





```

1 0001 0 MODULE CHECKPROT(%TITLE 'Queue protection checks'
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 | Job controller.
34 0034 1 |
35 0035 1 | ABSTRACT:
36 0036 1 | This module contains the routines that interface to the $CHKPRO service
37 0037 1 | for queue protection checking.
38 0038 1 |
39 0039 1 | ENVIRONMENT:
40 0040 1 | VAX/VMS user and kernel mode.
41 0041 1 | --
42 0042 1 |
43 0043 1 | AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
44 0044 1 |
45 0045 1 | MODIFIED BY:
46 0046 1 |
47 0047 1 | V03-001 MLJ0114 Martin L. Jack, 23-Jun-1983 4:57
48 0048 1 | Changes for job controller baselevel.
49 0049 1 |
50 0050 1 |**

```

CHECKPROT  
V04-000

Queue protection checks

N 13  
15-Sep-1984 23:59:44  
14-Sep-1984 12:36:58

VAX-11 Bliss-32 V4.0-742  
[JOBCTL.SRC]CHECKPROT.B32;1

```
.. 52      0051 1 REQUIRE 'SRCS:JOBCTLDEF';  
.. 53      1092 1  
.. 54      1093 1  
.. 55      1094 1 FORWARD ROUTINE  
.. 56      1095 1      CHKPRO,  
.. 57      1096 1      CHECK_R_ACCESS,  
.. 58      1097 1      CHECK_W_ACCESS,  
.. 59      1098 1      CHECK_E_ACCESS,  
.. 60      1099 1      CHECK_D_ACCESS;
```

```
62 1100 1 GLOBAL ROUTINE CHKPRO(ACCESS,OWNER,PROTECTION)=
63 1101 1
64 1102 1 !++
65 1103 1
66 1104 1 FUNCTIONAL DESCRIPTION:
67 1105 1 This routine executes a call to the $CHKPRO service.
68 1106 1
69 1107 1 INPUT PARAMETERS:
70 1108 1 ACCESS - Desired access.
71 1109 1 OWNER - Object ownership.
72 1110 1 PROTECTION - Object protection.
73 1111 1
74 1112 1 IMPLICIT INPUTS:
75 1113 1 MBX - Pointer to buffered mailbox message.
76 1114 1
77 1115 1 OUTPUT PARAMETERS:
78 1116 1 NONE
79 1117 1
80 1118 1 IMPLICIT OUTPUTS:
81 1119 1 NONE
82 1120 1
83 1121 1 ROUTINE VALUE:
84 1122 1 True if access is granted, false if denied.
85 1123 1
86 1124 1 SIDE EFFECTS:
87 1125 1 NONE
88 1126 1
89 1127 1 --
90 1128 1
91 1129 2 BEGIN
92 1130 2 LOCAL
93 1131 2 ITMLST: BBLOCK[64], ! $CHKPRO item list
94 1132 2 RIGHTS: VECTOR[2]; ! Rights vector
95 1133 2
96 1134 2
97 1135 2 ! OPER privilege implicitly grants all types of access.
98 1136 2
99 1137 2 IF .BBLOCK[MBX[ACMSQ_PRVMSK], PRV$V_OPER]
100 1138 2 THEN
101 1139 2 RETURN TRUE;
102 1140 2
103 1141 2
104 1142 2 ! Construct the item list and call $CHKPRO to check the access.
105 1143 2
106 1144 2 RIGHTS[0] = .MBX[ACMSL_UIC];
107 1145 2 RIGHTS[1] = 0;
108 1146 2 ITMLST[0,0,16,0] = 4; ! Desired access
109 1147 2 ITMLST[2,0,16,0] = CHPS ACCESS;
110 1148 2 ITMLST[4,0,32,0] = ACCESS;
111 1149 2 ITMLST[8,0,32,0] = 0;
112 1150 2 ITMLST[12,0,16,0] = 8; ! Requestor's privileges
113 1151 2 ITMLST[14,0,16,0] = CHPS PRIV;
114 1152 2 ITMLST[16,0,32,0] = MBX[ACMSQ_PRVMSK];
115 1153 2 ITMLST[20,0,32,0] = 0;
116 1154 2 ITMLST[24,0,16,0] = 8; ! Requestor's UIC
117 1155 2 ITMLST[26,0,16,0] = CHPS RIGHTS;
118 1156 2 ITMLST[28,0,32,0] = RIGHTS;
```

```

: 119      1157  2 ITMLST[32,0,32,0] = 0;
: 120      1158  2 ITMLST[36,0,16,0] = 4;
: 121      1159  2 ITMLST[38,0,16,0] = CHPS OWNER;
: 122      1160  2 ITMLST[40,0,32,0] = OWNER;
: 123      1161  2 ITMLST[44,0,32,0] = 0;
: 124      1162  2 ITMLST[48,0,16,0] = 4;
: 125      1163  2 ITMLST[50,0,16,0] = CHPS PROT;
: 126      1164  2 ITMLST[52,0,32,0] = PROTECTION;
: 127      1165  2 ITMLST[56,0,32,0] = 0;
: 128      1166  2 ITMLST[60,0,32,0] = 0;
: 129      1167  2
: 130      1168  2
: 131      1169  3 $CHKPRO(ITMLST=ITMLST)
: 132      1170  1 END;

