```
            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCC   TTTTTTTTTTTTTTT  LLL
            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCC   TTTTTTTTTTTTTTT  LLL
            JJJ     000000000     BBBBBBBBBBBB        CCCCCCCCCCC   TTTTTTTTTTTTTTT  LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
            JJJ  000         000  BBBBBBBBBBBB     CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
JJJ         JJJ  000         000  BBB         BBB  CCC                    TTT        LLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCC         TTT        LLLLLLLLLLLLLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCC         TTT        LLLLLLLLLLLLLL
   JJJJJJJJJ        000000000     BBBBBBBBBBBB        CCCCCCCCCCC         TTT        LLLLLLLLLLLLLL
```

```
_$2
Val
----
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
314
485
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

FILEID**CONTROL

```
  CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR        OOOOOO    LL
  CCCCCCCC    OOOOOO    NN      NN  TTTTTTTTTT  RRRRRRRR        OOOOOO    LL
CC          OO      OO  NN      NN      TT      RR      RR  OO      OO  LL
CC          OO      OO  NN      NN      TT      RR      RR  OO      OO  LL
CC          OO      OO  NNNN    NN      TT      RR      RR  OO      OO  LL
CC          OO      OO  NNNN    NN      TT      RR      RR  OO      OO  LL
CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    OO      OO  LL
CC          OO      OO  NN  NN  NN      TT      RRRRRRRR    OO      OO  LL
CC          OO      OO  NN    NNNN      TT      RR  RR      OO      OO  LL
CC          OO      OO  NN    NNNN      TT      RR  RR      OO      OO  LL
CC          OO      OO  NN      NN      TT      RR    RR    OO      OO  LL              ....
CC          OO      OO  NN      NN      TT      RR    RR    OO      OO  LL              ....
  CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR    OOOOOO    LLLLLLLLLL      ....
  CCCCCCCC    OOOOOO    NN      NN      TT      RR      RR    OOOOOO    LLLLLLLLLL      ....

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
```

```
0 MODULE CONTROL  (%TITLE 'Main control logic'
0                 MAIN = JOBCTL,
0                 IDENT = 'V04-002'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:
1 !       Job controller.
1 !
1 ! ABSTRACT:
1 !       This module contains the main control logic.
1 !
1 ! ENVIRONMENT:
1 !       VAX/VMS user and kernel mode.
1 !--
1 !
1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
1 !
1 ! MODIFIED BY:
1 !
1 !       V04-002 JAK0236         J A Krycka      14-Sep-1984
1 !               Collect more diagnostic information.
1 !
1 !       V04-001 JAK0236         J A Krycka      13-Sep-1984
1 !               Set control flags to have the job controller take the abort-
1 !               restart patch instead of the bugcheck patch on encountering a
1 !               severe (fata) error.
1 !
1 !       V03-012 JAK0230         J A Krycka      28-Aug-1984
1 !               Conditionally bypass close of the system job queue file on the
1 !               job controller bug check error path to preserve internal RMS
```

```
1 !                 data structures in the dump.
1 !
1 !         V03-011 JAK0219           J A Krycka          17-Jul-1984
1 !                 Track changes in JOBCTLDEF.REQ.
1 !
1 !         V03-010 JAK0215           J A Krycka          25-May-1984
1 !                 For the time being, bug check on encountering a severe error
1 !                 instead of aborting/restarting the job controller.
1 !
1 !         V03-009 JAK0213           J A Krycka          18-May-1984
1 !                 Make $ENQ/$DEQ lock errors reported via the JBC$_COMREMJBC code
1 !                 into severe (fatal) errors.
1 !
1 !         V03-008 JAK0209           J A Krycka          09-May-1984
1 !                 Log diagnostic information in the DIAG_TRACE vector.
1 !
1 !         V03-007 JAK0206           J A Krycka          06-May-1984
1 !                 Conditionally request image dump on creating the symbiont and
1 !                 (recreated) job controller processes depending on whether or not
1 !                 the job controller was initiated with image dump enabled.
1 !
1 !         V03-006 GRR3006           Gregory R. Robert        06-May-1984
1 !                 Enable image dumps for newly created job controller process.
1 !                 Also resignal errors so that the last chance handler will
1 !                 terminate the process as appropriate instead of calling $EXIT.
1 !
1 !         V03-005 MLJ0115           Martin L. Jack, 30-Jul-1983
1 !                 Changes for job controller baselevel.
1 !
1 !         V03-004 MLJ0114           Martin L. Jack, 23-Jun-1983
1 !                 Changes for job controller baselevel.
1 !
1 !         V03-003 MLJ0113           Martin L. Jack, 26-May-1983
1 !                 Changes for job controller baselevel.
1 !
1 !         V03-002 MLJ0112           Martin L. Jack, 29-Apr-1983
1 !                 Changes for job controller baselevel.
1 !
1 !         V03-001 MLJ0109           Martin L. Jack, 14-Apr-1983
1 !                 Changes for job controller baselevel.
1 !
1 !**
```

```
REQUIRE 'SRC$:JOBCTLDEF';
REQUIRE 'SHRLIB$:OPCDEFTMP';            ! To support internal diagnostic option


FORWARD ROUTINE
        JOBCTL:                         NOVALUE,
        RESET_USERNAME,
        SCHEDULE_NONAST:                NOVALUE,
        MAILBOX_NONAST:                 NOVALUE,
        MAILBOX_AST:                    NOVALUE,
        SIGNAL_FILE_ERROR:              NOVALUE,
        MAIN_HANDLER_ACTION,
        MAIN_HANDLER,
        BUG_CHECK_SYSTEM:               NOVALUE;


EXTERNAL ROUTINE
        ALLOCATE_MEMORY,
        CLOSE_ACCOUNTING_FILE:          NOVALUE,
        DEALLOCATE_MEMORY:              NOVALUE,
        DELETE_SYMBIONTS:               NOVALUE,
        GETQUI_SERVICE,
        INITIALIZE_RESTRICTED:          NOVALUE,
        LOCK_QUEUE_FILE:                NOVALUE,
        OPEN_ACCOUNTING_FILE:           NOVALUE,
        PROCESS_ACCOUNTING:             NOVALUE,
        PROCESS_DELETION:               NOVALUE,
        SEND_SERVICE_RESPONSE:          NOVALUE,
        SYS$SETDDIR,
        SNDACC_SERVICE,
        SNDJBC_SERVICE,
        SNDSMB_SERVICE,
        SYMBIONT_SERVICE:               NOVALUE,
        UNSOLICITED_INPUT:              NOVALUE,
        UNLOCK_QUEUE_FILE:              NOVALUE,
        WARM_START_EXISTING_FILE:       NOVALUE;


EXTERNAL
        SGN$GL_VMSD2:   BBLOCK[4]       ADDRESSING_MODE(GENERAL),
        CTL$GL_PHD:     REF BBLOCK      ADDRESSING_MODE(GENERAL),
        CTL$T_USERNAME: VECTOR[,BYTE]   ADDRESSING_MODE(GENERAL);


EXTERNAL LITERAL
        SYS$C_JOBCTLMB;


GLOBAL BIND
        NLA0_DESC=              $DESCRIPTOR('_NLA0:'),
        OPA0_DESC=              $DESCRIPTOR('_OPA0:'),
        JOBCTLMBX_DESC=         UPLIT(6, UPLIT BYTE ('_', LONG(SYS$C_JOBCTLMB), ':')),
        ALL_PRIVILEGES=         UPLIT(-1, -1);


BIND
        JOB_CONTROL_DESC=       $DESCRIPTOR('JOB_CONTROL'): BBLOCK;
```

```

MACRO

        ! Fields for non-AST work queue block.
        !
        NWB_FLINK=          0,0,32,0 %,     ! Forward link
        NWB_BLINK=          4,0,32,0 %,     ! Backward link
        NWB_ROUT=           8,0,32,0 %,     ! Routine address
        NWB_PAGES=          12,0,32,0 %,    ! Size of NWB in pages
        NWB_PARAM=          16,0,0,0 %;     ! Routine parameters


BUILTIN
        INSQUE,
        REMQUE,
        TESTBITCS;
```

```
1 ROUTINE JOBCTL: NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is the main entry point for the job controller.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       STATUS_1,                               ! Status return
2       STATUS_2;                               ! Status return
2 OWN
2       PRIVILEGE_MASK: BBLOCK[8] PSECT(CODE) PRESET(
2               [PRV$V_NOACNT]  = TRUE,
2               [PRV$V_ALTPRI]  = TRUE,
2               [PRV$V_BYPASS]  = TRUE,
2               [PRV$V_CMKRNL]  = TRUE,
2               [PRV$V_DETACH]  = TRUE,
2               [PRV$V_EXQUOTA] = TRUE,
2               [PRV$V_LOG_IO]  = TRUE,
2               [PRV$V_NETMBX]  = TRUE,
2               [PRV$V_OPER]    = TRUE,
2               [PRV$V_PSWAPM]  = TRUE,
2               [PRV$V_SETPRV]  = TRUE,
2               [PRV$V_TMPMBX]  = TRUE,
2               [PRV$V_WORLD]   = TRUE);
2
2 BUILTIN
2       FP;
2
2
2 ! Initialize error processing control flags.
2 !
2 FLAGS[FLAGS_V_READ_VMSD2] = TRUE;        ! Reset debugging control flags from VMSD2 sysgen parameter
2                                         ! On encountering a severe (fatal) error:
2 FLAGS[FLAGS_V_CLUSTER_SCRAM] = FALSE;    !  Synchronously bug check entire cluster
2 FLAGS[FLAGS_V_LEAVE_OPEN] = FALSE;       !  Leave system job queue file open on exit
```

```
FLAGS[FLAGS_V_BUGCHECK] = FALSE;           ! Bug check system vs abort/restart image
                                           ! Other debugging flags:
FLAGS[FLAGS_V_CS_QF_DEADLOCK] = FALSE;     !  Bug check cluster on queue file deadlock error
FLAGS[FLAGS_V_LOG_QF_REPAIR] = FALSE;      !  Log occurrence of on-the-fly repair of queue file
FLAGS[FLAGS_V_OMIT_QF_INIT] = FALSE;       !  Omit queue file initializaton after opening it

IF .FLAGS[FLAGS_V_READ_VMSD2]              ! Reset debugging control flags
    THEN FLAGS<17,15> = .SGN$GL_VMSD2<1,15>;


! Set the privileges that we need.
!
$SETPRV(ENBFLG=TRUE, PRVADR=PRIVILEGE_MASK);


! Define SYS$DISK to set the default device for image dumps.
!
$CRELOG(
    TBLFLG=LOG$C_PROCESS,
    LOGNAM=$DESCRIPTOR ('SYS$DISK'),
    EQLNAM=$DESCRIPTOR('SYS$SYSROOT:') );


! Set the default directory for image dumps.
! Note that optional arguments must be specified as zero.
!
SYS$SETDDIR ($DESCRIPTOR ('[SYSEXE]'), 0, 0);


! Get the image dump flag from the process header which will be used later
! to determine whether or not image dump will be requested when a symbiont
! (or the job controller itself upon termnation) is created.
!
IF (.CTL$GL_PHD[PHD$W_FLAGS] AND PHD$M_IMGDMP) EQL 0
THEN IMAGE_DUMP_STSFLG = 0
ELSE IMAGE_DUMP_STSFLG = PRC$M_IMGDMP;


