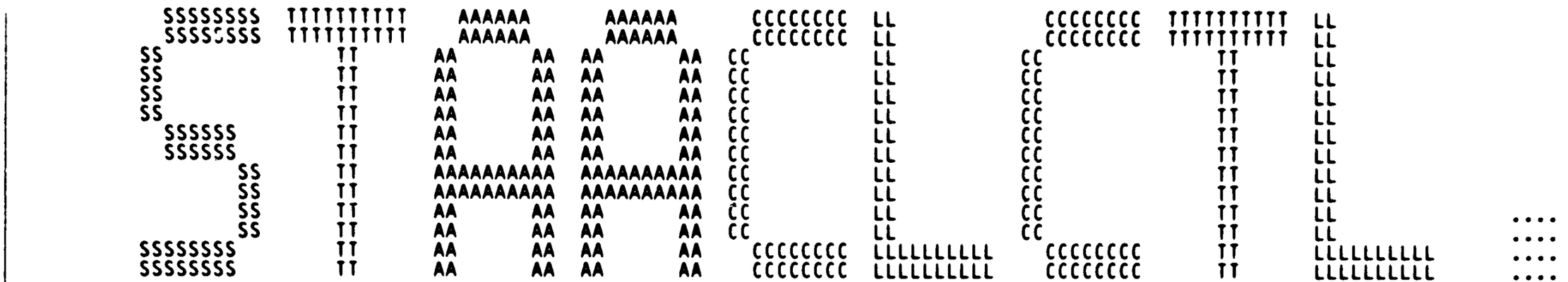
```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK         KKK  UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBB         BBB  AAA         AAA  CCC            KKK       KKK    UUU          UUU  PPP         PPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB     AAA         AAA  CCC            KKKKKKKKK        UUU          UUU  PPPPPPPPPPPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAAAAAAAAAAAAAA  CCC            KKK       KKK    UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBB         BBB  AAA         AAA  CCC            KKK          KKK  UUU          UUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
BBBBBBBBBBBB     AAA         AAA     CCCCCCCCCCC  KKK          KKK  UUUUUUUUUUUUUUUU  PPP
```


SSSSSSSS TTTTTTTTTT AAAAAA AAAAAA CCCCCCCC LL CCCCCCCC TTTTTTTTTT LL
SSSSSSSS TTTTTTTTTT AAAAAA AAAAAA CCCCCCCC LL CCCCCCCC TTTTTTTTTT LL
SS TT AA AA AA AA CC LL CC TT LL
SS TT AA AA AA AA CC LL CC TT LL
SS TT AA AA AA AA CC LL CC TT LL
SS TT AA AA AA AA CC LL CC TT LL
SSSSSS TT AA AA AA AA CC LL CC TT LL
SSSSSS TT AA AA AA AA CC LL CC TT LL
SS TT AAAAAAAAAA AAAAAAAAAA CC LL CC TT LL
SS TT AAAAAAAAAA AAAAAAAAAA CC LL CC TT LL
SS TT AA AA AA AA CC LL CC TT LL ....
SS TT AA AA AA AA CC LL CC TT LL ....
SSSSSSSS TT AA AA AA AA CCCCCCCC LLLLLLLLLL CCCCCCCC TT LLLLLLLLLL ....
SSSSSSSS TT AA AA AA AA CCCCCCCC LLLLLLLLLL CCCCCCCC TT 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

```
;     1          0001  0 MODULE STAACLCTL (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE         *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER    *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    18          0018  1 !*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY      *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT       *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS    *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:     BACKUP/RESTORE
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module contains the routines to manipulate the Access
;    38          0038  1 !       Control Lists and Access Control Entries.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       VAX/VMS user mode utilities.
;    43          0043  1 !
;    44          0044  1 !--
;    45          0045  1 !
;    46          0046  1 !
;    47          0047  1 ! AUTHOR:       L. Mark Pilant,         CREATION DATE:  3-Nov-1982  11:35
;    48          0048  1 !
;    49          0049  1 ! MODIFIED BY:
;    50          0050  1 !
;    51          0051  1 !       V03-008 LMP0272         L. Mark Pilant,         3-Jul-1984  10:32
;    52          0052  1 !               Change routines to use the common ACL subroutines where
;    53          0053  1 !               possible.
;    54          0054  1 !
;    55          0055  1 !       V03-006 ACG0415         Andrew C. Goldstein,    24-Apr-1984  18:42
;    56          0056  1 !               Misc ACL processing cleanups
;    57          0057  1 !
```

```
;   58          0058  1 ;        V03-005 ACG0382         Andrew C. Goldstein,    16-Dec-1983  17:08
;   59          0059  1 ;                Fix RVN usage in reading extension headers
;   60          0060  1 ;
;   61          0061  1 ;        V03-004 ACG0365         Andrew C. Goldstein,    11-Oct-1983  14:55
;   62          0062  1 ;                Return correct error status on failures
;   63          0063  1 ;
;   64          0064  1 ;        V03-003 ACG0313         Andrew C. Goldstein,    12-Feb-1983  17:07
;   65          0065  1 ;                Add routine subtitles
;   66          0066  1 ;
;   67          0067  1 ;        V03-002 LMP0067         L. Mark Pilant,         15-Dec-1982  15:19
;   68          0068  1 ;                Deallocate memory obtained for ACL segments when the file
;   69          0069  1 ;                is deaccessed.