```

```

! Object's UIC
! Object's protection
! Trailing zero

```

```

.TITLE CHECKPROT Queue protection checks
.IDENT \V04-000\
.PSECT COMMON,NOEXE, OVR,2

```

```

0000 DIAG_STORAGE BASE:
      .BLKB 0
0000 DIAG_TRACE:
      .BLKB 96
00060 DIAG_COUNT:
      .BLKB 96
000C0 DIAG_FLAGS:
      .BLKB 4
000C4 WORK_AREA:
      .BLKB 44
000F0 SNDJBC_COUNT:
      .BLKB 132
00174 GETQUI_COUNT:
      .BLKB 40
0019C SNDACC_COUNT:
      .BLKB 28
001B8 SNDSMB_COUNT:
      .BLKB 72
00200 DIAG_STORAGE END:
      .BLKB 0
00200 FLAGS: .BLKB 4
00204 IMAGE_DUMP_STSFLG:
      .BLKB 4
00208 THIS_SYSID:
      .BLKB 6
0020E .BLKB 2
00210 CUR_TIME:
      .BLKB 8
00218 HOURLY_TIME:
      .BLKB 8
00220 HOURLY_PARAMS:
      .BLKB 20
00234 SYMBIONT_COUNT:
      .BLKB 4
00238 QUEUE_REFERENCE_COUNT:
      .BLKB 4

```

0023C MBX\_MESSAGE\_COUNT:  
          .BLRB 4  
00240 MBX: .BLKB 4  
00244 MBX\_END: .BLKB 4  
00248 MEMORY\_FREE\_QUEUES:  
          .BLRB 40  
00270 NONAST\_WORK\_QUEUE:  
          .BLRB 8  
00278 BCB\_FREE\_LIST:  
          .BLKB 4  
0027C BCB\_ACTIVE\_LIST:  
          .B[KB 4  
00280 GQL\_FREE\_LIST:  
          .BLKB 4  
00284 GQL\_ACTIVE\_LIST:  
          .B[KB 4  
00288 OPEN\_GETQUI\_LIST:  
          .BLRB 4  
0028C PROCESS\_DATA\_LIST:  
          .BLKB 4  
00290 SYMBIONT\_CONTROL:  
          .BLKB 4  
00294 SPARE\_AREA:  
          .BLKB 12  
002A0 REMOTE\_REQUEST\_LKSB:  
          .BLKB 8  
002A8 QUEUE\_FILE\_LKSB:  
          .B[KB 8  
002B0 QUEUE\_LOCK\_LKSB:  
          .B[KB 8  
002B8 RSP: .BLKB 8  
002C0 JBC\_PRIORITY:  
          .BLKB 4  
002C4 JBC\_PRIVILEGES:  
          .BLKB 8  
002CC JBC\_QUOTAS:  
          .BLKB 66  
0030E .BLKB 2  
00310 JBC\_UIC: .BLKB 4  
00314 QUEUE\_FAB:  
          .BLKB 80  
00364 QUEUE\_RAB:  
          .BLKB 68  
003A8 QUEUE\_NAM:  
          .BLKB 96  
00408 QUEUE\_XAB:  
          .BLKB 88  
00460 QUEUE\_RSA:  
          .BLKB 255  
0055F .BLKB 1  
00560 QUEUE\_ALQ:  
          .BLKB 4  
00564 QUEUE\_MBF:  
          .BLKB 1  
00565 .BLKB 3  
00568 ACCOUNTING\_FABS:  
          .B[KB 8

