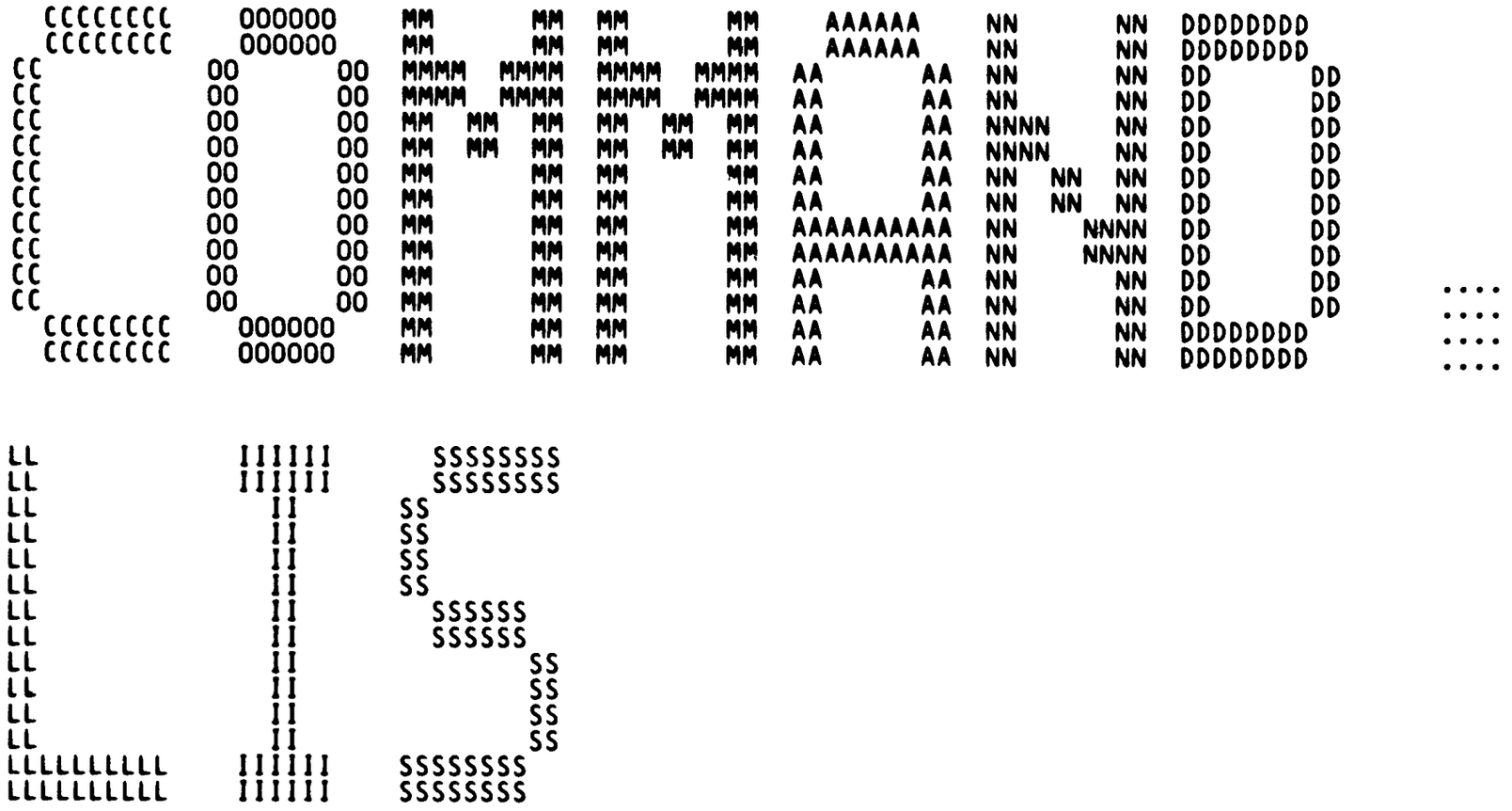
```
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBBBBBBBBBBB        AAAAAAAAA        CCCCCCCCCCC  KKK         KKK  UUU          UUU  PPPPPPPPPPPP
BBB         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
```


CCCCCCCC 000000 MM MM MM MM AAAAAA NN NN DDDDDDDD
CCCCCCCC 000000 MM MM MM MM AAAAAA NN NN DDDDDDDD
CC 00 00 MMMM MMMM MMMM MMMM AA AA NN NN DD DD
CC 00 00 MMMM MMMM MMMM MMMM AA AA NN NN DD DD
CC 00 00 MM MM MM MM MM MM AA AA NNNN NN DD DD
CC 00 00 MM MM MM MM MM MM AA AA NNNN NN DD DD
CC 00 00 MM MM MM MM AA AA NN NN NN DD DD
CC 00 00 MM MM MM MM AA AA NN NN NN DD DD
CC 00 00 MM MM MM MM AAAAAAAAAA NN NNNN DD DD
CC 00 00 MM MM MM MM AAAAAAAAAA NN NNNN DD DD
CC 00 00 MM MM MM MM AA AA NN NN DD DD
CCCCCCCC 000000 MM MM MM MM AA AA NN NN DDDDDDDD
CCCCCCCC 000000 MM MM MM MM AA AA NN NN DDDDDDDD
LL IIIIII SSSSSSSS
LL IIIIII SSSSSSSS
LL II SS
LL II SS
LL II SS
LL II SS
LL II SSSSSS
LL II SSSSSS
LL II SS
LL II SS
LL II SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```
;     1        0001  0 MODULE COMMAND  (%TITLE 'Command parser'
;     2        0002  0                 IDENT = 'V04-000'
;     3        0003  0                 ) =
;     4        0004  1 BEGIN
;     5        0005  1
;     6        0006  1 !
;     7        0007  1 !**********************************************************************
;     8        0008  1 !*                                                                    *
;     9        0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
;    10        0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
;    11        0011  1 !*  ALL RIGHTS RESERVED.                                           *
;    12        0012  1 !*                                                                    *
;    13        0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    14        0014  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    15        0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
;    16        0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    17        0017  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    18        0018  1 !*  TRANSFERRED.                                                       *
;    19        0019  1 !*                                                                    *
;    20        0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    21        0021  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    22        0022  1 !*  CORPORATION.                                                       *
;    23        0023  1 !*                                                                    *
;    24        0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    25        0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
;    26        0026  1 !*                                                                    *
;    27        0027  1 !*                                                                    *
;    28        0028  1 !**********************************************************************
;    29        0029  1 !
;    30        0030  1
;    31        0031  1 !++
;    32        0032  1 ! FACILITY:
;    33        0033  1 !       Backup/Restore
;    34        0034  1 !
;    35        0035  1 ! ABSTRACT:
;    36        0036  1 !       This module contains the routines that perform command parsing and
;    37        0037  1 !       analysis.
;    38        0038  1 !
;    39        0039  1 ! ENVIRONMENT:
;    40        0040  1 !       VAX/VMS user mode.
;    41        0041  1 !--
;    42        0042  1 !
;    43        0043  1 ! AUTHOR: M. Jack, CREATION DATE: 26-Aug-1980
;    44        0044  1 !
;    45        0045  1 ! MODIFIED BY:
;    46        0046  1 !
;    47        0047  1 !       V03-015 LY0520          Larry Yetto             5-AUG-1984 14:15
;    48        0048  1 !               QAR # 1212 .  Search lists with concealed devices do not
;    49        0049  1 !               work.  In order to fix this perform the SLPARSE parses with
;    50        0050  1 !               the NOCONCEAL flag set in the NAM.  Once this is done
;    51        0051  1 !               we will end up with ".][" etc. strings in the file
;    52        0052  1 !               spec.  BACKUP does not know how to deal with these strings
;    53        0053  1 !               so we will always strip out the ".][" from the file file
;    54        0054  1 !               name string.  This will result in BACKUP not being able to
;    55        0055  1 !               get below 8 level deep directories (ie. concealed name hides 8)
;    56        0056  1 !               but at least some cases will work.
;    57        0057  1 !
```

```
;    58          0058  1 ;       V03-014 LY0514          Larry Yetto             23-JUL-1984 15:37
;    59          0059  1 ;               Correct incremental restores so that they do not use
;    60          0060  1 ;               the current default directory to determine the placement
;    61          0061  1 ;               of the restored files.
;    62          0062  1 ;
;    63          0063  1 ;       V03-013 LY0486          Larry Yetto             27-APR-1984 10:55
;    64          0064  1 ;               Base Level QAR # 217  -  Fix accvio if no output
;    65          0065  1 ;               file spec
;    66          0066  1 ;
;    67          0067  1 ;       V03-012 LY0471          Larry Yetto              9-APR-1984 08:21
;    68          0068  1 ;               Modify the parsing algorithm for input file lists so
;    69          0069  1 ;               that it will work properly with search lists.
;    70          0070  1 ;
;    71          0071  1 ;       V03-011 LMP0140         L. Mark Pilant,         23-Aug-1983  8:20
;    72          0072  1 ;               Add support for alphanumeric UICs.
;    73          0073  1 ;
;    74          0074  1 ;       V03-010 JEP0003         J. Eric Pollack,        23-Apr-1983  10:53
;    75          0075  1 ;               Add support for encrypted savesets.
;    76          0076  1 ;
;    77          0077  1 ;       V03-009 ACG0313         Andrew C. Goldstein,    12-Feb-1983  16:05
;    78          0078  1 ;               Add routine subtitles
;    79          0079  1 ;
;    80          0080  1 ;       V03-008 ACG0314         Andrew C. Goldstein,    10-Feb-1983  22:28
;    81          0081  1 ;               Make /NOREWIND and /NOINITIALIZE default for save sets
;    82          0082  1 ;
;    83          0083  1 ;       V03-007 ACG0309         Andrew C. Goldstein,    19-Jan-1983  11:19
;    84          0084  1 ;               Fix test for interactive to use SYS$COMMAND properly
;    85          0085  1 ;               Also change tape block sizes to integral 512 byte units
;    86          0086  1 ;
;    87          0087  1 ;       V03-006 MLJ0095         Martin L. Jack, 26-Aug-1982  15:41
;    88          0088  1 ;               Use CLI$END_PARSE only in standalone version.
;    89          0089  1 ;
;    90          0090  1 ;       V03-005 MLJ48501        Martin L. Jack, 17-Aug-1982  15:30
;    91          0091  1 ;               Correct problem with wildcard version stickiness.
;    92          0092  1 ;
;    93          0093  1 ;       V03-004 MLJ47951        Martin L. Jack, 27-Jul-1982  11:45
;    94          0094  1 ;               Correct problem that caused the length of the /JOURNAL value to
;    95          0095  1 ;               be taken as the length of the /LABEL values.
;    96          0096  1 ;
;    97          0097  1 ;       V03-003 LMP0032         L. Mark Pilant,         21-Jun-1982  11:83
;    98          0098  1 ;               Add support for wildcard save set names on the RESTORE and
;    99          0099  1 ;               LIST operations.
;   100          0100  1 ;
;   101          0101  1 ;       V03-002 PCG0001         Peter George    18-Jun-1982  13:30
;   102          0102  1 ;               Check new CLI$PRESENT statuses for local qualifiers.
;   103          0103  1 ;
;   104          0104  1 ;       V03-001 MLJ0084         Martin L. Jack, 27-Mar-1982  17:00
;   105          0105  1 ;               Allow a save set to be written to a mailbox (including NL:).
;   106          0106  1 ;
;   107          0107  1 ;       V02-015 MLJ0081         Martin L. Jack, 26-Feb-1982  14:35
;   108          0108  1 ;               Correct setting of COM_INTERACTIVE for standalone and
;   109          0109  1 ;               STARTUP.COM environments.
;   110          0110  1 ;
;   111          0111  1 ;       V02-014 MLJ0073         Martin L. Jack, 19-Jan-1982  18:19
;   112          0112  1 ;               Implement /PROTECTION qualifier for save set protection.
;   113          0113  1 ;               Extend /LABEL qualifier to accept a list.
;   114          0114  1 ;
```

```
;  115     0115  1 !       V02-013 MLJ0062         Martin L. Jack, 7-Dec-1981  13:57
;  116     0116  1 !               Implement /INCREMENTAL.  Allow input volume of a /PHYSICAL
;  117     0117  1 !               operation to be mounted foreign or to be mounted Files-11 if
;  118     0118  1 !               the process has LOG_IO privilege.
;  119     0119  1 !
;  120     0120  1 !       V02-012 MLJ0054         Martin L. Jack, 15-Oct-1981  17:08
;  121     0121  1 !               Allow save sets to be network files.  Implement /VOLUME
;  122     0122  1 !               qualifier.  Implement /DELETE qualifier.  Integrate GET_VM
;  123     0123  1 !               and FREE_VM jacket routines.  Delete JPI_STS, add COM_INTERACT
;  124     0124  1 !               flag.
;  125     0125  1 !
;  126     0126  1 !       V02-011 MLJ0052         Martin L. Jack, 1-Oct-1981  13:16
;  127     0127  1 !               Implement /INTERCHANGE qualifier.
;  128     0128  1 !
;  129     0129  1 !       V02-010 MLJ0049         Martin L. Jack, 28-Sep-1981  22:06
;  130     0130  1 !               Correct test for /IMAGE in standalone environment.
;  131     0131  1 !
;  132     0132  1 !       V02-009 MLJ0043         Martin L. Jack, 8-Sep-1981  16:56
;  133     0133  1 !               Account for RMS logical device name change.  Install $GETSYI.
;  134     0134  1 !               Correct /SELECT and /EXCLUDE operation on [000000].
;  135     0135  1 !
;  136     0136  1 !       V02-008 MLJ0036         Martin L. Jack, 29-Aug-1981  16:00
;  137     0137  1 !               Additional error detection.
;  138     0138  1 !
;  139     0139  1 !       V02-007 MLJ0025         Martin L. Jack, 8-May-1981  10:55
;  140     0140  1 !               Reorganize qualifier database.  Remove /SEARCH qualifier.
;  141     0141  1 !               Use PL/I varying strings rather than dynamic strings.
;  142     0142  1 !
;  143     0143  1 !       V02-006 MLJ0022         Martin L. Jack, 21-Apr-1981  13:53
;  144     0144  1 !               Convert directory wildcarding to use RMS.
;  145     0145  1 !
;  146     0146  1 !       V02-005 MLJ0021         Martin L. Jack, 20-Apr-1981  21:50
;  147     0147  1 !               Implement /TRUNCATE qualifier.
;  148     0148  1 !
;  149     0149  1 !       V02-004 MLJ0015         Martin L. Jack, 6-Apr-1981  23:22
;  150     0150  1 !               Explicitly check for SS$_NOTRAN in translation of SYS$NODE.
;  151     0151  1 !               Remove unnecessary code to clear QUAL area.
;  152     0152  1 !
;  153     0153  1 !       V02-003 MLJ0010         Martin L. Jack, 25-Mar-1981  16:20
;  154     0154  1 !               Reorganize global storage.  Rename /OWNER to /OWNER_UIC.
;  155     0155  1 !               Remove unnecessary parts of qualifier database.  Add capability
;  156     0156  1 !               for listing concurrently with another operation.  Add checking
;  157     0157  1 !               for properly mounted volumes.  Replace /SEQUENTIAL by /SAVE_SET
;  158     0158  1 !               specified on a foreign volume.  Reparse to get save set name in
;  159     0159  1 !               all elements of a list.  Changes for standalone operation.
;  160     0160  1 !
;  161     0161  1 !       V02-002 MLJ0008         Martin L. Jack, 10-Mar-1981  11:44
;  162     0162  1 !               Rename /FILE to /SAVE_SET
;  163     0163  1 !
;  164     0164  1 !       V02-001 MLJ0005         Martin L. Jack, 26-Feb-1981  17:08
;  165     0165  1 !               Allow null input save set filename
;  166     0166  1 !
;  167     0167  1 !**
```

```
;  169          0168  1 REQUIRE 'SRC$:COMMON';
;  170          1274  1 LIBRARY 'SYS$LIBRARY:LIB';
;  171          1275  1 LIBRARY 'SYS$LIBRARY:TPAMAC';
;  172          1276  1
;  173          1277  1
;  174          1278  1 FORWARD ROUTINE
;  175          1279  1         CALL_TPARSE,                      ! Execute call to LIB$TPARSE
;  176          1280  1         PRESERVE_STRING:NOVALUE,          ! preserve string value
;  177          1281  1         GET_FC,                           ! Get a file context area
;  178          1282  1         COPY_FC:        NOVALUE,          ! Copy one FC to a new one
;  179          1283  1         INIT_FC:        NOVALUE,          ! Init an existing FC incl related name
;  180          1284  1         PARSE_EXC_SEL:  NOVALUE,          ! Parse /EXCLUDE and /SELECT
;  181          1285  1         EXPAND_SEARCH_LIST,               ! Expand a search list
;  182          1286  1         CHECK_SAV:      NOVALUE,          ! Check save set input/output spec
;  183          1287  1         CHECK_F11:      NOVALUE,          ! Check files-11 input/output spec
;  184          1288  1         CHECK_COMMON:   NOVALUE,          ! Check common input/output spec
;  185          1289  1         BUILD_QUAL_DESC:NOVALUE,          ! Build qualifier block descriptors
;  186          1290  1         CHECK:          NOVALUE,          ! Check input and output specifications
;  187          1291  1         COMMAND:        NOVALUE;          ! Parse BACKUP command
;  188          1292  1
;  189          1293  1
;  190          1294  1 EXTERNAL ROUTINE
;  191          1295  1         CLI$END_PARSE:  WEAK ADDRESSING_MODE(GENERAL),
;  192          1296  1                                           ! Declare end of parsing for STACLINT
;  193          1297  1         CLI$GET_VALUE:  ADDRESSING_MODE(GENERAL),
;  194          1298  1                                           ! Get a parameter or qualifier value
;  195          1299  1         CLI$PRESENT:    ADDRESSING_MODE(GENERAL),
;  196          1300  1                                           ! Determine if entity is present
;  197          1301  1         LIB$CVT_DTB:    ADDRESSING_MODE(GENERAL),
;  198          1302  1                                           ! Convert decimal string to binary
;  199          1303  1         LIB$CVT_TIME:   ADDRESSING_MODE(GENERAL),
;  200          1304  1                                           ! Convert absolute time to binary
;  201          1305  1         LIB$TPARSE:     ADDRESSING_MODE(GENERAL),
;  202          1306  1                                           ! Table-driven parser
;  203          1307  1         SYS$SETDFPROT:  ADDRESSING_MODE(GENERAL),
;  204          1308  1                                           ! Read/set default file protection
;  205          1309  1         CRYPTO_INIT:    NOVALUE WEAK,     ! Initialize for saveset encryption
;  206          1310  1         FILE_ERROR:     NOVALUE,          ! Signal file-related error
;  207          1311  1         GET_VM,                           ! Allocate virtual memory
;  208          1312  1         GET_ZERO_VM,                      ! Allocate and clear virtual memory
;  209          1313  1         FREE_VM:        NOVALUE ;         ! Deallocate virtual memory
;  210          1314  1
;  211          1315  1
;  212          1316  1 EXTERNAL LITERAL
;  213          1317  1         BACKUP$_ENCNOTSUP,
;  214          1318  1         BACKUP$_ENCQUAIGN,
;  215          1319  1         BACKUP$_INVQUAVAL,
;  216          1320  1         BACKUP$_INVDEVTYP,
;  217          1321  1         BACKUP$_PARSE,
;  218          1322  1         BACKUP$_INCDEVLIS,
;  219          1323  1         BACKUP$_ONEF11OUT,
;  220          1324  1         BACKUP$_CMPOUTF11,
;  221          1325  1         BACKUP$_SAVSETNAM,
;  222          1326  1         BACKUP$_ONEIMGDEV,
;  223          1327  1         BACKUP$_ONEF11DEV,
;  224          1328  1         BACKUP$_ONEPHYDEV,
;  225          1329  1         BACKUP$_PHYFILSPE,
```

