

[illegible]

• • • •

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL
LLLLLLLLLLL

SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSSS
SSSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```



```
1 0001 0 MODULE TRANQUEUE(XTITLE 'Translate queue name'
2 0002 0 IDENT = 'V04-000'
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1
7 0007 1 *****
8 0008 1 *
9 0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
10 0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
11 0011 1 * ALL RIGHTS RESERVED.
12 0012 1 *
13 0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
14 0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
15 0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
16 0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
17 0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
18 0018 1 * TRANSFERRED.
19 0019 1 *
20 0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
21 0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
22 0022 1 * CORPORATION.
23 0023 1 *
24 0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
25 0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
26 0026 1 *
27 0027 1 *
28 0028 1 *****
29 0029 1
30 0030 1
31 0031 1 ++
32 0032 1 FACILITY:
33 0033 1 CLI Utilities
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 This module contains a utility routine that performs the standard
37 0037 1 translation for queue names.
38 0038 1
39 0039 1 ENVIRONMENT:
40 0040 1 VAX/VMS user mode.
41 0041 1 --
42 0042 1
43 0043 1 AUTHOR: M. Jack, CREATION DATE: 11-Jan-1982
44 0044 1
45 0045 1 MODIFIED BY:
46 0046 1
47 0047 1 V03-001 MLJ0112 Martin L. Jack, 29-Apr-1983 0:34
48 0048 1 Extend to 31 character queue names.
49 0049 1
50 0050 1 **
```

TRANQUEUE
V04-000

Translate queue name

H 16
16-Sep-1984 00:30:55
14-Sep-1984 12:10:01

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]TRANQUEUE.B32;1

Page (2)

:	52	0051	1	LIBRARY 'SYSSLIBRARY:LIB';
:	53	0052	1	
:	54	0053	1	
:	55	0054	1	FORWARD ROUTINE
:	56	0055	1	TRAN_QUEUE;