00570 ACCOUNTING\_RABS:  
          .BCKB 8  
00578 ACCOUNT\_FAB\_A:  
          .BLRB 80  
005C8 ACCOUNT\_RAB\_A:  
          .BLRB 68  
0060C ACCOUNT\_NAM\_A:  
          .BLRB 96  
0066C ACCOUNT\_RSA\_A:  
          .BLRB 255  
0076B          .BCKB 1  
0076C ACCOUNT\_FAB\_B:  
          .BLRB 80  
007BC ACCOUNT\_RAB\_B:  
          .BLRB 68  
00800 ACCOUNT\_NAM\_B:  
          .BLRB 96  
00860 ACCOUNT\_RSA\_B:  
          .BLRB 255  
0095F          .BCKB 1  
00960 DIAG\_FAB:  
          .BCKB 80  
009B0 DIAG\_RAB:  
          .BCKB 68  
009F4 MBX\_CHAN:  
          .BCKB 4  
009F8 MBX\_IOSB:  
          .BCKB 8  
00A00 MBX\_BUFFER:  
          .BCKB 1024  
00E00 VALUE\_STORAGE\_BASE:  
          .BCKB 0  
00E00 ITEM\_PRESENT:  
          .BCKB 32  
00E20 VALUE\_GETQUI\_BASE:  
          .BCKB 0  
00E20 VALUE\_ACCOUNTING\_MESSAGE:  
          .BCKB 8  
00E26 VALUE\_ACCOUNTING\_TYPES:  
          .BCKB 4  
00E2A VALUE\_AFTER\_TIME:  
          .BLRB 8  
00E32 VALUE\_ALIGNMENT\_PAGES:  
          .BCKB 1  
00E33 VALUE\_BASE\_PRIORITY:  
          .BCKB 1  
00E34 VALUE\_BATCH\_INPUT:  
          .BLRB 6  
00E3A VALUE\_BATCH\_OUTPUT:  
          .BLRB 10  
00E44 VALUE\_BUFFER\_COUNT:  
          .BCKB 1  
00E45 VALUE\_CHARACTERISTIC\_NAME:  
          .BCKB 6  
00E4B VALUE\_CHARACTERISTIC\_NUMBER:  
          .BCKB 1  
00E4C VALUE\_CHARACTERISTICS:



00E5C VALUE\_CHECKPOINT\_DATA: .BLKB 16  
00E62 VALUE\_CLI: .BLKB 8  
00E68 VALUE\_CPU\_DEFAULT: .BLKB 6  
00E6C VALUE\_CPU\_LIMIT: .BLKB 4  
00E70 VALUE\_DESTINATION\_QUEUE: .BLKB 4  
00E78 VALUE\_DEVICE\_NAME: .BLKB 8  
00E7E VALUE\_ENTRY\_NUMBER: .BLKB 6  
00E82 VALUE\_ENTRY\_NUMBER\_OUTPUT: .BLKB 4  
00E8C VALUE\_EXTEND\_QUANTITY: .BLKB 10  
00E8E VALUE\_FILE\_COPIES: .BLKB 2  
00E8F VALUE\_FILE\_IDENTIFICATION: .BCKB 1  
00EB3 VALUE\_FILE\_SETUP\_MODULES: .BCKB 36  
00EB9 VALUE\_FILE\_SPECIFICATION: .BCKB 8  
00EBF VALUE\_FIRST\_PAGE: .BCKB 6  
00EC3 VALUE\_FORM\_DESCRIPTION: .BLKB 4  
00EC9 VALUE\_FORM\_LENGTH: .BCKB 6  
00ECA VALUE\_FORM\_MARGIN\_BOTTOM: .BCKB 1  
00ECB VALUE\_FORM\_MARGIN\_LEFT: .BCKB 2  
00ECD VALUE\_FORM\_MARGIN\_RIGHT: .BCKB 2  
00ECF VALUE\_FORM\_MARGIN\_TOP: .BCKB 1  
00ED0 VALUE\_FORM\_NAME: .BCKB 6  
00ED6 VALUE\_FORM\_NUMBER: .BCKB 4  
00EDA VALUE\_FORM: .BLKB 8  
00EE2 VALUE\_FORM\_SETUP\_MODULES: .BCKB 8  
00EE8 VALUE\_FORM\_STOCK: .BCKB 6  
00EEE VALUE\_FORM\_WIDTH: .BCKB 2  
00EF0 VALUE\_GENERIC\_TARGET: .BLKB 996  
012D4 VALUE\_JOB\_COPIES: .BLKB 1