! Execute a $GETJPI to get process information that will be needed later.
!
$GETJPIW(
    EFN=JBC$K_SYNC_EFN,
    ITMLST=UPLIT(
        WORD(4, JPI$_ASTLM),      LONG(JBC_QUOTAS[ 1,0,32,0], 0),
        WORD(4, JPI$_BIOLM),      LONG(JBC_QUOTAS[ 6,0,32,0], 0),
        WORD(4, JPI$_BYTLM),      LONG(JBC_QUOTAS[11,0,32,0], 0),
        WORD(4, JPI$_DIOLM),      LONG(JBC_QUOTAS[21,0,32,0], 0),
        WORD(4, JPI$_ENQLM),      LONG(JBC_QUOTAS[26,0,32,0], 0),
        WORD(4, JPI$_FILLM),      LONG(JBC_QUOTAS[31,0,32,0], 0),
        WORD(4, JPI$_PGFLQUOTA),LONG(JBC_QUOTAS[36,0,32,0], 0),
        WORD(4, JPI$_PRCLM),      LONG(JBC_QUOTAS[41,0,32,0], 0),
        WORD(4, JPI$_TQLM),       LONG(JBC_QUOTAS[46,0,32,0], 0),
        WORD(4, JPI$_DFWSCNT),    LONG(JBC_QUOTAS[51,0,32,0], 0),
        WORD(4, JPI$_WSAUTHEXT),LONG(JBC_QUOTAS[56,0,32,0], 0),
        WORD(4, JPI$_WSAUTH),     LONG(JBC_QUOTAS[61,0,32,0], 0),
        WORD(4, JPI$_PRIB),       LONG(JBC_PRIORITY, 0),
        WORD(8, JPI$_CURPRIV),    LONG(JBC_PRIVILEGES, 0),
```

```
                WORD(4, JPI$_UIC),      LONG(JBC_UIC, 0),
                LONG(0)));

        JBC_QUOTAS[ 0,0,8,0] = PQL$_ASTLM;
        JBC_QUOTAS[ 5,0,8,0] = PQL$_BIOLM;
        JBC_QUOTAS[10,0,8,0] = PQL$_BYTLM;
        JBC_QUOTAS[15,0,8,0] = PQL$_CPULM;
        JBC_QUOTAS[20,0,8,0] = PQL$_DIOLM;
        JBC_QUOTAS[25,0,8,0] = PQL$_ENQLM;
        JBC_QUOTAS[30,0,8,0] = PQL$_FILLM;
        JBC_QUOTAS[35,0,8,0] = PQL$_PGFLQUOTA;
        JBC_QUOTAS[40,0,8,0] = PQL$_PRCLM;
        JBC_QUOTAS[45,0,8,0] = PQL$_TQELM;
        JBC_QUOTAS[50,0,8,0] = PQL$_WSDEFAULT;
        JBC_QUOTAS[55,0,8,0] = PQL$_WSEXTENT;
        JBC_QUOTAS[60,0,8,0] = PQL$_WSQUOTA;
        JBC_QUOTAS[65,0,8,0] = PQL$_LISTEND;


        ! Establish the condition handler.
        !
        .FP = MAIN_HANDLER;


        ! Set the process name and the username to JOB_CONTROL.
        !
        $SETPRN(PRCNAM=JOB_CONTROL_DESC);
        $CMKRNL(ROUTIN=RESET_USERNAME);


        ! Assign a channel to the job controller input mailbox.
        !
        STATUS_1 = $ASSIGN(DEVNAM=JOBCTLMBX_DESC, CHAN=MBX_CHAN);
        IF NOT .STATUS_1 THEN SIGNAL(JBC$_OPEJBCMBX OR STS$K_SEVERE, 0, .STATUS_1);


        ! Get the current time.
        !
        $GETTIM(TIMADR=CUR_TIME);


        ! Initialize the restricted login hours subsystem.
        !
        INITIALIZE_RESTRICTED();


        ! Initialize the accounting manager.
        !
        OPEN_ACCOUNTING_FILE(FALSE);


        ! Initialize the static queue headers.
        !
        NONAST_WORK_QUEUE[0] = NONAST_WORK_QUEUE[1] = NONAST_WORK_QUEUE[0];
        INCR I FROM 0 TO %ALLOCATION(MEMORY_FREE_QUEUES)/4-2 BY 2 DO
            MEMORY_FREE_QUEUES[.I] = MEMORY_FREE_QUEUES[.I+1] = MEMORY_FREE_QUEUES[.I];
```

```

! Queue the first read from the mailbox.
!
FLAGS[FLAGS_V_READ_POSTED] = TRUE;
STATUS_2 = $QIO(
    FUNC=IO$_READVBLK,
    CHAN=.MBX_CHAN,
    IOSB=MBX_IOSB,
    ASTADR=MAILBOX_AST,
    P1=MBX_BUFFER,
    P2=%ALLOCATION(MBX_BUFFER));
IF NOT .STATUS_2 THEN SIGNAL(JBC$_REAJBCMBX OR STS$K_SEVERE, 0, .STATUS_2);


! Hibernate, waiting for non-AST work.
!
WHILE TRUE DO
    BEGIN
    LOCAL
        NWB:            REF BBLOCK;     ! Pointer to non-AST work block


    UNTIL REMQUE(.NONAST_WORK_QUEUE[0], NWB) DO
        BEGIN
        (.NWB[NWB_ROUT])(NWB[NWB_PARAM]);
        DEALLOCATE_MEMORY(.NWB, .NWB[NWB_PAGES]);
        END;
    $HIBER;
    END;
END;
```

```
                    .TITLE  CONTROL Main control logic
                    .IDENT  \V04-002\

                    .PSECT  COMMON,NOEXE,  OVR,2

00000 DIAG_STORAGE_BASE:
                    .BLKB   0
00000 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:
              .BLKB   4
00240 MBX:    .BLKB   4
00244 MBX_END:.BLKB   4
00248 MEMORY_FREE_QUEUES:
              .BLKB   40
00270 NONAST_WORK_QUEUE:
              .BLKB   8
00278 BCB_FREE_LIST:
              .BLKB   4
0027C BCB_ACTIVE_LIST:
              .BLKB   4
00280 GQL_FREE_LIST:
              .BLKB   4
00284 GQL_ACTIVE_LIST:
              .BLKB   4
00288 OPEN_GETQUI_LIST:
              .BLKB   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:
              .BLKB   8
002B0 QUEUE_LOCK_LKSB:
              .BLKB   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:
               .BLKB   8
00570 ACCOUNTING_RABS:
               .BLKB   8
00578 ACCOUNT_FAB_A:
               .BLKB   80
005C8 ACCOUNT_RAB_A:
               .BLKB   68
0060C ACCOUNT_NAM_A:
               .BLKB   96
0066C ACCOUNT_RSA_A:
               .BLKB   255
0076B          .BLKB   1
0076C ACCOUNT_FAB_B:
               .BLKB   80
007BC ACCOUNT_RAB_B:
               .BLKB   68
00800 ACCOUNT_NAM_B:
               .BLKB   96
00860 ACCOUNT_RSA_B:
               .BLKB   255
0095F          .BLKB   1
00960 DIAG_FAB:
               .BLKB   80
009B0 DIAG_RAB:
               .BLKB   68
009F4 MBX_CHAN:
               .BLKB   4
009F8 MBX_IOSB:
               .BLKB   8
00A00 MBX_BUFFER:
               .BLKB   1024
00E00 VALUE_STORAGE_BASE:
               .BLKB   0
00E00 ITEM_PRESENT:
               .BLKB   32
00E20 VALUE_GETQUI_BASE:
               .BLKB   0
00E20 VALUE_ACCOUNTING_MESSAGE:
               .BLKB   6
00E26 VALUE_ACCOUNTING_TYPES:
               .BLKB   4
00E2A VALUE_AFTER_TIME:
```

```
                .BLKB   8
00E32 VALUE_ALIGNMENT_PAGES:
                .BLKB   1
00E33 VALUE_BASE_PRIORITY:
                .BLKB   1
00E34 VALUE_BATCH_INPUT:
                .BLKB   6
00E3A VALUE_BATCH_OUTPUT:
                .BLKB   10
00E44 VALUE_BUFFER_COUNT:
                .BLKB   1
00E45 VALUE_CHARACTERISTIC_NAME:
                .BLKB   6
00E4B VALUE_CHARACTERISTIC_NUMBER:
                .BLKB   1
00E4C VALUE_CHARACTERISTICS:
                .BLKB   16
00E5C VALUE_CHECKPOINT_DATA:
                .BLKB   6
00E62 VALUE_CLI:
                .BLKB   6
00E68 VALUE_CPU_DEFAULT:
                .BLKB   4
00E6C VALUE_CPU_LIMIT:
                .BLKB   4
00E70 VALUE_DESTINATION_QUEUE:
                .BLKB   8
00E78 VALUE_DEVICE_NAME:
                .BLKB   6
00E7E VALUE_ENTRY_NUMBER:
                .BLKB   4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
                .BLKB   10
00E8C VALUE_EXTEND_QUANTITY:
                .BLKB   2
00E8E VALUE_FILE_COPIES:
                .BLKB   1
00E8F VALUE_FILE_IDENTIFICATION:
                .BLKB   36
00EB3 VALUE_FILE_SETUP_MODULES:
                .BLKB   6
00EB9 VALUE_FILE_SPECIFICATION:
                .BLKB   6
00EBF VALUE_FIRST_PAGE:
                .BLKB   4
00EC3 VALUE_FORM_DESCRIPTION:
                .BLKB   6
00EC9 VALUE_FORM_LENGTH:
                .BLKB   1
00ECA VALUE_FORM_MARGIN_BOTTOM:
                .BLKB   1
00ECB VALUE_FORM_MARGIN_LEFT:
                .BLKB   2
00ECD VALUE_FORM_MARGIN_RIGHT:
                .BLKB   2
00ECF VALUE_FORM_MARGIN_TOP:
                .BLKB   1
```

```
00ED0 VALUE_FORM_NAME:
              .BLKB   6
00ED6 VALUE_FORM_NUMBER:
              .BLKB   4
00EDA VALUE_FORM:
              .BLKB   8
00EE2 VALUE_FORM_SETUP_MODULES:
              .BLKB   6
00EE8 VALUE_FORM_STOCK:
              .BLKB   6
00EEE VALUE_FORM_WIDTH:
              .BLKB   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   6
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    6
                                   01369 VALUE_RELATIVE_PAGE:
                                                .BLKB    4
                                   0136D VALUE_RESERVED_INPUT_1:
                                                .BLKB    1
                                   0136E VALUE_RESERVED_INPUT_2:
                                                .BLKB    2
                                   01370 VALUE_RESERVED_INPUT_3:
                                                .BLKB    4
                                   01374 VALUE_RESERVED_INPUT_4:
                                                .BLKB    6
                                   0137A VALUE_RESERVED_OUTPUT_1:
                                                .BLKB    10
                                   01384 VALUE_RESERVED_OUTPUT_2:
                                                .BLKB    10
                                   0138E VALUE_SEARCH_STRING:
                                                .BLKB    6
                                   01394 VALUE_SCSNODE_NAME:
                                                .BLKB    6
                                   0139A VALUE_WSDEFAULT:
                                                .BLKB    2
                                   0139C VALUE_WSEXTENT:
                                                .BLKB    2
                                   0139E VALUE_WSQUOTA:
                                                .BLKB    2
                                   013A0 VALUE_STORAGE_END:
                                                .BLKB    0