```
;  226     1330  1          BACKUP$_IMGFILSPE,
;  227     1331  1          BACKUP$_INCFILSPE,
;  228     1332  1          BACKUP$_LISINPSAV,
;  229     1333  1          BACKUP$_MOUNTF11,
;  230     1334  1          BACKUP$_MOUNTFOR,
;  231     1335  1          BACKUP$_CONFQUAL,
;  232     1336  1          CLI$_PRESENT,           ! Explicit global /QUAL
;  233     1337  1          CLI$_NEGATED,           ! Explicit global /NOQUAL
;  234     1338  1          CLI$_ABSENT,            ! Implicit global /NOQUAL
;  235     1339  1          CLI$_LOCPRES,           ! Explicit local /QUAL
;  236     1340  1          CLI$_LOCNEG;            ! Explicit local /NOQUAL
;  237     1341  1
;  238     1342  1
;  239     1343  1 G$DEFINE();                      ! Define global area
;  240     1344  1
;  241     1345  1
;  242     1346  1 FORWARD
;  243     1347  1
;  244     1348  1          ! TPARSE tables, which are defined at the end of this module.
;  245     1349  1          !
;  246     1350  1          UIC_STATES:     VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  247     1351  1          UIC_KEYS:       VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  248     1352  1          OWN_STATES:     VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  249     1353  1          OWN_KEYS:       VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  250     1354  1          IGN_STATES:     VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  251     1355  1          IGN_KEYS:       VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  252     1356  1          BAK_STATES:     VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  253     1357  1          BAK_KEYS:       VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  254     1358  1          PRO_STATES:     VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE),
;  255     1359  1          PRO_KEYS:       VECTOR[0] ADDRESSING_MODE(LONG_RELATIVE);
;  256     1360  1
;  257     1361  1
;  258     1362  1 BIND
;  259     1363  1
;  260     1364  1          ! TPARSE parse-output variables.  To save space, these are defined
;  261     1365  1          ! to overlay OUTPUT_ATTBUF.
;  262     1366  1          !
;  263     1367  1          CONVERTED_UIC=  OUTPUT_ATTBUF,                          ! Converted UIC
;  264     1368  1          TPA_FLAGS=      OUTPUT_ATTBUF+8:        BLOCK[1],       ! Various flags
;  265     1369  1          TPA_PROTECTION= OUTPUT_ATTBUF:          VECTOR[,WORD],  ! Protection
;  266     1370  1          DEVCLASS=       OUTPUT_ATTBUF;                          ! For $GETDVI
;  267     1371  1
;  268     1372  1
;  269     1373  1 MACRO
;  270     1374  1
;  271     1375  1          ! Definitions of TPA_FLAGS.
;  272     1376  1          !
;  273     1377  1          TPA_WGROUP=     0,0,1,0 %,      ! Wild group number
;  274     1378  1          TPA_WMEMBER=    0,1,1,0 %,      ! Wild member number
;  275     1379  1          TPA_DEFAULT=    0,2,1,0 %,      ! DEFAULT specified
;  276     1380  1          TPA_ORIGINAL=   0,3,1,0 %,      ! ORIGINAL specified
;  277     1381  1          TPA_PARENT=     0,4,1,0 %,      ! PARENT specified
;  278     1382  1          TPA_UIC=        0,5,1,0 %,      ! UIC specified
;  279     1383  1          TPA_NOBACKUP=   0,6,1,0 %,      ! NOBACKUP specified
;  280     1384  1          TPA_INTERLOCK=  0,7,1,0 %;      ! INTERLOCK specified
;  281     1385  1
;  282     1386  1
```

```
;   283         1387  1 LITERAL
;   284         1388  1         TPA_M_WGROUP=   $FIELDMASK(TPA_WGROUP),
;   285         1389  1         TPA_M_WMEMBER=  $FIELDMASK(TPA_WMEMBER),
;   286         1390  1         TPA_M_DEFAULT=  $FIELDMASK(TPA_DEFAULT),
;   287         1391  1         TPA_M_ORIGINAL= $FIELDMASK(TPA_ORIGINAL),
;   288         1392  1         TPA_M_PARENT=   $FIELDMASK(TPA_PARENT),
;   289         1393  1         TPA_M_UIC=      $FIELDMASK(TPA_UIC),
;   290         1394  1         TPA_M_NOBACKUP= $FIELDMASK(TPA_NOBACKUP),
;   291         1395  1         TPA_M_INTERLOCK=$FIELDMASK(TPA_INTERLOCK);
;   292         1396  1
;   293         1397  1
;   294         1398  1 MACRO
;   295         1399  1         VAR_LENGTH=     0,0,16,0 %,     ! Length of varying string
;   296         1400  1         VAR_BODY=       2,0,0,0 %;      ! Body of varying string
;   297         1401  1
;   298         1402  1 OWN
;   299         1403  1         CHK_FLAGS:      BBLOCK[4];      ! Flags for CHECK routines
;   300         1404  1 MACRO
;   301         1405  1         CHK_FLAGS_COUNT=        0,0,8,0 %,      ! Count of items in list
;   302         1406  1         CHK_FLAGS_F11=          1,0,1,0 %,      ! True if Files-11 seen
;   303         1407  1         CHK_FLAGS_SAV=          1,1,1,0 %,      ! True if save set seen
;   304         1408  1         CHK_FLAGS_SAVE_SET=     1,2,1,0 %,      ! True if /SAVE_SET
;   305         1409  1         CHK_FLAGS_SRCH_LIST=    1,3,1,0 %;      ! True if search decoding is needed
;   306         1410  1
;   307         1411  1
```

```
;   309          1412  1 %SBTTL 'CALL_TPARSE - call LIB$TPARSE'
;   310          1413  1 ROUTINE CALL_TPARSE(SRC,STATES,KEYS)=
;   311          1414  1 !
;   312          1415  1 !++
;   313          1416  1 !
;   314          1417  1 ! FUNCTIONAL DESCRIPTION:
;   315          1418  1 !       This routine executes a call to LIB$TPARSE.
;   316          1419  1 !
;   317          1420  1 ! INPUT PARAMETERS:
;   318          1421  1 !       SRC             - Pointer to the string to be parsed (PL/I varying).
;   319          1422  1 !       STATES          - State table parameter for TPARSE.
;   320          1423  1 !       KEYS            - Keyword table parameter for TPARSE.
;   321          1424  1 !
;   322          1425  1 ! IMPLICIT INPUTS:
;   323          1426  1 !       NONE
;   324          1427  1 !
;   325          1428  1 ! OUTPUT PARAMETERS:
;   326          1429  1 !       NONE
;   327          1430  1 !
;   328          1431  1 ! IMPLICIT OUTPUTS:
;   329          1432  1 !       NONE
;   330          1433  1 !
;   331          1434  1 ! ROUTINE VALUE:
;   332          1435  1 !       As returned by LIB$TPARSE.
;   333          1436  1 !
;   334          1437  1 ! SIDE EFFECTS:
;   335          1438  1 !       NONE
;   336          1439  1 !
;   337          1440  1 !--
;   338          1441  1
;   339          1442  2 BEGIN
;   340          1443  2 MAP
;   341          1444  2         SRC:            REF BBLOCK;             ! Pointer to varying string
;   342          1445  2 LOCAL
;   343          1446  2         TPA_PARAM:      BBLOCK[TPA$K_LENGTH0];  ! TPARSE parameter block
;   344          1447  2
;   345          1448  2
;   346          1449  2 CH$FILL(0, TPA$K_LENGTH0, TPA_PARAM);
;   347          1450  2 TPA_PARAM[TPA$L_COUNT] = TPA$K_COUNT0;
;   348          1451  2 TPA_PARAM[TPA$L_OPTIONS] = TPA$M_ABBREV;
;   349          1452  2 TPA_PARAM[TPA$L_STRINGCNT] = .SRC[VAR_LENGTH];
;   350          1453  2 TPA_PARAM[TPA$L_STRINGPTR] = SRC[VAR_BODY];
;   351          1454  2 LIB$TPARSE(TPA_PARAM, .STATES, .KEYS)
;   352          1455  1 END;


                                .TITLE  COMMAND Command parser
                                .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_BCB:
                .BLKB   4
00104 RWSV_IN_SEQ:
                .BLKB   4
00108 RWSV_IN_SEQ_0:
                .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_NTV:
                        .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

                .PSECT  DATA,NOEXE,2

00000 CHK_FLAGS:
                .BLKB   4

        CONVERTED_UIC=      OUTPUT_ATTBUF
        TPA_FLAGS=          OUTPUT_ATTBUF+8
        TPA_PROTECTION=     OUTPUT_ATTBUF
        DEVCLASS=           OUTPUT_ATTBUF
                .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                .EXTRN  LIB$CVT_DTB, LIB$CVT_TIME
                .EXTRN  LIB$TPARSE, SYS$SETDFPROT
                .EXTRN  FILE_ERROR, GET_VM
```

```
                                .EXTRN  GET_ZERO_VM, FREE_VM
                                .EXTRN  BACKUP$_ENCNOTSUP
                                .EXTRN  BACKUP$_ENCQUAIGN
                                .EXTRN  BACKUP$_INVQUAVAL
                                .EXTRN  BACKUP$_INVDEVTYP
                                .EXTRN  BACKUP$_PARSE, BACKUP$_INCDEVLIS
                                .EXTRN  BACKUP$_ONEF11OUT
                                .EXTRN  BACKUP$_CMPOUTF11
                                .EXTRN  BACKUP$_SAVSETNAM
                                .EXTRN  BACKUP$_ONEIMGDEV
                                .EXTRN  BACKUP$_ONEF11DEV
                                .EXTRN  BACKUP$_ONEPHYDEV
                                .EXTRN  BACKUP$_PHYFILSPE
                                .EXTRN  BACKUP$_IMGFILSPE
                                .EXTRN  BACKUP$_INCFILSPE
                                .EXTRN  BACKUP$_LISINPSAV
                                .EXTRN  BACKUP$_MOUNTF11
                                .EXTRN  BACKUP$_MOUNTFOR
                                .EXTRN  BACKUP$_CONFQUAL
                                .EXTRN  CLI$_PRESENT, CLI$_NEGATED
                                .EXTRN  CLI$_ABSENT, CLI$_LOCPRES
                                .EXTRN  CLI$_LOCNEG
                                .WEAK   CLI$END_PARSE, CRYPTO_INIT

                                .PSECT  CODE,NOWRT,2

                        003C 00000 CALL_TPARSE:
                                .WORD   Save R2,R3,R4,R5                        ; 1413
                             5E 24  C2 00002    SUBL2   #36, SP
        24          00       6E 00  2C 00005    MOVC5   #0, (SP), #0, #36, TPA_PARAM  ; 1449
                             6E     0000A
                             6E 08  D0 0000B    MOVL    #8, TPA_PARAM               ; 1450
                          04 AE     02  D0 0000E    MOVL    #2, TPA_PARAM+4         ; 1451
                          08 AE  04 BC  3C 00012    MOVZWL  @SRC, TPA_PARAM+8       ; 1452
                    0C AE 04 AC     02  C1 00017    ADDL3   #2, SRC, TPA_PARAM+12   ; 1453
                             7E 08  AC  7D 0001D    MOVQ    STATES, -(SP)           ; 1454
                             08 AE  9F  00021    PUSHAB  TPA_PARAM
                 00000000G   00     03  FB 00024    CALLS   #3, LIB$TPARSE
                                    04  0002B    RET                             ; 1455

; Routine Size:  44 bytes,    Routine Base:  CODE + 0000
```

```
;  354          1456  1 %SBTTL 'PRESERVE_STRING - save string buffer'
;  355          1457  1 ROUTINE PRESERVE_STRING(SRC,DST): NOVALUE=
;  356          1458  1
;  357          1459  1 !++
;  358          1460  1 !
;  359          1461  1 ! FUNCTIONAL DESCRIPTION:
;  360          1462  1 !       This routine saves the contents of the string buffer in dynamically
;  361          1463  1 !       allocated memory.
;  362          1464  1 !
;  363          1465  1 ! INPUT PARAMETERS:
;  364          1466  1 !       SRC             - Pointer to input string (PL/I varying).
;  365          1467  1 !       DST             - Pointer to descriptor that is initialized to
;  366          1468  1 !                         describe the copy.
;  367          1469  1 !
;  368          1470  1 ! IMPLICIT INPUTS:
;  369          1471  1 !       NONE
;  370          1472  1 !
;  371          1473  1 ! OUTPUT PARAMETERS:
;  372          1474  1 !       NONE
;  373          1475  1 !
;  374          1476  1 ! IMPLICIT OUTPUTS:
;  375          1477  1 !       NONE
;  376          1478  1 !
;  377          1479  1 ! ROUTINE VALUE:
;  378          1480  1 !       NONE
;  379          1481  1 !
;  380          1482  1 ! SIDE EFFECTS:
;  381          1483  1 !       Dynamic memory is allocated.
;  382          1484  1 !
;  383          1485  1 !--
;  384          1486  1
;  385          1487  2 BEGIN
;  386          1488  2 MAP
;  387          1489  2         SRC:            REF BBLOCK,     ! Pointer to PL/I varying string
;  388          1490  2         DST:            REF BBLOCK;     ! Pointer to descriptor
;  389          1491  2
;  390          1492  2
;  391          1493  2 ! Initialize output descriptor.
;  392          1494  2 !
;  393          1495  2 DST[DSC$W_LENGTH] = .SRC[VAR_LENGTH];
;  394          1496  2 DST[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  395          1497  2 DST[DSC$B_CLASS] = DSC$K_CLASS_S;
;  396          1498  2
;  397          1499  2
;  398          1500  2 IF .SRC[VAR_LENGTH] NEQ 0
;  399          1501  2 THEN
;  400          1502  3     BEGIN
;  401          1503  3
;  402          1504  3     ! Allocate memory to hold string.
;  403          1505  3     !
;  404          1506  3     DST[DSC$A_POINTER] = GET_VM(.SRC[VAR_LENGTH]);
;  405          1507  3
;  406          1508  3
;  407          1509  3     ! Copy string to allocated memory.
;  408          1510  3     !
;  409          1511  3     CH$MOVE(.SRC[VAR_LENGTH], SRC[VAR_BODY], .DST[DSC$A_POINTER]);
;  410          1512  2     END;
```

```
;  411          1513  1 END;


                                                003C 00000 PRESERVE_STRING:
                                                           .WORD   Save R2,R3,R4,R5                ; 1457
                                    52      08  AC  D0 00002          MOVL    DST, R2             ; 1495
                                    53      04  AC  D0 00006          MOVL    SRC, R3             ;
                                    62          63  B0 0000A          MOVW    (R3), (R2)          ;
                                02  A2    010E  8F  B0 0000D          MOVW    #270, 2(R2)         ; 1496
                                              63  B5 00013          TSTW    (R3)                ; 1500
                                              14  13 00015          BEQL    1$                  ;
                                    7E        63  3C 00017          MOVZWL  (R3), -(SP)         ; 1506
                          00000000G 00        01  FB 0001A          CALLS   #1, GET_VM          ;
                                04  A2        50  D0 00021          MOVL    R0, 4(R2)           ;
                          04  B2    02  A3    63  28 00025          MOVC3   (R3), 2(R3), @4(R2) ; 1511
                                              04  0002B 1$:         RET                         ; 1513

; Routine Size:  44 bytes,    Routine Base:  CODE + 002C
```

```
;  413          1514  1 %SBTTL 'GET_FC - initialize file context area'
;  414          1515  1 ROUTINE GET_FC(SRC)=
;  415          1516  1
;  416          1517  1 !++
;  417          1518  1 !
;  418          1519  1 ! FUNCTIONAL DESCRIPTION:
;  419          1520  1 !       This routine gets a file context area, initializing it.
;  420          1521  1 !
;  421          1522  1 ! INPUT PARAMETERS:
;  422          1523  1 !       SRC             - Pointer to filename string (PL/I varying).
;  423          1524  1 !
;  424          1525  1 ! IMPLICIT INPUTS:
;  425          1526  1 !       NONE
;  426          1527  1 !
;  427          1528  1 ! OUTPUT PARAMETERS:
;  428          1529  1 !       NONE
;  429          1530  1 !
;  430          1531  1 ! IMPLICIT OUTPUTS:
;  431          1532  1 !       NONE
;  432          1533  1 !
;  433          1534  1 ! ROUTINE VALUE:
;  434          1535  1 !       Address of the allocated memory.
;  435          1536  1 !
;  436          1537  1 ! SIDE EFFECTS:
;  437          1538  1 !       LIB$GET_VM errors may be signalled.
;  438          1539  1 !
;  439          1540  1 !--
;  440          1541  1
;  441          1542  2 BEGIN
;  442          1543  2 MAP
;  443          1544  2         SRC:            REF BBLOCK;     ! Pointer to PL/I varying string
;  444          1545  2 LOCAL
;  445          1546  2         DESC:           BBLOCK[8],      ! Descriptor for preserved string
;  446          1547  2         AREA:           REF BBLOCK;     ! Pointer to allocated area
;  447          1548  2
;  448          1549  2
;  449          1550  2 PRESERVE_STRING(.SRC, DESC);           ! Get dynamic copy of string
;  450          1551  2 AREA = GET_ZERO_VM(FC_S_AREA);
;  451        P 1552  2 $FAB_INIT(
;  452        P 1553  2     FAB=AREA[FC_FAB],
;  453        P 1554  2     FNA=.DESC[DSC$A_POINTER],
;  454        P 1555  2     FNS=.DESC[DSC$W_LENGTH],
;  455          1556  2     NAM=AREA[FC_NAM]);
;  456        P 1557  2 $RAB_INIT(
;  457        P 1558  2     RAB=AREA[FC_RAB],
;  458          1559  2     FAB=AREA[FC_FAB]);
;  459        P 1560  2 $NAM_INIT(
;  460        P 1561  2     NAM=AREA[FC_NAM],
;  461        P 1562  2     RLF=AREA[FC_RLF],
;  462        P 1563  2     ESA=AREA[FC_ESA],
;  463        P 1564  2     ESS=NAM$C_MAXRSS,
;  464        P 1565  2     RSA=AREA[FC_RSA],
;  465          1566  2     RSS=NAM$C_MAXRSS);
;  466        P 1567  2 $NAM_INIT(
;  467          1568  2     NAM=AREA[FC_RLF]);
;  468          1569  2 .AREA
;  469          1570  1 END;
```

```
                                                  01FC 00000 GET_FC:  .WORD   Save R2,R3,R4,R5,R6,R7,R8                ;1515
                                        5E                 08  C2 00002        SUBL2   #8, SP                          ;
                                                           5E  DD 00005        PUSHL   SP                              ;1550
                                                     04    AC  DD 00007        PUSHL   SRC                             ;
                                                C6   AF      02  FB 0000A        CALLS   #2, PRESERVE_STRING             ;
                                                     7E 0354 8F  3C 0000E        MOVZWL  #852, -(SP)                     ;1551
                                          00000000G  00      01  FB 00013        CALLS   #1, GET_ZERO_VM                 ;
                                                     57      50  D0 0001A        MOVL    R0, AREA                        ;
                0050    8F          00               6E      00  2C 0001D        MOVC5   #0, (SP), #0, #80, (AREA)       ;1556
                                                                67     00024                                            |
                                                     67 5003 8F  B0 00025        MOVW    #20483, (AREA)                  |
                                                16   A7      02  90 0002A        MOVB    #2, 22(AREA)                    |
                                                1F   A7      02  90 0002E        MOVB    #2, 31(AREA)                    |
                                                     56 0094 C7  9E 00032        MOVAB   148(R7), R6                     |
                                                28   A7      56  D0 00037        MOVL    R6, 40(AREA)                    |
                                                2C   A7   04 AE  D0 0003B        MOVL    DESC+4, 44(AREA)                |
                                                34   A7      6E  90 00040        MOVB    DESC, 52(AREA)                  |
                                                     58   50 A7  9E 00044        MOVAB   80(AREA), R8                    ;1559
                0044    8F          00               6E      00  2C 00048        MOVC5   #0, (SP), #0, #68, (R8)         |
                                                                68     0004F                                            |
                                                     68 4401 8F  B0 00050        MOVW    #17409, (R8)                    |
                                                3C   A8      57  D0 00055        MOVL    AREA, 60(R8)                    |
                0060    8F          00               6E      00  2C 00059        MOVC5   #0, (SP), #0, #96, (R6)         ;1566
                                                                66     00060                                            |
                                                     66 6002 8F  B0 00061        MOVW    #24578, (R6)                    |
                                                02   A6      01  8E 00066        MNEGB   #1, 2(R6)                       |
                                                04   A6 0254 C7  9E 0006A        MOVAB   596(R7), 4(R6)                  |
                                                0A   A6      01  8E 00070        MNEGB   #1, 10(R6)                      |
                                                0C   A6 0154 C7  9E 00074        MOVAB   340(R7), 12(R6)                 |
                                                     58 00F4 C7  9E 0007A        MOVAB   244(R7), R8                     |
                                                10   A6      58  D0 0007F        MOVL    R8, 16(R6)                      |
                0060    8F          00               6E      00  2C 00083        MOVC5   #0, (SP), #0, #96, (R8)         ;1568
                                                                68     0008A                                            |
                                                     68 6002 8F  B0 0008B        MOVW    #24578, (R8)                    |
                                                     50      57  D0 00090        MOVL    AREA, R0                        ;1570
                                                                04 00093        RET                                     ;