! Translate queue name


```
58 0056 1 GLOBAL ROUTINE TRAN_QUEUE(INPUT_DESC,OUTPUT_STRING)=
59 0057 1
60 0058 1 ++
61 0059 1
62 0060 1 FUNCTIONAL DESCRIPTION:
63 0061 1 This routine performs the standard translation of queue names.
64 0062 1
65 0063 1 INPUT PARAMETERS:
66 0064 1
67 0065 1 INPUT_DESC - Address of a descriptor for the input string. The
68 0066 1 class must be unspecified or fixed.
69 0067 1
70 0068 1 OUTPUT_STRING - Pointer to a 16-byte area to receive the counted
71 0069 1 ASCII result, padded with binary zeros. Typically
72 0070 1 this is an area in the $SNSMB parameter.
73 0071 1
74 0072 1 IMPLICIT INPUTS:
75 0073 1 NONE
76 0074 1
77 0075 1 OUTPUT PARAMETERS:
78 0076 1 NONE
79 0077 1
80 0078 1 IMPLICIT OUTPUTS:
81 0079 1 NONE
82 0080 1
83 0081 1 ROUTINE VALUE:
84 0082 1 $$$_NORMAL Successful completion
85 0083 1 $$$_IVLOGNAM Input string longer than LOG$C_NAMLENGTH characters, or
86 0084 1 resulting string longer than 31 characters
87 0085 1
88 0086 1 SIDE EFFECTS:
89 0087 1 NONE
90 0088 1
91 0089 1 --
92 0090 1
93 0091 2 BEGIN
94 0092 2 MAP
95 0093 2 INPUT_DESC: REF BLOCK[ BYTE], ! Input string
96 0094 2 OUTPUT_STRING: REF VECTOR[,BYTE]; ! Output buffer
97 0095 2 LOCAL
98 0096 2 STATUS, ! Status return
99 0097 2 LOGNAM: VECTOR[2], ! Descriptor for input string
100 0098 2 RSLBUF: VECTOR[2], ! Descriptor for $TRNLOG result
101 0099 2 TRNLOG_BUFFER: VECTOR[LOG$C_NAMLENGTH,BYTE]; ! $TRNLOG result buffer
102 0100 2
103 0101 2
104 0102 2 ! Pick up input string descriptor.
105 0103 2
106 0104 2 IF .INPUT_DESC[DSC$W_LENGTH] GTRU LOG$C_NAMLENGTH THEN RETURN $$$_IVLOGNAM;
107 0105 2 LOGNAM[0] = .INPUT_DESC[DSC$W_LENGTH];
108 0106 2 LOGNAM[1] = .INPUT_DESC[DSC$A_POINTER];
109 0107 2 STATUS = $$$_NORMAL;
110 0108 2
111 0109 2
112 0110 2 ! Loop for 10 logical name translations. The last iteration is only for
113 0111 2 upcasing and blank-compressing the tenth translation, if any.
114 0112 2
```



```
115 0113 2 DECR I FROM 10 TO 0 DO
116 0114 BEGIN
117 0115 LOCAL
118 0116 OUTPUT_CURSOR: REF VECTOR[BYTE];
119 0117
120 0118
121 0119 ! Upcase and remove spaces, tabs, and nulls from the string.
122 0120
123 0121 OUTPUT_CURSOR = TRNLOG_BUFFER;
124 0122 INCR N FROM 0 TO .LOGNAM[0]-1 DO
125 0123 BEGIN
126 0124 LOCAL
127 0125 C: BYTE;
128 0126
129 0127 C = .VECTOR[.LOGNAM[1], .N, .BYTE];
130 0128 IF .C NEQ %C' ' AND .C NEQ %O'011' AND .C NEQ 0
131 0129 THEN
132 0130 BEGIN
133 0131 IF .C GEQU %C'a' AND .C LEQU %C'z' THEN C = .C - %C'a' + %C'A';
134 0132 OUTPUT_CURSOR[0] = .C;
135 0133 OUTPUT_CURSOR = .OUTPUT_CURSOR + 1;
136 0134 END;
137 0135 END;
138 0136
139 0137
140 0138 ! Build a descriptor for the upcased and blank-compressed result and
141 0139 remove a trailing colon, if any.
142 0140
143 0141 LOGNAM[1] = TRNLOG_BUFFER;
144 0142 LOGNAM[0] = .OUTPUT_CURSOR - .LOGNAM[1];
145 0143 IF .LOGNAM[0] NEQ 0
146 0144 THEN
147 0145 IF .OUTPUT_CURSOR[-1] EQL %C':' THEN LOGNAM[0] = .LOGNAM[0] - 1;
148 0146
149 0147
150 0148 ! If this is the eleventh iteration or there was no translation on the
151 0149 previous iteration, we are done.
152 0150
153 0151 IF .I EQL 0 OR .STATUS<0,16> EQL $$$_NOTRAN THEN EXITLOOP;
154 0152
155 0153
156 0154 ! Attempt translation.
157 0155
158 0156 RSLBUF[0] = LOG$C_NAMLENGTH;
159 0157 RSLBUF[1] = TRNLOG_BUFFER;
160 0158 STATUS = $TRNLOG(LOGNAM=LOGNAM, RSLLEN=RSLBUF, RSLBUF=RSLBUF);
161 0159 IF NOT .STATUS THEN RETURN .STATUS;
162 0160
163 0161
164 0162 ! Establish the result as the new input.
165 0163
166 0164 LOGNAM[0] = .RSLBUF[0];
167 0165 LOGNAM[1] = .RSLBUF[1];
168 0166 END;
169 0167
170 0168
171 0169 2 ! Ensure that the result will fit and return it as a counted ASCII string.
```



```
: 172      0170 2 | padded with binary zeros to a total of 32 bytes.
: 173      0171 2 |
: 174      0172 2 IF .RSLBUF[0] GTRU 31 THEN RETURN SS$_IVLOGNAM;
: 175      0173 2 OUTPUT_STRING[0] = .RSLBUF[0];
: 176      0174 2 CH$COPY(.RSLBUF[0], TRNLOG_BUFFER, 0, 31, OUTPUT_STRING[1]);
: 177      0175 2 SS$_NORMAL
: 178      0176 1 END;
```

.TITLE TRANQUEUE Translate queue name
.IDENT \V04-000\

.EXTRN SYS\$TRNLOG

.PSECT \$CODE\$,NOWRT,2

.ENTRY TRAN_QUEUE, Save R2,R3,R4,R5

MOVAB -80(SP), SP

MOVL INPUT_DESC, R0

CMPW (R0), #64

BLEQU 1\$

BRW 8\$

MOVZWL (R0), LOGNAM

MOVL 4(R0), LOGNAM+4

MOVL #1, STATUS

MOVL #10, I

MOVAB TRNLOG_BUFFER, OUTPUT_CURSOR

MNEGL #1, N

BRB 5\$

MOVB @LOGNAM+4[N], C

CMPB C, #32

BEQL 5\$

CMPB C, #9

BEQL 5\$

TSTB C

BEQL 5\$

CMPB C, #97

BLSSU 4\$

CMPB C, #122

BGTRU 4\$

SUBB2 #32, C

MOVB C, (OUTPUT_CURSOR)+

AOBLSS LOGNAM, N, -3\$

MOVAB TRNLOG_BUFFER, LOGNAM+4

SUBL3 LOGNAM+4, OUTPUT_CURSOR, LOGNAM

BEQL 6\$

CMPB -1(OUTPUT_CURSOR), #58

BNEQ 6\$

DECL LOGNAM

TSTL I

BEQL 7\$

CMPW STATUS, #1577

BEQL 7\$

MOVZBL #64, RSLBUF

MOVAB TRNLOG_BUFFER, RSLBUF+4

CLRQ -(SP)