                                                .PSECT   CODE,NOWRT,2

      3A  30  41  4C  4E  5F  00000 P.AAB:     .ASCII   \_NLA0:\                              ;
                              00006            .BLKB    2
                   00000006   00008 P.AAA:     .LONG    6                                     ;
                   00000000'  0000C            .ADDRESS P.AAB                                 ;
      3A  30  41  50  4F  5F  00010 P.AAD:     .ASCII   \_OPA0:\                              ;
                              00016            .BLKB    2
                   00000006   00018 P.AAC:     .LONG    6                                     ;
                   00000000'  0001C            .ADDRESS P.AAD                                 ;
                         5F   00020 P.AAF:     .ASCII   \_\                                   ;
                   00000000G  00021            .LONG    SYS$C_JOBCTLMB                        ;
                         3A   00025            .ASCII   \:\                                   ;
                              00026            .BLKB    2
                   00000006   00028 P.AAE:     .LONG    6                                     ;
                   00000000'  0002C            .ADDRESS P.AAF                                 ;
         FFFFFFFF  FFFFFFFF   00030 P.AAG:     .LONG    -1, -1                                ;
```

```
                     4C 4F 52 54 4E 4F 43 5F 42 4F 4A  00038 P.AAI:   .ASCII   \JOB_CONTROL\                  ;
                                                       00043            .BLKB    1
                                         0000000B      00044 P.AAH:   .LONG    11                             ;
                                         00000000'     00048            .ADDRESS P.AAI                          ;
                                      20 1D F2 A1      0004C PRIVILEGE_MASK:
                                                                        .BYTE    -95, -14, 29, 32               ;
                                                       00050            .BLKB    4
                        4B 53 49 44 24 53 59 53        00054 P.AAK:   .ASCII   \SYS$DISK\                     ;
                                         00000008      0005C P.AAJ:   .LONG    8                              ;
                                         00000000'     00060            .ADDRESS P.AAK                          ;
                  3A 54 4F 4F 52 53 59 53 24 53 59 53  00064 P.AAM:   .ASCII   \SYS$SYSROOT:\                 ;
                                         0000000C      00070 P.AAL:   .LONG    12                             ;
                                         00000000'     00074            .ADDRESS P.AAM                          ;
                        5D 45 58 45 53 59 53 5B        00078 P.AAO:   .ASCII   \[SYSEXE]\                     ;
                                         00000008      00080 P.AAN:   .LONG    8                              ;
                                         00000000'     00084            .ADDRESS P.AAO                          ;
                                    0409 0004          00088 P.AAP:   .WORD    4, 1033                        ;
                                         00000000'     0008C            .ADDRESS JBC_QUOTAS+1                   ;
                                         00000000      00090            .LONG    0                              ;
                                    0310 0004          00094            .WORD    4, 784                         ;
                                         00000000'     00098            .ADDRESS JBC_QUOTAS+6                   ;
                                         00000000      0009C            .LONG    0                              ;
                                    031A 0004          000A0            .WORD    4, 794                         ;
                                         00000000'     000A4            .ADDRESS JBC_QUOTAS+11                  ;
                                         00000000      000A8            .LONG    0                              ;
                                    0313 0004          000AC            .WORD    4, 787                         ;
                                         00000000'     000B0            .ADDRESS JBC_QUOTAS+21                  ;
                                         00000000      000B4            .LONG    0                              ;
                                    0320 0004          000B8            .WORD    4, 800                         ;
                                         00000000'     000BC            .ADDRESS JBC_QUOTAS+26                  ;
                                         00000000      000C0            .LONG    0                              ;
                                    040F 0004          000C4            .WORD    4, 1039                        ;
                                         00000000'     000C8            .ADDRESS JBC_QUOTAS+31                  ;
                                         00000000      000CC            .LONG    0                              ;
                                    040E 0004          000D0            .WORD    4, 1038                        ;
                                         00000000'     000D4            .ADDRESS JBC_QUOTAS+36                  ;
                                         00000000      000D8            .LONG    0                              ;
                                    0408 0004          000DC            .WORD    4, 1032                        ;
                                         00000000'     000E0            .ADDRESS JBC_QUOTAS+41                  ;
                                         00000000      000E4            .LONG    0                              ;
                                    0410 0004          000E8            .WORD    4, 1040                        ;
                                         00000000'     000EC            .ADDRESS JBC_QUOTAS+46                  ;
                                         00000000      000F0            .LONG    0                              ;
                                    0403 0004          000F4            .WORD    4, 1027                        ;
                                         00000000'     000F8            .ADDRESS JBC_QUOTAS+51                  ;
                                         00000000      000FC            .LONG    0                              ;
                                    0417 0004          00100            .WORD    4, 1047                        ;
                                         00000000'     00104            .ADDRESS JBC_QUOTAS+56                  ;
                                         00000000      00108            .LONG    0                              ;
                                    0401 0004          0010C            .WORD    4, 1025                        ;
                                         00000000'     00110            .ADDRESS JBC_QUOTAS+61                  ;
                                         00000000      00114            .LONG    0                              ;
                                    0309 0004          00118            .WORD    4, 777                         ;
                                         00000000'     0011C            .ADDRESS JBC_PRIORITY                   ;
                                         00000000      00120            .LONG    0                              ;
                                    0400 0008          00124            .WORD    8, 1024                        ;
                                         00000000'     00128            .ADDRESS JBC_PRIVILEGES                 ;
```

```
                                  00000000  0012C           .LONG    0                                         ;
                              0304  0004  00130           .WORD    4, 772                                    ;
                                  00000000' 00134           .ADDRESS JBC_UIC                                 ;
                                  00000000  00138           .LONG    0                                         ;
                                  00000000  0013C           .LONG    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
                                                 NLAO_DESC==            P.AAA
                                                 OPAO_DESC==            P.AAC
                                                 JOBCTLMBX_DESC==       P.AAE
                                                 ALL_PRIVILEGES==       P.AAG
                                                 JOB_CONTROL_DESC=      P.AAH
                                                            .EXTRN   ALLOCATE_MEMORY
                                                            .EXTRN   CLOSE_ACCOUNTING_FILE
                                                            .EXTRN   DEALLOCATE_MEMORY
                                                            .EXTRN   DELETE_SYMBIONTS
                                                            .EXTRN   GETQUI_SERVICE, INITIALIZE_RESTRICTED
                                                            .EXTRN   LOCK_QUEUE_FILE
                                                            .EXTRN   OPEN_ACCOUNTING_FILE
                                                            .EXTRN   PROCESS_ACCOUNTING
                                                            .EXTRN   PROCESS_DELETION
                                                            .EXTRN   SEND_SERVICE_RESPONSE
                                                            .EXTRN   SYS$SETDDIR, SNDACC_SERVICE
                                                            .EXTRN   SNDJBC_SERVICE, SNDSMB_SERVICE
                                                            .EXTRN   SYMBIONT_SERVICE
                                                            .EXTRN   UNSOLICITED_INPUT
                                                            .EXTRN   UNLOCK_QUEUE_FILE
                                                            .EXTRN   WARM_START_EXISTING_FILE
                                                            .EXTRN   SGN$GL_VMSD2, CTL$GL_PHD
                                                            .EXTRN   CTL$T_USERNAME, SYS$C_JOBCTLMB
                                                            .EXTRN   SYS$SETPRV, SYS$CRELOG
                                                            .EXTRN   SYS$GETJPIW, SYS$SETPRN
                                                            .EXTRN   SYS$CMKRNL, SYS$ASSIGN
                                                            .EXTRN   SYS$GETTIM, SYS$QIO
                                                            .EXTRN   SYS$HIBER

                                        003C 00000 JOBCTL:  .WORD    Save R2,R3,R4,R5                          ; 1454
                  55 00000000G      00   9E 00002           MOVAB    LIB$SIGNAL, R5                            ;
                  54      FEFF      CF   9E 00009           MOVAB    PRIVILEGE_MASK, R4                        ;
                  53 00000000'      EF   9E 0000E           MOVAB    FLAGS, R3                                 ;
             02   A3                01   88 00015           BISB2    #1, FLAGS+2                               ; 1507
             02   A3        7E      8F   8A 00019           BICB2    #126, FLAGS+2                             ; 1515
                  0F        02      A3   E9 0001E           BLBC     FLAGS+2, 1$                               ; 1517
   50 00000000G   00        0F      01   EF 00022           EXTZV    #1, #15, SGN$GL_VMSD2, R0                 ; 1518
02    A3          0F                01   50   F0 0002B           INSV     R0, #1, #15, FLAGS+2                      ;
                                    7E   7C 00031 1$:       CLRQ     -(SP)                                     ; 1523
                                    54   DD 00033           PUSHL    R4                                        ;
                                    01   DD 00035           PUSHL    #1                                        ;
           00000000G  00            04   FB 00037           CALLS    #4, SYS$SETPRV                            ;
                                    7E   D4 0003E           CLRL     -(SP)                                     ; 1531
```