;   70          0070  1 ;
;   71          0071  1 ;
;   72          0072  1 ;**
```

```
;    74          0073  1 REQUIRE 'SRC$:COMMON';
;    75          1179  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    76          1180  1 REQUIRE 'LIB$:BACKDEF';
;    77          1630  1
;    78          1631  1
;    79          1632  1 FORWARD ROUTINE
;    80          1633  1         ACL_DISPATCH,                  ! ACL dispatch routine
;    81          1634  1         ACL_BUILDACL,                  ! Build the ACL structures
;    82          1635  1         ALLOC_PAGED,                   ! Jacket routine for ACLSUBR
;    83          1636  1         DALLOC_PAGED;                  ! Jacket routine for ACLSUBR
;    84          1637  1
;    85          1638  1 EXTERNAL ROUTINE
;    86          1639  1         ACL_ADDENTRY,                  ! Add an Access Control Entry
;    87          1640  1         ACL_DELETEACL,                 ! Delete the entire ACL
;    88          1641  1         ACL_READACL,                   ! Read the entire ACL
;    89          1642  1         ACL_ACLLENGTH,                 ! Return the length of the ACL
;    90          1643  1         GET_ZERO_VM,                   ! Allocate & clear memory
;    91          1644  1         FREE_VM,                       ! Return allocated memory
;    92          1645  1         READ_HEADER,                   ! Read header specified by FCB
;    93          1646  1         SWITCH_VOLUME;                 ! Set the corrent volume
;    94          1647  1
;    95          1648  1 G$DEFINE ();                           ! Define global common area
;    96          1649  1
;    97          1650  1 BUILTIN
;    98          1651  1         INSQUE,
;    99          1652  1         REMQUE;
```

```
;  101          1653  1 %SBTTL 'ACL_DISPATCH - dispatch ACL operation'
;  102          1654  1 GLOBAL ROUTINE ACL_DISPATCH (CODE, COUNT, ACE, FIB) =
;  103          1655  1
;  104          1656  1 !++
;  105          1657  1 !
;  106          1658  1 ! FUNCTIONAL DESCRIPTION:
;  107          1659  1 !
;  108          1660  1 !       This routine is called whenever any ACL processing is to be done.
;  109          1661  1 !       The code is checked for validity and if valid, the appropriate routine
;  110          1662  1 !       is called to manipulate the ACL's and ACE's.
;  111          1663  1 !
;  112          1664  1 ! CALLING SEQUENCE:
;  113          1665  1 !       ACL_DISPATCH (ARG1, ARG2, ARG3, ARG4)
;  114          1666  1 !
;  115          1667  1 ! INPUT PARAMETERS:
;  116          1668  1 !       ARG1: attribute code to determine the action to perform
;  117          1669  1 !       ARG2: size of the attribute
;  118          1670  1 !       ARG3: address of the user Access Control Entry
;  119          1671  1 !       ARG4: address of the FIB
;  120          1672  1 !
;  121          1673  1 ! IMPLICIT INPUTS:
;  122          1674  1 !       CURRENT_MTL: address of the current file info block
;  123          1675  1 !
;  124          1676  1 ! OUTPUT PARAMETERS:
;  125          1677  1 !       none
;  126          1678  1 !
;  127          1679  1 ! IMPLICIT OUTPUTS:
;  128          1680  1 !       none
;  129          1681  1 !
;  130          1682  1 ! ROUTINE VALUE:
;  131          1683  1 !       1 if successful
;  132          1684  1 !       0 otherwise
;  133          1685  1 !
;  134          1686  1 ! SIDE EFFECTS:
;  135          1687  1 !       Appropriate action routine called, possible header modification may
;  136          1688  1 !       result.
;  137          1689  1 !
;  138          1690  1 !--
;  139          1691  1
;  140          1692  2 BEGIN
;  141          1693  2
;  142          1694  2 MAP
;  143          1695  2         ACE             : REF BBLOCK,           ! Address of the user ACE
;  144          1696  2         FIB             : REF BBLOCK;           ! Address of the FIB
;  145          1697  2
;  146          1698  2 LOCAL
;  147          1699  2         ACL_CONTEXT,                            ! Address of ACL context cell
;  148          1700  2         DUMMY,                                  ! Throw away cell
;  149          1701  2         LOCAL_STATUS;                           ! Status of attribute processing
;  150          1702  2
;  151          1703  2 ! Switch context to the volume of the specified RVN.
;  152          1704  2
;  153          1705  2 SWITCH_VOLUME (.CURRENT_MTL[MTL_FID_RVN]);
;  154          1706  2
;  155          1707  2 ! Determine where the ACL context is to go.
;  156          1708  2
;  157          1709  2 DUMMY = 0;
```

```
;  158          1710  2 IF .FIB NEQ 0
;  159          1711  2 THEN ACL_CONTEXT = FIB[FIB$L_ACLCTX]
;  160          1712  2 ELSE ACL_CONTEXT = DUMMY;
;  161          1713  2
;  162          1714  2 ! Dispatch to the appropriate action routine based upon the attribute code.
