ARG2, ARG3 )
115 0837 1
116 0838 1 : INPUT:
117 0839 1
118 0840 1 : ARG1 : Address of message
119 0841 1 : ARG2 : Length of message
120 0842 1 : ARG3 : Message dependent parameter
121 0843 1
122 0844 1 : IMPLICIT INPUT:
123 0845 1
124 0846 1 : None.
125 0847 1
126 0848 1 : OUTPUT:
127 0849 1
128 0850 1 : None.
129 0851 1
130 0852 1 : IMPLICIT OUTPUT:
131 0853 1
132 0854 1 : None.
133 0855 1
134 0856 1 : SIDE EFFECTS:
135 0857 1
136 0858 1 : Messages will be sent to remote nodes.
137 0859 1
138 0860 1 : ROUTINE VALUE:
139 0861 1
140 0862 1 : Status from comm primitive.
141 0863 1
142 0864 1 : --
143 0865 1
144 0866 2 BEGIN : Start of SEND_CLUSTER
145 0867 2
146 0868 2 LOCAL
147 0869 2 : CSD : REF BBLOCK,
148 0870 2 : STATUS : LONG;
149 0871 2
150 0872 2 LITERAL
151 0873 2 : PRM_LEN = 4; : Length of message parameter
152 0874 2
153 0875 2
154 0876 2 : Trap anything weird, and turn it into a return
155 0877 2
156 0878 2 ENABLE
157 0879 2 SEND_CLUSTER_HANDLER;
158 0880 2

```

```

159 0881 2 |
160 0882 | Allocate a template CSD block for the transfer. Common fields in the
161 0883 | CSD are initialized by the allocate routine.
162 0884 |
163 0885 | STATUS = EXE$ALLOC_CSD (CSD$K_LENGTH + .MSG_LEN + PRM_LEN; CSD);
164 0886 | IF NOT .STATUS
165 0887 | THEN
166 0888 |     RETURN .STATUS;
167 0889 |
168 0890 |
169 0891 | Set the message dependent fields in the CSD
170 0892 |
171 0893 | CSD [CSD$W_CODE] = CSD$K_MOUNT; | Set the MOUNT client code
172 0894 | CSD [CSD$L_P1] = .MSG_PRM; | Set parameter into packet
173 0895 | CSD [CSD$L_SENOFF] = CSD [CSD$L_P2] - .CSD; | Store offset to the actual message
174 0896 | CSD [CSD$L_SENLEN] = .MSG_LEN; | Store size of message
175 0897 | CH$MOVE (.MSG_LEN, .MSG_PTR, CSD [CSD$L_P2]); | Move the message into the CSD
176 0898 | CSD [CSD$L_RECVOFF] = CSD [CSD$L_RECVLEN] = 0; | We do not want a reply
177 0899 | CSD [CSD$A_ASTADR] = 0; | No completion AST routine
178 0900 | CSD [CSD$L_CSID] = -1; | Set CSP broadcast function
179 0901 |
180 0902 |
181 0903 | Now send the message
182 0904 |
183 0905 | STATUS = EXE$CSP_BRDCST (.CSD); | Send the CSP message with
184 0906 | | the template CSD
185 0907 | EXE$DEALLOC_CSD (.CSD); | Deallocate template CSD
186 0908 |
187 0909 | RETURN .STATUS;
188 0910 | 1 END; | End of SEND_CLUSTER

```