```
                         24    A4    9F 00040            PUSHAB  P.AAL                                    :
                         10    A4    9F 00043            PUSHAB  P.AAJ                                    :
                               02    DD 00046            PUSHL   #2                                       :
      00000000G  00            04    FB 00048            CALLS   #4, SYS$CRELOG                           :
                               7E    7C 0004F            CLRQ    -(SP)                                    : 1537
                         34    A4    9F 00051            PUSHAB  P.AAN                                    :
      00000000G  EF            03    FB 00054            CALLS   #3, SYS$SETDDIR                          :
                 50 00000000G  00    D0 0005B            MOVL    CTL$GL_PHD, R0                           : 1544
 05         36   A0            05    E0 00062            BBS     #5, 54(R0), 2$                           :
                         04    A3    D4 00067            CLRL    IMAGE_DUMP_STSFLG                        : 1545
                               06    11 0006A            BRB     3$                                       :
           04    A3      0800  8F    3C 0006C  2$:       MOVZWL  #2048, IMAGE_DUMP_STSFLG                 : 1546
                               7E    7C 00072  3$:       CLRQ    -(SP)                                    : 1569
                               7E    D4 00074            CLRL    -(SP)                                    :
                         3C    A4    9F 00076            PUSHAB  P.AAP                                    :
                               7E    7C 00079            CLRQ    -(SP)                                    :
                               01    DD 0007B            PUSHL   #1                                       :
      00000000G  00            07    FB 0007D            CALLS   #7, SYS$GETJPIW                          :
           00CC  C3            01    90 00084            MOVB    #1, JBC_QUOTAS                           : 1571
           00D1  C3            02    90 00089            MOVB    #2, JBC_QUOTAS+5                         : 1572
           00D6  C3            03    90 0008E            MOVB    #3, JBC_QUOTAS+10                        : 1573
           00DB  C3            04    90 00093            MOVB    #4, JBC_QUOTAS+15                        : 1574
           00E0  C3            05    90 00098            MOVB    #5, JBC_QUOTAS+20                        : 1575
           00E5  C3            0C    90 0009D            MOVB    #12, JBC_QUOTAS+25                       : 1576
           00EA  C3            06    90 000A2            MOVB    #6, JBC_QUOTAS+30                        : 1577
           00EF  C3            07    90 000A7            MOVB    #7, JBC_QUOTAS+35                        : 1578
           00F4  C3            08    90 000AC            MOVB    #8, JBC_QUOTAS+40                        : 1579
           00F9  C3            09    90 000B1            MOVB    #9, JBC_QUOTAS+45                        : 1580
           00FE  C3            0B    90 000B6            MOVB    #11, JBC_QUOTAS+50                       : 1581
           0103  C3            0D    90 000BB            MOVB    #13, JBC_QUOTAS+55                       : 1582
           0108  C3            0A    90 000C0            MOVB    #10, JBC_QUOTAS+60                       : 1583
                       010D    C3    94 000C5            CLRB    JBC_QUOTAS+65                            : 1584
                 6D   0000V    CF    9E 000C9            MOVAB   MAIN_HANDLER, (FP)                       : 1589
                         F8    A4    9F 000CE            PUSHAB  JOB_CONTROL_DESC                         : 1594
      00000000G  00            01    FB 000D1            CALLS   #1, SYS$SETPRN                           :
                               7E    D4 000D8            CLRL    -(SP)                                    : 1595
                      0000V    CF    9F 000DA            PUSHAB  RESET_USERNAME                           :
      00000000G  00            02    FB 000DE            CALLS   #2, SYS$CMKRNL                           :
                               7E    7C 000E5            CLRQ    -(SP)                                    : 1600
                       07F4    C3    9F 000E7            PUSHAB  MBX_CHAN                                 :
                         DC    A4    9F 000EB            PUSHAB  JOBCTLMBX_DESC                           :
      00000000G  00            04    FB 000EE            CALLS   #4, SYS$ASSIGN                           :
                 0D            50    E8 000F5            BLBS    STATUS_1, 4$                             : 1601
                               50    DD 000F8            PUSHL   STATUS_1                                 :
                               7E    D4 000FA            CLRL    -(SP)                                    :
                   00048434    8F    DD 000FC            PUSHL   #295988                                  :
                 65            03    FB 00102            CALLS   #3, LIB$SIGNAL                           :
                         10    A3    9F 00105  4$:       PUSHAB  CUR_TIME                                 : 1606
      00000000G  00            01    FB 00108            CALLS   #1, SYS$GETTIM                           :
      00000000G  EF            00    FB 0010F            CALLS   #0, INITIALIZE_RESTRICTED                : 1611
                               7E    D4 00116            CLRL    -(SP)                                    : 1616
      00000000G  EF            01    FB 00118            CALLS   #1, OPEN_ACCOUNTING_FILE                 :
                 50      70    A3    9E 0011F            MOVAB   NONAST_WORK_QUEUE, R0                    : 1621
             74  A3            50    D0 00123            MOVL    R0, NONAST_WORK_QUEUE+4                  :
             70  A3            50    D0 00127            MOVL    R0, NONAST_WORK_QUEUE                    :
                               50    D4 0012B            CLRL    I                                        : 1622
                 51      48 A340     DE 0012D  5$:       MOVAL   MEMORY_FREE_QUEUES[I], R1                : 1623
```

```
                         4C A340         51 D0 00132         MOVL    R1, MEMORY_FREE_QUEUES+4[I]           :
                         48 A340         51 D0 00137         MOVL    R1, MEMORY_FREE_QUEUES[I]             :
     FFEB          50         02         08 F1 0013C         ACBL    #8, #2, I, 5$                         :
                              63         02 88 00142         BISB2   #2, FLAGS                             : 1628
                                         7E 7C 00145         CLRQ    -(SP)                                 : 1635
                                         7E 7C 00147         CLRQ    -(SP)                                 :
                              7E   0400  8F 3C 00149         MOVZWL  #1024, -(SP)                          :
                                   0800  C3 9F 0014E         PUSHAB  MBX_BUFFER                            :
                                         7E D4 00152         CLRL    -(SP)                                 :
                                  0000V  CF 9F 00154         PUSHAB  MAILBOX_AST                           :
                                   07F8  C3 9F 00158         PUSHAB  MBX_IOSB                              :
                                         31 DD 0015C         PUSHL   #49                                   :
                                   07F4  C3 DD 0015E         PUSHL   MBX_CHAN                              :
                                         7E D4 00162         CLRL    -(SP)                                 :
                    00000000G 00         0C FB 00164         CALLS   #12, SYS$QIO                          :
                              0D         50 E8 0016B         BLBS    STATUS_2, 6$                          : 1636
                                         50 DD 0016E         PUSHL   STATUS_2                              :
                                         7E D4 00170         CLRL    -(SP)                                 :
                               0004844C  8F DD 00172         PUSHL   #296012                               :
                              65         03 FB 00178         CALLS   #3, LIB$SIGNAL                        :
                              52     70  B3 0F 0017B 6$:     REMQUE  @NONAST_WORK_QUEUE, NWB               : 1647
                                         15 1D 0017F         BVS     7$                                    :
                                     10  A2 9F 00181         PUSHAB  16(NWB)                               : 1649
                         08   B2         01 FB 00184         CALLS   #1, @8(NWB)                           :
                                     0C  A2 DD 00188         PUSHL   12(NWB)                               : 1650
                                         52 DD 0018B         PUSHL   NWB                                   :
                    00000000G EF         02 FB 0018D         CALLS   #2, DEALLOCATE_MEMORY                 :
                                         E5 11 00194         BRB     6$                                    : 1647
                    00000000G 00         00 FB 00196 7$:     CALLS   #0, SYS$HIBER                         : 1651
                                         DC 11 0019D         BRB     6$                                    : 1641

; Routine Size:  415 bytes,    Routine Base:  CODE + 0140
```

```
;   1 ROUTINE RESET_USERNAME=
;   1
;   1 !++
;   1 !
;   1 ! FUNCTIONAL DESCRIPTION:
;   1 !       This routine sets the process username to JOB_CONTROL.  It executes in
;   1 !       kernel mode.
;   1 !
;   1 ! INPUT PARAMETERS:
;   1 !       NONE
;   1 !
;   1 ! IMPLICIT INPUTS:
;   1 !       NONE
;   1 !
;   1 ! OUTPUT PARAMETERS:
;   1 !       NONE
;   1 !
;   1 ! IMPLICIT OUTPUTS:
;   1 !       CTL$T_USERNAME set to JOB_CONTROL.
;   1 !
;   1 ! ROUTINE VALUE:
;   1 !       SS$_NORMAL
;   1 !
;   1 ! SIDE EFFECTS:
;   1 !       NONE
;   1 !
;   1 !--
;   1
;   2 BEGIN
;   2 CH$COPY(
;   2     .JOB_CONTROL_DESC[DSC$W_LENGTH], .JOB_CONTROL_DESC[DSC$A_POINTER],
;   2     %C' ',
;   2     12, CTL$T_USERNAME);
;   2 SS$_NORMAL
;   1 END;


                                             003C 00000 RESET_USERNAME:
                                                               .WORD   Save R2,R3,R4,R5                          ; 1655
        0C          20     FD60    DF      FD5F    CF   2C 00002       MOVC5   JOB_CONTROL_DESC, @JOB_CONTROL_DESC+4, -  ; 1684
                                   00000000G  00           0000B               #32, #12, CTL$T_USERNAME                  ;
                                   50                 01   D0 00010       MOVL    #1, R0                                    ; 1689
                                                           04 00013       RET                                               ;

; Routine Size:  20 bytes,    Routine Base:  CODE + 02DF
```

```
1 GLOBAL ROUTINE SCHEDULE_NONAST(ROUT,LENGTH,ADDRESS): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is called by an AST routine to schedule a specified routine
1 !       to execute at non-AST level.  The requests are processed first in, first
1 !       out.
1 !
1 ! INPUT PARAMETERS:
1 !       ROUT            - Address of the routine to be executed.
1 !       LENGTH          - (Optional) Descriptor for parameter data.
1 !       ADDRESS         -
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 LOCAL
2       NWB:            REF BBLOCK;     ! Pointer to non-AST work block
2 BUILTIN
2       ACTUALCOUNT;
2
2
2 IF ACTUALCOUNT() LSSU 2
2 THEN
3     BEGIN
3     NWB = ALLOCATE_MEMORY();
3     NWB[NWB_ROUT] = .ROUT;
3     NWB[NWB_PAGES] = 1;
3     END
2 ELSE
3     BEGIN
3     LOCAL
3       NP;
3
3     NP = (.LENGTH + 16 + 511) / 512;
3     NWB = ALLOCATE_MEMORY(.NP);
3     NWB[NWB_ROUT] = .ROUT;
3     NWB[NWB_PAGES] = .NP;
3     CH$MOVE(.LENGTH, .ADDRESS, NWB[NWB_PARAM]);
2     END;
2
```

```
;  471          1747  2
;  472          1748  2 IF INSQUE(.NWB, .NONAST_WORK_QUEUE[1]) THEN $WAKE();
;  473          1749  1 END;


                                                                  .EXTRN   SYS$WAKE

                                                  00FC 00000      .ENTRY   SCHEDULE_NONAST, Save R2,R3,R4,R5,R6,R7      ; 1690
                         57 00000000G    EF    9E 00002           MOVAB    ALLOCATE_MEMORY, R7                          ;
                         02              6C    91 00009           CMPB     (AP), #2                                     ; 1728
                                         11    1E 0000C           BGEQU    1$                                           ;
                         67              00    FB 0000E           CALLS    #0, ALLOCATE_MEMORY                          ; 1731
                         56              50    D0 00011           MOVL     R0, NWB                                      ;
                 08      A6         04   AC    D0 00014           MOVL     ROUT, 8(NWB)                                 ; 1732
                 0C      A6              01    D0 00019           MOVL     #1, 12(NWB)                                  ; 1733
                                         29    11 0001D           BRB      2$                                           ; 1728
         50      08      AC 0000020F     8F    C1 0001F 1$:       ADDL3    #527, LENGTH, R0                             ; 1740
         52              50 00000200     8F    C7 00028           DIVL3    #512, R0, NP                                 ;
                                         52    DD 00030           PUSHL    NP                                           ; 1741
                         67              01    FB 00032           CALLS    #1, ALLOCATE_MEMORY                          ;
                         56              50    D0 00035           MOVL     R0, NWB                                      ;
                 08      A6         04   AC    D0 00038           MOVL     ROUT, 8(NWB)                                 ; 1742
                 0C      A6              52    D0 0003D           MOVL     NP, 12(NWB)                                  ; 1743
    10     A6    0C      BC         08   AC    28 00041           MOVC3    LENGTH, @ADDRESS, 16(NWB)                    ; 1744
               00000000' FF              66    0E 00048 2$:       INSQUE   (NWB), @NONAST_WORK_QUEUE+4                  ; 1748
                                         09    12 0004F           BNEQ     3$                                           ;
                                         7E    7C 00051           CLRQ     -(SP)                                        ;
               00000000G 00              02    FB 00053           CALLS    #2, SYS$WAKE                                 ;
                                               04 0005A 3$:       RET                                                   ; 1749