012D5 VALUE\_JOB\_LIMIT:  
          .BLKB 1  
012D6 VALUE\_JOB\_NAME:  
          .BLKB 6  
012DC VALUE\_JOB\_RESET\_MODULES:  
          .BLKB 6  
012E2 VALUE\_JOB\_SIZE\_MAXIMUM:  
          .BLKB 4  
012E6 VALUE\_JOB\_SIZE\_MINIMUM:  
          .BLKB 4  
012EA VALUE\_JOB\_STATUS\_OUTPUT:  
          .BLKB 10  
012F4 VALUE\_LAST\_PAGE:  
          .BLKB 4  
012F8 VALUE\_LIBRARY\_SPECIFICATION:  
          .BLKB 6  
012FE VALUE\_LOG\_QUEUE:  
          .BLKB 8  
01306 VALUE\_LOG\_SPECIFICATION:  
          .BLKB 6  
0130C VALUE\_NOTE:  
          .BLKB 6  
01312 VALUE\_OPERATOR\_REQUEST:  
          .BLKB 6  
01318 VALUE\_OWNER\_UIC:  
          .BLKB 4  
0131C VALUE\_PAGE\_SETUP\_MODULES:  
          .BLKB 8  
01322 VALUE\_PARAMETER\_1:  
          .BLKB 6  
01328 VALUE\_PARAMETER\_2:  
          .BLKB 6  
0132E VALUE\_PARAMETER\_3:  
          .BLKB 6  
01334 VALUE\_PARAMETER\_4:  
          .BLKB 6  
0133A VALUE\_PARAMETER\_5:  
          .BLKB 6  
01340 VALUE\_PARAMETER\_6:  
          .BLKB 6  
01346 VALUE\_PARAMETER\_7:  
          .BLKB 6  
0134C VALUE\_PARAMETER\_8:  
          .BLKB 6  
01352 VALUE\_PRIORITY:  
          .BLKB 1  
01353 VALUE\_PROCESSOR:  
          .BLKB 6  
01359 VALUE\_PROTECTION:  
          .BLKB 4  
0135D VALUE\_QUEUE:  
          .BLKB 6  
01363 VALUE\_QUEUE\_FILE\_SPECIFICATION:  
          .BLKB 8  
01369 VALUE\_RELATIVE\_PAGE:  
          .BLKB 4  
0136D VALUE\_RESERVED\_INPUT\_1:

```

0136E VALUE_RESERVED_INPUT_2:
      .BLKB 1
01370 VALUE_RESERVED_INPUT_3:
      .BLKB 2
01374 VALUE_RESERVED_INPUT_4:
      .BLKB 4
0137A VALUE_RESERVED_OUTPUT_1:
      .BLKB 6
01384 VALUE_RESERVED_OUTPUT_2:
      .BLKB 10
0138E VALUE_SEARCH_STRING:
      .BLKB 10
01394 VALUE_SC$NODE_NAME:
      .BLKB 6
0139A VALUE_WSDEFAULT:
      .BLKB 6
0139C VALUE_W$EXTENT:
      .BLKB 2
0139E VALUE_W$QUOTA:
      .BLKB 2
013A0 VALUE_STORAGE_END:
      .BLKB 0

```

```

JBC$_CLOSEOUT= 266328
JBC$_NOCMKRNL= 272388
JBC$_NOOPER= 272532
JBC$_NOSYSNAM= 272404
JBC$_OPENIN= 266392
JBC$_OPENOUT= 266400
JBC$_READERR= 266416
JBC$_WRITEERR= 266448
      .EXTRN SYSSCHKPRO