;  163          1715  2
;  164          1716  3 LOCAL_STATUS = (
;  165          1717  3     CASE .CODE FROM ATR$C_ADDACLENT TO ATR$C_READACE OF
;  166          1718  3     SET
;  167          1719  3         [ATR$C_ADDACLENT]:
;  168          1720  4             BEGIN
;  169          1721  4             CURRENT_MTL[MTL_NEW_ACL] = TRUE;
;  170          1722  4             DUMMY = %X'00FFFFFF';
;  171          1723  4             ACL_ADDENTRY (CURRENT_MTL[MTL_ACLFL], DUMMY, .COUNT, .ACE)
;  172          1724  3             END;
;  173          1725  3
;  174          1726  3         [ATR$C_DELETEACL]:
;  175          1727  4             BEGIN
;  176          1728  4             IF .CURRENT_MTL[MTL_ACLFL] EQL 0
;  177          1729  4             THEN
;  178          1730  5                 BEGIN
;  179          1731  5                 CURRENT_MTL[MTL_ACLFL] = CURRENT_MTL[MTL_ACLFL];
;  180          1732  5                 CURRENT_MTL[MTL_ACLBL] = CURRENT_MTL[MTL_ACLFL];
;  181          1733  4                 END;
;  182          1734  4             CURRENT_MTL[MTL_NEW_ACL] = TRUE;
;  183          1735  4             ACL_DELETEACL (CURRENT_MTL[MTL_ACLFL], 0)
;  184          1736  3             END;
;  185          1737  3
;  186          1738  3         [ATR$C_READACL]:
;  187          1739  4             BEGIN
;  188          1740  4             IF .CURRENT_MTL[MTL_ACLFL] EQL CURRENT_MTL[MTL_ACLFL]
;  189          1741  4             THEN
;  190          1742  5                 BEGIN
;  191          1743  5                 CH$FILL (0, .COUNT, .ACE);
;  192          1744  5                 RETURN 1;
;  193          1745  4                 END;
;  194          1746  4             ACL_READACL (CURRENT_MTL[MTL_ACLFL], .ACL_CONTEXT, .COUNT, .ACE)
;  195          1747  3             END;
;  196          1748  3
;  197          1749  3         [ATR$C_ACLLENGTH]:
;  198          1750  3             ACL_ACLLENGTH (CURRENT_MTL[MTL_ACLFL], .ACL_CONTEXT, .COUNT, .ACE);
;  199          1751  3
;  200          1752  3         [INRANGE,OUTRANGE]:     SS$_BADATTRIB;
;  201          1753  3
;  202          1754  2     TES);
;  203          1755  2
;  204          1756  2 IF .FIB NEQ 0 THEN FIB[FIB$L_ACL_STATUS] = .LOCAL_STATUS;
;  205          1757  2
;  206          1758  2 RETURN 1;
;  207          1759  2
;  208          1760  1 END;                                    ! End of routine ACL_DISPATCH

                                                                .TITLE  STAACLCTL
                                                                .IDENT  \V04-000\

                                                                .PSECT  COMMON,NOEXE,  OVR,2
```

```
00000 GLOBAL_BASE:
                .BLKB   0
00000 FREE_LIST:
                .BLKB   8
00008 INPUT_WAIT:
                .BLKB   8
00010 REREAD_WAIT:
                .BLKB   8
00018 OUTPUT_WAIT:
                .BLKB   8
00020 JPI_UIC:.BLKB     4
00024 JPI_USERNAME:
                .BLKB   12
00030 JPI_DATE:
                .BLKB   8
00038 JPI_NODE_DESC:
                .BLKB   8
00040 JPI_CURPRIV:
                .BLKB   8
00048 SYI_VERSION:
                .BLKB   4
0004C SYI_SID:.BLKB     4
00050 RWSV_HOLD_LIST:
                .BLKB   8
00058 RWSV_CRC16:
                .BLKB   64
00098 RWSV_AUTODIN:
                .BLKB   64
000D8 RWSV_FILESET_ID:
                .BLKB   8
000E0 RWSV_VOLUME_ID:
                .BLKB   12
000EC RWSV_VOL_NUMBER:
                .BLKB   2
000EE RWSV_SEG_NUMBER:
                .BLKB   2
000F0 RWSV_FILE_NUMBER:
                .BLKB   4
000F4 RWSV_SAVE_QUAL:
                .BLKB   4
000F8 RWSV_SAVE_FAB.
                .BLKB   4
000FC RWSV_CHAN:
                .BLKB   4
00100 RWSV_XOR_BCC:
                .BLKB   4
00104 RWSV_IN_SEQ:
                .BLKB   4
00108 RWSV_IN_SEQ_):
                .BLKB   4
0010C RWSV_IN_XOR_SEQ:
                .BLKB   4
00110 RWSV_IN_XOR_RFA:
                .BLKB   6
00116 RWSV_LOOKAHEAD:
                .BLKB   1
```

```
00117 RWSV_XORSIZE:
                .BLKB   1
00118 RWSV_IN_GROUP_SIZE:
                .BLKB   4
0011C RWSV_IN_ERRORS:
                .BLKB   2
0011E RWSV_IN_XORUSE:
                .BLKB   2
00120 RWSV_IN_ORGERR:
                .BLKB   8
00128 RWSV_IN_VBN:
                .BLKB   4
0012C RWSV_IN_VBN_0:
                .BLKB   4
00130 RWSV_ALLOC:
                .BLKB   4
00134 RWSV_EOF:
                .BLKB   4
00138 RWSV_OUT_SEQ:
                .BLKB   4
0013C RWSV_OUT_VBN:
                .BLKB   4
00140 RWSV_OUT_BLOCK_COUNT:
                .BLKB   4
00144 RWSV_OUT_ERRORS:
                .BLKB   2
00146 RWSV_SEQ_ERRORS:
                .BLKB   2
00148 RWSV_OUT_GROUP_COUNT:
                .BLKB   1
00149 RWSV_PADDING:
                .BLKB   3
0014C QUAL:     .BLKB   112
001BC COM_SSNAME:
                .BLKB   8
001C4 COM_VALID_TYPES:
                .BLKB   2
001C6 COM_FLAGS:
                .BLKB   2
001C8 COM_PADDING:
                .BLKB   1
001C9 COM_BUFF_COUNT:
                .BLKB   1
001CA COM_I_SETCOUNT:
                .BLKB   1
001CB COM_O_SETCOUNT:
                .BLKB   1
001CC COM_I_STRUCNAME:
                .BLKB   12
001D8 COM_O_STRUCNAME:
                .BLKB   12
001E4 COM_O_BSRDATE:
                .BLKB   8
001EC ALT_SSNAME:
                .BLKB   32
0020C INPUT_FUNC:
                .BLKB   1
```

```
0020D INPUT_RTYPE:
                .BLKB   1
0020E OUTPUT_FUNC:
                .BLKB   1
0020F FAST_STRUCLEV:
                .BLKB   1
00210 INPUT_BEG:
                .BLKB   0
00210 INPUT_CHAN:
                .BLKB   4
00214 INPUT_FLAGS:
                .BLKB   2
00216 INPUT_PADDING:
                .BLKB   2
00218 INPUT_FAB:
                .BLKB   4
0021C INPUT_NAM:
                .BLKB   4
00220 INPUT_BCB:
                .BLKB   4
00224 INPUT_QUAL:
                .BLKB   4
00228 INPUT_BAD:
                .BLKB   4
0022C INPUT_BLOCK:
                .BLKB   4
00230 INPUT_MAXBLOCK:
                .BLKB   4
00234 INPUT_MEDIA_ID:
                .BLKB   4
00238 INPUT_NAMEDESC:
                .BLKB   8
00240 INPUT_STATBLK:
                .BLKB   8
00248 INPUT_HDR_BEG:
                .BLKB   0
00248 INPUT_CREDATE:
                .BLKB   8
00250 INPUT_REVDATE:
                .BLKB   8
00258 INPUT_EXPDATE:
                .BLKB   8
00260 INPUT_BAKDATE:
                .BLKB   8
00268 INPUT_FILEOWNER:
                .BLKB   4
0026C INPUT_FILECHAR:
                .BLKB   4
00270 INPUT_RECATTR:
                .BLKB   32
00290 INPUT_HDR_END:
                .BLKB   0
00290 INPUT_END:
                .BLKB   0
00290 INPUT_PROC_LIST:
                .BLKB   4
00294 INPUT_PLACEMENT:
```

```
                        .BLKB   8
0029C INPUT_VBN_LIST:
                        .BLKB   8
002A4 INPUT_PLACE_LEN:
                        .BLKB   2
002A6 INPUT_PADDING_2:
                        .BLKB   2
002A8 OUTPUT_BEG:
                        .BLKB   0
002A8 OUTPUT_CHAN:
                        .BLKB   4
002AC OUTPUT_FLAGS:
                        .BLKB   2
002AE OUTPUT_PADDING:
                        .BLKB   2
002B0 OUTPUT_FAB:
                        .BLKB   4
002B4 OUTPUT_NAM:
                        .BLKB   4
002B8 OUTPUT_BCB:
                        .BLKB   4
002BC OUTPUT_QUAL:
                        .BLKB   4
002C0 OUTPUT_BAD:
                        .BLKB   4
002C4 OUTPUT_BLOCK:
                        .BLKB   4
002C8 OUTPUT_MAXBLOCK:
                        .BLKB   4
002CC OUTPUT_DEVGEOM:
                        .BLKB   8
002D4 OUTPUT_ATTBUF:
                        .BLKB   144
00364 OUTPUT_END:
                        .BLKB   0
00364 LIST_TOTFILES:
                        .BLKB   4
00368 LIST_TOTSIZE:
                        .BLKB   4
0036C VERIFY_FAB:
                        .BLKB   4
00370 VERIFY_USE_COUNT:
                        .BLKB   4
00374 VERIFY_QUAL:
                        .BLKB   4
00378 COMPARE_BCB:
                        .BLKB   4
0037C FAST_BUFFER:
                        .BLKB   4
00380 FAST_BUFFER_SIZE:
                        .BLKB   4
00384 FAST_RVN:
                        .BLKB   1
00385 FAST_PADDING:
                        .BLKB   1
00386 DIR_VERLIMIT:
                        .BLKB   2
```

```
00388 FAST_VOL_BEG:
                .BLKB   0
00388 FAST_IMAP_SIZE:
                .BLKB   4
0038C FAST_IMAP:
                .BLKB   4
00390 FAST_HDR_OFFSET:
                .BLKB   4
00394 FAST_BOOT_LBN:
                .BLKB   4
00398 FAST_VOL_END:
                .BLKB   0
00398 JOUR_BUFFER:
                .BLKB   4
0039C JOUR_DIR:
                .BLKB   4
003A0 JOUR_HIBLK:
                .BLKB   4
003A4 JOUR_EFBLK:
                .BLKB   4
003A8 JOUR_INBLK:
                .BLKB   4
003AC JOUR_FFBYTE:
                .BLKB   2
003AE JOUR_INBYTE:
                .BLKB   2
003B0 JOUR_STRUCT_LEV:
                .BLKB   2
003B2 JOUR_COUNT:
                .BLKB   1
003B3 JOUR_REVERSE:
                .BLKB   1
003B4 JOUR_EXSZ:
                .BLKB   2
003B6 JOUR_PADDING:
                .BLKB   2
003B8 CHKPT_HIGH_SP:
                .BLKB   4
003BC CHKPT_LOW_SP:
                .BLKB   4
003C0 CHKPT_STACK:
                .BLKB   4
003C4 CHKPT_VARS:
                .BLKB   4
003C8 CHKPT_STATUS:
                .BLKB   4
003CC DIR_BEG:.BLKB     0
003CC DIR_CHAN:
                .BLKB   4
003D0 DIR_NAM:.BLKB     4
003D4 DIR_DEV_DESC:
                .BLKB   4
003D8 DIR_SEL_DIR:
                .BLKB   8
003E0 DIR_SEL_NIV:
                .BLKB   8
003E8 DIR_STRUCLEV:
```

```
                        .BLKB   1
003E9 DIR_LEVELS:
                        .BLKB   1
003EA DIR_FLAGS:
                        .BLKB   1
003EB DIR_STATUS:
                        .BLKB   1
003EC DIR_STRING:
                        .BLKB   320
0052C DIR_STACK:
                        .BLKB   612
00790 DIR_SP: .BLKB     4
00794 DIR_SEL_LATEST:
                        .BLKB   4
00798 DIR_END:.BLKB     0
00798 DIR_SCANLIMIT:
                        .BLKB   36
007BC INPUT_MTL:
                        .BLKB   4
007C0 OUTPUT_MTL:
                        .BLKB   4
007C4 CURRENT_MTL:
                        .BLKB   4
007C8 CURRENT_VCB:
                        .BLKB   4
007CC CURRENT_WCB:
                        .BLKB   4
007D0 ACL_FIB_DESCR:
                        .BLKB   8
007D8 ACL_FIB:.BLKB     64
00818 ACL_LENGTH:
                        .BLKB   4
0081C ACL_BUFFER:
                        .BLKB   4
00820 CRYP_IN_CONTEXT:
                        .BLKB   4
00824 CRYP_OU_CONTEXT:
                        .BLKB   4
00828 CRYP_DA_CONTEXT:
                        .BLKB   4
0082C CRYP_DATA_ENCIV:
                        .BLKB   8
00834 CRYP_DATA_CODE:
                        .BLKB   4
00838 CRYP_DATA_KEY:
                        .BLKB   8
00840 CRYP_DATA_IV:
                        .BLKB   8
00848 CRYP_DATA_CKSM:
                        .BLKB   4

                        .EXTRN  ACL_ADDENTRY, ACL_DELETEACL
                        .EXTRN  ACL_READACL, ACL_ACLLENGTH
                        .EXTRN  GET_ZERO_VM, FREE_VM
                        .EXTRN  READ_HEADER, SWITCH_VOLUME

                        .PSECT  CODE,NOWRT,2
```

```
                                                 0FFC 00000          .ENTRY  ACL_DISPATCH, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1654
                                                                             R10,R11
                                          5B 00000000'   EF  9E 00002        MOVAB   CURRENT_MTL, R11                    ;
                                          5E             04  C2 00009        SUBL2   #4, SP                              ;
                                          50             6B  D0 0000C        MOVL    CURRENT_MTL, R0                     ; 1705
                                          7E       1C    A0  9A 0000F        MOVZBL  28(R0), -(SP)                       ;
                                00000000G 00             01  FB 00013        CALLS   #1, SWITCH_VOLUME                   ;