; Routine Size:  148 bytes,    Routine Base:  CODE + 0058

;  470          1571  1
```

```
;  472          1572  1 %SBTTL 'COPY_FC - copy file context area'
;  473          1573  1 ROUTINE COPY_FC ( FC : REF BBLOCK,
;  474          1574  1                   NEW_FC : REF BBLOCK, FILE_LEN, FILE_ADDR ) : NOVALUE =
;  475          1575  1
;  476          1576  1 !++
;  477          1577  1 !
;  478          1578  1 ! FUNCTIONAL DESCRIPTION:
;  479          1579  1 !       This routine allocates a file context area and copies the
;  480          1580  1 !       file context area passed as a parameter into it
;  481          1581  1 !
;  482          1582  1 ! INPUT PARAMETERS:
;  483          1583  1 !       FC              - File context block to make a copy of
;  484          1584  1 !       NEW_FC          - File context block to copy to
;  485          1585  1 !       FILE_LEN        - length of file name string
;  486          1586  1 !       FILE_ADDR       - address of file name string
;  487          1587  1 !
;  488          1588  1 ! IMPLICIT INPUTS:
;  489          1589  1 !       NONE
;  490          1590  1 !
;  491          1591  1 ! OUTPUT PARAMETERS:
;  492          1592  1 !       NONE
;  493          1593  1 !
;  494          1594  1 ! IMPLICIT OUTPUTS:
;  495          1595  1 !       NONE
;  496          1596  1 !
;  497          1597  1 ! ROUTINE VALUE:
;  498          1598  1 !
;  499          1599  1 ! SIDE EFFECTS:
;  500          1600  1 !       LIB$GET_VM errors may be signalled.
;  501          1601  1 !
;  502          1602  1 !--
;  503          1603  1
;  504          1604  2 BEGIN
;  505          1605  2
;  506          1606  2 LITERAL
;  507          1607  2     LEN1 = %CHARCOUNT ('.]['),
;  508          1608  2     LEN2 = %CHARCOUNT ('.><'),
;  509          1609  2     LEN3 = %CHARCOUNT ('.>['),
;  510          1610  2     LEN4 = %CHARCOUNT ('.]<');
;  511          1611  2
;  512          1612  2 BIND
;  513          1613  2     STR1 = UPLIT BYTE ('.]['),
;  514          1614  2     STR2 = UPLIT BYTE ('.><'),
;  515          1615  2     STR3 = UPLIT BYTE ('.>['),
;  516          1616  2     STR4 = UPLIT BYTE ('.]<') ;
;  517          1617  2
;  518          1618  2 LOCAL
;  519          1619  2     STR1_PTR            : REF VECTOR [,BYTE] ,
;  520          1620  2     STR2_PTR            : REF VECTOR [,BYTE] ,
;  521          1621  2     STR3_PTR            : REF VECTOR [,BYTE] ,
;  522          1622  2     STR4_PTR            : REF VECTOR [,BYTE] ,
;  523          1623  2     NEW_FILE_LEN        : LONG ,
;  524          1624  2     WORK_FILE_NAME      : REF VECTOR [,BYTE] ,
;  525          1625  2     FILE_NAME           : REF BBLOCK ,          ! Pointer to FNS buffer
;  526          1626  2     FAB                 : REF BBLOCK ,
;  527          1627  2     NAM                 : REF BBLOCK ,
;  528          1628  2     RAB                 : REF BBLOCK ,
```

```
;  529          1629  2     CONTINUE            : BYTE ;
;  530          1630  2     STR_FOUND           : BYTE ;
;  531          1631  2
;  532          1632  2
;  533          1633  2 FAB = FC [ FC_FAB ] ;
;  534          1634  2
;  535          1635  2 !
;  536          1636  2 ! Allocate memory for the file name string and copy the string.
;  537          1637  2 ! ******************************************************************************
;  538          1638  2 ! ******************************************************************************
;  539          1639  2 ! In the process of coppyng it strip out all "."[" , "."<", ">[" ,and ">."
;  540          1640  2 ! strings.  This puts a restriction on BACKUP since we will no longer be able
;  541          1641  2 ! to deal with a concraled device which hides 8 directorie levels but at the
;  542          1642  2 ! present time there is no time for a more elegant solution.  This problem
;  543          1643  2 ! was reported in QAR #1212 from the FT2 data base.  Search lists with
;  544          1644  2 ! concealed devices do not work.
;  545          1645  2 ! ******************************************************************************
;  546          1646  2 ! ******************************************************************************
;  547          1647  2 !
;  548          1648  2
;  549          1649  2 WORK_FILE_NAME = GET_VM ( .FILE_LEN ) ;
;  550          1650  2 CH$MOVE ( .FILE_LEN, .FILE_ADDR, .WORK_FILE_NAME ) ;
;  551          1651  2 NEW_FILE_LEN = .FILE_LEN ;
;  552          1652  2
;  553          1653  2 CONTINUE = 1 ;
;  554          1654  2 STR_FOUND = 0 ;
;  555          1655  2 WHILE .CONTINUE DO
;  556          1656  3     BEGIN
;  557          1657  3     STR1_PTR = CH$FIND_SUB ( .NEW_FILE_LEN, .WORK_FILE_NAME, LEN1, STR1 );
;  558          1658  3     STR2_PTR = CH$FIND_SUB ( .NEW_FILE_LEN, .WORK_FILE_NAME, LEN2, STR2 );
;  559          1659  3     STR3_PTR = CH$FIND_SUB ( .NEW_FILE_LEN, .WORK_FILE_NAME, LEN3, STR3 );
;  560          1660  3     STR4_PTR = CH$FIND_SUB ( .NEW_FILE_LEN, .WORK_FILE_NAME, LEN4, STR4 );
;  561          1661  3
;  562          1662  3     IF NOT CH$FAIL (.STR1_PTR)
;  563          1663  3     THEN
;  564          1664  4         BEGIN
;  565          1665  4         STR_FOUND = 1 ;
;  566          1666  4         STR1_PTR = .STR1_PTR + 1 ;      ! Skip past "."
;  567          1667  4         NEW_FILE_LEN = .NEW_FILE_LEN - LEN1 + 1 ;
;  568          1668  4         CH$MOVE ( (.WORK_FILE_NAME + .NEW_FILE_LEN - .STR1_PTR),
;  569          1669  4                   .STR1_PTR + LEN1 - 1,
;  570          1670  4                   .STR1_PTR ) ;
;  571          1671  4         END
;  572          1672  3     ELSE IF NOT CH$FAIL (.STR2_PTR)
;  573          1673  3     THEN
;  574          1674  4         BEGIN
;  575          1675  4         STR_FOUND = 1 ;
;  576          1676  4         STR2_PTR = .STR2_PTR + 1 ;      ! Skip past "."
;  577          1677  4         NEW_FILE_LEN = .NEW_FILE_LEN - LEN2 + 1 ;
;  578          1678  4         CH$MOVE ( (.WORK_FILE_NAME + .NEW_FILE_LEN - .STR2_PTR),
;  579          1679  4                   .STR2_PTR + LEN2 - 1,
;  580          1680  4                   .STR2_PTR ) ;
;  581          1681  4         END
;  582          1682  3     ELSE IF NOT CH$FAIL ( .STR3_PTR )
;  583          1683  3     THEN
;  584          1684  4         BEGIN
;  585          1685  4         STR_FOUND = 1 ;
```

```
;  586          1686  4          STR3_PTR = .STR3_PTR + 1 ;          ! Skip past "."
;  587          1687  4          NEW_FILE_LEN = .NEW_FILE_LEN - LEN3 + 1;
;  588          1688  4          CH$MOVE ( (.WORK_FILE_NAME + .NEW_FILE_LEN - .STR3_PTR),
;  589          1689  4                    .STR3_PTR + LEN3 - 1,
;  590          1690  4                    .STR3_PTR ) ;
;  591          1691  4          STR2_PTR = CH$FIND_CH ( .WORK_FILE_NAME + .NEW_FILE_LEN - .STR3_PTR,
;  592          1692  4                                  .STR3_PTR,
;  593          1693  4                                  %C']' ) ;
;  594          1694  4          IF NOT CH$FAIL (.STR2_PTR)
;  595          1695  4          THEN
;  596          1696  4              STR2_PTR[0] = %C'>' ;
;  597          1697  4          END
;  598          1698  3      ELSE IF NOT CH$FAIL ( .STR4_PTR )
;  599          1699  3      THEN
;  600          1700  4          BEGIN
;  601          1701  4          STR_FOUND = 1 ;
;  602          1702  4          STR4_PTR = .STR4_PTR + 1 ;          ! Skip past "."
;  603          1703  4          NEW_FILE_LEN = .NEW_FILE_LEN - LEN4 + 1 ;
;  604          1704  4          CH$MOVE ( (.WORK_FILE_NAME + .NEW_FILE_LEN - .STR4_PTR),
;  605          1705  4                    .STR4_PTR + LEN4 - 1 ,
;  606          1706  4                    .STR4_PTR ) ;
;  607          1707  4          STR2_PTR = CH$FIND_CH ( .WORK_FILE_NAME + .NEW_FILE_LEN - .STR4_PTR,
;  608          1708  4                                  .STR4_PTR,
;  609          1709  4                                  %C'>' ) ;
;  610          1710  4          IF NOT CH$FAIL (.STR2_PTR)
;  611          1711  4          THEN
;  612          1712  4              STR2_PTR[0] = %C']' ;
;  613          1713  4          END
;  614          1714  3      ELSE CONTINUE = 0 ;
;  615          1715  3
;  616          1716  2      END ;
;  617          1717  2
;  618          1718  2  IF .STR_FOUND
;  619          1719  2  THEN
;  620          1720  3      BEGIN
;  621          1721  3      FILE_NAME = GET_VM ( .NEW_FILE_LEN ) ;
;  622          1722  3      CH$MOVE ( .NEW_FILE_LEN, .WORK_FILE_NAME, .FILE_NAME ) ;
;  623          1723  3      FREE_VM ( .FILE_LEN, .WORK_FILE_NAME ) ;
;  624          1724  3      END
;  625          1725  2  ELSE
;  626          1726  2      FILE_NAME = .WORK_FILE_NAME ;
;  627          1727  2
;  628          1728  2  CH$MOVE ( FC_S_AREA, .FC, .NEW_FC ) ;
;  629          1729  2
;  630          1730  2  FAB = NEW_FC [ FC_FAB ] ;
;  631          1731  2  NAM = NEW_FC [ FC_NAM ] ;
;  632          1732  2  RAB = NEW_FC [ FC_RAB ] ;
;  633          1733  2
;  634          1734  2  FAB[FAB$B_FNS] = .NEW_FILE_LEN ;
;  635          1735  2  FAB[FAB$L_FNA] = .FILE_NAME ;
;  636          1736  2  FAB[FAB$L_NAM] = .NAM ;
;  637          1737  2
;  638          1738  2  RAB[RAB$L_FAB] = .FAB ;
;  639          1739  2
;  640          1740  2  NAM[NAM$L_RLF]  = NEW_FC [ FC_RLF ] ;
;  641          1741  2  NAM[NAM$L_ESA]  = NEW_FC [ FC_ESA ] ;
;  642          1742  2  NAM[NAM$B_ESS]  = NAM$C_MAXRSS ;
```

```
;  643          1743  2 NAM[NAM$L_RSA]  = NEW_FC [ FC_RSA ] ;
;  644          1744  2 NAM[NAM$L_NODE] = .NAM[NAM$L_ESA] ;
;  645          1745  2 NAM[NAM$L_DEV]  = .NAM[NAM$L_NODE]  + .NAM[NAM$B_NODE] ;
;  646          1746  2 NAM[NAM$L_DIR]  = .NAM[NAM$L_DEV]  + .NAM[NAM$B_DEV] ;
;  647          1747  2 NAM[NAM$L_NAME] = .NAM[NAM$L_DIR]  + .NAM[NAM$B_DIR] ;
;  648          1748  2 NAM[NAM$L_TYPE] = .NAM[NAM$L_NAME] + .NAM[NAM$B_NAME] ;
;  649          1749  2 NAM[NAM$L_VER]  = .NAM[NAM$L_TYPE] + .NAM[NAM$B_TYPE] ;
;  650          1750  2
;  651          1751  2 NAM[NAM$B_RSS] = NAM$C_MAXRSS ;
;  652          1752  2 NAM[NAM$V_SLPARSE] = 0 ;
;  653          1753  2 NAM[NAM$V_NOCONCEAL] = 0 ;
;  654          1754  2 NAM[NAM$V_SVCTX] = 0 ;
;  655          1755  2 NAM[NAM$L_WCC] = 0 ;
;  656          1756  2
;  657          1757  2 RETURN ;
;  658          1758  2
;  659          1759  1 END;


                                                5B  5D  2E  000EC P.AAA:  .ASCII  \.][\                                 ;
                                                3C  3E  2E  000EF P.AAB:  .ASCII  \.><\                                 ;
                                                5B  3E  2E  000F2 P.AAC:  .ASCII  \.>[\                                 ;
                                                3C  5D  2E  000F5 P.AAD:  .ASCII  \.]<\                                 ;

                                                                  STR1=             P.AAA
                                                                  STR2=             P.AAB
                                                                  STR3=             P.AAC
                                                                  STR4=             P.AAD


                                                      0FFC 00000 COPY_FC:.WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11  ; 1573
                                                5E          10  C2 00002           SUBL2    #16, SP                      ;
                                          0C    AE        04  AC  D0 00005           MOVL     FC, FAB                      ; 1633
                                                          0C  AC  DD 0000A           PUSHL    FILE_LEN                     ; 1649
                                    00000000G   00            01  FB 0000D           CALLS    #1, GET_VM                   ;
                                                56            50  D0 00014           MOVL     R0, WORK_FILE_NAME           ;
                                    66    10    BC        0C  AC  28 00017           MOVC3    FILE_LEN, @FILE_ADDR, (WORK_FILE_NAME) ; 1650
                                                57        0C  AC  D0 0001D           MOVL     FILE_LEN, NEW_FILE_LEN       ; 1651
                                          08    AE            01  90 00021           MOVB     #1, CONTINUE                 ; 1653
                                                          04  AE  94 00025           CLRB     STR_FOUND                    ; 1654
                                                03        08  AE  E8 00028 1$:       BLBS     CONTINUE, 2$                 ; 1655
                                                                00D3  31 0002C       BRW      15$                          ;
                                    66    57    C1        AF      03  39 0002F 2$:   MATCHC   #3, STR1, NEW_FILE_LEN, (WORK_FILE_NAME) ; 1657
                                                                    03  13 00035       BEQL     3$                           ;
                                                53                  03  D0 00037       MOVL     #3, R3                       ;
                                                                    53  D7 0003A 3$:   DECL     R3                           ;
                                                6E                  73  3E 0003C       MOVAW    -(R3), STR1_PTR              ;
                                    66    57    B4        AF      03  39 0003F       MATCHC   #3, STR2, NEW_FILE_LEN, (WORK_FILE_NAME) ; 1658
                                                                    03  13 00045       BEQL     4$                           ;
                                                53                  03  D0 00047       MOVL     #3, R3                       ;
                                                                    53  D7 0004A 4$:   DECL     R3                           ;
                                                58                  73  3E 0004C       MOVAW    -(R3), STR2_PTR              ;
                                    66    57    A7        AF      03  39 0004F       MATCHC   #3, STR3, NEW_FILE_LEN, (WORK_FILE_NAME) ; 1659
                                                                    03  13 00055       BEQL     5$                           ;
                                                53                  03  D0 00057       MOVL     #3, R3                       ;
                                                                    53  D7 0005A 5$:   DECL     R3                           ;
```

```
                          5B          73  3E 0005C          MOVAW   -(R3), STR3_PTR                                     ;
      66        57    9A  AF          03  39 0005F          MATCHC  #3, STR4, NEW_FILE_LEN, (WORK_FILE_NAME)            ; 1660
                                      03  13 00065          BEQL    6$                                                  ;
                          53          03  D0 00067          MOVL    #3, R3                                              ;
                                      53  D7 0006A 6$:      DECL    R3                                                  ;
                          5A          73  3E 0006C          MOVAW   -(R3), STR4_PTR                                     ;
                                      6E  D5 0006F          TSTL    STR1_PTR                                            ; 1662
                                      1A  13 00071          BEQL    8$                                                  ;
                    04    AE          01  90 00073          MOVB    #1, STR_FOUND                                       ; 1665
                                      6E  D6 00077          INCL    STR1_PTR                                            ; 1666
                                      57  D7 00079          DECL    NEW_FILE_LEN                                        ; 1667
                          50        7746  9E 0007B          MOVAB   -(NEW_FILE_LEN)[WORK_FILE_NAME], R0                 ; 1668
                          50          6E  C2 0007F          SUBL2   STR1_PTR, R0                                        ;
                59        6E          02  C1 00082          ADDL3   #2, STR1_PTR, R9                                    ; 1670
         00     BE        69          50  28 00086          MOVC3   R0, (R9), @STR1_PTR                                 ;
                                      9B  11 0008B 7$:      BRB     1$                                                  ; 1662
                                      58  D5 0008D 8$:      TSTL    STR2_PTR                                            ; 1672
                                      16  13 0008F          BEQL    9$                                                  ;
                    04    AE          01  90 00091          MOVB    #1, STR_FOUND                                       ; 1675
                                      58  D6 00095          INCL    STR2_PTR                                            ; 1676
                                      57  D7 00097          DECL    NEW_FILE_LEN                                        ; 1677
                          50        7746  9E 00099          MOVAB   -(NEW_FILE_LEN)[WORK_FILE_NAME], R0                 ; 1678
                          50          58  C2 0009D          SUBL2   STR2_PTR, R0                                        ;
                68    02  A8          50  28 000A0          MOVC3   R0, 2(STR2_PTR), (STR2_PTR)                         ; 1680
                                      59  11 000A5          BRB     14$                                                 ; 1672
                                      5B  D5 000A7 9$:      TSTL    STR3_PTR                                            ; 1682
                                      27  13 000A9          BEQL    11$                                                 ;
                    04    AE          01  90 000AB          MOVB    #1, STR_FOUND                                       ; 1685
                                      5B  D6 000AF          INCL    STR3_PTR                                            ; 1686
                                      57  D7 000B1          DECL    NEW_FILE_LEN                                        ; 1687
                          59        7746  9E 000B3          MOVAB   -(NEW_FILE_LEN)[WORK_FILE_NAME], R9                 ; 1688
                          59          5B  C2 000B7          SUBL2   STR3_PTR, R9                                        ;
                6B    02  AB          59  28 000BA          MOVC3   R9, 2(STR3_PTR), (STR3_PTR)                         ; 1690
                6B        59      5D  8F  3A 000BF          LOCC    #93, R9, (STR3_PTR)                                 ; 1691
                                      02  12 000C4          BNEQ    10$                                                 ;
                                      51  D4 000C6          CLRL    R1                                                  ;
                          58          51  D0 000C8 10$:     MOVL    R1, STR2_PTR                                        ;
                                      BE  13 000CB          BEQL    7$                                                  ; 1694
                          68          3E  90 000CD          MOVB    #62, (STR2_PTR)                                     ; 1696
                                      B9  11 000D0          BRB     7$                                                  ; 1682
                                      5A  D5 000D2 11$:     TSTL    STR4_PTR                                            ; 1698
                                      27  13 000D4          BEQL    13$                                                 ;
                    04    AE          01  90 000D6          MOVB    #1, STR_FOUND                                       ; 1701
                                      5A  D6 000DA          INCL    STR4_PTR                                            ; 1702
                                      57  D7 000DC          DECL    NEW_FILE_LEN                                        ; 1703
                          59        7746  9E 000DE          MOVAB   -(NEW_FILE_LEN)[WORK_FILE_NAME], R9                 ; 1704
                          59          5A  C2 000E2          SUBL2   STR4_PTR, R9                                        ;
                6A    02  AA          59  28 000E5          MOVC3   R9, 2(STR4_PTR), (STR4_PTR)                         ; 1706
                6A        59          3E  3A 000EA          LOCC    #62, R9, (STR4_PTR)                                 ; 1707
                                      02  12 000EE          BNEQ    12$                                                 ;
                                      51  D4 000F0          CLRL    R1                                                  ;
                          58          51  D0 000F2 12$:     MOVL    R1, STR2_PTR                                        ;
                                      94  13 000F5          BEQL    7$                                                  ; 1710
                          68      5D  8F  90 000F7          MOVB    #93, (STR2_PTR)                                     ; 1712
                                      8E  11 000FB          BRB     7$                                                  ; 1698
                                  08  AE  94 000FD 13$:     CLRB    CONTINUE                                            ; 1714
                                      89  11 00100 14$:     BRB     7$                                                  ; 1655
```