CLRL -(SP)

```
003C 00000
      5E      B0 AE 9E 00002
      50      04 AC D0 00006
0040 8F      60 B1 0000A
      03 1B 0000F
      008F 31 00011
      60 3C 00014 1$:
      4C AE      04 A0 D0 00018
      AE      01 D0 0001D
      50      0A D0 00020
      54      6E 9E 00023 2$:
      53      01 CE 00026
      52      25 11 00029
      51      4C BE42 90 0002B 3$:
      20      51 91 00030
      09      1B 13 00033
      51 91 00035
      16 13 00038
      51 95 0003A
      12 13 0003C
      51 91 0003E
      09 1F 00042
      7A 8F      51 91 00044
      51 91 00048
      20 82 0004A
      51 90 0004D 4$:
      D6      48 AE F2 00050 5$:
      52      6E 9E 00055
      4C AE      4C AE C3 00059
      53      09 13 0005F
      3A      FF A3 91 00061
      03 12 00065
      48 AE      48 AE D7 00067
      54 D5 0006A 6$:
      2F 13 0006C
      50 B1 0006E
      0629 8F      28 13 00073
      40 AE      40 8F 9A 00075
      44 AE      6E 9E 0007A
      7E 7C 0007E
      7E D4 00080
```

```
: 0056
: 0104
: 0105
: 0106
: 0107
: 0113
: 0121
: 0127
: 0128
: 0131
: 0132
: 0122
: 0141
: 0142
: 0143
: 0145
: 0151
: 0156
: 0157
: 0158
```

TRANQUEUE
V04-000

Translate queue name

L 16
16-Sep-1984 00:30:55
14-Sep-1984 12:10:01

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]TRANQUEUE.B32;1

Page 6
(3)

		4C	AE	9F	00082	PUSHAB	RSLBUF			
		50	AE	9F	00085	PUSHAB	RSLBUF			
		5C	AE	9F	00088	PUSHAB	LOGNAM			
	00000000G	00	06	FB	0008B	CALLS	#6, SYS\$TRNLOG			
		27	50	E9	00092	BLBC	STATUS, 10\$		0159	
	48	AE	40	AE	7D	00095	MOVQ	RSLBUF, LOGNAM	0164	
		86	54	F4	0009A	SOBGEQ	I, 2\$		0113	
		1F	40	AE	D1	0009D	7\$:	CMP	RSLBUF, #31	0172
			06	1B	000A1	BLEQU	9\$			
		50	0154	8F	3C	000A3	8\$:	MOVZWL	#340, R0	
				04	000A8	RET				
		50	08	AC	D0	000A9	9\$:	MOVL	OUTPUT_STRING, R0	0173
		60	40	AE	90	000AD		MOVB	RSLBUF, (R0)	
1F	00	6E	40	AE	2C	000B1		MOVCS	RSLBUF, TRNLOG_BUFFER, #0, #31, 1(R0)	0174
			01	A0		000B7				
		50		01	D0	000B9		MOVL	#1, R0	0176
				04	000BC	10\$:	RET			

; Routine Size: 189 bytes, Routine Base: \$CODE\$ + 0000

TRANQUEUE
V04-000

Translate queue name

M 16
16-Sep-1984 00:30:55
14-Sep-1984 12:10:01

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]TRANQUEUE.B32;1

Page 7
(4)

: 180
: 181
0177 1 END
0178 0 ELUDOM

PSECT SUMMARY

:
: Name Bytes Attributes
: \$CODE\$ 189 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

:
: File ----- Symbols ----- Pages Processing
: Total Loaded Percent Mapped Time
: _\$255\$DUA28:[SYSLIB]LIB.L32;1 18619 9 0 1000 00:01.9

COMMAND QUALIFIERS

:
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:TRANQUEUE/OBJ=OBJ\$:TRANQUEUE MSRC\$:TRANQUEUE/UPDATE=(ENH\$:TRANQUEUE)

: Size: 189 code + 0 data bytes
: Run Time: 00:06.5
: Elapsed Time: 00:24.7
: Lines/CPU Min: 1638
: Lexemes/CPU-Min: 13665
: Memory Used: 96 pages
: Compilation Complete

0059

AH-BT13A-SE
VAX/VMS V4.

**DIGITAL
CONFIDE**

EQUIPMENT
TIAL AND

CORPORATION
PROPRIETARY