                                                         6E  D4 0001A        CLRL    DUMMY                               ; 1709
                                          58       10    AC  D0 0001C        MOVL    FIB, R8                             ; 1710
                                                         59  D4 00020        CLRL    R9                                  ;
                                                         58  D5 00022        TSTL    R8                                  ;
                                                         08  13 00024        BEQL    1$                                  ;
                                                         59  D6 00026        INCL    R9                                  ;
                                          56       30    A8  9E 00028        MOVAB   48(R8), ACL_CONTEXT                 ; 1711
                                                         03  11 0002C        BRB     2$                                  ;
                                          56             6E  9E 0002E 1$:    MOVAB   DUMMY, ACL_CONTEXT                  ; 1712
                                    08    1F       04    AC  CF 00031 2$:    CASEL   CODE, #31, #8                       ; 1717
                0012          0012          0012          0018       00036 3$:    .WORD   5$-3$,-                             ;
                0085          005C          0039          0012       0003E        4$-3$,-                             ;
                                                          0012       00046        4$-3$,-                             ;
                                                                                  4$-3$,-                             ;
                                                                                  4$-3$,-                             ;
                                                                                  6$-3$,-                             ;
                                                                                  8$-3$,-                             ;
                                                                                  10$-3$,-                            ;
                                                                                  4$-3$                               ;
                                          5A                 34  D0 00048 4$:    MOVL    #52, LOCAL_STATUS           ;
                                                           0081  31 0004B        BRW     12$                         ;
                                          50                 6B  D0 0004E 5$:    MOVL    CURRENT_MTL, R0             ; 1721
                                    31    A0                 02  88 00051        BISB2   #2, 49(R0)                  ;
                                          6E  00FFFFFF       8F  D0 00055        MOVL    #16777215, DUMMY            ; 1722
                                          7E             08  AC  7D 0005C        MOVQ    COUNT, -(SP)                ; 1723
                                                         08  AE  9F 00060        PUSHAB  DUMMY                       ;
                                                         10  A0  9F 00063        PUSHAB  16(R0)                      ;
                                00000000G 00                 04  FB 00066        CALLS   #4, ACL_ADDENTRY            ;
                                                             5D  11 0006D        BRB     11$                         ;
                                          50                 6B  D0 0006F 6$:    MOVL    CURRENT_MTL, R0             ; 1728
                                          51             10  A0  9E 00072        MOVAB   16(R0), R1                  ;
                                                             61  D5 00076        TSTL    (R1)                        ;
                                                             07  12 00078        BNEQ    7$                          ;
                                          61                 51  D0 0007A        MOVL    R1, (R1)                    ; 1731
                                    14    A0                 51  D0 0007D        MOVL    R1, 20(R0)                  ; 1732
                                    31    A0                 02  88 00081 7$:    BISB2   #2, 49(R0)                  ; 1734
                                                             7E  D4 00085        CLRL    -(SP)                       ; 1735
                                                             51  DD 00087        PUSHL   R1                          ;
                                00000000G 00                 02  FB 00089        CALLS   #2, ACL_DELETEACL           ;
                                                             3A  11 00090        BRB     11$                         ;
                                          57                 6B  D0 00092 8$:    MOVL    CURRENT_MTL, R7             ; 1740
                                          50             10  A7  9E 00095        MOVAB   16(R7), R0                  ;
                                          50             10  A7  D1 00099        CMPL    16(R7), R0                  ;
                                                             0A  12 0009D        BNEQ    9$                          ;
                08    AC          00    6E                 00  2C 0009F        MOVC5   #0, (SP), #0, COUNT, @ACE           ; 1743
                                                         0C  BC       000A5                                              ;
                                                             2D  11 000A7        BRB     13$                         ; 1744
                                          7E             08  AC  7D 000A9 9$:    MOVQ    COUNT, -(SP)                ; 1746
                                                             56  DD 000AD        PUSHL   ACL_CONTEXT                 ;
```

```