```
                                 1E      04    AE  E9 00102 15$:    BLBC    STR_FOUND, 16$                                  ; 1718
                                               57  DD 00106        PUSHL   NEW_FILE_LEN                                    ; 1721
                    00000000G    00             01  FB 00108        CALLS   #1, GET_VM                                      ;
                                 58             50  D0 0010F        MOVL    R0, FILE_NAME                                   ;
                68               66             57  28 00112        MOVC3   NEW_FILE_LEN, (WORK_FILE_NAME), (FILE_NAME)     ; 1722
                                               56  DD 00116        PUSHL   WORK_FILE_NAME                                  ; 1723
                                       0C      AC  DD 00118        PUSHL   FILE_LEN                                        ;
                    00000000G    00             02  FB 0011B        CALLS   #2, FREE_VM                                     ;
                                               03  11 00122        BRB     17$                                             ; 1718
                                 58             56  D0 00124 16$:    MOVL    WORK_FILE_NAME, FILE_NAME                       ; 1726
                                 56       08    AC  D0 00127 17$:    MOVL    NEW_FC, R6                                      ; 1728
                66         04    BC     0354   8F  28 0012B        MOVC3   #852, @FC, (R6)                                 ;
                           0C    AE             56  D0 00132        MOVL    R6, FAB                                         ; 1730
                                 50     0094    C6  9E 00136        MOVAB   148(R6), NAM                                    ; 1731
                                 51       50    A6  9E 0013B        MOVAB   80(R6), RAB                                     ; 1732
                52         0C    AE             34  C1 0013F        ADDL3   #52, FAB, R2                                    ; 1734
                                               62  57  90 00144        MOVB    NEW_FILE_LEN, (R2)                              ;
                52         0C    AE             2C  C1 00147        ADDL3   #44, FAB, R2                                    ; 1735
                                               62  58  D0 0014C        MOVL    FILE_NAME, (R2)                                 ;
                52         0C    AE             28  C1 0014F        ADDL3   #40, FAB, R2                                    ; 1736
                                               62  50  D0 00154        MOVL    NAM, (R2)                                       ;
                           3C    A1       0C    AE  D0 00157        MOVL    FAB, 60(RAB)                                    ; 1738
                           10    A0     00F4    C6  9E 0015C        MOVAB   244(R6), 16(NAM)                                ; 1740
                           0C    A0     0154    C6  9E 00162        MOVAB   340(R6), 12(NAM)                                ; 1741
                           0A    A0             01  8E 00168        MNEGB   #1, 10(NAM)                                     ; 1742
                           04    A0     0254    C6  9E 0016C        MOVAB   596(R6), 4(NAM)                                 ; 1743
                           40    A0       0C    A0  D0 00172        MOVL    12(NAM), 64(NAM)                                ; 1744
                                 51       38    A0  9A 00177        MOVZBL  56(NAM), R1                                     ; 1745
                           44    A0       40 B041  9E 0017B        MOVAB   @64(NAM)[R1], 68(NAM)                           ;
                                 51       39    A0  9A 00181        MOVZBL  57(NAM), R1                                     ; 1746
                           48    A0       44 B041  9E 00185        MOVAB   @68(NAM)[R1], 72(NAM)                           ;
                                 51       3A    A0  9A 0018B        MOVZBL  58(NAM), R1                                     ; 1747
                           4C    A0       48 B041  9E 0018F        MOVAB   @72(NAM)[R1], 76(NAM)                           ;
                                 51       3B    A0  9A 00195        MOVZBL  59(NAM), R1                                     ; 1748
                           50    A0       4C B041  9E 00199        MOVAB   @76(NAM)[R1], 80(NAM)                           ;
                                 51       3C    A0  9A 0019F        MOVZBL  60(NAM), R1                                     ; 1749
                           54    A0       50 B041  9E 001A3        MOVAB   @80(NAM)[R1], 84(NAM)                           ;
                           02    A0             01  8E 001A9        MNEGB   #1, 2(NAM)                                      ; 1751
                           08    A0             30  8A 001AD        BICB2   #48, 8(NAM)                                     ; 1753
                           33    A0       80    8F  8A 001B1        BICB2   #128, 51(NAM)                                   ; 1754
                                          30    A0  D4 001B6        CLRL    48(NAM)                                         ; 1755
                                                    04 001B9        RET                                                     ; 1759

; Routine Size:  442 bytes,    Routine Base:  CODE + 00F8

;  660          1760  1
```

```
;  662          1761  1 %SBTTL 'INIT_FC - init file context area'
;  663          1762  1 ROUTINE INIT_FC ( FC : REF BBLOCK , FILE_LEN, FILE_ADDR ) : NOVALUE =
;  664          1763  1
;  665          1764  1 !++
;  666          1765  1 !
;  667          1766  1 ! FUNCTIONAL DESCRIPTION:
;  668          1767  1 !       This routine is used to reinitialize an already existing FC
;  669          1768  1 !       for use parsing another file name string.  The name from the
;  670          1769  1 !       previous file parse is propagated in the related name block
;  671          1770  1 !       so that the name parsing will be good and sticky.
;  672          1771  1 !
;  673          1772  1 ! INPUT PARAMETERS:
;  674          1773  1 !       FC              - File context block from last parse
;  675          1774  1 !       FILE_LEN        - Length of file name string
;  676          1775  1 !       FILE_ADDR       - Address of file name string
;  677          1776  1 !
;  678          1777  1 ! IMPLICIT INPUTS:
;  679          1778  1 !       NONE
;  680          1779  1 !
;  681          1780  1 ! OUTPUT PARAMETERS:
;  682          1781  1 !       NONE
;  683          1782  1 !
;  684          1783  1 ! IMPLICIT OUTPUTS:
;  685          1784  1 !
;  686          1785  1 !       FC will be reinitialized and there will be a new related nam
;  687          1786  1 !       block hanging off the FC's NAM block.
;  688          1787  1 !
;  689          1788  1 ! ROUTINE VALUE:
;  690          1789  1 !
;  691          1790  1 ! SIDE EFFECTS:
;  692          1791  1 !       LIB$GET_VM errors may be signalled.
;  693          1792  1 !
;  694          1793  1 !--
;  695          1794  1
;  696          1795  2 BEGIN
;  697          1796  2
;  698          1797  2 LOCAL
;  699          1798  2     DESC        : BBLOCK [8] ,
;  700          1799  2     NEW_RNAM    : REF BBLOCK ,          ! Pointer to allocated area
;  701          1800  2     RLF_NAME    : REF BBLOCK ,          ! Pointer to FNS buffer
;  702          1801  2     FILE_NAME   : REF BBLOCK ,          ! Pointer to FNA buffer
;  703          1802  2     RLF_LEN     : BYTE ,                ! File name length holder
;  704          1803  2     FAB         : REF BBLOCK ,
;  705          1804  2     NAM         : REF BBLOCK ;
;  706          1805  2
;  707          1806  2
;  708          1807  2 FAB = FC [ FC_FAB ] ;
;  709          1808  2 NAM = FC [ FC_NAM ] ;
;  710          1809  2
;  711          1810  2 !
;  712          1811  2 ! Get memory for the new FNS
;  713          1812  2 !
;  714          1813  2 FILE_NAME = GET_VM ( .FILE_LEN ) ;
;  715          1814  2 CH$MOVE ( .FILE_LEN, .FILE_ADDR, .FILE_NAME ) ;
;  716          1815  2
;  717          1816  2
;  718          1817  2 !
```

```
;  719          1818  2 ! Allocate memory for the new related NAM block, init it
;  720          1819  2 ! and link it into the chain.
;  721          1820  2 !
;  722          1821  2 NEW_RNAM = GET_VM (NAM$C_BLN) ;
;  723          1822  2
;  724          1823  2 RLF_LEN = .FAB[FAB$B_FNS] ;
;  725          1824  2 RLF_NAME = GET_VM ( .RLF_LEN ) ;
;  726          1825  2 CH$MOVE ( .RLF_LEN, .FAB[FAB$L_FNA], .RLF_NAME ) ;
;  727          1826  2
;  728        P 1827  2 $NAM_INIT (
;  729        P 1828  2       NAM = .NEW_RNAM,
;  730        P 1829  2       RLF = .NAM[NAM$L_RLF] ,
;  731        P 1830  2       RSS = .RLF_LEN ,
;  732        P 1831  2       RSA = .RLF_NAME
;  733          1832  2       ) ;
;  734          1833  2 NEW_RNAM[NAM$B_RSL] = .RLF_LEN ;
;  735          1834  2
;  736          1835  2 NAM[NAM$L_RLF] = .NEW_RNAM ;
;  737          1836  2
;  738          1837  2 !
;  739          1838  2 ! Now give up the memory that the old FNA used and get a new
;  740          1839  2 ! hunk of the correct length.
;  741          1840  2 !
;  742          1841  2 FREE_VM ( .FAB[FAB$B_FNS], .FAB[FAB$L_FNA] ) ;
;  743          1842  2
;  744          1843  2 !
;  745          1844  2 ! Now init the FAB
;  746          1845  2 !
;  747        P 1846  2 $FAB_INIT(
;  748        P 1847  2       FAB = .FAB ,
;  749        P 1848  2       FNA = .FILE_NAME,
;  750        P 1849  2       FNS = .FILE_LEN,
;  751          1850  2       NAM = .NAM );
;  752          1851  2
;  753          1852  2
;  754          1853  2 RETURN ;
;  755          1854  2
;  756          1855  1 END;


                                    0FFC 00000 INIT_FC:.WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 1762
                          5E          08   C2 00002          SUBL2   #8, SP                                  ;
                          57    04    AC   D0 00005          MOVL    FC, FAB                                 ; 1807
                 58       04    AC 00000094 8F   C1 00009    ADDL3   #148, FC, NAM                           ; 1808
                                       08   AC   DD 00012    PUSHL   FILE_LEN                                ; 1813
                    00000000G  00          01   FB 00015    CALLS   #1, GET_VM                              ;
                          5B          50   D0 0001C          MOVL    R0, FILE_NAME                           ;
                 6B       0C    BC       08   AC   28 0001F MOVC3   FILE_LEN, @FILE_ADDR, (FILE_NAME)       ; 1814
                          7E          60   8F   9A 00025    MOVZBL  #96, -(SP)                              ; 1821
                    00000000G  00          01   FB 00029    CALLS   #1, GET_VM                              ;
                          56          50   D0 00030          MOVL    R0, NEW_RNAM                            ;
                          59    34    A7   90 00033          MOVB    52(FAB), RLF_LEN                        ; 1823
                          7E          59   9A 00037          MOVZBL  RLF_LEN, -(SP)                          ; 1824
                    00000000G  00          01   FB 0003A    CALLS   #1, GET_VM                              ;
```

```
                                5A                  50  D0 00041        MOVL    R0, RLF_NAME                            ;
                                50                  59  9A 00044        MOVZBL  RLF_LEN, R0                             ; 1825
                          6A    2C    B7            50  28 00047        MOVC3   R0, @44(FAB), (RLF_NAME)                ;
    0060  8F              00          6E            00  2C 0004C        MOVC5   #0, (SP), #0, #96, (NEW_RNAM)           ; 1832
                                                    66     00053                                                        ;
                                66      6002        8F  B0 00054        MOVW    #24578, (NEW_RNAM)                      ;
                          02    A6                  59  90 00059        MOVB    RLF_LEN, 2(NEW_RNAM)                    ;
                          04    A6                  5A  D0 0005D        MOVL    RLF_NAME, 4(NEW_RNAM)                   ;
                          10    A6      10          A8  D0 00061        MOVL    16(NAM), 16(NEW_RNAM)                   ;
                          03    A6                  59  90 00066        MOVB    RLF_LEN, 3(NEW_RNAM)                    ; 1833
                          10    A8                  56  D0 0006A        MOVL    NEW_RNAM, 16(NAM)                       ; 1835
                                      2C    A7      DD  0006E        PUSHL   44(FAB)                                 ; 1841
                                7E    34    A7      9A  00071        MOVZBL  52(FAB), -(SP)                          ;
                    00000000G   00          02      FB  00075        CALLS   #2, FREE_VM                             ;
    0050  8F              00          6E            00  2C 0007C        MOVC5   #0, (SP), #C, #80, (FAB)                ; 1850
                                                    67     00083                                                        ;
                                67      5003        8F  B0 00084        MOVW    #20483, (FAB)                           ;
                          16    A7                  02  90 00089        MOVB    #2, 22(FAB)                             ;
                          1F    A7                  02  90 0008D        MOVB    #2, 31(FAB)                             ;
                          28    A7                  58  D0 00091        MOVL    NAM, 40(FAB)                            ;
                          2C    A7                  5B  D0 00095        MOVL    FILE_NAME, 44(FAB)                      ;
                          34    A7      08          AC  90 00099        MOVB    FILE_LEN, 52(FAB)                       ;
                                                        04 0009E        RET                                             ; 1855

; Routine Size:  159 bytes,    Routine Base:  CODE + 02B2

;  757          1856  1
```

```
;  759          1857  1 %SBTTL 'PARSE_EXC_SEL - parse EXCLUDE or SELECT qualifier'
;  760          1858  1 ROUTINE PARSE_EXC_SEL(SRC,QUAL_DESC,ESL_DESC): NOVALUE=
;  761          1859  1
;  762          1860  1 !++
;  763          1861  1 !
;  764          1862  1 ! FUNCTIONAL DESCRIPTION:
;  765          1863  1 !       This routine parses and checks the value of an EXCLUDE or SELECT
;  766          1864  1 !       qualifier.
;  767          1865  1 !
;  768          1866  1 ! INPUT PARAMETERS:
;  769          1867  1 !       SRC             - Pointer to the string to be parsed (PL/I varying).
;  770          1868  1 !       QUAL_DESC       - Descriptor for the qualifier name.
;  771          1869  1 !
;  772          1870  1 ! IMPLICIT INPUTS:
;  773          1871  1 !       NONE
;  774          1872  1 !
;  775          1873  1 ! OUTPUT PARAMETERS:
;  776          1874  1 !       ESL_DESC        - Rewritten with the expanded string resulting
;  777          1875  1 !                         from the $PARSE operation.
;  778          1876  1 !
;  779          1877  1 ! IMPLICIT OUTPUTS:
;  780          1878  1 !       NONE
;  781          1879  1 !
;  782          1880  1 ! ROUTINE VALUE:
;  783          1881  1 !       NONE
;  784          1882  1 !
;  785          1883  1 ! SIDE EFFECTS:
;  786          1884  1 !       NONE
;  787          1885  1 !
;  788          1886  1 !--
;  789          1887  1
;  790          1888  2 BEGIN
;  791          1889  2 MAP
;  792          1890  2         SRC:            REF BBLOCK,     ! Pointer to PL/I varying string
;  793          1891  2         ESL_DESC:       REF BBLOCK;     ! Pointer to descriptor
;  794          1892  2 LOCAL
;  795          1893  2         FAB:            BBLOCK[FAB$C_BLN],      ! Local FAB
;  796          1894  2         NAM:            BBLOCK[NAM$C_BLN],      ! Local NAM block
;  797          1895  2         ESA:            BBLOCK[NAM$C_MAXRSS];   ! Local expanded string
;  798          1896  2
;  799          1897  2
;  800          1898  2 ! Initialize the local RMS structures we use to do this $PARSE.
;  801          1899  2 !
;  802        P 1900  2 $FAB_INIT(FAB=FAB,
;  803        P 1901  2     DNA=UPLIT BYTE('[000000...]*.*;*'),
;  804        P 1902  2     DNS=%CHARCOUNT('[000000...]*.*;*'),
;  805        P 1903  2     FNA=SRC[VAR_BODY],
;  806        P 1904  2     FNS=.SRC[VAR_LENGTH],
;  807          1905  2     NAM=NAM);
;  808        P 1906  2 $NAM_INIT(NAM=NAM,
;  809        P 1907  2     ESA=ESA,
;  810          1908  2     ESS=NAM$C_MAXRSS);
;  811          1909  2
;  812          1910  2
;  813          1911  2 ! Execute the $PARSE.
;  814          1912  2 !
;  815          1913  2 $PARSE(FAB=FAB);
```

```
;  816          1914  2
;  817          1915  2
;  818          1916  2 ! If an error we are not willing to accept occurs, signal it.
;  819          1917  2 !
;  820          1918  2 IF
;  821          1919  2     .NAM[NAM$B_ESL] EQL 0 OR
;  822          1920  3     (.NAM[NAM$[_FNB] AND
;  823          1921  4         (NAM$M_EXP_DEV OR
;  824          1922  4          NAM$M_NODE OR
;  825          1923  2          NAM$M_QUOTED)) NEQ 0
;  826          1924  2 THEN
;  827          1925  2     SIGNAL(BACKUP$_INVQUAVAL, 3, .SRC[VAR_LENGTH], SRC[VAR_BODY], .QUAL_DESC);
;  828          1926  2
;  829          1927  2
;  830          1928  2 ! Initialize output descriptor.
;  831          1929  2 !
;  832          1930  2 ESL_DESC[DSC$W_LENGTH] = .NAM[NAM$B_ESL];
;  833          1931  2 ESL_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  834          1932  2 ESL_DESC[DSC$B_CLASS] = DSC$K_CLASS_S;
;  835          1933  2
;  836          1934  2
;  837          1935  2 ! Allocate memory to hold string.
;  838          1936  2 !
;  839          1937  2 ESL_DESC[DSC$A_POINTER] = GET_VM(.NAM[NAM$B_ESL]);
;  840          1938  2
;  841          1939  2
;  842          1940  2 ! Copy the expanded string to ESL_DESC.
;  843          1941  2 !
;  844          1942  2 CH$MOVE(
;  845          1943  2     .NAM[NAM$B_ESL],
;  846          1944  2     ESA,
;  847          1945  2     .ESL_DESC[DSC$A_POINTER]);
;  848          1946  2
;  849          1947  2
;  850          1948  2 ! Do another $PARSE on the null string.  This will delete the internal RMS
;  851          1949  2 ! context, if any.
;  852          1950  2 !
;  853          1951  2 FAB[FAB$B_FNS] = 0;
;  854          1952  2 FAB[FAB$B_DNS] = 0;
;  855          1953  2 $PARSE(FAB=FAB);
;  856          1954  1 END;