; Routine Size:  91 bytes,    Routine Base:  CODE + 02F3
```

```
ROUTINE MAILBOX_NONAST(MSG): NOVALUE=

!++
!
! FUNCTIONAL DESCRIPTION:
!	This routine is scheduled to execute by the completion AST routine for
!	a mailbox read.  It processes the buffered message.
!
! INPUT PARAMETERS:
!	MSG		- Pointer to message IOSB followed by message.
!
! IMPLICIT INPUTS:
!	NONE
!
! OUTPUT PARAMETERS:
!	NONE
!
! IMPLICIT OUTPUTS:
!	NONE
!
! ROUTINE VALUE:
!	NONE
!
! SIDE EFFECTS:
!	NONE
!
!--

BEGIN
LOCAL
	STATUS;				! Status return


! Initialize global pointers to the message.
!
MBX = .MSG + 8;
MBX_END = .MBX + .MBX[ACM$W_MSGLEN];

IF .FLAGS[FLAGS_V_READ_VMSD2]		! Reset debug control flags
    THEN FLAGS<17,15> = .SGN$GL_VMSD2<1,15>;

DIAG_TRACE[0] = .MBX[MSG_W_TYPE];	! ***** diagnostic info *****
DIAG_TRACE[1] = 0;			! ***** diagnostic info *****
DIAG_COUNT[0] = DIAG_COUNT[1] = DIAG_COUNT[2] = DIAG_COUNT[3] = 0;
DIAG_COUNT[4] = DIAG_COUNT[5] = DIAG_COUNT[6] = DIAG_COUNT[7] = 0;


! Process the message.
!
CASE .MBX[MSG_W_TYPE] FROM MSG$_DELPROC TO MSG$_GETQUI OF
    SET


    [INRANGE, OUTRANGE]:
	SIGNAL(JBC$_INVMSG OR STS$K_ERROR);

```

```
        [MSG$_DELPROC]:
            BEGIN
            COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
            PROCESS_DELETION();
            END;


        [MSG$_SNDSMB]:
            BEGIN
            COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
            RSP[RSP_W_TYPE] = MSG$_SMBRSP;
            RSP[RSP_W_ENTRY_NUMBER] = 0;
            RSP[RSP_L_STATUS] = SNDSMB_SERVICE();
            UNLOCK_QUEUE_FILE();
            SEND_SERVICE_RESPONSE();
            END;


        [MSG$_SMBINI]:
            BEGIN
            $GETTIM(TIMADR=CUR_TIME);
            SYMBIONT_SERVICE();
            END;


        [MSG$_SMBDON]:                                      ! Connection manager message
            BEGIN
            $GETTIM(TIMADR=CUR_TIME);
            IF .QUEUE_FAB[FAB$W_IFI] NEQ 0
            THEN
                BEGIN
                LOCAL
                    SYSID:  BBLOCK[6];

                CH$COPY(.MBX[8,0,8,0], MBX[9,0,0,0], %C' ', 6, SYSID);
                LOCK_QUEUE_FILE();
                WARM_START_EXISTING_FILE(SYSID);
                UNLOCK_QUEUE_FILE();
                END;
            END;


        [MSG$_SNDACC]:
            BEGIN
            COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
            RSP[RSP_W_TYPE] = MSG$_ACCRSP;
            RSP[RSP_W_ENTRY_NUMBER] = 0;
            RSP[RSP_L_STATUS] = SNDACC_SERVICE();
            SEND_SERVICE_RESPONSE();
            END;


        [MSG$_PURPROC, MSG$_DELIMAG, MSG$_PURIMAG]:
            BEGIN
            COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
            PROCESS_ACCOUNTING();
            END;
```

```
  2
  2
  2     [MSG$_SNDJBC]:
  3         BEGIN
  3         COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
  3         RSP[RSP_W_ENTRY_NUMBER] = 0;
  3         RSP[RSP_L_STATUS] = SNDJBC_SERVICE();
  3         UNLOCK_QUEUE_FILE();
  3         SEND_SERVICE_RESPONSE();
  2         END;
  2
  2
  2     [MSG$_GETQUI]:
  3         BEGIN
  3         COPY_TIME(MBX[ACM$Q_SYSTIME], CUR_TIME);
  3         RSP[RSP_W_ENTRY_NUMBER] = 0;
  3         RSP[RSP_L_STATUS] = GETQUI_SERVICE();
  3         UNLOCK_QUEUE_FILE();
  3         SEND_SERVICE_RESPONSE();
  2         END;
  2
  2
  2     TES;
  2
  2
  2 DIAG_COUNT[12] = .DIAG_COUNT[12] + .DIAG_COUNT[0];
  2 DIAG_COUNT[13] = .DIAG_COUNT[13] + .DIAG_COUNT[1];
  2 DIAG_COUNT[14] = .DIAG_COUNT[14] + .DIAG_COUNT[2];
  2 DIAG_COUNT[15] = .DIAG_COUNT[15] + .DIAG_COUNT[3];
  2 DIAG_COUNT[16] = .DIAG_COUNT[16] + .DIAG_COUNT[4];
  2 DIAG_COUNT[17] = .DIAG_COUNT[17] + .DIAG_COUNT[5];
  2 DIAG_COUNT[18] = .DIAG_COUNT[18] + .DIAG_COUNT[6];
  2 DIAG_COUNT[19] = .DIAG_COUNT[19] + .DIAG_COUNT[7];
  2
  2
  2 ! Queue another read from the mailbox if there is no read in progress -- it
  2 ! was shut down because the maximum was exceeded.
  2 !
  2 MBX_MESSAGE_COUNT = .MBX_MESSAGE_COUNT - 1;
  2 IF TESTBITCS(FLAGS[FLAGS_V_READ_POSTED])
  2 THEN
  3     BEGIN
  3     LOCAL
  3         STATUS;                     ! Status return
  3
P 3     STATUS = $QIO(
P 3         FUNC=IO$_READVBLK,
P 3         CHAN=.MBX_CHAN,
P 3         IOSB=MBX_IOSB,
P 3         ASTADR=MAILBOX_AST,
P 3         P1=MBX_BUFFER,
  3         P2=%ALLOCATION(MBX_BUFFER));
  3     IF NOT .STATUS THEN SIGNAL(JBC$_REAJBCMBX OR STS$K_SEVERE, 0, .STATUS);
  2     END;
  1 END;
```

```
                                              03FC 00000 MAILBOX_NONAST:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 1750
                      59 00000000G   EF   9E 00002             MOVAB   UNLOCK_QUEUE_FILE, R9                           ;
                      58 00000000G   00   9E 00009             MOVAB   SYS$GETTIM, R8                                  ;
                      57 00000000G   00   9E 00010             MOVAB   LIB$SIGNAL, R7                                  ;
                      56 00000000'   EF   9E 00017             MOVAB   CUR_TIME, R6                                    ;
                      5E             08   C2 0001E             SUBL2   #8, SP                                          ;
        30   A6   04  AC             08   C1 00021             ADDL3   #8, MSG, MBX                                    ; 1785
                      52        30   A6   D0 00027             MOVL    MBX, R2                                         ; 1786
                      50        FA   A2   3C 0002B             MOVZWL  -6(R2), R0                                      ;
        34   A6       52             50   C1 0002F             ADDL3   R0, R2, MBX_END                                 ;
                      0F        F2   A6   E9 00034             BLBC    FLAGS+2, 1$                                     ; 1788
   50 00000000G   00  0F             01   EF 00038             EXTZV   #1, #15, SGN$GL_VMSD2, R0                       ; 1789
F2   A6           0F  01             50   F0 00041             INSV    R0, #1, #15, FLAGS+2                            ;
             FDF0     C6             62   3C 00047 1$:         MOVZWL  (R2), DIAG_TRACE                                ; 1791
                           FDF4      C6   D4 0004C             CLRL    DIAG_TRACE+4                                    ; 1792
                           FE58      C6   7C 00050             CLRQ    DIAG_COUNT+8                                    ; 1793
                           FE50      C6   7C 00054             CLRQ    DIAG_COUNT                                      ;
                           FE68      C6   7C 00058             CLRQ    DIAG_COUNT+24                                   ; 1794
                           FE60      C6   7C 0005C             CLRQ    DIAG_COUNT+16                                   ;
                  0D  03             62   AF 00060             CASEW   (R2), #3, #13                                   ; 1799
  001C      001C    0034       0027          00064 2$:         .WORD   4$-2$,-                                         ;
  0083      0054    0046       001C          0006C                     5$-2$,-                                         ;
  001C      009A    009A       009A          00074                     3$-2$,-                                         ;
                    00B8       00A7          0007C                     3$-2$,-                                         ;
                                                                       3$-2$,-                                         ;
                                                                       6$-2$,-                                         ;
                                                                       7$-2$,-                                         ;
                                                                       8$-2$,-                                         ;
                                                                       9$-2$,-                                         ;
                                                                       9$-2$,-                                         ;
                                                                       9$-2$,-                                         ;
                                                                       3$-2$,-                                         ;
                                                                       11$-2$,-                                        ;
                                                                       13$-2$                                          ;
                   00048422  8F   DD 00080 3$:         PUSHL   #295970                                         ; 1804
                      67             01   FB 00086             CALLS   #1, LIB$SIGNAL                                  ;
                                     7E   11 00089             BRB     10$                                             ;
                      66        3C   A2   7D 0008B 4$:         MOVQ    60(R2), CUR_TIME                                ; 1809
              00000000G   EF         00   FB 0008F             CALLS   #0, PROCESS_DELETION                            ; 1810
                                     71   11 00096             BRB     10$                                             ; 1799
                      66        3C   A2   7D 00098 5$:         MOVQ    60(R2), CUR_TIME                                ; 1816
                   00A8   C6         20   D0 0009C             MOVL    #32, RSP                                        ; 1817
              00000000G   EF         00   FB 000A1             CALLS   #0, SNDSMB_SERVICE                              ; 1819
                                     70   11 000A8             BRB     12$                                             ;
                                     56   DD 000AA 6$:         PUSHL   R6                                              ; 1827
                      68             01   FB 000AC             CALLS   #1, SYS$GETTIM                                  ;
              00000000G   EF         00   FB 000AF             CALLS   #0, SYMBIONT_SERVICE                            ; 1828
                                     51   11 000B6             BRB     10$                                             ; 1799
                                     56   DD 000B8 7$:         PUSHL   R6                                              ; 1834
                      68             01   FB 000BA             CALLS   #1, SYS$GETTIM                                  ;
                           0106      C6   B5 000BD             TSTW    QUEUE_FAB+2                                     ; 1835
                                     77   13 000C1             BEQL    16$                                             ;
                      50        30   A6   D0 000C3             MOVL    MBX, R0                                         ; 1841
```