                                 10   A7  9F 000AF          PUSHAB  16(R7)                                  ;
                00000000G  00         04  FB 000B2          CALLS   #4, ACL_READACL                         ;
                                      11  11 000B9          BRB     11$                                     ;
                           7E    08   AC  7D 000BB 10$:     MOVQ    COUNT, -(SP)                            ; 1750
                                      56  DD 000BF          PUSHL   ACL_CONTEXT                             ;
                      7E           6B   10  C1 000C1          ADDL3   #16, CURRENT_MTL, -(SP)                 ;
                00000000G  00         04  FB 000C5          CALLS   #4, ACL_ACLLENGTH                       ;
                           5A         50  D0 000CC 11$:     MOVL    R0, LOCAL_STATUS                        ;
                           04         59  E9 000CF 12$:     BLBC    R9, 13$                                 ; 1756
                     34    A8         5A  D0 000D2          MOVL    LOCAL_STATUS, 52(R8)                    ;
                           50         01  D0 000D6 13$:     MOVL    #1, R0                                  ; 1758
                                         04 000D9          RET                                             ; 1760

; Routine Size:  218 bytes,    Routine Base:  CODE + 0000
```

```
;  210          1761  1 %SBTTL 'ACL_BUILDACL - build in memory ACL from headers'
;  211          1762  1 GLOBAL ROUTINE ACL_BUILDACL =
;  212          1763  1
;  213          1764  1 !++
;  214          1765  1 !
;  215          1766  1 ! FUNCTIONAL DESCRIPTION:
;  216          1767  1 !
;  217          1768  1 !       This routine builds the in core ACL from the file headers.
;  218          1769  1 !
;  219          1770  1 ! CALLING SEQUENCE:
;  220          1771  1 !       none
;  221          1772  1 !
;  222          1773  1 ! INPUT PARAMETERS:
;  223          1774  1 !       none
;  224          1775  1 !
;  225          1776  1 ! IMPLICIT INPUTS:
;  226          1777  1 !       CURRENT_MTL: for ACL queue head & header address
;  227          1778  1 !
;  228          1779  1 ! OUTPUT PARAMETERS:
;  229          1780  1 !       none
;  230          1781  1 !
;  231          1782  1 ! IMPLICIT OUTPUTS:
;  232          1783  1 !       none
;  233          1784  1 !
;  234          1785  1 ! ROUTINE VALUE:
;  235          1786  1 !       1 if successful
;  236          1787  1 !       LBC otherwise
;  237          1788  1 !
;  238          1789  1 ! SIDE EFFECTS:
;  239          1790  1 !       The in core ACL is created from the file headers.
;  240          1791  1 !
;  241          1792  1 !--
;  242          1793  1
;  243          1794  2 BEGIN
;  244          1795  2
;  245          1796  2 LOCAL
;  246          1797  2         STATUS,                                 ! General status value
;  247          1798  2         RVN,                                    ! RVN of current header
;  248          1799  2         ACE             : REF BBLOCK,           ! Address of current ACE
;  249          1800  2         ACL_LENGTH,                             ! Size of header segment ACL
;  250          1801  2         ACL_POINTER     : REF BBLOCK,           ! Pointer to current segment
;  251          1802  2         HEADER          : REF BBLOCK,           ! Address of current header
;  252          1803  2         LOCAL_HEADER    : BBLOCK [512],         ! Storage for file header
;  253          1804  2         EXT_HEADER_FID  : BBLOCK [6];           ! Extension header file-ID
;  254          1805  2
;  255          1806  2 HEADER = .CURRENT_MTL[MTL_HEADER];
;  256          1807  2 RVN = .CURRENT_MT[[MTL_FID_RVN];
;  257          1808  2
;  258          1809  2 ! Build an in core ACL from the file headers
;  259          1810  2
;  260          1811  2 WHILE 1
;  261          1812  2 DO
;  262          1813  3     BEGIN
;  263          1814  3     IF .HEADER[FH2$B_ACOFFSET]*2 LSS $BYTEOFFSET (FH2$W_CHECKSUM)
;  264          1815  3     THEN
;  265          1816  4         BEGIN
;  266          1817  4         ACE = .HEADER + .HEADER[FH2$B_ACOFFSET]*2;
```

```
;  267          1818  4          ACL_LENGTH = 0;
;  268          1819  4          DO
;  269          1820  5              BEGIN
;  270          1821  5              ACL_LENGTH = .ACL_LENGTH + .ACE[ACE$B_SIZE];
;  271          1822  5              ACE = .ACE + .ACE[ACE$B_SIZE];
;  272          1823  5              END
;  273          1824  4          UNTIL .ACE[ACE$B_SIZE] EQL 0
;  274          1825  4             OR .ACE GEQA HEADER[FH2$W_CHECKSUM];
;  275          1826  4          ACL_POINTER = GET_ZERO_VM (ACL$C_LENGTH + .ACL_LENGTH);
;  276          1827  4          CH$MOVE (.ACL_LENGTH, .HEADER + .HEADER[FH2$B_ACOFFSET]*2,
;  277          1828  4                                ACL_POINTER[ACL$L_LIST]);
;  278          1829  4          ACL_POINTER[ACL$W_SIZE] = ACL$C_LENGTH + .ACL_LENGTH;
;  279          1830  4          INSQUE (.ACL_POINTER, .CURRENT_MTL[MTL_ACLBL]);
;  280          1831  3          END;
;  281          1832  3      CH$MOVE (6, HEADER[FH2$W_EXT_FID], EXT_HEADER_FID);
;  282          1833  3      IF .EXT_HEADER_FID[FID$W_NUM] EQL 0
;  283          1834  3      AND .EXT_HEADER_FID[FID$B_NMX] EQL 0
;  284          1835  3      THEN EXITLOOP;
;  285          1836  3      IF .EXT_HEADER_FID[FID$B_RVN] EQL 0
;  286          1837  3      THEN EXT_HEADER_FID[FID$B_RVN] = .RVN
;  287          1838  3      ELSE RVN = .EXT_HEADER_FID[FID$B_RVN];
;  288          1839  3      STATUS = READ_HEADER (EXT_HEADER_FID, LOCAL_HEADER);
;  289          1840  3      IF NOT .STATUS THEN RETURN .STATUS;
;  290          1841  3      HEADER = LOCAL_HEADER;
;  291          1842  2      END;
;  292          1843  2
;  293          1844  2  RETURN 1;
;  294          1845  2
;  295          1846  1  END;                                          ! End of routine ACL_BUILDACL
```

```