3B 2A 2E 2A 5D 2E 2E 2E 30 30 30 30 30 30 5B  00351 P.AAE:  .ASCII  \[000000...]*.*;*\                          ;
                                           2A  00360                                                          ;

                                                         .EXTRN  SYS$PARSE

                                           00FC  00000 PARSE_EXC_SEL:
                                                         .WORD   Save R2,R3,R4,R5,R6,R7                       ; 1858
                                 57 00000000G  00  9E  00002         MOVAB   SYS$PARSE, R7                    ;
                                 5E     FE50   CE  9E  00009         MOVAB   -432(SP), SP                     ;
    0050 8F           00         6E            00  2C  0000E         MOVC5   #0, (SP), #0, #80, $RMS_PTR      ; 1905
                                       B0      AD      00015                                                 ;
                           B0 AD      5003      8F  B0  00017         MOVW    #20483, $RMS_PTR                 ;
                           C6 AD                02  90  0001D         MOVB    #2, $RMS_PTR+22                  ;
```

```
                         CF    AD               02  90 00021           MOVB     #2, $RMS_PTR+31                      ;
                         D8    AD        FF50   CD  9E 00025           MOVAB    NAM, $RMS_PTR+40                     ;
               56        04    AC               02  C1 0002B           ADDL3    #2, SRC, R6                          ;
                         DC    AD               56  D0 00030           MOVL     R6, $RMS_PTR+44                      ;
                         E0    AD          B9   AF  9E 00034           MOVAB    P.AAE, $RMS_PTR+48                   ;
                         E4    AD          04   BC  90 00039           MOVB     @SRC, $RMS_PTR+52                    ;
                         E5    AD               10  90 0003E           MOVB     #16, $RMS_PTR+53                     ;
  0060   8F    00              6E               00  2C 00042           MOVC5    #0, (SP), #0, #96, $RMS_PTR          ; 1908
                                         FF50   CD     00049                                                         ;
                       FF50    CD        6002   8F  B0 0004C           MOVW     #24578, $RMS_PTR                     ;
                       FF5A    CD               01  8E 00053           MNEGB    #1, $RMS_PTR+10                      ;
                       FF5C    CD               6E  9E 00058           MOVAB    ESA, $RMS_PTR+12                     ;
                                           B0   AD  9F 0005D           PUSHAB   FAB                                  ; 1913
                               67               01  FB 00060           CALLS    #1, SYS$PARSE                        ;
                               53        FF5B   CD  9A 00063           MOVZBL   NAM+11, R3                           ; 1919
                                                0A  13 00068           BEQL     1$                                   ;
                   00060080    8F          84   AD  D3 0006A           BITL     NAM+52, #393344                      ; 1923
                                                18  13 00072           BEQL     2$                                   ;
                                           08   AC  DD 00074 1$:       PUSHL    QUAL_DESC                            ; 1925
                                                56  DD 00077           PUSHL    R6                                   ;
                               7E          04   BC  3C 00079           MOVZWL   @SRC, -(SP)                          ;
                                                03  DD 0007D           PUSHL    #3                                   ;
                                    00000000G   8F  DD 0007F           PUSHL    #BACKUP$_INVQUAVAL                   ;
                   00000000G   00               05  FB 00085           CALLS    #5, LIB$SIGNAL                       ;
                               52          0C   AC  D0 0008C 2$:       MOVL     ESL_DESC, R2                         ; 1930
                               62               53  B0 00090           MOVW     R3, (R2)                             ;
                         02    A2        010E   8F  B0 00093           MOVW     #270, 2(R2)                          ; 1931
                                                53  DD 00099           PUSHL    R3                                   ; 1937
                   00000000G   00               01  FB 0009B           CALLS    #1, GET_VM                           ;
                         04    A2               50  D0 000A2           MOVL     R0, 4(R2)                            ;
          04   B2              6E               53  28 000A6           MOVC3    R3, ESA, @4(R2)                      ; 1945
                                           E4   AD  B4 000AB           CLRW     FAB+52                               ; 1951
                                           B0   AD  9F 000AE           PUSHAB   FAB                                  ; 1953
                               67               01  FB 000B1           CALLS    #1, SYS$PARSE                        ;
                                                    04 000B4           RET                                           ; 1954

; Routine Size:  181 bytes,    Routine Base:  CODE + 0361


;  857          1955  1
```

```
;  859          1956  1 %SBTTL 'EXPAND_SEARCH_LIST '
;  860          1957  1 ROUTINE EXPAND_SEARCH_LIST ( FC  : REF BBLOCK,
;  861          1958  1                             PTR : REF BBLOCK, LAST_PTR : REF BBLOCK ) =
;  862          1959  1
;  863          1960  1 !++
;  864          1961  1 !
;  865          1962  1 ! FUNCTIONAL DESCRIPTION:
;  866          1963  1 !       This routine is used to expand a search list into individual
;  867          1964  1 !       directory names.  The block pointed to by PTR will be removed
;  868          1965  1 !       from the linked list and replaced by n blocks, one each for
;  869          1966  1 !       each value of the search list.
;  870          1967  1 !
;  871          1968  1 ! INPUT PARAMETERS:
;  872          1969  1 !       FC              - address of file context block used for $PARSE
;  873          1970  1 !       PTR             - address of input qualifier context block
;  874          1971  1 !       LAST_PTR        - address of last block (previous block in list)
;  875          1972  1 !
;  876          1973  1 ! IMPLICIT INPUTS:
;  877          1974  1 !       NONE
;  878          1975  1 !
;  879          1976  1 ! OUTPUT PARAMETERS:
;  880          1977  1 !       NONE
;  881          1978  1 !
;  882          1979  1 ! IMPLICIT OUTPUTS:
;  883          1980  1 !       If a search list was found then the forward pointer will be modified
;  884          1981  1 !       to point to the new elements and the old .PTR will be deallocated.
;  885          1982  1 !
;  886          1983  1 ! ROUTINE VALUE:
;  887          1984  1 !       If a search list was found the address of the new input qualifier
;  888          1985  1 !       block else .PTR
;  889          1986  1 !
;  890          1987  1 ! SIDE EFFECTS:
;  891          1988  1 !       NONE
;  892          1989  1 !
;  893          1990  1 !--
;  894          1991  1
;  895          1992  2 BEGIN
;  896          1993  2
;  897          1994  2 LOCAL
;  898          1995  2     FAB         : REF BBLOCK ,
;  899          1996  2     NAM         : REF BBLOCK ,
;  900          1997  2     WORK_FC     : REF BBLOCK ,
;  901          1998  2     NEW_FAB     : REF BBLOCK ,
;  902          1999  2     NEW_NAM     : REF BBLOCK ,
;  903          2000  2     LAST        : REF BBLOCK ,
;  904          2001  2     NEW_LIST    : REF BBLOCK ,
;  905          2002  2     NEW_ITEM    : REF BBLOCK ,
;  906          2003  2     OPEN_COUNT  : LONG ,
;  907          2004  2     WCCSAV      : LONG ,
;  908          2005  2     STATUS      : LONG ;
;  909          2006  2
;  910          2007  2 LAST = 0 ;
;  911          2008  2 WORK_FC = .FC ;
;  912          2009  2 FAB = WORK_FC [FC_FAB] ;
;  913          2010  2 NAM = WORK_FC [FC_NAM] ;
;  914          2011  2 OPEN_COUNT = .CHK_FLAGS[CHK_FLAGS_COUNT] ;
;  915          2012  2
```

```
;  916          2013  2 !
;  917          2014  2 ! Parse the string and if it is not a search list then return
;  918          2015  2 !
;  919          2016  2 NAM[NAM$V_SLPARSE] = 1 ;
;  920          2017  2 NAM[NAM$V_NOCONCEAL] = 1 ;
;  921          2018  2 WCCSAV = .NAM[NAM$L_WCC] ;
;  922          2019  2 NAM[NAM$L_WCC] = 0 ;
;  923          2020  2 STATUS = $PARSE ( FAB = .FAB ) ;
;  924          2021  3 IF NOT ( .STATUS AND
;  925          2022  3         .NAM[NAM$V_SEARCH_LIST] )
;  926          2023  2 THEN
;  927          2024  3     BEGIN
;  928          2025  3     !
;  929          2026  3     ! Clean up the work FC
;  930          2027  3     !
;  931          2028  3     NAM[NAM$V_SVCTX] = 0 ;
;  932          2029  3     NAM[NAM$V_SLPARSE] = 0 ;
;  933          2030  3     NAM[NAM$V_NOCONCEAL] = 0 ;
;  934          2031  3     NAM[NAM$L_WCC] = .WCCSAV;
;  935          2032  3     RETURN 0 ;
;  936          2033  2     END ;
;  937          2034  2
;  938          2035  2 !
;  939          2036  2 ! We have a search list so loop through all values
;  940          2037  2 !
;  941          2038  2 WHILE .STATUS DO
;  942          2039  3     BEGIN
;  943          2040  3
;  944          2041  3     !
;  945          2042  3     ! Allocate a new block and insert it in
;  946          2043  3     ! the new list
;  947          2044  3     !
;  948          2045  3     NEW_ITEM = GET_ZERO_VM ( QUAL_S_INPU ) ;
;  949          2046  3     IF .LAST EQL 0
;  950          2047  3         THEN NEW_LIST = .NEW_ITEM
;  951          2048  3         ELSE
;  952          2049  4             BEGIN
;  953          2050  4             LAST [QUAL_NEXT] = .NEW_ITEM ;
;  954          2051  4             CHK_FLAGS [CHK_FLAGS_COUNT] = .CHK_FLAGS [CHK_FLAGS_COUNT] + 1 ;
;  955          2052  3             END ;
;  956          2053  3     LAST = .NEW_ITEM ;
;  957          2054  3
;  958          2055  3     !
;  959          2056  3     ! Copy over the data
;  960          2057  3     !
;  961          2058  3     NEW_ITEM[QUAL_PARA_FC] = GET_VM ( FC_S_AREA ) ;
;  962          2059  3     COPY_FC ( .WORK_FC, .NEW_ITEM[QUAL_PARA_FC],
;  963          2060  3               .NAM[NAM$B_ESL], .NAM[NAM$L_ESA] );
;  964          2061  3
;  965          2062  3     !
;  966          2063  3     ! Reparse the copy
;  967          2064  3     !
;  968          2065  3     NEW_FAB = BBLOCK [.NEW_ITEM[QUAL_PARA_FC], FC_FAB ] ;
;  969          2066  3     NEW_NAM = BBLOCK [.NEW_ITEM[QUAL_PARA_FC], FC_NAM ] ;
;  970          2067  3
;  971          2068  4     IF NOT $PARSE(FAB=.NEW_FAB)
;  972          2069  3     ! _N
```

```
;  973     2070  3          FILE_ERROR(BACKUP$_PARSE, .NEW_FAB, .NEW_FAB[FAB$L_STS], .NEW_FAB[FAB$L_STV]);
;  974     2071
;  975     2072  2      !
;  976     2073  2      ! Check the validity of the resultant spec.
;  977     2074  2      !
;  978     2075  2      CHECK_F11 ( .NEW_NAM, FALSE ) ;
;  979     2076  2      CHECK_COMMON ( .NEW_FAB, FALSE ) ;
;  980     2077  2
;  981     2078  2      !
;  982     2079  2      ! Set up the string descriptors in the new list element
;  983     2080  2      !
;  984     2081  2      BUILD_QUAL_DESC ( .NEW_ITEM, .NEW_NAM ) ;
;  985     2082  2
;  986     2083  2      !
;  987     2084  2      ! Get the next element of the search list
;  988     2085  2      !
;  989     2086  2      STATUS = $PARSE ( FAB = .FAB ) ;
;  990     2087  2
;  991     2088  2      END ;
;  992     2089  2
;  993     2090  2  !
;  994     2091  2  ! Insert the new list where the old element came from
;  995     2092  2  !
;  996     2093  2  NEW_ITEM [QUAL_NEXT] = .PTR [QUAL_NEXT] ;
;  997     2094  2  IF .OPEN_COUNT EQL 1
;  998     2095  2      THEN .LAST_PTR = .NEW_LIST
;  999     2096  2      ELSE LAST_PTR [QUAL_NEXT] = .NEW_LIST ;
; 1000     2097  2
; 1001     2098  2  !
; 1002     2099  2  ! Clean up the work FC
; 1003     2100  2  !
; 1004     2101  2  NAM[NAM$V_SVCTX] = 0 ;
; 1005     2102  2  NAM[NAM$V_SLPARSE] = 0 ;
; 1006     2103  2  NAM[NAM$V_NOCONCEAL] = 0 ;
; 1007     2104  2  NAM[NAM$L_WCC] = .WCCSAV ;
; 1008     2105  2
; 1009     2106  2  !
; 1010     2107  2  ! Now clean up the old item
; 1011     2108  2  !
; 1012     2109  2  FAB = BBLOCK [.PTR[QUAL_PARA_FC], FC_FAB ] ;
; 1013     2110  2  FREE_VM ( .FAB[FAB$B_FNS], .FAB[FAB$L_FNA] ) ;
; 1014     2111  2  FREE_VM ( FC_S_AREA, .PTR[QUAL_PARA_FC] ) ;
; 1015     2112  2  FREE_VM ( QUAL_S_INPU, .PTR ) ;
; 1016     2113  2
; 1017     2114  2  RETURN .NEW_ITEM ;
; 1018     2115  2
; 1019     2116  1  END ;


                                 0FFC 00000 EXPAND_SEARCH_LIST:
                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 1957
                         5E              0C  C2 00002           SUBL2   #12, SP                 ;
                                         59  D4 00005           CLRL    LAST                    ; 2007
                         5A        04    AC  D0 00007           MOVL    FC, WORK_FC             ; 2008
```

```
                         57             5A  D0 0000B             MOVL    WORK_FC, FAB                          : 2009
                         53      0094   CA  9E 0000E             MOVAB   148(R10), NAM                         : 2010
                   04    AE 00000000'   EF  9A 00013             MOVZBL  CHK_FLAGS, OPEN_COUNT                 : 2011
                         56        08   A3  9E 0001B             MOVAB   8(NAM), R6                            : 2016
                         66             30  88 0001F             BISB2   #48, (R6)                             : 2017
                         55        30   A3  9E 00022             MOVAB   48(NAM), R5                           : 2018
                         5B             65  D0 00026             MOVL    (R5), WCCSAV                          :
                                        65  D4 00029             CLRL    (R5)                                  : 2019
                                        57  DD 0002B             PUSHL   FAB                                   : 2020
            00000000G    00             01  FB 0002D             CALLS   #1, SYS$PARSE                         :
                         6E             50  D0 00034             MOVL    R0, STATUS                            :
                         05             6E  E9 00037             BLBC    STATUS, 1$                            : 2021
        0E         35    A3             03  E0 0003A             BBS     #3, 53(NAM), 2$                       : 2022
                   03    A5        80   8F  8A 0003F 1$:         BICB2   #128, 3(R5)                           : 2028
                         66             30  8A 00044             BICB2   #48, (R6)                             : 2030
                         65             5B  D0 00047             MOVL    WCCSAV, (R5)                          : 2031
                                      00E6  31 0004A             BRW     8$                                    : 2032
                         03             6E  E8 0004D 2$:         BLBS    STATUS, 3$                            : 2038
                                      0099  31 00050             BRW     7$                                    :
                                        28  DD 00053 3$:         PUSHL   #40                                   : 2045
            00000000G    00             01  FB 00055             CALLS   #1, GET_ZERO_VM                       :
                         52             50  D0 0005C             MOVL    R0, NEW_ITEM                          :
                                        59  D5 0005F             TSTL    LAST                                  : 2046
                                        06  12 00061             BNEQ    4$                                    :
                   08    AE             52  D0 00063             MOVL    NEW_ITEM, NEW_LIST                    : 2047
                                        09  11 00067             BRB     5$                                    :
                         69             52  D0 00069 4$:         MOVL    NEW_ITEM, (LAST)                      : 2050
                            00000000'   EF  96 0006C             INCB    CHK_FLAGS                             : 2051
                         59             52  D0 00072 5$:         MOVL    NEW_ITEM, LAST                        : 2053
                         7E      0354   8F  3C 00075             MOVZWL  #852, -(SP)                           : 2058
            00000000G    00             01  FB 0007A             CALLS   #1, GET_VM                            :
                   04    A2             50  D0 00081             MOVL    R0, 4(NEW_ITEM)                       :
                                   0C   A3  DD 00085             PUSHL   12(NAM)                               : 2060
                         7E        0B   A3  9A 00088             MOVZBL  11(NAM), -(SP)                        :
                                   04   A2  DD 0008C             PUSHL   4(NEW_ITEM)                           : 2059
                                        5A  DD 0008F             PUSHL   WORK_FC                               :
                 FC4C    CF             04  FB 00091             CALLS   #4, COPY_FC                           :
                         54        04   A2  D0 00096             MOVL    4(NEW_ITEM), NEW_FAB                  : 2065
        58         04    A2 00000094    8F  C1 0009A             ADDL3   #148, 4(NEW_ITEM), NEW_NAM            : 2066
                                        54  DD 000A3             PUSHL   NEW_FAB                               : 2068
            00000000G    00             01  FB 000A5             CALLS   #1, SYS$PARSE                         :
                         13             50  E8 000AC             BLBS    R0, 6$                                :
                         7E        08   A4  7D 000AF             MOVQ    8(NEW_FAB), -(SP)                     : 2070
                                        54  DD 000B3             PUSHL   NEW_FAB                               :
                            00000000G   8F  DD 000B5             PUSHL   #BACKUP$_PARSE                        :
            00000000G    00             04  FB 000BB             CALLS   #4, FILE_ERROR                        :
                                        7E  D4 000C2 6$:         CLRL    -(SP)                                 : 2075
                                        58  DD 000C4             PUSHL   NEW_NAM                               :
                0000V    CF             02  FB 000C6             CALLS   #2, CHECK_F11                         :
                                        7E  D4 000CB             CLRL    -(SP)                                 : 2076
                                        54  DD 000CD             PUSHL   NEW_FAB                               :
                0000V    CF             02  FB 000CF             CALLS   #2, CHECK_COMMON                      :
                                 0104   8F  BB 000D4             PUSHR   #^M<R2,R8>                            : 2081
                0000V    CF             02  FB 000D8             CALLS   #2, BUILD_QUAL_DESC                   :
                                        57  DD 000DD             PUSHL   FAB                                   : 2086
            00000000G    00             01  FB 000DF             CALLS   #1, SYS$PARSE                         :
                         6E             50  D0 000E6             MOVL    R0, STATUS                            :
```

```
                            FF61  31 000E9          BRW     2$                          ; 2038
             53        08     AC  D0 000EC  7$:     MOVL    PTR, R3                     ; 2093
             62               63  D0 000F0          MOVL    (R3), (NEW_ITEM)            ;
        0C   BC        08     AE  D0 000F3          MOVL    NEW_LIST, @LAST_PTR         ; 2096
        03   A5        80     8F  8A 000F8          BICB2   #128, 3(R5)                 ; 2101
             66               30  8A 000FD          BICB2   #48, (R6)                   ; 2103
             65               5B  D0 00100          MOVL    WCCSAV, (R5)                ; 2104
             57        04     A3  D0 00103          MOVL    4(R3), FAB                  ; 2109
                       2C     A7  DD 00107          PUSHL   44(FAB)                     ; 2110
             7E        34     A7  9A 0010A          MOVZBL  52(FAB), -(SP)              ;
   00000000G 00               02  FB 0010E          CALLS   #2, FREE_VM                 ;
                       04     A3  DD 00115          PUSHL   4(R3)                       ; 2111
             7E      0354     8F  3C 00118          MOVZWL  #852, -(SP)                 ;
   00000000G 00               02  FB 0011D          CALLS   #2, FREE_VM                 ;
                              53  DD 00124          PUSHL   R3                          ; 2112
                              28  DD 00126          PUSHL   #40                         ;
   00000000G 00               02  FB 00128          CALLS   #2, FREE_VM                 ;
             50               52  D0 0012F          MOVL    NEW_ITEM, R0                ; 2114
                                  04 00132          RET                                 ;
                              50  D4 00133  8$:     CLRL    R0                          ; 2116
                                  04 00135          RET                                 ;