```

.TITLE CLUSUTIL
.IDENT \V04-000\

.EXTRN LIB$SIG_TO_RET, EXE$ALLOC_CSD
.EXTRN EXE$CSP_BRDCST, EXE$DEALLOC_CSD

.PSECT Z$DISMOUNT,NOWRT, SHR, PIC,2

```

				00FC 00000	.ENTRY SEND_CLUSTER, Save R2,R3,R4,R5,R6,R7	
		6D	005D	CF DE 00002	MOVAL 28, 7(FP)	: 0825
51	08	AC	00000056	8F C1 00007	ADDL3 #86, MSG_LEN, R1	: 0866
			00000000G	00 16 00010	JSB EXE\$ALLOC_CSD	: 0885
		57		50 D0 00016	MOVL R0, STATUS	
		56		52 D0 00019	MOVL R2, R6	
		40		57 E9 0001C	BLBC STATUS, 18	: 0886
		0C	A6	06 B0 0001F	MOVW #6, 12(CSD)	: 0893
		52	A6	AC D0 00023	MOVL MSG_PRM, 82(CSD)	: 0894
52		56	0C	56 C3 00028	SUBL3 CSD, CSD, R2	: 0895
		16	A6	A2 9E 0002C	MOVAB 86(R2), 22(CSD)	
		12	A6	AC D0 00031	MOVL MSG_LEN, 18(CSD)	: 0896
			52	56 D0 00036	MOVL CSD, R2	: 0897
56	A2	04	BC	AC 28 00039	MOVCL3 MSG_LEN, @MSG_PTR, 86(R2)	
				1A A6 7C 00040	CLRQ 26(CSD)	: 0898
				22 A6 D4 00043	CLRL 34(CSD)	: 0899
		0E	A6	01 CE 00046	MNEGL #1, 14(CSD)	: 0900

CLUSUTIL
V04-000

H 1
15-Sep-1984 23:41:53 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:20:02 [DISMOU.SRC]CLUSUTIL.B32;1

Page 6
(3)

52			56	D0	0004A	MOVL	CSD, R2	:	0905	
	00000000G		00	16	0004D	JSB	EXE\$CSP_BRDCST	:		
57			50	D0	00053	MOVL	R0, STATUS	:		
50			56	D0	00056	MOVL	CSD, R0	:	0907	
	00000000G		00	16	00059	JSB	EXE\$DEALLOC_CSD	:		
50			57	D0	0005F	1\$:	MOVL	STATUS, R0	:	0909
				04	00062		RET	:	0910	
				0000	00063	2\$:	.WORD	Save nothing	:	0866
			7E	D4	00065		CLRL	-(SP)	:	
			5E	DD	00067		PUSHL	SP	:	
	7E	04	AC	7D	00069		MOVQ	4(AP), -(SP)	:	
0000V	CF		03	FB	0006D		CALLS	#3, SEND_CLUSTER_HANDLER	:	
			04		00072		RET	:		

; Routine Size: 115 bytes, Routine Base: Z\$DISMOUNT + 0000

; 189 0911 1


```

: 191 0912 1
: 192 0913 1 ROUTINE SEND_CLUSTER_HANDLER (SIG : REF BBLOCK, MECH : REF BBLOCK) =
: 193 0914 1
: 194 0915 1 :++
: 195 0916 1
: 196 0917 1 FUNCTIONAL DESCRIPTION:
: 197 0918 1
: 198 0919 1 This routine intercepts kernel mode signals, the signal is
: 199 0920 1 converted to a return.
: 200 0921 1
: 201 0922 1 INPUTS:
: 202 0923 1
: 203 0924 1 SIG : Signal argument list
: 204 0925 1 MECH : Mechanism argument list
: 205 0926 1
: 206 0927 1 IMPLICIT INPUTS:
: 207 0928 1
: 208 0929 1 None.
: 209 0930 1
: 210 0931 1 OUTPUTS:
: 211 0932 1
: 212 0933 1 None.
: 213 0934 1
: 214 0935 1 IMPLICIT OUTPUTS:
: 215 0936 1
: 216 0937 1 None.
: 217 0938 1
: 218 0939 1 ROUTINE VALUE:
: 219 0940 1
: 220 0941 1 None.
: 221 0942 1
: 222 0943 1 SIDE EFFECTS:
: 223 0944 1
: 224 0945 1 A return is made to the caller with the error status.
: 225 0946 1
: 226 0947 1 --
: 227 0948 1
: 228 0949 2 BEGIN
: 229 0950 2
: 230 0951 2 Convert the signal to a return
: 231 0952 2
: 232 0953 2 RETURN LIB$SIG_TO_RET (.SIG, .MECH);
: 233 0954 2
: 234 0955 1 END;
! End of SEND_CLUSTER_HANDLER

```

```

0000 0000 SEND_CLUSTER_HANDLER:
: 0913 .WORD Save nothing
: 0953 MOVQ SIG, -(SP)
: CALLS #2, LIB$SIG_TO_RET
: 0955 RET

```

; Routine Size: 14 bytes, Routine Base: Z\$DISMOUNT + 0073

CLUSUTIL
V04-000

¹
15-Sep-1984 23:41:53
14-Sep-1984 12:20:02

VAX-11 Bliss-32 V4.0-742
[DISMOU.SRC]CLUSUTIL.B32;1

Page 8
(4)

: 235 0956 1
: 236 0957 1