                                    0FFC 00000          .ENTRY  ACL_BUILDACL, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1762
                                                                R10,R11
                          5E     FDF8    CE  9E 00002         MOVAB   -520(SP), SP
                          50 00000000'   EF  D0 00007         MOVL    CURRENT_MTL, R0                       ; 1806
                          56       0C    A0  D0 0000E         MOVL    12(R0), HEADER
                          5A       1C    A0  9A 00012         MOVZBL  28(R0), RVN                           ; 1807
                          52       02    A6  9A 00016 1$:     MOVZBL  2(HEADER), R2                         ; 1814
                          52              02  C4 0001A         MULL2   #2, R2
                000001FE  8F              52  D1 0001D         CMPL    R2, #510
                                          46  18 00024         BGEQ    4$
                59        56              52  C1 00026         ADDL3   R2, HEADER, ACE                       ; 1817
                                          58  D4 0002A         CLRL    ACL_LENGTH                            ; 1818
                          50              69  9A 0002C 2$:     MOVZBL  (ACE), R0                             ; 1821
                          58              50  C0 0002F         ADDL2   R0, ACL_LENGTH
                          50              69  9A 00032         MOVZBL  (ACE), R0                             ; 1822
                          59              50  C0 00035         ADDL2   R0, ACE
                                          69  95 00038         TSTB    (ACE)                                 ; 1824
                                          0A  13 0003A         BEQL    3$
                          50     01FE     C6  9E 0003C         MOVAB   510(R6), R0                           ; 1825
                          50              59  D1 00041         CMPL    ACE, R0
                                          E6  1F 00044         BLSSU   2$
                             0C           A8  9F 00046 3$:     PUSHAB  12(ACL_LENGTH)                        ; 1826
                00000000G 00              01  FB 00049         CALLS   #1, GET_ZERO_VM
```

```
                                57              50  D0 00050          MOVL    R0, ACL_POINTER                                 ;
                            OC  A7          6246    58  28 00053      MOVC3   ACL_LENGTH, (R2)[HEADER], 12(ACL_POINTER)       ; 1828
                                50          OC  A8  9E 00059          MOVAB   12(ACL_LENGTH), R0                              ; 1829
                            08  A7              50  B0 0005D          MOVW    R0, 8(ACL_POINTER)                              ;
                                50 00000000'    EF  D0 00061          MOVL    CURRENT_MTL, R0                                 ; 1830
                            14  B0              67  0E 00068          INSQUE  (ACL_POINTER), @20(R0)                          ;
                        6E  0E  A6              06  28 0006C 4$:      MOVC3   #6, T4(HEADER), EXT_HEADER_FID                  ; 1832
                                                6E  B5 00071          TSTW    EXT_HEADER_FID                                  ; 1833
                                                05  12 00073          BNEQ    5$                                              ;
                                            05  AE  95 00075          TSTB    EXT_HEADER_FID+5                                ; 1834
                                                2D  13 00078          BEQL    9$                                              ;
                                            04  AE  95 0007A 5$:      TSTB    EXT_HEADER_FID+4                                ; 1836
                                                06  12 0007D          BNEQ    6$                                              ;
                            04  AE              5A  90 0007F          MOVB    RVN, EXT_HEADER_FID+4                           ; 1837
                                                04  11 00083          BRB     7$                                              ;
                                5A          04  AE  9A 00085 6$:      MOVZBL  EXT_HEADER_FID+4, RVN                           ; 1838
                                            08  AE  9F 00089 7$:      PUSHAB  LOCAL_HEADER                                    ; 1839
                                            04  AE  9F 0008C          PUSHAB  EXT_HEADER_FID                                  ;
                    00000000G   00              02  FB 0008F          CALLS   #2, READ_HEADER                                 ;
                                5B              50  D0 00096          MOVL    R0, STATUS                                      ;
                                04              5B  E8 00099          BLBS    STATUS, 8$                                      ; 1840
                                50              5B  D0 0009C          MOVL    STATUS, R0                                      ;
                                                    04 0009F          RET                                                     ;
                                56          08  AE  9E 000A0 8$:      MOVAB   LOCAL_HEADER, HEADER                            ; 1841
                                            FF6F    31 000A4          BRW     1$                                              ; 1811
                                50              01  D0 000A7 9$:      MOVL    #1, R0                                          ; 1844
                                                    04 000AA          RET                                                     ; 1846

; Routine Size:  171 bytes,    Routine Base:  CODE + 00DA
```

```
;  297          1847  1 %SBTTL 'ALLOC_PAGED - memory allocator for ACLSUBR'
;  298          1848  1 GLOBAL ROUTINE ALLOC_PAGED (SIZE, DUMMY) =
;  299          1849  1
;  300          1850  1 !++
;  301          1851  1 !