```

.PSECT CODE,NOWRT,2

```

04      06      5E      B8      AE 00000000      0000 00000      .ENTRY CHKPRO, Save nothing      : 1100
      50      00000000      EF D0 00006      MOVAB -72(SP), SP      : 1137
      AO      02      E1 0000D      MOVL MBX, R0      :
      50      01      D0 00012      BBC #2, 6(R0), 1$      : 1139
      6E      0C      A0 D0 00016      MOVL #1, R0      :
      04      AE D4 0001A      RET      : 1144
08      AE 00010004      8F D0 0001D      MOVL 12(R0), RIGHTS      : 1145
OC      AE      04      AC 9E 00025      CLRL RIGHTS+4      : 1146
      10      AE D4 0002A      MOVL #65540, ITMLST      : 1148
14      AE 00030008      8F D0 0002D      MOVAB ACCESS, ITMLST+4      : 1149
18      AE      04      A0 9E 00035      CLRL ITMLST+8      : 1150
      1C      AE D4 0003A      MOVL #196616, ITMLST+12      : 1152
20      AE 00060008      8F D0 0003D      MOVAB 4(R0), ITMLST+16      : 1153
24      AE      04      6E 9E 00045      CLRL ITMLST+20      : 1154
      28      AE D4 00049      MOVL #393224, ITMLST+24      : 1156
2C      AE 000C0004      8F D0 0004C      MOVAB RIGHTS, ITMLST+28      : 1157
30      AE      08      AE D4 00049      CLRL ITMLST+32      : 1158
      34      AE D4 00054      MOVL #786436, ITMLST+36      : 1160
38      AE 000D0004      8F D0 0005C      MOVAB OWNER, ITMLST+40      : 1161
      8F      D0 0005C      CLRL ITMLST+44      : 1162
      MOVL #851972, ITMLST+48      :

```

CHECKPROT  
V04-000

Queue protection checks

I 14  
15-Sep-1984 23:59:44 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:36:58 [JOBCTL.SRC]CHECKPROT.B32;1

Page 10  
(3)

3C	AE	0C	AC	9E	00064
		40	AE	7C	00069
		08	AE	9F	0006C
00000000G	00	01	FB	0006F	
			04	00076	

MOVAB	PROTECTION, ITMLST+52
CLRQ	ITMLST+56
PUSHAB	ITMLST
CALLS	#1, SYS\$CHKPRO
RET	

: 1164  
: 1165  
: 1169  
: 1170

: Routine Size: 119 bytes, Routine Base: CODE + 0000

CC  
VC

```
134 1171 1 GLOBAL ROUTINE CHECK_R_ACCESS(SMQ,SJH)=
135 1172 1
136 1173 1 ++
137 1174 1
138 1175 1 FUNCTIONAL DESCRIPTION:
139 1176 1 This routine checks whether the requestor has R access to the specified
140 1177 1 job in the specified queue. OPER privilege and E access to the queue
141 1178 1 implicitly grant R access.
142 1179 1
143 1180 1 INPUT PARAMETERS:
144 1181 1 SMQ - Pointer to SMQ.
145 1182 1 SJH - Pointer to SJH.
146 1183 1
147 1184 1 IMPLICIT INPUTS:
148 1185 1 MBX - Pointer to buffered mailbox message.
149 1186 1
150 1187 1 OUTPUT PARAMETERS:
151 1188 1 NONE
152 1189 1
153 1190 1 IMPLICIT OUTPUTS:
154 1191 1 NONE
155 1192 1
156 1193 1 ROUTINE VALUE:
157 1194 1 True if access is granted, false if denied.
158 1195 1
159 1196 1 SIDE EFFECTS:
160 1197 1 NONE
161 1198 1
162 1199 1 --
163 1200 1
164 1201 2 BEGIN
165 1202 2 MAP
166 1203 2 SMQ: REF BBLOCK, ! Pointer to SMQ
167 1204 2 SJH: REF BBLOCK; ! Pointer to SJH
168 1205 2
169 1206 2
170 1207 2 ! Test for R access to the job.
171 1208 2
172 1209 2 IF CHKPRO(%B'0001', .SJH[SJH$L_UIC], .SMQ[SMQ$W_PROTECTION])
173 1210 2 THEN
174 1211 2 RETURN TRUE;
175 1212 2
176 1213 2
177 1214 2 ! Test for E access to the queue.
178 1215 2
179 1216 2 IF CHKPRO(%B'0100', .SMQ[SMQ$L_OWNER_UIC], .SMQ[SMQ$W_PROTECTION])
180 1217 2 THEN
181 1218 2 RETURN TRUE;
182 1219 2
183 1220 2
184 1221 2 FALSE
185 1222 1 END;
```