```
                       51      08  A0  9A 000C7        MOVZBL  8(R0), R1                              ;
       06      20      09  A0      51  2C 000CB        MOVC5   R1, 9(R0), #32, #6, SYSID              ;
                                   6E     000D1                                                       ;
          00000000G  EF            00  FB 000D2        CALLS   #0, LOCK_QUEUE_FILE                    ; 1842
                                   5E  DD 000D9        PUSHL   SP                                     ; 1843
          00000000G  EF            01  FB 000DB        CALLS   #1, WARM_START_EXISTING_FILE           ;
                     69            00  FB 000E2        CALLS   #0, UNLOCK_QUEUE_FILE                  ; 1844
                                   53  11 000E5        BRB     16$                                    ; 1799
                     66      3C    A2  7D 000E7 8$:    MOVQ    60(R2), CUR_TIME                       ; 1851
               00A8  C6            21  D0 000EB        MOVL    #33, RSP                               ; 1852
          00000000G  EF            00  FB 000F0        CALLS   #0, SNDACC_SERVICE                     ; 1854
               00AC  C6            50  D0 000F7        MOVL    R0, RSP+4                              ;
                                   35  11 000FC        BRB     15$                                    ; 1855
                     66      3C    A2  7D 000FE 9$:    MOVQ    60(R2), CUR_TIME                       ; 1861
          00000000G  EF            00  FB 00102        CALLS   #0, PROCESS_ACCOUNTING                 ; 1862
                                   2F  11 00109 10$:   BRB     16$                                    ; 1799
                     66      3C    A2  7D 0010B 11$:   MOVQ    60(R2), CUR_TIME                       ; 1868
                           00AA    C6  B4 0010F        CLRW    RSP+2                                  ; 1869
          00000000G  EF            00  FB 00113        CALLS   #0, SNDJBC_SERVICE                     ; 1870
                                   0F  11 0011A 12$:   BRB     14$                                    ;
                     66      3C    A2  7D 0011C 13$:   MOVQ    60(R2), CUR_TIME                       ; 1878
                           00AA    C6  B4 00120        CLRW    RSP+2                                  ; 1879
          00000000G  EF            00  FB 00124        CALLS   #0, GETQUI_SERVICE                     ; 1880
               00AC  C6            50  D0 0012B 14$:   MOVL    R0, RSP+4                              ;
                     69            00  FB 00130        CALLS   #0, UNLOCK_QUEUE_FILE                  ; 1881
          00000000G  EF            00  FB 00133 15$:   CALLS   #0, SEND_SERVICE_RESPONSE              ; 1882
               FE80  C6    FE50    C6  C0 0013A 16$:   ADDL2   DIAG_COUNT, DIAG_COUNT+48              ; 1889
               FE84  C6    FE54    C6  C0 00141        ADDL2   DIAG_COUNT+4, DIAG_COUNT+52            ; 1890
               FE88  C6    FE58    C6  C0 00148        ADDL2   DIAG_COUNT+8, DIAG_COUNT+56            ; 1891
               FE8C  C6    FE5C    C6  C0 0014F        ADDL2   DIAG_COUNT+12, DIAG_COUNT+60           ; 1892
               FE90  C6    FE60    C6  C0 00156        ADDL2   DIAG_COUNT+16, DIAG_COUNT+64           ; 1893
               FE94  C6    FE64    C6  C0 0015D        ADDL2   DIAG_COUNT+20, DIAG_COUNT+68           ; 1894
               FE98  C6    FE68    C6  C0 00164        ADDL2   DIAG_COUNT+24, DIAG_COUNT+72           ; 1895
               FE9C  C6    FE6C    C6  C0 0016B        ADDL2   DIAG_COUNT+28, DIAG_COUNT+76           ; 1896
                             2C    A6  D7 00172        DECL    MBX_MESSAGE_COUNT                      ; 1902
              36     F0  A6        01  E2 00175        BBSS    #1, FLAGS, 17$                         ; 1903
                                   7E  7C 0017A        CLRQ    -(SP)                                  ; 1915
                                   7E  7C 0017C        CLRQ    -(SP)                                  ;
                     7E    0400    8F  3C 0017E        MOVZWL  #1024, -(SP)                           ;
                           07F0    C6  9F 00183        PUSHAB  MBX_BUFFER                             ;
                                   7E  D4 00187        CLRL    -(SP)                                  ;
                           0000V   CF  9F 00189        PUSHAB  MAILBOX_AST                            ;
                           07E8    C6  9F 0018D        PUSHAB  MBX_IOSB                               ;
                                   31  DD 00191        PUSHL   #49                                    ;
                           07E4    C6  DD 00193        PUSHL   MBX_CHAN                               ;
                                   7E  D4 00197        CLRL    -(SP)                                  ;
          00000000G  00            0C  FB 00199        CALLS   #12, SYS$QIO                           ;
                     0D            50  E8 001A0        BLBS    STATUS, 17$                            ; 1916
                                   50  DD 001A3        PUSHL   STATUS                                 ;
                                   7E  D4 001A5        CLRL    -(SP)                                  ;
                        0004844C   8F  DD 001A7        PUSHL   #296012                                ;
                     67            03  FB 001AD        CALLS   #3, LIB$SIGNAL                         ;
                                       04 001B0 17$:   RET                                            ; 1918

; Routine Size:  433 bytes,    Routine Base:  CODE + 034E
```

```
ROUTINE MAILBOX_AST: NOVALUE=

!++

! FUNCTIONAL DESCRIPTION:
!       This is the completion AST routine for a mailbox read.  It processes
!       the message just read and queues another read.
!
! INPUT PARAMETERS:
!       NONE
!
! IMPLICIT INPUTS:
!       MBX_IOSB            - IOSB for read.
!       MBX_BUFFER          - Buffer containing data.
!
! OUTPUT PARAMETERS:
!       NONE
!
! IMPLICIT OUTPUTS:
!       NONE
!
! ROUTINE VALUE:
!       NONE
!
! SIDE EFFECTS:
!       NONE
!
!--

BEGIN
LOCAL
        STATUS;                                 ! Status return


IF NOT .MBX_IOSB[0]
THEN
    SIGNAL(JBC$_REAJBCMBX OR STS$K_ERROR, 0, .MBX_IOSB[0])
ELSE
    IF .MBX_BUFFER[MSG_W_TYPE] EQL MSG$_TRMUNSOLIC
    OR .MBX_BUFFER[MSG_W_TYPE] EQL MSG$_CRUNSOLIC
    THEN
        UNSOLICITED_INPUT()
    ELSE
        BEGIN
        SCHEDULE_NONAST(MAILBOX_NONAST, .MBX_IOSB[1] + 8, MBX_IOSB);
        MBX_MESSAGE_COUNT = .MBX_MESSAGE_COUNT + 1;
        END;


! Queue another read from the mailbox.
!
IF .MBX_MESSAGE_COUNT LSSU JBC$K_MAXBUFMSG
THEN
    BEGIN
    STATUS = $QIO(
        FUNC=IO$_READVBLK,
        CHAN=.MBX_CHAN,
```

```
;   702      P 1976  3           IOSB=MBX_IOSB,
;   703      P 1977  3           ASTADR=MAILBOX_AST,
;   704      P 1978  3           P1=MBX_BUFFER,
;   705      P 1979  3           P2=%ALLOCATION(MBX_BUFFER));
;   706        1980  3       IF NOT .STATUS THEN SIGNAL(JBC$_REAJBCMBX OR STS$K_SEVERE, 0, .STATUS);
;   707        1981  3       END
;   708        1982  2 ELSE
;   709        1983  2       FLAGS[FLAGS_V_READ_POSTED] = FALSE;
;   710        1984  1 END;


                                         000C 00000 MAILBOX_AST:
                                                          .WORD   Save R2,R3                          ; 1919
               53 00000000G     00    9E 00002           MOVAB   LIB$SIGNAL, R3
               52 00000000'     EF    9E 00009           MOVAB   MBX_IOSB, R2
               10               62    E8 00010           BLBS    MBX_IOSB, 1$                        ; 1953
               7E               62    3C 00013           MOVZWL  MBX_IOSB, -(SP)                     ; 1955
                                7E    D4 00016           CLRL    -(SP)
                  0004844A      8F    DD 00018           PUSHL   #296010
               63               03    FB 0001E           CALLS   #3, LIB$SIGNAL
                                2B    11 00021           BRB     4$
               01         08    A2    B1 00023 1$:       CMPW    MBX_BUFFER, #1                      ; 1957
                                06    13 00027           BEQL    2$
               02         08    A2    B1 00029           CMPW    MBX_BUFFER, #2                      ; 1958
                                09    12 0002D           BNEQ    3$
         00000000G EF                 00    FB 0002F 2$:       CALLS   #0, UNSOLICITED_INPUT           ; 1960
                                16    11 00036           BRB     4$
                                52    DD 00038 3$:       PUSHL   R2                                  ; 1963
               7E         02    A2    3C 0003A           MOVZWL  MBX_IOSB+2, -(SP)
               6E               08    C0 0003E           ADDL2   #8, (SP)
                        FE0A    CF    9F 00041           PUSHAB  MAILBOX_NONAST
            FDAA  CF                  03    FB 00045           CALLS   #3, SCHEDULE_NONAST
                        F844    C2    D6 0004A           INCL    MBX_MESSAGE_COUNT                   ; 1964
         00000064  8F           F844  C2    D1 0004E 4$:       CMPL    MBX_MESSAGE_COUNT, #100         ; 1970
                                32    1E 00057           BGEQU   5$
                                7E    7C 00059           CLRQ    -(SP)                               ; 1979
                                7E    7C 0005B           CLRQ    -(SP)
               7E       0400    8F    3C 0005D           MOVZWL  #1024, -(SP)
                          08    A2    9F 00062           PUSHAB  MBX_BUFFER
                                7E    D4 00065           CLRL    -(SP)
                          96    AF    9F 00067           PUSHAB  MAILBOX_AST
                                52    DD 0006A           PUSHL   R2
                                31    DD 0006C           PUSHL   #49
                          FC    A2    DD 0006E           PUSHL   MBX_CHAN
                                7E    D4 00071           CLRL    -(SP)
         00000000G 00                 0C    FB 00073           CALLS   #12, SYS$QIO
               13               50    E8 0007A           BLBS    STATUS, 6$                          ; 1980
                                50    DD 0007D           PUSHL   STATUS
                                7E    D4 0007F           CLRL    -(SP)
                  0004844C      8F    DD 00081           PUSHL   #296012
               63               03    FB 00087           CALLS   #3, LIB$SIGNAL
                                      04 0008A           RET                                         ; 1970
            F808  C2                  02    8A 0008B 5$:       BICB2   #2, FLAGS                       ; 1983
                                      04 00090 6$:       RET                                         ; 1984
```

```
; Routine Size:  145 bytes,    Routine Base:  CODE + 04FF
```

```
1 GLOBAL ROUTINE SIGNAL_FILE_ERROR(MESSAGE,FAB,FRAB): NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine handles signalling of file-related errors.
1 !
1 ! INPUT PARAMETERS:
1 !       MESSAGE              - Message code to be signalled.
1 !       FAB                  - FAB for the file, to obtain the file specification.
1 !       FRAB                 - FAB or RAB that sustained the error, to obtain
1 !                              STS and STV.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       Message signalled.
1 !
1 !--
1
2 BEGIN
2 MAP
2       FAB:            REF BBLOCK,     ! Pointer to FAB
2       FRAB:           REF BBLOCK;     ! Pointer to FAB or RAB
2 LOCAL
2       NAM:            REF BBLOCK,     ! Pointer to NAM block
2       DESC:           VECTOR[2];      ! Descriptor for file specification
2
2
2 ! Set up the file name descriptor.
2 !
2 NAM = .FAB[FAB$L_NAM];
2 IF .NAM[NAM$B_RSL] NEQ 0
2 THEN
3     BEGIN
3     DESC[0] = .NAM[NAM$B_RSL];
3     DESC[1] = .NAM[NAM$L_RSA];
3     END
2 ELSE IF .NAM[NAM$B_ESL] NEQ 0
2 THEN
3     BEGIN
3     DESC[0] = .NAM[NAM$B_ESL];
3     DESC[1] = .NAM[NAM$L_ESA];
3     END
2 ELSE
3     BEGIN
3     DESC[0] = .FAB[FAB$B_FNS];
```