;  302          1852  1 ! FUNCTIONAL DESCRIPTION:
;  303          1853  1 !       This routine is a simple jacket routine for GET_ZERO_VM.  It is
;  304          1854  1 !       needed by the ACL management routines in ACLSUBR (from the XQP).
;  305          1855  1 !
;  306          1856  1 ! INPUT PARAMETERS:
;  307          1857  1 !       SIZE            - Size of the block to allocate
;  308          1858  1 !       DUMMY           - Unused by BACKUP.
;  309          1859  1 !
;  310          1860  1 ! OUTPUT PARAMETERS:
;  311          1861  1 !       NONE
;  312          1862  1 !
;  313          1863  1 ! ROUINE VALUE:
;  314          1864  1 !       Address of allocated area.
;  315          1865  1 !
;  316          1866  1 ! SIDE EFFECTS:
;  317          1867  1 !       If allocation fails, a fatal error is signalled.
;  318          1868  1 !
;  319          1869  1 !--
;  320          1870  1
;  321          1871  2 BEGIN
;  322          1872  2
;  323          1873  2 RETURN GET_ZERO_VM (.SIZE);
;  324          1874  2
;  325          1875  1 END;                                    ! End of routine ALLOC_PAGED


                                        0000 00000      .ENTRY  ALLOC_PAGED, Save nothing       ; 1848
                                  04    AC   DD 00002   PUSHL   SIZE                            ; 1873
                        00000000G   00    01   FB 00005   CALLS   #1, GET_ZERO_VM               ;
                                        04 0000C        RET                                     ; 1875

; Routine Size:  13 bytes,    Routine Base:  CODE + 0185
```

```
;  327          1876  1 %SBTTL 'DALLOC_PAGED - memory  deallocator for ACLSUBR'
;  328          1877  1 GLOBAL ROUTINE DALLOC_PAGED (AREA, DUMMY) =
;  329          1878  1
;  330          1879  1 !++
;  331          1880  1 !
;  332          1881  1 ! FUNCTIONAL DESCRIPTION:
;  333          1882  1 !       This routine is a simple jacket routine for FREE_VM.  It is
;  334          1883  1 !       needed by the ACL management routines in ACLSUBR (from the XQP).
;  335          1884  1 !
;  336          1885  1 ! INPUT PARAMETERS:
;  337          1886  1 !       AREA            - Address of the block to deallocate
;  338          1887  1 !       DUMMY           - Unused by BACKUP.
;  339          1888  1 !
;  340          1889  1 ! OUTPUT PARAMETERS:
;  341          1890  1 !       NONE
;  342          1891  1 !
;  343          1892  1 ! ROUTINE VALUE:
;  344          1893  1 !       NONE
;  345          1894  1 !
;  346          1895  1 ! SIDE EFFECTS:
;  347          1896  1 !       If deallocation fails, a fatal error is signalled.
;  348          1897  1 !
;  349          1898  1 !--
;  350          1899  1
;  351          1900  2 BEGIN
;  352          1901  2
;  353          1902  2 MAP
;  354          1903  2         AREA            : REF $BBLOCK;          ! Address of block to free up
;  355          1904  2
;  356          1905  2 FREE_VM (.AREA[ACL$W_SIZE], .AREA);
;  357          1906  2 RETURN 1;
;  358          1907  2
;  359          1908  1 END;                                            ! End of routine DALLOC_PAGED



                                                0000 00000      .ENTRY  DALLOC_PAGED, Save nothing              ; 1877
                                  50    04  AC  D0 00002      MOVL    AREA, R0                                ; 1905
                                              50  DD 00006      PUSHL   R0                                      ;
                                  7E    08  A0  3C 00008      MOVZWL  8(R0), -(SP)                            ;
                        00000000G 00          02  FB 0000C      CALLS   #2, FREE_VM                             ;
                                  50          01  D0 00013      MOVL    #1, R0                                  ; 1906
                                                  04 00016      RET                                             ; 1908

; Routine Size:  23 bytes,    Routine Base:  CODE + 0192

;  360          1909  1
;  361          1910  1 END
;  362          1911  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  COMMON                        2124  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                           425  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        21         0      1000          00:02.3
;
;
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:STAACLCTL/OBJ=OBJ$:STAACLCTL MSRC$:STAACLCTL/UPDATE=(ENH$:STAACLCTL)
;
; Size:         425 code + 2124 data bytes
; Run Time:       00:22.3
; Elapsed Time:   01:13.7
; Lines/CPU Min:   5134
; Lexemes/CPU-Min: 42298
; Memory Used:  284 pages
; Compilation Complete
```

STAACP
LIS

STAACCTL
LIS