CHECKPROT  
V04-000

Queue protection checks

K 14  
15-Sep-1984 23:59:44  
14-Sep-1984 12:36:58

VAX-11 Bliss-32 V4.0-742  
[JOBCTL.SRC]CHECKPROT.B32;1

Page 12  
(4)

			0004	00000	.ENTRY	CHECK R_ACCESS, Save R2	
52	04	AC	DO	00002	MOVL	SMQ, R2	1171
7E	0104	C2	3C	00006	MOVZWL	260(R2), -(SP)	1209
50	08	AC	DO	00008	MOVL	SJH, R0	
	0144	C0	DD	0000F	PUSHL	324(R0)	
		01	DD	00013	PUSHL	#1	
FF6F	CF	03	FB	00015	CALLS	#3, CHKPRO	
	13	50	E8	0001A	BLBS	R0, 1\$	
	7E	0104	C2	3C	0001D	MOVZWL	260(R2), -(SP)
		00D0	C2	DD	00022	PUSHL	208(R2)
			04	DD	00026	PUSHL	#4
FF5C	CF	03	FB	00028	CALLS	#3, CHKPRO	
	04	50	E9	0002D	BLBC	R0, 2\$	
	50	01	DO	00030	MOVL	#1, R0	1218
			04	00033	RET		
		50	D4	00034	CLRL	R0	1222
			04	00036	RET		

; Routine Size: 55 bytes, Routine Base: CODE + 0077

CC  
VC

```

187 1223 1 GLOBAL ROUTINE CHECK_W_ACCESS(SMQ)=
188 1224 1
189 1225 1  ++
190 1226 1
191 1227 1  FUNCTIONAL DESCRIPTION:
192 1228 1      This routine checks whether the requestor has W access to the specified
193 1229 1      queue. OPER privilege and E access to the queue implicitly grant
194 1230 1      W access.
195 1231 1
196 1232 1  INPUT PARAMETERS:
197 1233 1      SMQ          - Pointer to SMQ.
198 1234 1
199 1235 1  IMPLICIT INPUTS:
200 1236 1      MBX          - Pointer to buffered mailbox message.
201 1237 1
202 1238 1  OUTPUT PARAMETERS:
203 1239 1      NONE
204 1240 1
205 1241 1  IMPLICIT OUTPUTS:
206 1242 1      NONE
207 1243 1
208 1244 1  ROUTINE VALUE:
209 1245 1      True if access is granted, false if denied.
210 1246 1
211 1247 1  SIDE EFFECTS:
212 1248 1      NONE
213 1249 1
214 1250 1  --
215 1251 1
216 1252 2 BEGIN
217 1253 2 MAP
218 1254 2      SMQ:          REF BBLOCK;      ! Pointer to SMQ
219 1255 2
220 1256 2
221 1257 2 ! Test for W access to the queue.
222 1258 2
223 1259 2 IF CHKPRO(%B'0010', .SMQ[SMQ$L_OWNER_UIC], .SMQ[SMQ$W_PROTECTION])
224 1260 2 THEN
225 1261 2     RETURN TRUE;
226 1262 2
227 1263 2
228 1264 2 ! Test for E access to the queue.
229 1265 2
230 1266 2 IF CHKPRO(%B'0100', .SMQ[SMQ$L_OWNER_UIC], .SMQ[SMQ$W_PROTECTION])
231 1267 2 THEN
232 1268 2     RETURN TRUE;
233 1269 2
234 1270 2
235 1271 2 FALSE
236 1272 1 END;

```

```

52      04      0004 0000      .ENTRY CHECK_W_ACCESS, Save R2      : 1223
          AC      DO 00002      MOVL   SMQ, R2      : 1259

```

CHECKPROT  
V04-000

Queue protection checks

M 14  
15-Sep-1984 23:59:44 VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 12:36:58 [JOBCTL.SRC]CHECKPROT.B32;1

Page 14  
(5)