```
;  769       2042  3      DESC[1] = .FAB[FAB$L_FNA];
;  770       2043  2      END;
;  771       2044  2
;  772       2045  2
;  773       2046  2   ! Signal the message.
;  774       2047  2   !
;  775       2048  2   SIGNAL(.MESSAGE, 1, DESC, .FRAB[FAB$L_STS], .FRAB[FAB$L_STV]);
;  776       2049  1   END;



                                       0000 00000       .ENTRY  SIGNAL_FILE_ERROR, Save nothing          ; 1985
                     5E           08     C2 00002       SUBL2   #8, SP                                   ;
                     51      08   AC     D0 00005       MOVL    FAB, R1                                  ; 2026
                     50      28   A1     D0 00009       MOVL    40(R1), NAM                              ;
                             03   A0     95 0000D       TSTB    3(NAM)                                   ; 2027
                                  0B     13 00010       BEQL    1$                                       ;
                     6E      03   A0     9A 00012       MOVZBL  3(NAM), DESC                             ; 2030
                04   AE      04   A0     D0 00016       MOVL    4(NAM), DESC+4                           ; 2031
                                  19     11 0001B       BRB     3$                                       ; 2027
                             0B   A0     95 0001D 1$:   TSTB    11(NAM)                                  ; 2033
                                  0B     13 00020       BEQL    2$                                       ;
                     6E      0B   A0     9A 00022       MOVZBL  11(NAM), DESC                            ; 2036
                04   AE      0C   A0     D0 00026       MOVL    12(NAM), DESC+4                          ; 2037
                                  09     11 0002B       BRB     3$                                       ; 2033
                     6E      34   A1     9A 0002D 2$:   MOVZBL  52(R1), DESC                             ; 2041
                04   AE      2C   A1     D0 00031       MOVL    44(R1), DESC+4                           ; 2042
                     50      0C   AC     D0 00036 3$:   MOVL    FRAB, R0                                 ; 2048
                     7E      08   A0     7D 0003A       MOVQ    8(R0), -(SP)                             ;
                             08   AE     9F 0003E       PUSHAB  DESC                                     ;
                                  01     DD 00041       PUSHL   #1                                       ;
                             04   AC     DD 00043       PUSHL   MESSAGE                                  ;
         00000000G   00           05     FB 00046       CALLS   #5, LIB$SIGNAL                           ;
                                         04 0004D       RET                                              ; 2049

; Routine Size:  78 bytes,    Routine Base:  CODE + 0590
```

```
1 ROUTINE MAIN_HANDLER_ACTION(MSG_DESC)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This is an action routine for the $PUTMSG that issues a signalled
1 !       message to the system console.  It writes the record to the operator
1 !       via OPCOM or via broadcast.
1 !
1 ! INPUT PARAMETERS:
1 !       MSG_DESC        - Descriptor for message.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       FALSE, to signal $PUTMSG not to write the message.
1 !
1 ! SIDE EFFECTS:
1 !       NONE
1 !
1 !--
1
2 BEGIN
2 MAP
2       MSG_DESC:       REF BBLOCK;     ! Descriptor for message text
2 LOCAL
2       LENGTH:         WORD,           ! Length of message, minimized
2       OPC_BUFFER:     BBLOCK[$BYTEOFFSET(OPC$L_MS_TEXT) + 132],
2                                       ! Buffer for OPCOM message
2       OPC_DESC:       VECTOR[2],      ! Descriptor for message buffer
2       STATUS;                         ! Status return
2
2
2 ! Set up the OPCOM message buffer.
2 !
2 OPC_BUFFER[OPC$B_MS_TYPE] = OPC$_RQ_RQST;
2 OPC_BUFFER[OPC$B_MS_TARGET] = OPC$M_NM_CENTRL;
2 OPC_BUFFER[OPC$W_MS_STATUS] = 0;
2 OPC_BUFFER[OPC$L_MS_RQSTID] = 0;
2 LENGTH = .MSG_DESC[DSC$W_LENGTH];
2 IF .LENGTH GTRU 132 THEN LENGTH = 132;
2 CH$MOVE(.LENGTH, .MSG_DESC[DSC$A_POINTER], OPC_BUFFER[OPC$L_MS_TEXT]);
2 OPC_DESC[0] = $BYTEOFFSET(OPC$L_MS_TEXT) + .LENGTH;
2 OPC_DESC[1] = OPC_BUFFER;
2
2
2 ! Try to send the message by OPCOM.  If this fails, send a broadcast to the
2 ! system console.
2 !
2 STATUS = $SNDOPR(MSGBUF=OPC_DESC);
```

```
;  835          2107  2 IF NOT .STATUS OR .STATUS EQL OPC$_NOPERATOR
;  836          2108  2 THEN
;  837        P 2109  2     $BRKTHRU(
;  838        P 2110  2         MSGBUF=.MSG_DESC,
;  839        P 2111  2         SENDTO=OPA0_DESC,
;  840        P 2112  2         SNDTYP=BRK$C_DEVICE,
;  841          2113  2         TIMOUT=10);
;  842          2114  2
;  843          2115  2
;  844          2116  2 ! Return FALSE, to signal $PUTMSG not to write the message.
;  845          2117  2 !
;  846          2118  2 FALSE
;  847          2119  1 END;


                                                        .EXTRN  SYS$SNDOPR, SYS$BRKTHRU

                                          00FC 00000 MAIN_HANDLER_ACTION:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7                   ; 2050
                          5E     FF6C  CE  9E 00002     MOVAB   -148(SP), SP                             ;
                  08      AE     0103  8F  3C 00007     MOVZWL  #259, OPC_BUFFER                         ; 2092
                                   0C  AE  D4 0000D     CLRL    OPC_BUFFER+4                             ; 2095
                          57       04  AC  D0 00010     MOVL    MSG_DESC, R7                             ; 2096
                          56           67  B0 00014     MOVW    (R7), LENGTH                             ;
                0084      8F           56  B1 00017     CMPW    LENGTH, #132                             ; 2097
                                       04  1B 0001C     BLEQU   1$                                       ;
                          56       84  8F  9B 0001E     MOVZBW  #132, LENGTH                             ;
        10   AE     04    B7           56  28 00022 1$: MOVC3   LENGTH, @4(R7), OPC_BUFFER+8             ; 2098
                          6E           56  3C 00028     MOVZWL  LENGTH, OPC_DESC                         ; 2099
                          6E           08  C0 0002B     ADDL2   #8, OPC_DESC                             ;
                  04      AE       08  AE  9E 0002E     MOVAB   OPC_BUFFER, OPC_DESC+4                   ; 2100
                                       7E  D4 00033     CLRL    -(SP)                                    ; 2106
                                   04  AE  9F 00035     PUSHAB  OPC_DESC                                 ;
           00000000G      00           02  FB 00038     CALLS   #2, SYS$SNDOPR                           ;
                          09           50  E9 0003F     BLBC    STATUS, 2$                               ; 2107
            00058061      8F           50  D1 00042     CMPL    STATUS, #360545                          ;
                                       1A  12 00049     BNEQ    3$                                       ;
                                       7E  7C 0004B 2$: CLRQ    -(SP)                                    ; 2113
                                       0A  DD 0004D     PUSHL   #10                                      ;
                                       7E  7C 0004F     CLRQ    -(SP)                                    ;
                                       20  DD 00051     PUSHL   #32                                      ;
                          7E           01  7D 00053     MOVQ    #1, -(SP)                                ;
                                 F9E0  CF  9F 00056     PUSHAB  OPA0_DESC                                ;
                                       57  DD 0005A     PUSHL   R7                                       ;
                                       7E  D4 0005C     CLRL    -(SP)                                    ;
           00000000G      00           0B  FB 0005E     CALLS   #11, SYS$BRKTHRU                         ;
                                       50  D4 00065 3$: CLRL    R0                                       ; 2119
                                           04 00067     RET                                              ;

; Routine Size:  104 bytes,    Routine Base:  CODE + 05DE
```

```
1 ROUTINE MAIN_HANDLER(SIG,MECH)=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine is the condition handler for the main routine.  It
1 !       intercepts signals and writes the message to the operator.
1 !
1 ! INPUT PARAMETERS:
1 !       Standard VMS condition handler parameters.
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       SS$_CONTINUE
1 !
1 ! SIDE EFFECTS:
1 !       If the condition is fatal, the image exits.
1 !
1 !--
1
2 BEGIN
2 MAP
2       SIG:            REF BBLOCK,     ! Signal parameters
2       MECH:           REF BBLOCK;     ! Mechanism parameters
2 LOCAL
2       OPC_BUFFER:     $BBLOCK[OPC$K_SHUTDOWN_MIN_SIZE],       ! Data structures
2       OPC_DESC:       VECTOR [2]                              !  to support
2                       PRESET ([0] = OPC$K_SHUTDOWN_MIN_SIZE,  !  cluster-wide
2                               [1] = OPC_BUFFER);              !  take down
2
2
2 ! Adjust the signal parameter count to remove the PC and PSL, and call $PUTMSG
2 ! to issue the message to the operator.
2 !
2 SIG[CHF$L_SIG_ARGS] = .SIG[CHF$L_SIG_ARGS] - 2;
2 $PUTMSG(MSGVEC=.SIG, ACTRTN=MAIN_HANDLER_ACTION);
2
2
2 ! If the condition is not a job controller message, force the severity to
2 ! fatal.  It must be a real system exception.
2 !
2 IF .BBLOCK[SIG[CHF$L_SIG_NAME], STS$V_FAC_NO] NEQ JBC$_FACILITY
2 THEN
2     BBLOCK[SIG[CHF$L_SIG_NAME], STS$V_SEVERITY] = STS$K_SEVERE;
2
2
2 ! Turn $ENQ/$DEQ errors reported via the JBC$_COMREMJBC code into fatal errors.
2 !
3 IF .SIG[CHF$L_SIG_NAME] EQL (JBC$_COMREMJBC OR STS$K_ERROR)
```

```
THEN
    BBLOCK[SIG[CHF$L_SIG_NAME], STS$V_SEVERITY] = STS$K_SEVERE;


! If the exception was fatal, exit the image.  Otherwise, signal the error and
! continue. Note that certain control flags can alter this behavior.
!
DIAG_TRACE[14] = .SIG[CHF$L_SIG_NAME];  ! ***** diagnostic info *****
IF .BBLOCK[SIG[CHF$L_SIG_NAME], STS$V_SEVERITY] EQL STS$K_SEVERE
THEN
    BEGIN