```

238 0958 1
239 0959 1 GLOBAL ROUTINE IN_CLUSTER =
240 0960 1
241 0961 1 :++
242 0962 1
243 0963 1 FUNCTIONAL DESCRIPTION:
244 0964 1
245 0965 1 This routine determines if the system is running in a
246 0966 1 cluster environment.
247 0967 1
248 0968 1 CALLING SEQUENCE:
249 0969 1
250 0970 1 IN_CLUSTER ()
251 0971 1
252 0972 1 INPUT:
253 0973 1
254 0974 1 None.
255 0975 1
256 0976 1 IMPLICIT INPUT:
257 0977 1
258 0978 1 None.
259 0979 1
260 0980 1 OUTPUT:
261 0981 1
262 0982 1 None.
263 0983 1
264 0984 1 IMPLICIT OUTPUT:
265 0985 1
266 0986 1 None.
267 0987 1
268 0988 1 SIDE EFFECTS:
269 0989 1
270 0990 1 None.
271 0991 1
272 0992 1 ROUTINE VALUE:
273 0993 1
274 0994 1 0 : If not running in a cluster
275 0995 1 1 : If running in a cluster
276 0996 1 Otherwise: System service error status
277 0997 1
278 0998 1 --
279 0999 1
280 1000 2 BEGIN ! Start of IN_CLUSTER
281 1001 2
282 1002 2 OWN
283 1003 2 CLUSTER_FLAG : LONG,
284 1004 2 SYI_NODE : VECTOR [4, LONG] ! GETSYI list
285 1005 2 INITIAL ((SYI$ CLUSTER_MEMBER*16 OR 4),
286 1006 2 CLUSTER_FLAG,
287 1007 2 0);
288 1008 2 LOCAL
289 1009 2 STATUS : LONG;
290 1010 2
291 1011 2
292 1012 2 Get system information to see if we are in a cluster.
293 1013 2
294 1014 2 IF NOT (STATUS = $GETSYIW (ITMLST=SYI_NODE))

```

```

: 295      1015      2      THEN RETURN .STATUS;
: 296      1016      2
: 297      1017      2
: 298      1018      2      Return value of cluster_flag
: 299      1019      2
: 300      1020      2      IF .CLUSTER_FLAG
: 301      1021      2      THEN
: 302      1022      2          RETURN 1
: 303      1023      2      ELSE
: 304      1024      2          RETURN 0;
: 305      1025      2
: 306      1026      1      END;

```

! End of IN_CLUSTER

```

.PSECT $OWNS,NOEXE,2
0000 CLUSTER_FLAG:
      .BLKB 4
10CF0004 00004 SYI_NODE:
      .LONG 282001412
00000000' 00008 .ADDRESS CLUSTER_FLAG
00000000 0000C .LONG 0
      .BLKB 4
.EXTRN SYSSGETSYIW
.PSECT Z$DISMOUNT,NOWRT, SHR, PIC,2

```

```

0000 0000 .ENTRY IN_CLUSTER, Save nothing
7E 7C 00002 CLRQ -(SP)
7E D4 00004 CLRL -(SP)
0000' CF 9F 00006 PUSHAB SYI_NODE
7E 7C 0000A CLRQ -(SP)
7E D4 0000C CLRL -(SP)
00000000G 00 07 FB 0000E CALLS #7, SYSSGETSYIW
0B 50 E9 00015 BLBC STATUS, 2$
04 0000' CF E9 00018 BLBC CLUSTER_FLAG, 1$
50 01 D0 0001D MOVL #1, R0
04 00020 RET
50 D4 00021 1$: CLRL R0
04 00023 2$: RET
: 0959
: 1014
:
:
: 1020
: 1024
:
: 1026

```

; Routine Size: 36 bytes. Routine Base: Z\$DISMOUNT + 0081

```

: 307      1027      1
: 308      1028      1
: 309      1029      1      END
: 310      1030      0      ELUDOM

```

! End of CLUSUTIL

PSECT SUMMARY

Name	Bytes	Attributes
ZSDISMOUNT	165	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
\$OWNS	20	NOVEC, WRT, RD, NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

file	----- Symbols -----		Pages Mapped	Processing Time
	Total	Loaded Percent		
\$_S255\$DUA28:[SYSLIB]LIB.L32;1	18619	12 0	1000	00:02.0

COMMAND QUALIFIERS

BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:CLUSUTIL/OBJ=OBJ\$:CLUSUTIL MSRCS :CLUSUTIL/UPDATE=(ENHS :CLUSUTIL)

```

: 311          1031  0
: Size:        165 code + 20 data bytes
: Run Time:    00:14.1
: Elapsed Time: 00:37.4
: Lines/CPU Min: 4399
: Lexemes/CPU-Min: 35620
: Memory Used: 114 pages
: Compilation Complete

```