	7E	0104	C2	3C	00006	MOVZWL	260(R2), -(SP)
		00D0	C2	DD	0000B	PUSHL	208(R2)
FF3C	CF		02	DD	0000F	PUSHL	#2
	13		03	FB	00011	CALLS	#3, CHKPRO
	7E	0104	50	E8	00016	BLBS	RO, 1\$
		00D0	C2	3C	00019	MOVZWL	260(R2), -(SP)
			C2	DD	0001E	PUSHL	208(R2)
			04	DD	00022	PUSHL	#4
FF29	CF		03	FB	00024	CALLS	#3, CHKPRO
	04		50	E9	00029	BLBC	RO, 2\$
	50		01	D0	0002C	MOVL	#1, RO
				04	0002F	RET	
			50	D4	00030	CLRL	RO
			04	00032	RET		

1266

1268

1272

; Routine Size: 51 bytes, Routine Base: CODE + 00AE



```

: 238 1273 1 GLOBAL ROUTINE CHECK_E_ACCESS(SMQ)=
: 239 1274 1
: 240 1275 1 |++
: 241 1276 1
: 242 1277 1 FUNCTIONAL DESCRIPTION:
: 243 1278 1 This routine checks whether the requestor has E access to the specified
: 244 1279 1 queue. OPER privilege implicitly grants E access.
: 245 1280 1
: 246 1281 1 INPUT PARAMETERS:
: 247 1282 1 SMQ - Pointer to SMQ.
: 248 1283 1
: 249 1284 1 IMPLICIT INPUTS:
: 250 1285 1 MBX - Pointer to buffered mailbox message.
: 251 1286 1
: 252 1287 1 OUTPUT PARAMETERS:
: 253 1288 1 NONE
: 254 1289 1
: 255 1290 1 IMPLICIT OUTPUTS:
: 256 1291 1 NONE
: 257 1292 1
: 258 1293 1 ROUTINE VALUE:
: 259 1294 1 True if access is granted, false if denied.
: 260 1295 1
: 261 1296 1 SIDE EFFECTS:
: 262 1297 1 NONE
: 263 1298 1
: 264 1299 1 --
: 265 1300 1
: 266 1301 2 BEGIN
: 267 1302 2 MAP
: 268 1303 2 SMQ: REF BBLOCK; ! Pointer to SMQ
: 269 1304 2
: 270 1305 2
: 271 1306 2 ! Test for E access to the queue.
: 272 1307 2
: 273 1308 2 IF CHKPRO(%B'0100', .SMQ[SMQ$SL_OWNER_UIC], .SMQ[SMQ$W_PROTECTION])
: 274 1309 2 THEN
: 275 1310 2 RETURN TRUE;
: 276 1311 2
: 277 1312 2
: 278 1313 2 FALSE
: 279 1314 1 END;

```

```

: 1273
: 1308
: 1310
: 1314

```

50	04	AC	D0	00002	.ENTRY	CHECK_E_ACCESS, Save nothing	:	1273
7E	0104	CO	3C	00006	MOVL	SMQ, R0	:	1308
	00D0	CO	DD	0000B	MOVZWL	260(R0), -(SP)	:	
		04	DD	0000F	PUSHL	208(R0)	:	
FF09	CF	03	FB	00011	PUSHL	#4	:	
	04	50	E9	00016	CALLS	#3, CHKPRO	:	
	50	01	D0	00019	BLBC	R0, 1\$	:	1310
			C4	0001C	MOVL	#1, R0	:	
		50	D4	0001D	RET		:	1314
				1\$:	CLRL	R0	:	

CHECKPROT  
V04-000

Queue protection checks

B 15  
15-Sep-1984 23:59:44  
14-Sep-1984 12:36:58

VAX-11 Bliss-32 V4.0-742  
[JOBCTL.SRC]CHECKPROT.B32;1

Page 16  
(6)

COI  
VOI

04 0001F

RET

;