    ! For debugging purposes, conditionally force a synchronous cluster-wide
    ! crash.
    !
    IF .FLAGS[FLAGS_V_CLUSTER_SCRAM] THEN
        BEGIN
        OPC_BUFFER[OPC$B_RQSTCODE] = OPC$_X_SHUTDOWN;
        OPC_BUFFER[OPC$L_RQ_OPTIONS] = OPC$M_CLUSTER;
        IF(WORK_AREA[2] = $SNDOPR(MSGBUF=OPC_DESC))
        THEN WHILE TRUE DO
            BEGIN
            $HIBER;                               ! Wait for system to crash ...
            END;
        END;


    ! Delete all symbiont processes.
    !
    DELETE_SYMBIONTS();


    ! Close the accounting file.
    !
    CLOSE_ACCOUNTING_FILE();


    ! Close the system job queue file unless it is desired to leave it open
    ! for debugging purposes to preserve internal RMS data structures.
    !
    IF NOT .FLAGS[FLAGS_V_LEAVE_OPEN] THEN $CLOSE(FAB=QUEUE_FAB);


    ! Conditionally bug check the system instead of aborting/restarting the job
    ! controller.
    !
    IF .FLAGS[FLAGS_V_BUGCHECK] THEN $CMKRNL(ROUTIN=BUG_CHECK_SYSTEM);


    ! Make the process name null to avoid a name conflict with the new process.
    !
    $SETPRN();


    ! Create a new job controller process.
    !
```

```
;  963       P 2234  3         $CREPRC(
;  964       P 2235  3             IMAGE=$DESCRIPTOR('SYS$SYSTEM:JOBCTL.EXE'),
;  965       P 2236  3             INPUT=OPA0_DESC,
;  966       P 2237  3             OUTPUT=OPA0_DESC,
;  967       P 2238  3             ERROR=OPA0_DESC,
;  968       P 2239  3             PRVADR=JBC_PRIVILEGES,
;  969       P 2240  3             QUOTA=JBC_QUOTAS,
;  970       P 2241  3             PRCNAM=JOB_CONTROL_DESC,
;  971       P 2242  3             BASPRI=.JBC_PRIORITY,
;  972       P 2243  3             STSFLG=.IMAGE_DUMP_STSFLG,
;  973         2244  3             UIC=.JBC_UIC);
;  974         2245  3
;  975         2246  3
;  976         2247  3         ! Resignal the error to allow last chance handler to write the dump. Then
;  977         2248  3         ! the process will exit after the dump has been written. In addition,
;  978         2249  3         ! further display of the message will be inhibited so that it is displayed
;  979         2250  3         ! only once.
;  980         2251  3         !
;  981         2252  3         SIG[CHF$L_SIG_NAME] = .SIG[CHF$L_SIG_NAME] OR STS$M_INHIB_MSG;
;  982         2253  3         RETURN SS$_RESIGNAL
;  983         2254  2         END;
;  984         2255  2
;  985         2256  2
;  986         2257  2 SS$_CONTINUE
;  987         2258  1 END;


43 42 4F 4A 3A 4D 45 54 53 59 53 24 53 59 53  00646 P.AAR:  .ASCII  \SYS$SYSTEM:JOBCTL.EXE\                ;
                              45 58 45 2E 4C 54  00655                                                        ;
                                                 0065B          .BLKB   1
                                     00000015  0065C P.AAQ:  .LONG   21                                      ;
                                     00000000' 00660          .ADDRESS P.AAR                                  ;

                                                                .EXTRN  SYS$PUTMSG, SYS$CLOSE
                                                                .EXTRN  SYS$CREPRC

                                          001C 00000 MAIN_HANDLER:
                                                                .WORD   Save R2,R3,R4                      ; 2120
                 54        F9AE   CF  9E 00002          MOVAB   OPA0_DESC, R4                              ;
                 53   00000000'   EF  9E 00007          MOVAB   FLAGS, R3                                  ;
                 5E               20  C2 0000E          SUBL2   #32, SP                                    ;
                                  1A  DD 00011          PUSHL   #26                                        ; 2156
           04    AE          08   AE  9E 00013          MOVAB   OPC_BUFFER, OPC_DESC+4                     ;
                 52          04   AC  D0 00018          MOVL    SIG, R2                                    ; 2162
                 62               02  C2 0001C          SUBL2   #2, (R2)                                   ;
                                  7E  7C 0001F          CLRQ    -(SP)                                      ; 2163
                           FF55   CF  9F 00021          PUSHAB  MAIN_HANDLER_ACTION                        ;
                                  52  DD 00025          PUSHL   R2                                         ;
        00000000G  00             04  FB 00027          CALLS   #4, SYS$PUTMSG                             ;
                   52             04  C0 0002E          ADDL2   #4, R2                                     ; 2169
04       02  A2    0C             00  ED 00031          CMPZV   #0, #12, 2(R2), #4                         ;
                                  05  13 00037          BEQL    1$                                         ;
62           03    00             04  F0 00039          INSV    #4, #0, #3, (R2)                           ; 2171
        00048412   8F             62  D1 0003E 1$:      CMPL    (R2), #295954                              ; 2176
                                  05  12 00045          BNEQ    2$                                         ;
62           03    00             04  F0 00047          INSV    #4, #0, #3, (R2)                           ; 2178
```

```
                       FE38  C3      62  D0 0004C 2$:    MOVL    (R2), DIAG_TRACE+56                    ; 2184
         04      62          03      00  ED 00051        CMPZV   #0, #3, (R2), #4                       ; 2185
                                     03  13 00056        BEQL    3$                                     ;
                                   009A  31 00058        BRW     8$                                     ;
                 25     02   A3      01  E1 0005B 3$:    BBC     #1, FLAGS+2, 5$                        ; 2193
                        08   AE      10  90 00060        MOVB    #16, OPC_BUFFER                        ; 2195
                        0E   AE      01  D0 00064        MOVL    #1, OPC_BUFFER+6                       ; 2196
                                     7E  D4 00068        CLRL    -(SP)                                  ; 2197
                             04      AE  9F 0006A        PUSHAB  OPC_DESC                               ;
                  00000000G  00      02  FB 0006D        CALLS   #2, SYS$SNDOPR                         ;
                       FECC  C3      50  D0 00074        MOVL    R0, WORK_AREA+8                        ;
                             09      50  E9 00079        BLBC    R0, 5$                                 ;
                  00000000G  00      00  FB 0007C 4$:    CALLS   #0, SYS$HIBER                          ; 2199
                                     F7  11 00083        BRB     4$                                     ; 2198
                  00000000G  EF      00  FB 00085 5$:    CALLS   #0, DELETE_SYMBIONTS                   ; 2207
                  00000000G  EF      00  FB 0008C        CALLS   #0, CLOSE_ACCOUNTING_FILE              ; 2212
                 0B     02   A3      02  E0 00093        BBS     #2, FLAGS+2, 6$                        ; 2218
                           0114      C3  9F 00098        PUSHAB  QUEUE_FAB                              ;
                  00000000G  00      01  FB 0009C        CALLS   #1, SYS$CLOSE                          ;
                 0D     02   A3      03  E1 000A3 6$:    BBC     #3, FLAGS+2, 7$                        ; 2224
                                     7E  D4 000A8        CLRL    -(SP)                                  ;
                          0000V      CF  9F 000AA        PUSHAB  BUG_CHECK_SYSTEM                       ;
                  00000000G  00      02  FB 000AE        CALLS   #2, SYS$CMKRNL                         ;
                                     7E  D4 000B5 7$:    CLRL    -(SP)                                  ; 2229
                  00000000G  00      01  FB 000B7        CALLS   #1, SYS$SETPRN                         ;
                                     7E  D4 000BE        CLRL    -(SP)                                  ; 2244
                             04      A3  DD 000C0        PUSHL   IMAGE_DUMP_STSFLG                      ;
                                     7E  D4 000C3        CLRL    -(SP)                                  ;
                           0110      C3  DD 000C5        PUSHL   JBC_UIC                                ;
                           00C0      C3  DD 000C9        PUSHL   JBC_PRIORITY                           ;
                             2C      A4  9F 000CD        PUSHAB  JOB_CONTROL_DESC                       ;
                           00CC      C3  9F 000D0        PUSHAB  JBC_QUOTAS                             ;
                           00C4      C3  9F 000D4        PUSHAB  JBC_PRIVILEGES                         ;
                                     54  DD 000D8        PUSHL   R4                                     ;
                                     54  DD 000DA        PUSHL   R4                                     ;
                                     54  DD 000DC        PUSHL   R4                                     ;
                           0644      C4  9F 000DE        PUSHAB  P.AAQ                                  ;
                                     7E  D4 000E2        CLRL    -(SP)                                  ;
                  00000000G  00      0D  FB 000E4        CALLS   #13, SYS$CREPRC                        ;
                        03   A2      10  88 000EB        BISB2   #16, 3(R2)                             ; 2252
                             50    0918  8F  3C 000EF    MOVZWL  #2328, R0                              ; 2253
                                         04 000F4        RET                                            ;
                             50      01  D0 000F5 8$:    MOVL    #1, R0                                 ; 2258
                                         04 000F8        RET                                            ;

; Routine Size:  249 bytes,    Routine Base:  CODE + 0664
```

```
1 ROUTINE BUG_CHECK_SYSTEM: NOVALUE=
1
1 !++
1 !
1 ! FUNCTIONAL DESCRIPTION:
1 !       This routine crashes the system via a bug check.
1 !
1 ! INPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !       NONE
1 !
1 ! OUTPUT PARAMETERS:
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !       NONE
1 !
1 ! ROUTINE VALUE:
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !       The system crashes!!!
1 !
1 !--
1
2 BEGIN
2 BUG_CHECK(UNXINTEXC);
1 END;
```

```
                                .EXTRN  BUGS_UNXINTEXC

      0000 00000 BUG_CHECK_SYSTEM:
                                .WORD   Save nothing                    ; 2259
      FEFF 00002                BUGW                                    ; 2287
     0000* 00004                .WORD   <BUG$_UNXINTEXC!4>              ;
        04 00006                RET                                     ; 2288

; Routine Size:  7 bytes,    Routine Base:  CODE + 075D
```

```
; 1020          2289  1 END
; 1021          2290  0 ELUDOM



                                                                 .EXTRN  LIB$SIGNAL

;                                      PSECT SUMMARY
;
;       Name                        Bytes                       Attributes
;
;  COMMON                              5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                                1892  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                               Library Statistics
;
;                                          -------- Symbols --------      Pages      Processing
;       File                               Total    Loaded   Percent      Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619      150         0       1000        00:01.4



;                                     COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CONTROL/OBJ=OBJ$:CONTROL MSRC$:CONTROL/UPDATE=(ENH$:CONTROL)

; Size:           1542 code + 5374 data bytes
; Run Time:          00:30.8
; Elapsed Time:      01:47.7
; Lines/CPU Min:     4458
; Lexemes/CPU-Min: 48716
; Memory Used:  348 pages
; Compilation Complete
```

0191 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BATCH
LIS

BROADCAST
LIS

BUFFERS
LIS

CONTROL
LIS

CHECKPROT
LIS

ASYNCHRON
LIS

0192 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

GETQUI
LIS

DELETION
LIS

EXECUTOR
LIS

INITQUEUE
LIS