; Routine Size:  310 bytes,    Routine Base:  CODE + 0416


; 1020          2117  1
; 1021          2118  1
```

```
;  1023         2119  1 %SBTTL 'CHECK_SAV - check save set input and output lists'
;  1024         2120  1 ROUTINE CHECK_SAV( NAM : REF BBLOCK, OUTPUT_SIDE ): NOVALUE=
;  1025         2121  1
;  1026         2122  1 !++
;  1027         2123  1 !
;  1028         2124  1 ! FUNCTIONAL DESCRIPTION:
;  1029         2125  1 !       This routine validitys the save set input and output lists.
;  1030         2126  1 !
;  1031         2127  1 ! INPUT PARAMETERS:
;  1032         2128  1 !       NAM             - address of NAM
;  1033         2129  1 !       OUTPUT_SIDE     - True if output, false if input.
;  1034         2130  1 !
;  1035         2131  1 ! IMPLICIT INPUTS:
;  1036         2132  1 !       NONE
;  1037         2133  1 !
;  1038         2134  1 ! OUTPUT PARAMETERS:
;  1039         2135  1 !       NONE
;  1040         2136  1 !
;  1041         2137  1 ! IMPLICIT OUTPUTS:
;  1042         2138  1 !       NONE
;  1043         2139  1 !
;  1044         2140  1 ! ROUTINE VALUE:
;  1045         2141  1 !       NONE
;  1046         2142  1 !
;  1047         2143  1 ! SIDE EFFECTS:
;  1048         2144  1 !       Messages may be signalled if errors are detected.
;  1049         2145  1 !
;  1050         2146  1 !--
;  1051         2147  1
;  1052         2148  2 BEGIN
;  1053         2149  2
;  1054         2150  2 IF .OUTPUT_SIDE
;  1055         2151  3 THEN (IF .NAM[NAM$V_WILDCARD] THEN SIGNAL(BACKUP$_SAVSETNAM);)
;  1056         2152  2 ELSE
;  1057         2153  3     BEGIN
;  1058         2154  3     INPUT_FLAGS[INPUT_REWOUND] = FALSE;
;  1059         2155  3     IF .NAM[NAM$V_WILDCARD]
;  1060         2156  3     THEN INPUT_FLAGS[INPUT_WILDSAVE] = TRUE
;  1061         2157  3     ELSE INPUT_FLAGS[INPUT_WILDSAVE] = FALSE;
;  1062         2158  2     END;
;  1063         2159  2
;  1064         2160  2 IF
;  1065         2161  3     BEGIN
;  1066         2162  3     IF
;  1067         2163  4         (.NAM[NAM$L_FNB] AND
;  1068         2164  5             (NAM$M_NODE OR NAM$M_QUOTED OR
;  1069         2165  5             NAM$M_EXP_DIR OR NAM$M_EXP_NAME OR
;  1070         2166  3             NAM$M_EXP_TYPE OR NAM$M_EXP_VER)) NEQ 0
;  1071         2167  3     THEN
;  1072         2168  3         .CHK_FLAGS[CHK_FLAGS_COUNT] NEQ 1               ! non-null must be first
;  1073         2169  3     ELSE
;  1074         2170  3         .CHK_FLAGS[CHK_FLAGS_COUNT] EQL 1 AND .OUTPUT_SIDE
;  1075         2171  3                                           ! null must not be first output
;  1076         2172  3     END
;  1077         2173  2 THEN
;  1078         2174  2     SIGNAL(BACKUP$_SAVSETNAM); ! save set name must be in first spec
;  1079         2175  2
```

```
;  1080     2176  2
;  1081     2177  2 IF .CHK_FLAGS[CHK_FLAGS_F11]
;  1082     2178  2     THEN SIGNAL(BACKUP$_INCDEVLIS); ! inconsistent devices in list
;  1083     2179  2
;  1084     2180  2
;  1085     2181  2 IF .OUTPUT_SIDE
;  1086     2182  2 THEN
;  1087     2183  3     BEGIN
;  1088     2184  3     IF .QUAL[QUAL_ISAV]
;  1089     2185  3         THEN SIGNAL(BACKUP$_ONEF11DEV); ! one F11 device
;  1090     2186  3     IF .QUAL[QUAL_COMP]
;  1091     2187  3         THEN SIGNAL(BACKUP$_CMPOUTF11); ! /COMPARE requires F11 output
;  1092     2188  2     END;
;  1093     2189  2
;  1094     2190  2 RETURN ;
;  1095     2191  2
;  1096     2192  1 END ;


                                007C 00000 CHECK_SAV:
                                           .WORD   Save R2,R3,R4,R5,R6                          ; 2120
                     56 00000000G   8F  D0 00002       MOVL    #BACKUP$_SAVSETNAM, R6          ;
                     55 00000000'   EF  9E 00009       MOVAB   CHK_FLAGS, R5                   ;
                     54 00000000'   EF  9E 00010       MOVAB   INPUT_FLAGS, R4                 ;
                     53 00000000G   00  9E 00017       MOVAB   LIB$SIGNAL, R3                  ;
                     52         04  AC  D0 0001E       MOVL    NAM, R2                         ; 2151
                     0B         08  AC  E9 00022       BLBC    OUTPUT_SIDE, 1$                 ; 2150
                     19         35  A2  E9 00026       BLBC    53(R2), 3$                      ; 2151
                                    56  DD 0002A       PUSHL   R6                              ;
                     63             01  FB 0002C       CALLS   #1, LIB$SIGNAL                  ;
                                    12  11 0002F       BRB     3$                              ; 2150
                     64         80  8F  8A 00031  1$:  BICB2   #128, INPUT_FLAGS              ; 2154
                     06         35  A2  E9 00035       BLBC    53(R2), 2$                      ; 2155
                     64         40  8F  88 00039       BISB2   #64, INPUT_FLAGS                ; 2156
                                    04  11 0003D       BRB     3$                              ;
                     64         40  8F  8A 0003F  2$:  BICB2   #64, INPUT_FLAGS               ; 2157
           00060047  8F         34  A2  D3 00043  3$:  BITL    52(R2), #393287                 ; 2166
                                    07  13 0004B       BEQL    4$                              ;
                     01             65  91 0004D       CMPB    CHK_FLAGS, #1                   ; 2168
                                    10  13 00050       BEQL    6$                              ;
                                    09  11 00052       BRB     5$                              ;
                     01             65  91 00054  4$:  CMPB    CHK_FLAGS, #1                   ; 2170
                                    09  12 00057       BNEQ    6$                              ;
                     05         08  AC  E9 00059       BLBC    OUTPUT_SIDE, 6$                 ;
                                    56  DD 0005D  5$:  PUSHL   R6                              ; 2174
                     63             01  FB 0005F       CALLS   #1, LIB$SIGNAL                  ;
                     09         01  A5  E9 00062  6$:  BLBC    CHK_FLAGS+1, 7$                 ; 2177
                           00000000G  8F  DD 00066       PUSHL   #BACKUP$_INCDEVLIS              ; 2178
                     63             01  FB 0006C       CALLS   #1, LIB$SIGNAL                  ;
                     1E         08  AC  E9 0006F  7$:  BLBC    OUTPUT_SIDE, 9$                 ; 2181
           09        FF47  C4         05  E1 00073       BBC     #5, QUAL+15, 8$                 ; 2184
                           00000000G  8F  DD 00079       PUSHL   #BACKUP$_ONEF11DEV              ; 2185
                     63             01  FB 0007F       CALLS   #1, LIB$SIGNAL                  ;
                               FF40  C4  95 00082  8$:  TSTB    QUAL+8                          ; 2186
```

```
                                        09  18 00086          BGEQ    9$                                                      ;
                                 00000000G  8F  DD 00088          PUSHL   #BACKUP$_CMPOUTF11                                      ; 2187
                                        63  01  FB 0008E          CALLS   #1, LIB$SIGNAL                                          ;
                                            04 00091 9$:         RET                                                             ; 2192

; Routine Size:  146 bytes,    Routine Base:  CODE + 054C

; 1097          2193  1
```

```
;  1099     2194  1 %SBTTL 'CHECK_F11 - check F11 format'
;  1100     2195  1 ROUTINE CHECK_F11( NAM : REF BBLOCK, OUTPUT_SIDE ): NOVALUE=
;  1101     2196  1
;  1102     2197  1 !++
;  1103     2198  1 !
;  1104     2199  1 ! FUNCTIONAL DESCRIPTION:
;  1105     2200  1 !       This routine validity checks files 11 input and output lists
;  1106     2201  1 !
;  1107     2202  1 ! INPUT PARAMETERS:
;  1108     2203  1 !       NAM             - address of NAM
;  1109     2204  1 !       OUTPUT_SIDE     - True if output, false if input.
;  1110     2205  1 !
;  1111     2206  1 ! IMPLICIT INPUTS:
;  1112     2207  1 !       NONE
;  1113     2208  1 !
;  1114     2209  1 ! OUTPUT PARAMETERS:
;  1115     2210  1 !       NONE
;  1116     2211  1 !
;  1117     2212  1 ! IMPLICIT OUTPUTS:
;  1118     2213  1 !       NONE
;  1119     2214  1 !
;  1120     2215  1 ! ROUTINE VALUE:
;  1121     2216  1 !       NONE
;  1122     2217  1 !
;  1123     2218  1 ! SIDE EFFECTS:
;  1124     2219  1 !       Messages may be signalled if errors are detected.
;  1125     2220  1 !
;  1126     2221  1 !--
;  1127     2222  1
;  1128     2223  2 BEGIN
;  1129     2224  2
;  1130     2225  2 IF .QUAL[QUAL_PHYS]
;  1131     2226  2 THEN
;  1132     2227  3     BEGIN
;  1133     2228  3     CHK_FLAGS [ CHK_FLAGS_SRCH_LIST ] = 0 ;     ! No search lists for physical
;  1134     2229  3     IF .CHK_FLAGS[CHK_FLAGS_COUNT] GTR 1
;  1135     2230  3         THEN SIGNAL(BACKUP$_ONEPHYDEV); ! one /PHYSICAL device
;  1136     2231  4     IF (.NAM[NAM$L_FNB] AND
;  1137     2232  5         (NAM$M_EXP_DIR OR NAM$M_EXP_NAME OR
;  1138     2233  3          NAM$M_EXP_TYPE OR NAM$M_EXP_VER)) NEQ 0
;  1139     2234  3         THEN SIGNAL(BACKUP$_PHYFILSPE); ! bad /PHYSICAL spec
;  1140     2235  3     END
;  1141     2236  2 ELSE IF .QUAL[QUAL_IMAG] OR .COM_FLAGS[COM_STANDALONE]
;  1142     2237  2 THEN
;  1143     2238  3     BEGIN
;  1144     2239  3     IF NOT .COM_FLAGS[COM_STANDALONE]
;  1145     2240  3     THEN
;  1146     2241  3         CHK_FLAGS [ CHK_FLAGS_SRCH_LIST ] = 0 ; ! No search lists for image
;  1147     2242  3     IF .CHK_FLAGS[CHK_FLAGS_COUNT] GTR 1 AND NOT .OUTPUT_SIDE AND NOT .COM_FLAGS[COM_STANDALONE]
;  1148     2243  3         THEN SIGNAL(BACKUP$_ONEIMGDEV); ! one /IMAGE device
;  1149     2244  4     IF (.NAM[NAM$L_FNB] AND
;  1150     2245  5         (NAM$M_EXP_DIR OR NAM$M_EXP_NAME OR
;  1151     2246  3          NAM$M_EXP_TYPE OR NAM$M_EXP_VER)) NEQ 0
;  1152     2247  3         THEN SIGNAL(BACKUP$_IMGFILSPE); ! bad /IMAGE spec
;  1153     2248  3     END
;  1154     2249  2 ELSE IF .QUAL[QUAL_INCR]
;  1155     2250  2 THEN
```

```
;  1156          2251  3     BEGIN
;  1157          2252  4     IF (.NAM[NAM$L_FNB] AND
;  1158          2253  5         (NAM$M_EXP_DIR OR NAM$M_EXP_NAME OR
;  1159          2254  5          NAM$M_EXP_TYPE OR NAM$M_EXP_VER)) NEQ 0
;  1160          2255  5         THEN SIGNAL(BACKUP$_INCFILSPE); ! bad /INCREMENTAL spec
;  1161          2256  2     END;
;  1162          2257  2
;  1163          2258  2
;  1164          2259  2 IF .CHK_FLAGS[CHK_FLAGS_SAV]
;  1165          2260  2     THEN SIGNAL(BACKUP$_INCDEVLIS); ! inconsistent devices in list
;  1166          2261  2
;  1167          2262  2
;  1168          2263  2 IF .OUTPUT_SIDE
;  1169          2264  2 THEN
;  1170          2265  3     BEGIN
;  1171          2266  3     IF
;  1172          2267  3         .CHK_FLAGS[CHK_FLAGS_COUNT] GTR 1 AND
;  1173          2268  3         NOT .QUAL[QUAL_IMAG] AND
;  1174          2269  3         NOT .COM_FLAGS[COM_STANDALONE]
;  1175          2270  3     THEN
;  1176          2271  3         SIGNAL(BACKUP$_ONEF11OUT); ! too many outputs
;  1177          2272  3     END
;  1178          2273  2 ELSE
;  1179          2274  3     BEGIN
;  1180          2275  3     IF .QUAL[QUAL_LIST] AND .QUAL[QUAL_OUTP_LIST] EQL 0
;  1181          2276  3         THEN SIGNAL(BACKUP$_LISINPSAV); ! LIST input must be saveset
;  1182          2277  2     END;
;  1183          2278  2
;  1184          2279  2 RETURN ;
;  1185          2280  2
;  1186          2281  1 END ;


                                        001C 00000 CHECK_F11:
                                                           .WORD   Save R2,R3,R4                          ; 2195
                                54 00000000G   00  9E 00002        MOVAB   LIB$SIGNAL, R4                 ;
                                53 00000000'   EF  9E 00009        MOVAB   CHK_FLAGS+1, R3                ;
                                52 00000000'   EF  9E 00010        MOVAB   QUAL+8, R2                     ;
                    25    04    A2             05  E1 00017        BBC     #5, QUAL+12, 2$                ; 2225
                                63             08  8A 0001C        BICB2   #8, CHK_FLAGS+1                ; 2228
                                01       FF    A3  91 0001F        CMPB    CHK_FLAGS, #1                  ; 2229
                                               09  1B 00023        BLEQU   1$                             ;
                                00000000G      8F  DD 00025        PUSHL   #BACKUP$_ONEPHYDEV             ; 2230
                                64             01  FB 0002B        CALLS   #1, LIB$SIGNAL                 ;
                                50       04    AC  D0 0002E 1$:    MOVL    NAM, R0                        ; 2231
                          47    8F       34    A0  93 00032        BITB    52(R0), #71                    ; 2233
                                               5E  13 00037        BEQL    8$                             ;
                                00000000G      8F  DD 00039        PUSHL   #BACKUP$_PHYFILSPE             ; 2234
                                               53  11 0003F        BRB     7$                             ;
                    05    02    A2             03  E0 00041 2$:    BBS     #3, QUAL+10, 3$                ; 2236
                    33    72    A2             01  E1 00046        BBC     #1, COM_FLAGS, 6$              ;
                    03    72    A2             01  E0 0004B 3$:    BBS     #1, COM_FLAGS, 4$              ; 2239
                                63             08  8A 00050        BICB2   #8, CHK_FLAGS+1                ; 2241
                                01       FF    A3  91 00053 4$:    CMPB    CHK_FLAGS, #1                  ; 2242
```

```
                                12  1B 00057          BLEQU   5$                         ;
             0E           08    AC  E8 00059          BLBS    OUTPUT_SIDE, 5$            ;
   09   72   A2                 01  E0 0005D          BBS     #1, COM_FLAGS, 5$          ;
                 00000000G      8F  DD 00062          PUSHL   #BACKUP$_ONEIMGDEV         ; 2243
             64                 01  FB 00068          CALLS   #1, LIB$SIGNAL             ;
             50           04    AC  D0 0006B 5$:      MOVL    NAM, R0                    ; 2244
        47   8F           34    A0  93 0006F          BITB    52(R0), #71                ; 2246
                                21  13 00074          BEQL    8$                         ;
                 00000000G      8F  DD 00076          PUSHL   #BACKUP$_IMGFILSPE         ; 2247
                                16  11 0007C          BRB     7$                         ;
   14   02   A2                 04  E1 0007E 6$:      BBC     #4, QUAL+10, 8$            ; 2249
             50           04    AC  D0 00083          MOVL    NAM, R0                    ; 2252
        47   8F           34    A0  93 00087          BITB    52(R0), #71                ; 2254
                                09  13 0008C          BEQL    8$                         ;
                 00000000G      8F  DD 0008E          PUSHL   #BACKUP$_INCFILSPE         ; 2255
             64                 01  FB 00094 7$:      CALLS   #1, LIB$SIGNAL             ;
   09        63                 01  E1 00097 8$:      BBC     #1, CHK_FLAGS+1, 9$        ; 2259
                 00000000G      8F  DD 0009B          PUSHL   #BACKUP$_INCDEVLIS         ; 2260
             64                 01  FB 000A1          CALLS   #1, LIB$SIGNAL             ;
             18           08    AC  E9 000A4 9$:      BLBC    OUTFUT_SIDE, 10$           ; 2263
             01           FF    A3  91 000A8          CMPB    CHK_FLAGS, #1              ; 2267
                                24  1B 000AC          BLEQU   12$                        ;
   1F   02   A2                 03  E0 000AE          BBS     #3, QUAL+10, 12$           ; 2268
   1A   72   A2                 01  E0 000B3          BBS     #1, COM_FLAGS, 12$         ; 2269
                 00000000G      8F  DD 000B8          PUSHL   #BACKUP$_ONEF11OUT         ; 2271
                                0F  11 000BE          BRB     11$                        ;
             0E           03    A2  E9 000C0 10$:     BLBC    QUAL+11, 12$               ; 2275
                          FC    A2  D5 000C4          TSTL    QUAL+4                     ;
                                09  12 000C7          BNEQ    12$                        ;
                 00000000G      8F  DD 000C9          PUSHL   #BACKUP$_LISINPSAV         ; 2276
             64                 01  FB 000CF 11$:     CALLS   #1, LIB$SIGNAL             ;
                                    04 000D2 12$:     RET                                ; 2281