; Routine Size: 32 bytes. Routine Base: CODE + 00E1

```
281 1315 1 GLOBAL ROUTINE CHECK_D_ACCESS(SMQ,SJH)=
282 1316 1
283 1317 1 |++
284 1318 1 |
285 1319 1 | FUNCTIONAL DESCRIPTION:
286 1320 1 |   This routine checks whether the requestor has D access to the specified
287 1321 1 |   job in the specified queue. UPER privilege and E access to the queue
288 1322 1 |   implicitly grant D access.
289 1323 1 |
290 1324 1 | INPUT PARAMETERS:
291 1325 1 |   SMQ           - Pointer to SMQ.
292 1326 1 |   SJH           - Pointer to SJH.
293 1327 1 |
294 1328 1 | IMPLICIT INPUTS:
295 1329 1 |   MBX           - Pointer to buffered mailbox message.
296 1330 1 |
297 1331 1 | OUTPUT PARAMETERS:
298 1332 1 |   NONE
299 1333 1 |
300 1334 1 | IMPLICIT OUTPUTS:
301 1335 1 |   NONE
302 1336 1 |
303 1337 1 | ROUTINE VALUE:
304 1338 1 |   True if access is granted, false if denied.
305 1339 1 |
306 1340 1 | SIDE EFFECTS:
307 1341 1 |   NONE
308 1342 1 |
309 1343 1 | --
310 1344 1 |
311 1345 2 BEGIN
312 1346 2 MAP
313 1347 2     SMQ:           REF BBLOCK,      ! Pointer to SMQ
314 1348 2     SJH:           REF BBLOCK;     ! Pointer to SJH
315 1349 2
316 1350 2
317 1351 2 ! Test for D access to the job.
318 1352 2 !
319 1353 2 IF CHKPRO(%B'1000', .SJH[SJH$L_UIC], .SMQ[SMQ$W_PROTECTION])
320 1354 2 THEN
321 1355 2     RETURN TRUE;
322 1356 2
323 1357 2
324 1358 2 ! Test for E access to the queue.
325 1359 2 !
326 1360 2 IF CHKPRO(%B'0100', .SMQ[SMQ$L_OWNER_UIC], .SMQ[SMQ$W_PROTECTION])
327 1361 2 THEN
328 1362 2     RETURN TRUE;
329 1363 2
330 1364 2
331 1365 2 FALSE
332 1366 1 END;
```

CHECKPROT  
V04-000

Queue protection checks

D 15  
15-Sep-1984 23:59:44  
14-Sep-1984 12:36:58

VAX-11 Bliss-32 V4.0-742  
[JOBCTL.SRC]CHECKPROT.B32;1

Page 18  
(7)

CC  
VC

			0004 00000	.ENTRY	CHECK_D_ACCESS, Save R2	:	1315
	52	04	AC D0 00002	MOVL	SMQ, R2	:	1353
	7E	0104	C2 3C 00006	MOVZWL	260(R2), -(SP)	:	
	50	08	AC D0 0000B	MOVL	SJH, R0	:	
		0144	C0 DD 0000F	PUSHL	324(R0)	:	
			08 DD 00013	PUSHL	#8	:	
FEE5	CF		03 FB 00015	CALLS	#3, CHKPRO	:	
	13		50 E8 0001A	BLBS	R0, 1\$	:	
	7E	0104	C2 3C 0001D	MOVZWL	260(R2), -(SP)	:	1360
		00D0	C2 DD 00022	PUSHL	208(R2)	:	
			04 DD 00026	PUSHL	#4	:	
FED2	CF		03 FB 00028	CALLS	#3, CHKPRO	:	
	04		50 E9 0002D	BLBC	R0, 2\$	:	
	50		01 D0 00030 1\$:	MOVL	#1, R0	:	1362
			04 00033	RET		:	
			50 D4 00034 2\$:	CLRL	R0	:	1366
			04 00036	RET		:	

; Routine Size: 55 bytes, Routine Base: CODE + 0101

CHECKPROT  
V04-000

Queue protection checks

E 15  
15-Sep-1984 23:59:44  
14-Sep-1984 12:36:58

VAX-11 Bliss-32 V4.0-742  
[JOBCTL.SRC]CHECKPROT.B32;1

Page 19  
(8)

: 334 1367 1 END  
: 335 1368 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	5024	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, QVR, NOPIC, ALIGN(2)
CODE	312	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	43	0	1000	00:01.4

COMMAND QUALIFIERS

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

: Size: 312 code + 5024 data bytes  
: Run Time: 00:13.7  
: Elapsed Time: 00:50.7  
: Lines/CPU Min: 5991  
: Lexemes/CPU-Min: 49703  
: Memory Used: 212 pages  
: Compilation Complete

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

BATCH  
LIS

BROADCAST  
LIS

BUFFERS  
LIS

CONTROL  
LIS

CHECKPROT  
LIS

ASYNCHRON  
LIS