; Routine Size:  211 bytes,    Routine Base:  CODE + 05DE

; 1187          2282  1
```

```
;  1189          2283  1 %SBTTL 'CHECK_COMMON - perform common check'
;  1190          2284  1 ROUTINE CHECK_COMMON( FAB : REF BBLOCK, OUTPUT_SIDE ): NOVALUE=
;  1191          2285  1
;  1192          2286  1 !++
;  1193          2287  1 !
;  1194          2288  1 ! FUNCTIONAL DESCRIPTION:
;  1195          2289  1 !       This routine performs the common checks for input and output lists.
;  1196          2290  1 !
;  1197          2291  1 ! INPUT PARAMETERS:
;  1198          2292  1 !       FAB             - address of FAB
;  1199          2293  1 !       OUTPUT_SIDE     - True if output, false if input.
;  1200          2294  1 !
;  1201          2295  1 ! IMPLICIT INPUTS:
;  1202          2296  1 !       NONE
;  1203          2297  1 !
;  1204          2298  1 ! OUTPUT PARAMETERS:
;  1205          2299  1 !       NONE
;  1206          2300  1 !
;  1207          2301  1 ! IMPLICIT OUTPUTS:
;  1208          2302  1 !       NONE
;  1209          2303  1 !
;  1210          2304  1 ! ROUTINE VALUE:
;  1211          2305  1 !       NONE
;  1212          2306  1 !
;  1213          2307  1 ! SIDE EFFECTS:
;  1214          2308  1 !       Messages may be signalled if errors are detected.
;  1215          2309  1 !
;  1216          2310  1 !--
;  1217          2311  1
;  1218          2312  2 BEGIN
;  1219          2313  2
;  1220          2314  2 IF NOT .COM_FLAGS[COM_STANDALONE]
;  1221          2315  2 THEN
;  1222          2316  3     IF  (
;  1223          2317  3         .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
;  1224          2318  3     OR
;  1225          2319  3         NOT .CHK_FLAGS[CHK_FLAGS_SAVE_SET] AND
;  1226          2320  4         (.QUAL[QUAL_PHYS] OR
;  1227          2321  4          .QUAL[QUAL_IMAG] AND .OUTPUT_SIDE AND NOT .QUAL[QUAL_COMP])
;  1228          2322  3         )
;  1229          2323  2     THEN
;  1230          2324  3         BEGIN
;  1231          2325  3         !
;  1232          2326  3         ! Tape save set, physical volume, or image output.
;  1233          2327  3         ! Must be mounted foreign.
;  1234          2328  3         !
;  1235          2329  3         IF
;  1236          2330  3             NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR] AND
;  1237          2331  4             NOT (.QUAL[QUAL_PHYS] AND NOT .OUTPUT_SIDE AND
;  1238          2332  4                 .JPI_CURPRIV[PRV$V_LOG_IO])
;  1239          2333  3         THEN
;  1240          2334  3             FILE_ERROR(BACKUP$_MOUNTFOR, .FAB);
;  1241          2335  3         END
;  1242          2336  2     ELSE IF NOT .CHK_FLAGS[CHK_FLAGS_SAVE_SET]
;  1243          2337  2     THEN
;  1244          2338  3         BEGIN
;  1245          2339  3         !
```

```
;  1246          2340  3          ! Files.
;  1247          2341  3          ! Must be mounted Files-11.
;  1248          2342  3          !
;  1249          2343  3          IF .BBLOCK[FAB[FAB$L_DEV], DEV$V_FOR]
;  1250          2344  3          THEN
;  1251          2345  3              FILE_ERROR(BACKUP$_MOUNTF11, .FAB);
;  1252          2346  2          END;
;  1253          2347  2
;  1254          2348  2 RETURN ;
;  1255          2349  2
;  1256          2350  1 END ;


                                                000C 00000 CHECK_COMMON:
                                                                .WORD   Save R2,R3                              ; 2284
                                    53 00000000'    EF  9E 00002        MOVAB   CHK_FLAGS+1, R3                 ;
                                    52 00000000'    EF  9E 00009        MOVAB   QUAL+12, R2                     ;
                          52        6E    A2              01  E0 00010    BBS     #1, COM_FLAGS, 5$               ; 2314
                                    50          04    AC  D0 00015        MOVL    FAB, R0                         ; 2317
                          16        40    A0              05  E0 00019    BBS     #5, 64(R0), 1$                  ;
                          45              63              02  E0 0001E    BBS     #2, CHK_FLAGS+1, 5$             ; 2319
                          0E              62              05  E0 00022    BBS     #5, QUAL+12, 1$                 ; 2320
                          25        FE    A2              03  E1 00026    BBC     #3, QUAL+10, 3$                 ; 2321
                                    21          08    AC  E9 0002B        BLBC    OUTPUT_SIDE, 3$                 ;
                                                FC    A2  95 0002F        TSTB    QUAL+8                          ;
                                                1C    19  00032           BLSS    3$                              ;
                                    2F          43    A0  E8 00034 1$:    BLBS    67(R0), 5$                      ; 2330
                                0A              62              05  E1 00038    BBC     #5, QUAL+12, 2$         ; 2331
                                    06          08    AC  E8 0003C        BLBS    OUTPUT_SIDE, 2$                 ;
                                              FEE8    C2  95 00040        TSTB    JPI_CURPRIV                     ; 2332
                                                21  19  00044           BLSS    5$                              ;
                                                50  DD  00046 2$:       PUSHL   R0                              ; 2334
                                    00000000G   8F  DD  00048           PUSHL   #BACKUP$_MOUNTFOR               ;
                                                10  11  0004E           BRB     4$                              ;
                                13              63              02  E0 00050 3$:    BBS     #2, CHK_FLAGS+1, 5$     ; 2336
                                    0F          43    A0  E9 00054        BLBC    67(R0), 5$                      ; 2343
                                                50  DD  00058           PUSHL   R0                              ; 2345
                                    00000000G   8F  DD  0005A           PUSHL   #BACKUP$_MOUNTF11               ;
                          00000000G 00              02  FB 00060 4$:    CALLS   #2, FILE_ERROR                  ;
                                                    04  00067 5$:       RET                                     ; 2350

; Routine Size:  104 bytes,    Routine Base:  CODE + 06B1

;  1257          2351  1
```

```
;  1259          2352  1 %SBTTL 'BUILD_QUAL_DESC - Build qualifier block descriptors'
;  1260          2353  1 ROUTINE BUILD_QUAL_DESC( PTR : REF BBLOCK, NAM : REF BBLOCK ): NOVALUE=
;  1261          2354  1
;  1262          2355  1 !++
;  1263          2356  1 !
;  1264          2357  1 ! FUNCTIONAL DESCRIPTION:
;  1265          2358  1 !       This routine fills in the descriptors in the qualifier block
;  1266          2359  1 !
;  1267          2360  1 ! INPUT PARAMETERS:
;  1268          2361  1 !       PTR             - Address of block
;  1269          2362  1 !       NAM             - Address of NAM block
;  1270          2363  1 !
;  1271          2364  1 ! IMPLICIT INPUTS:
;  1272          2365  1 !       NONE
;  1273          2366  1 !
;  1274          2367  1 ! OUTPUT PARAMETERS:
;  1275          2368  1 !       NONE
;  1276          2369  1 !
;  1277          2370  1 ! IMPLICIT OUTPUTS:
;  1278          2371  1 !       NONE
;  1279          2372  1 !
;  1280          2373  1 ! ROUTINE VALUE:
;  1281          2374  1 !       NONE
;  1282          2375  1 !
;  1283          2376  1 ! SIDE EFFECTS:
;  1284          2377  1 !
;  1285          2378  1 !--
;  1286          2379  1
;  1287          2380  2 BEGIN
;  1288          2381  2
;  1289          2382  2 ! Set up expanded-string descriptors.
;  1290          2383  2 !
;  1291          2384  2 BBLOCK[PTR[QUAL_EXP_DESC], DSC$W_LENGTH] = .NAM[NAM$B_ESL];
;  1292          2385  2 BBLOCK[PTR[QUAL_EXP_DESC], DSC$A_POINTER] = .NAM[NAM$L_ESA];
;  1293          2386  2 BBLOCK[PTR[QUAL_DEV_DESC], DSC$W_LENGTH] = .NAM[NAM$B_DEV];
;  1294          2387  2 BBLOCK[PTR[QUAL_DEV_DESC], DSC$A_POINTER] = .NAM[NAM$L_DEV];
;  1295          2388  2 BBLOCK[PTR[QUAL_DVI_DESC], DSC$W_LENGTH] = .(NAM[NAM$T_DVI])<0,8>;
;  1296          2389  2 BBLOCK[PTR[QUAL_DVI_DESC], DSC$A_POINTER] = NAM[NAM$T_DVI] + 1;
;  1297          2390  2
;  1298          2391  2 RETURN ;
;  1299          2392  2
;  1300          2393  1 END ;


                                0004 00000 BUILD_QUAL_DESC:
                                           .WORD   Save R2                                          ; 2353
                     50    04   AC  7D 00002          MOVQ    PTR, R0                             ; 2384
                     52    08   A0  9E 00006          MOVAB   8(R0), R2                           ;
                     62    0B   A1  9B 0000A          MOVZBW  11(R1), (R2)                        ;
               04    A2    0C   A1  D0 0000E          MOVL    12(R1), 4(R2)                       ; 2385
                     52    10   A0  9E 00013          MOVAB   16(R0), R2                          ; 2386
                     62    39   A1  9B 00017          MOVZBW  57(R1), (R2)                        ;
               04    A2    44   A1  D0 0001B          MOVL    68(R1), 4(R2)                       ; 2387
                     50          18  C0 00020          ADDL2   #24, R0                             ; 2388
```

```
                                 60        14    A1  9B 00023         MOVZBW  20(R1), (R0)                                    ;
                              04 A0        15    A1  9E 00027         MOVAB   21(R1), 4(R0)                                   ; 2389
                                                     04 0002C         RET                                                     ; 2393

; Routine Size:  45 bytes,    Routine Base:  CODE + 0719

; 1301          2394  1
```

```
;1303    2395  1 %SBTTL 'CHECK - check input and output lists'
;1304    2396  1 ROUTINE CHECK ( ROOT : REF BBLOCK, OUTPUT_SIDE ): NOVALUE=
;1305    2397  1
;1306    2398  1 !++
;1307    2399  1 !
;1308    2400  1 ! FUNCTIONAL DESCRIPTION:
;1309    2401  1 !       This routine validity-checks the input and output lists.
;1310    2402  1 !
;1311    2403  1 ! INPUT PARAMETERS:
;1312    2404  1 !       ROOT            - Address of root of the input or output list.
;1313    2405  1 !       OUTPUT_SIDE     - True if output, false if input.
;1314    2406  1 !
;1315    2407  1 ! IMPLICIT INPUTS:
;1316    2408  1 !       NONE
;1317    2409  1 !
;1318    2410  1 ! OUTPUT PARAMETERS:
;1319    2411  1 !       NONE
;1320    2412  1 !
;1321    2413  1 ! IMPLICIT OUTPUTS:
;1322    2414  1 !       NONE
;1323    2415  1 !
;1324    2416  1 ! ROUTINE VALUE:
;1325    2417  1 !       NONE
;1326    2418  1 !
;1327    2419  1 ! SIDE EFFECTS:
;1328    2420  1 !       Messages may be signalled if errors are detected.
;1329    2421  1 !
;1330    2422  1 !--
;1331    2423  1
;1332    2424  2 BEGIN
;1333    2425  2
;1334    2426  2 LITERAL
;1335    2427  2     ALL_WILD_LEN =      %CHARCOUNT('*.*;*'),
;1336    2428  2     WILD_VERS_LEN =     %CHARCOUNT(';*');
;1337    2429  2
;1338    2430  2 BIND
;1339    2431  2     ALL_WILD =          UPLIT BYTE('*.*;*'),
;1340    2432  2     WILD_VERS =         UPLIT BYTE(';*');
;1341    2433  2
;1342    2434  2 LOCAL
;1343    2435  2     FAB                 : REF BBLOCK,
;1344    2436  2     NAM                 : REF BBLOCK,
;1345    2437  2     FC                  : REF BBLOCK,
;1346    2438  2     WORK_FAB            : REF BBLOCK,
;1347    2439  2     WORK_NAM            : REF BBLOCK,
;1348    2440  2     WORK_FC             : REF BBLOCK,
;1349    2441  2     PTR                 : REF BBLOCK,   ! Cursor for list
;1350    2442  2     LAST_PTR            : REF BBLOCK;   ! Previous Cursor for list
;1351    2443  2
;1352    2444  2 LAST_PTR = .ROOT;
;1353    2445  2 PTR = ..ROOT;
;1354    2446  2 CHK_FLAGS = 0;
;1355    2447  2
;1356    2448  2 !
;1357    2449  2 ! If no files then return
;1358    2450  2 !
;1359    2451  2 IF .PTR EQL 0
```

```
;  1360          2452  2 THEN
;  1361          2453  3     BEGIN
;  1362          2454  3     QUAL[QUAL_OF11] = .CHK_FLAGS[CHK_FLAGS_F11];
;  1363          2455  3     QUAL[QUAL_OSAV] = .CHK_FLAGS[CHK_FLAGS_SAV];
;  1364          2456  3     RETURN ;
;  1365          2457  2     END ;
;  1366          2458  2
;  1367          2459  2 !
;  1368          2460  2 ! Allocate an FC block to do the work with and initialize
;  1369          2461  2 ! it to match the first FC in the list
;  1370          2462  2 !
;  1371          2463  2 FC = .PTR[QUAL_PARA_FC] ;
;  1372          2464  2 FAB = FC[FC_FAB] ;
;  1373          2465  2 NAM = FC[FC_NAM] ;
;  1374          2466  2 WORK_FC = GET_VM ( FC_S_AREA ) ;
;  1375          2467  2 COPY_FC ( .FC, .WORK_FC, .FAB[FAB$B_FNS], .FAB[FAB$L_FNA] ) ;
;  1376          2468  2 WORK_FAB = WORK_FC[FC_FAB] ;
;  1377          2469  2 WORK_NAM = WORK_FC[FC_NAM] ;
;  1378          2470  2 WORK_NAM[NAM$L_RLF] = 0 ;
;  1379          2471  2
;  1380          2472  2
;  1381          2473  2 ! Parse search lists for input
;  1382          2474  2 !
;  1383          2475  2 IF NOT .OUTPUT_SIDE
;  1384          2476  2 THEN
;  1385          2477  2     CHK_FLAGS[CHK_FLAGS_SRCH_LIST] = 1 ;
;  1386          2478  2
;  1387          2479  2 WHILE .PTR NEQ 0 DO
;  1388          2480  3     BEGIN
;  1389          2481  3
;  1390          2482  3     IF .CHK_FLAGS[CHK_FLAGS_COUNT] NEQ 0
;  1391          2483  3     THEN
;  1392          2484  4         BEGIN
;  1393          2485  4         !
;  1394          2486  4         ! This is not the first time through
;  1395          2487  4         ! so reinit the FC
;  1396          2488  4         !
;  1397          2489  4         FC = .PTR[QUAL_PARA_FC] ;
;  1398          2490  4         FAB = FC[FC_FAB] ;
;  1399          2491  4         NAM = FC[FC_NAM] ;
;  1400          2492  4         INIT_FC ( .WORK_FC, .FAB[FAB$B_FNS], .FAB[FAB$L_FNA] ) ;
;  1401          2493  4         END ;
;  1402          2494  3
;  1403          2495  3     CHK_FLAGS[CHK_FLAGS_COUNT] = .CHK_FLAGS[CHK_FLAGS_COUNT] + 1;
;  1404          2496  3
;  1405          2497  3     IF .OUTPUT_SIDE
;  1406          2498  3         THEN CHK_FLAGS[CHK_FLAGS_SAVE_SET] = .QUAL[QUAL_O_SAVE]
;  1407          2499  3         ELSE CHK_FLAGS[CHK_FLAGS_SAVE_SET] = .QUAL[QUAL_I_SAVE];
;  1408          2500  3
;  1409          2501  3     IF NOT $PARSE(FAB=.WORK_FAB)
;  1410          2502  3     THEN
;  1411          2503  3         IF .WORK_FAB[FAB$L_STS] EQL RMS$_DNF
;  1412          2504  3         THEN
;  1413          2505  3             WORK_FAB[FAB$L_DEV] = DEV$M_FOD OR DEV$M_MNT
;  1414          2506  3         ELSE
;  1415          2507  3             FILE_ERROR(BACKUP$_PARSE, .WORK_FAB,
;  1416          2508  3                        .WORK_FAB[FAB$L_STS], .WORK_FAB[FAB$L_STV]);
```

```
;  1417          2509  3
;  1418          2510  3
;  1419          2511  3      IF
;  1420          2512  3          NOT .BBLOCK[WORK_FAB[FAB$L_DEV], DEV$V_FOD] AND
;  1421          2513  5          NOT ((.BBLOCK[WORK_FAB[FAB$L_DEV], DEV$V_NET] OR
;  1422          2514  4              .BBLOCK[WORK_FAB[FAB$L_DEV], DEV$V_MBX]) AND .CHK_FLAGS[CHK_FLAGS_SAVE_SET])
;  1423          2515  3      THEN
;  1424          2516  3          FILE_ERROR(BACKUP$_INVDEVTYP, .WORK_FAB);
;  1425          2517  3
;  1426          2518  3
;  1427          2519  3      IF
;  1428          2520  3          .BBLOCK[WORK_FAB[FAB$L_DEV], DEV$V_SQD] OR
;  1429          2521  3          .CHK_FLAGS[CHK_FLAGS_SAVE_SET]
;  1430          2522  3      THEN
;  1431          2523  4          BEGIN
;  1432          2524  4          CHK_FLAGS[CHK_FLAGS_SAV] = TRUE;
;  1433          2525  4          IF .BBLOCK[WORK_FAB[FAB$L_DEV], DEV$V_FOR]
;  1434          2526  4          THEN
;  1435          2527  5              BEGIN
;  1436          2528  5              IF .CHK_FLAGS[CHK_FLAGS_COUNT] EQL 1
;  1437          2529  5              THEN
;  1438          2530  6                  BEGIN
;  1439          2531  6                  WORK_FAB[FAB$B_DNS] = %CHARCOUNT('[000000]');
;  1440          2532  6                  WORK_FAB[FAB$L_DNA] = UPLIT BYTE('[000000]');
;  1441          2533  6                  $PARSE(FAB=.WORK_FAB);
;  1442          2534  5                  END;
;  1443          2535  5              END
;  1444          2536  4          ELSE
;  1445          2537  4              QUAL[QUAL_SS_FILE] = TRUE;
;  1446          2538  4
;  1447          2539  4          CHECK_SAV ( .WORK_NAM, .OUTPUT_SIDE ) ;
;  1448          2540  4          CHECK_COMMON ( .WORK_FAB, .OUTPUT_SIDE ) ;
;  1449          2541  4
;  1450          2542  4          !
;  1451          2543  4          ! Now copy the parsed FC to the list
;  1452          2544  4          !
;  1453          2545  4          COPY_FC ( .WORK_FC, .FC, .WORK_NAM[NAM$B_ESL], .WORK_NAM[NAM$L_ESA] ) ;
;  1454          2546  4
;  1455          2547  4          ! Set up expanded-string descriptors.
;  1456          2548  4          !
;  1457          2549  4          BUILD_QUAL_DESC ( .PTR, .NAM ) ;
;  1458          2550  4
;  1459          2551  4          END
;  1460          2552  3      ELSE
;  1461          2553  4          BEGIN
;  1462          2554  4          LOCAL
;  1463          2555  4              NEW_PTR     : LONG,
;  1464          2556  4              WILD_RLF    : REF BBLOCK,
;  1465          2557  4              RNAM        : REF BBLOCK;
;  1466          2558  4
;  1467          2559  4          NEW_PTR = 0 ;
;  1468          2560  4
;  1469          2561  4          CHK_FLAGS[CHK_FLAGS_F11] = TRUE;
;  1470          2562  4
;  1471          2563  4          IF .CHK_FLAGS[CHK_FLAGS_COUNT] EQL 1
;  1472          2564  4          THEN
;  1473          2565  5              BEGIN
```

```
;1474     2566  5              !
;1475     2567  5              ! The first related file name must be *.*;*
;1476     2568  5              ! and the default name string must be *.*;* for
;1477     2569  5              ! the first round and then the default must be changed
;1478     2570  5              ! to ;* for each successive round so that
;1479     2571  5              ! stickyness works properly and unless an explicit
;1480     2572  5              ! version is requested all versions will be saved.
;1481     2573  5              !
;1482     2574  5              WILD_RLF = GET_VM ( ALL_WILD_LEN ) ;
;1483     2575  5              CH$MOVE ( ALL_WILD_LEN, ALL_WILD, .WILD_RLF );
;1484     2576  5              RNAM = GET_VM (NAM$C_BLN) ;
;1485   P 2577  5              $NAM_INIT (
;1486   P 2578  5                  NAM = .RNAM,
;1487   P 2579  5                  RLF = .WORK_NAM[NAM$L_RLF],
;1488   P 2580  5                  RSS = ALL_WILD_LEN ,
;1489   P 2581  5                  RSA = .WILD_RLF
;1490     2582  5                  ) ;
;1491     2583  5              RNAM[NAM$B_RSL] = ALL_WILD_LEN ;
;1492     2584  5              WORK_NAM[NAM$L_RLF] = .RNAM ;
;1493     2585  5              WORK_FAB[FAB$B_DNS] = ALL_WILD_LEN ;
;1494     2586  5              WORK_FAB[FAB$L_DNA] = ALL_WILD;
;1495     2587  5              END
;1496     2588  4          ELSE
;1497     2589  5              BEGIN
;1498     2590  5              WORK_FAB[FAB$B_DNS] = WILD_VERS_LEN ;
;1499     2591  5              WORK_FAB[FAB$L_DNA] = WILD_VERS;
;1500     2592  4              END ;
;1501     2593  4
;1502     2594  4          CHECK_F11 ( .WORK_NAM, .OUTPUT_SIDE ) ;
;1503     2595  4
;1504     2596  4          IF .CHK_FLAGS [CHK_FLAGS_SRCH_LIST]
;1505     2597  4          THEN
;1506     2598  4              ! Expand the search list.  If zero is returned then no search
;1507     2599  4              ! list was present.  If this is a search list then it will
;1508     2600  4              ! be expanded and the block pointed to by PTR will be replaced
;1509     2601  4              ! by a list, one for each expansion.  The routine value will
;1510     2602  4              ! also be the address of the last element in the list.
;1511     2603  4              !
;1512     2604  4              NEW_PTR = EXPAND_SEARCH_LIST ( .WORK_FC, .PTR, .LAST_PTR ) ;
;1513     2605  4
;1514     2606  4          IF .NEW_PTR NEQ 0
;1515     2607  4          THEN PTR = .NEW_PTR
;1516     2608  4          ELSE
;1517     2609  5              BEGIN
;1518     2610  5
;1519     2611  5              LOCAL
;1520     2612  5                  STATUS : LONG ;
;1521     2613  5
;1522     2614  5              STATUS = $PARSE(FAB=.WORK_FAB) ;
;1523     2615  5
;1524     2616  5              CHECK_COMMON ( .WORK_FAB, .OUTPUT_SIDE ) ;
;1525     2617  5
;1526     2618  5              !
;1527     2619  5              ! Now copy the parsed FC to the list
;1528     2620  5              !
;1529     2621  5              IF NOT .STATUS
;1530     2622  5              THEN
```

```
; 1531          2623  6                 BEGIN
; 1532          2624  6                 IF .WORK_FAB[FAB$L_STS] EQL RMS$_DNF
; 1533          2625  6                 THEN
; 1534          2626  7                     BEGIN
; 1535          2627  7                     WORK_NAM[NAM$V_SYNCHK] = 1 ;        ! Parse with no IO
; 1536          2628  7                     STATUS = $PARSE(FAB=.WORK_FAB) ;
; 1537          2629  7                     WORK_NAM[NAM$V_SYNCHK] = 0 ;
; 1538          2630  6                     END ;
; 1539          2631  6                 IF NOT .STATUS
; 1540          2632  6                 THEN
; 1541          2633  6                     FILE_ERROR(BACKUP$_PARSE, .WORK_FAB,
; 1542          2634  6                                .WORK_FAB[FAB$L_STS], .WORK_FAB[FAB$L_STV]);
; 1543          2635  5                 END ;
; 1544          2636  5
; 1545          2637  6             IF (.QUAL[QUAL_INCR] AND .OUTPUT_SIDE)
; 1546          2638  5             THEN
; 1547          2639  5                 COPY_FC ( .WORK_FC, .FC,
; 1548          2640  5                           .WORK_FAB[FAB$B_FNS], .WORK_FAB[FAB$L_FNA] )
; 1549          2641  5             ELSE
; 1550          2642  5                 COPY_FC ( .WORK_FC, .FC,
; 1551          2643  5                           .WORK_NAM[NAM$B_ESL], .WORK_NAM[NAM$L_ESA] ) ;
; 1552          2644  5
; 1553          2645  5                 ! Set up expanded-string descriptors.
; 1554          2646  5                 !
; 1555          2647  5                 BUILD_QUAL_DESC ( .PTR, .NAM ) ;
; 1556          2648  5
; 1557          2649  4                 END ;
; 1558          2650  3         END;
; 1559          2651  3
; 1560          2652  3
; 1561          2653  3         ! Get next list element
; 1562          2654  3         !
; 1563          2655  3         LAST_PTR = .PTR ;
; 1564          2656  3         PTR = .PTR[QUAL_NEXT];
; 1565          2657  2         END;
; 1566          2658  2
; 1567          2659  2
; 1568          2660  2 IF .OUTPUT_SIDE
; 1569          2661  2 THEN
; 1570          2662  3     BEGIN
; 1571          2663  3     QUAL[QUAL_OF11] = .CHK_FLAGS[CHK_FLAGS_F11];
; 1572          2664  3     QUAL[QUAL_OSAV] = .CHK_FLAGS[CHK_FLAGS_SAV];
; 1573          2665  3     END
; 1574          2666  2 ELSE
; 1575          2667  3     BEGIN
; 1576          2668  3     QUAL[QUAL_IF11] = .CHK_FLAGS[CHK_FLAGS_F11];
; 1577          2669  3     QUAL[QUAL_ISAV] = .CHK_FLAGS[CHK_FLAGS_SAV];
; 1578          2670  2     END;
; 1579          2671  2
; 1580          2672  2 !
; 1581          2673  2 ! Clean up our work space.
; 1582          2674  2 ! First deallocate all of the RNAM's
; 1583          2675  2 !
; 1584          2676  2 NAM = .WORK_NAM[NAM$L_RLF] ;
; 1585          2677  2 WHILE .NAM NEQ 0 DO
; 1586          2678  3     BEGIN
; 1587          2679  3     PTR = .NAM[NAM$L_RLF] ;
```

```
;  1588          2680  3     IF .NAM[NAM$L_RSA] NEQ 0
;  1589          2681  3     THEN
;  1590          2682  3         FREE_VM ( .NAM[NAM$B_RSS], .NAM[NAM$L_RSA] ) ;
;  1591          2683  3     FREE_VM ( NAM$C_BLN, .NAM ) ;
;  1592          2684  3     NAM = .PTR ;
;  1593          2685  2     END ;
;  1594          2686  2
;  1595          2687  2 !
;  1596          2688  2 ! now the FC
;  1597          2689  2 !
;  1598          2690  2 IF .WORK_FAB[FAB$L_FNA] NEQ 0
;  1599          2691  2 THEN
;  1600          2692  2     FREE_VM ( .WORK_FAB[FAB$B_FNS], .WORK_FAB[FAB$L_FNA] ) ;
;  1601          2693  2 FREE_VM ( FC_S_AREA, .WORK_FC ) ;
;  1602          2694  2
;  1603          2695  1 END;


                                                  2A  3B  2A  2E  2A  00746 P.AAF:  .ASCII  \*.*;*\                          ;
                                                              2A  3B  0074B P.AAG:  .ASCII  \;*\                             ;
                                              5D  30  30  30  30  30  30  5B  0074D P.AAH:  .ASCII  \[000000]\                 ;

                                                                              ALL_WILD=           P.AAF
                                                                              WILD_VERS=          P.AAG

                                                                        0FFC 00000 CHECK:  .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11  ; 2396
                                                          5E                14  C2 00002       SUBL2  #20, SP                          ;
                                                                  04        AC  DD 00005       PUSHL  ROOT                             ; 2444
                                                                  04        BC  DD 00008       PUSHL  @ROOT                            ; 2445
                                                              00000000'     EF  D4 0000B       CLRL   CHK_FLAGS                        ; 2446
                                                                            6E  D5 00011       TSTL   PTR                              ; 2451
                                                                            20  12 00013       BNEQ   1$                               ;
00000000'  EF                 01                          06 00000000'    EF  F0 00015       INSV   CHK_FLAGS+1, #6, #1, QUAL+15     ; 2454
           50 00000000'       EF                          01              01  EF 00022       EXTZV  #1, #1, CHK_FLAGS+1, R0          ; 2455
00000000'  EF                 01                          07              50  F0 0002B       INSV   R0, #7, #1, QUAL+15              ;
                                                                            04  00034       RET                                     ; 2453
                              50                          6E              04  C1 00035 1$:   ADDL3  #4, PTR, R0                      ; 2463
                                                          57              60  D0 00039       MOVL   (R0), FC                         ;
                                                          5B              57  D0 0003C       MOVL   FC, FAB                          ; 2464
                                                          56        0094  C7  9E 0003F       MOVAB  148(R7), NAM                     ; 2465
                                                          7E        0354  8F  3C 00044       MOVZWL #852, -(SP)                      ; 2466
                                  00000000G               00              01  FB 00049       CALLS  #1, GET_VM                       ;
                                         18               AE              50  D0 00050       MOVL   R0, WORK_FC                      ;
                                                                  2C        AB  DD 00054       PUSHL  44(FAB)                          ; 2467
                                                          7E      34        AB  9A 00057       MOVZBL 52(FAB), -(SP)                   ;
                                                                  20        AE  DD 0005B       PUSHL  WORK_FC                          ;
                                                                            57  DD 0005E       PUSHL  FC                               ;
                                             F93E         CF                  04  FB 00060       CALLS  #4, COPY_FC                      ;
                                                          59      18        AE  D0 00065       MOVL   WORK_FC, WORK_FAB                ; 2468
                              5A          18              AE  00000094      8F  C1 00069       ADDL3  #148, WORK_FC, WORK_NAM          ; 2469
                                                                  10        AA  D4 00072       CLRL   16(WORK_NAM)                     ; 2470
                                          0C              AE      08        AC  D0 00075       MOVL   OUTPUT_SIDE, 12(SP)              ; 2475
                                                          07      0C        AE  E8 0007A       BLBS   12(SP), 2$                       ;
                                  00000000'               EF              08  88 0007E       BISB2  #8, CHK_FLAGS+1                  ; 2477
                                                 08       AE      40        A9  9E 00085 2$:   MOVAB  64(WORK_FAB), 8(SP)              ; 2512
```

```
                                               6E  D5 0008A 3$:     TSTL    PTR                                  ; 2479
                                               03  12 0008C         BNEQ    4$                                   ;
                                             021E  31 0008E         BRW     26$                                  ;
                                 00000000'     EF  95 00091 4$:     TSTB    CHK_FLAGS                            ; 2482
                                               1E  13 00097         BEQL    5$                                   ;
                     50                6E      04  C1 00099         ADDL3   #4, PTR, R0                          ; 2489
                                       57      60  D0 0009D         MOVL    (R0), FC                             ;
                                       5B      57  D0 000A0         MOVL    FC, FAB                              ; 2490
                                       56    0094  C7  9E 000A3     MOVAB   148(R7), NAM                         ; 2491
                                               2C  AB  DD 000A8     PUSHL   44(FAB)                              ; 2492
                                       7E      34  AB  9A 000AB     MOVZBL  52(FAB), -(SP)                       ;
                                               20  AE  DD 000AF     PUSHL   WORK_FC                              ;
                              FAA6     CF      03  FB 000B2         CALLS   #3, INIT_FC                          ;
                                 00000000'     EF  96 000B7 5$:     INCB    CHK_FLAGS                            ; 2495
                                       0B      0C  AE  E9 000BD     BLBC    12(SP), 6$                           ; 2497
            50 00000000'     EF                01  02  EF 000C1     EXTZV   #2, #1, QUAL+13, R0                  ; 2498
                                               09  11 000CA         BRB     7$                                   ;
            50 00000000'     EF                01  01  EF 000CC 6$: EXTZV   #1, #1, QUAL+13, R0                  ; 2499
00000000'  EF            01                02  50  F0 000D5 7$:     INSV    R0, #2, #1, CHK_FLAGS+1              ;
                                               59  DD 000DE         PUSHL   WORK_FAB                             ; 2501
                         00000000G     00      01  FB 000E0         CALLS   #1, SYS$PARSE                        ;
                                       27      50  E8 000E7         BLBS    R0, 9$                               ;
                         0001C04A      8F      08  A9  D1 000EA     CMPL    8(WORK_FAB), #114762                 ; 2503
                                               0A  12 000F2         BNEQ    8$                                   ;
                              40       A9 00084000 8F  D0 000F4     MOVL    #540672, 64(WORK_FAB)                ; 2505
                                               13  11 000FC         BRB     9$                                   ;
                                       7E      08  A9  7D 000FE 8$: MOVQ    8(WORK_FAB), -(SP)                   ; 2508
                                               59  DD 00102         PUSHL   WORK_FAB                             ; 2507
                                 00000000G     8F  DD 00104         PUSHL   #BACKUP$_PARSE                       ;
                         00000000G     00      04  FB 0010A         CALLS   #4, FILE_ERROR                       ;
                     21        08      BE      0E  E0 00111 9$:     BBS     #14, @8(SP), 12$                     ; 2512
                     05        08      BE      0D  E0 00116         BBS     #13, @8(SP), 10$                     ; 2513
                     08        08      BE      14  E1 0011B         BBC     #20, @8(SP), 11$                     ; 2514
                     0F  00000000'     EF      02  E0 00120 10$:    BBS     #2, CHK_FLAGS+1, 12$                 ;
                                               59  DD 00128 11$:    PUSHL   WORK_FAB                             ; 2516
                                 00000000G     8F  DD 0012A         PUSHL   #BACKUP$_INVDEVTYP                   ;
                         0000[illegible]       00      02  FB 00130 CALLS   #2, FILE_ERROR                       ;
                     08        [illegible]     BE      05  E0 00137 12$: BBS #5, @8(SP), 13$                     ; 2520
                     48  00000000'     EF      02  E1 0013C         BBC     #2, CHK_FLAGS+1, 16$                 ; 2521
                         00000000'     EF      02  88 00144 13$:    BISB2   #2, CHK_FLAGS+1                      ; 2524
                     1E        08      BE      18  E1 0014B         BBC     #24, @8(SP), 14$                     ; 2525
                                       01 00000000' EF  91 00150     CMPB    CHK_FLAGS, #1                        ; 2528
                                               1C  12 00157         BNEQ    15$                                  ;
                               35      A9      08  90 00159         MOVB    #8, 53(WORK_FAB)                     ; 2531
                               30      A9    FE97  CF  9E 0015D     MOVAB   P.AAH, 48(WORK_FAB)                  ; 2532
                                               59  DD 00163         PUSHL   WORK_FAB                             ; 2533
                         00000000G     00      01  FB 00165         CALLS   #1, SYS$PARSE                        ;
                                               07  11 0016C         BRB     15$                                  ; 2525
                         00000000'     EF      08  88 0016E 14$:    BISB2   #8, QUAL+15                          ; 2537
                                               0C  AE  DD 00175 15$: PUSHL  12(SP)                               ; 2539
                                               5A  DD 00178         PUSHL   WORK_NAM                             ;
                              FC78     CF      02  FB 0017A         CALLS   #2, CHECK_SAV                        ;
                                               0C  AE  DD 0017F     PUSHL   12(SP)                               ; 2540
                                               59  DD 00182         PUSHL   WORK_FAB                             ;
                              FDD3     CF      02  FB 00184         CALLS   #2, CHECK_COMMON                     ;
                                             00FF  31 00189         BRW     23$                                  ; 2545
                                               14  AE  D4 0018C 16$: CLRL   NEW_PTR                              ; 2559
```

```
              00000000'  EF                01 88 0018F          BISB2   #1, CHK_FLAGS+1                      ; 2561
              01 00000000'  EF             91 00196             CMPB    CHK_FLAGS, #1                        ; 2563
                            51             12 0019D             BNEQ    17$
                            05             DD 0019F             PUSHL   #5                                   ; 2574
              00000000G  00                01 FB 001A1          CALLS   #1, GET_VM
                     10  AE                50 D0 001A8          MOVL    R0, WILD_RLF
        10   BE    FE40  CF                05 28 001AC          MOVC3   #5, ALL_WILD, @WILD_RLF              ; 2575
                         7E         60     8F 9A 001B3          MOVZBL  #96, -(SP)                           ; 2576
              00000000G  00                01 FB 001B7          CALLS   #1, GET_VM
                         58                50 D0 001BE          MOVL    R0, RNAM
  0060   8F          00  6E                00 2C 001C1          MOVC5   #0, (SP), #0, #96, (RNAM)            ; 2582
                         68                      001C8
                         68       6002     8F B0 001C9          MOVW    #24578, (RNAM)
                     02  A8                05 90 001CE          MOVB    #5, 2(RNAM)
                     04  A8         10     AE D0 001D2          MOVL    WILD_RLF, 4(RNAM)
                     10  A8         10     AA D0 001D7          MOVL    16(WORK_NAM), 16(RNAM)
                     03  A8                05 90 001DC          MOVB    #5, 3(RNAM)                          ; 2583
                     10  AA                58 D0 001E0          MOVL    RNAM, 16(WORK_NAM)                   ; 2584
                     35  A9                05 90 001E4          MOVB    #5, 53(WORK_FAB)                     ; 2585
                     30  A9       FE05     CF 9E 001E8          MOVAB   ALL_WILD, 48(WORK_FAB)               ; 2586
                                           0A 11 001EE          BRB     18$                                  ; 2563
                     35  A9                02 90 001F0 17$:     MOVB    #2, 53(WORK_FAB)                     ; 2590
                     30  A9       FDFE     CF 9E 001F4          MOVAB   WILD_VERS, 48(WORK_FAB)              ; 2591
                                    0C     AE DD 001FA 18$:     PUSHL   12(SP)                               ; 2594
                                           5A DD 001FD          PUSHL   WORK_NAM
                   FC85  CF                02 FB 001FF          CALLS   #2, CHECK_F11
           12 00000000'  EF                03 E1 00204          BBC     #3, CHK_FLAGS+1, 19$                 ; 2596
                                    04     AE DD 0020C          PUSHL   LAST_PTR                             ; 2604
                                    04     AE DD 0020F          PUSHL   PTR
                                    20     AE DD 00212          PUSHL   WORK_FC
                   FAA7  CF                03 FB 00215          CALLS   #3, EXPAND_SEARCH_LIST
                     14  AE                50 D0 0021A          MOVL    R0, NEW_PTR
                                    14     AE D5 0021E 19$:     TSTL    NEW_PTR                              ; 2606
                                           06 13 00221          BEQL    20$
                         6E         14     AE D0 00223          MOVL    NEW_PTR, PTR                         ; 2607
                                           7D 11 00227          BRB     25$
                                           59 DD 00229 20$:     PUSHL   WORK_FAB                             ; 2614
              00000000G  00                01 FB 0022B          CALLS   #1, SYS$PARSE
                         52                50 D0 00232          MOVL    R0, STATUS
                                    0C     AE DD 00235          PUSHL   12(SP)                               ; 2616
                                           59 DD 00238          PUSHL   WORK_FAB
                   FD1D  CF                02 FB 0023A          CALLS   #2, CHECK_COMMON
                         34                52 E8 0023F          BLBS    STATUS, 22$                          ; 2621
              0001C04A   8F         08     A9 D1 00242          CMPL    8(WORK_FAB), #114762                 ; 2624
                                           14 12 0024A          BNEQ    21$
                     08  AA                08 88 0024C          BISB2   #8, 8(WORK_NAM)                      ; 2627
                                           59 DD 00250          PUSHL   WORK_FAB                             ; 2628
              00000000G  00                01 FB 00252          CALLS   #1, SYS$PARSE
                         52                50 D0 00259          MOVL    R0, STATUS
                     08  AA                08 8A 0025C          BICB2   #8, 8(WORK_NAM)                      ; 2629
                         13                52 E8 00260 21$:     BLBS    STATUS, 22$                          ; 2631
                         7E         08     A9 7D 00263          MOVQ    8(WORK_FAB), -(SP)                   ; 2634
                                           59 DD 00267          PUSHL   WORK_FAB                             ; 2633
                              00000000G    8F DD 00269          PUSHL   #BACKUP$_PARSE
              00000000G  00                04 FB 0026F          CALLS   #4, FILE_ERROR
           0D 00000000'  EF                04 E1 00276 22$:     BBC     #4, QUAL+10, 23$                     ; 2637
                         09         0C     AE E9 0027E          BLBC    12(SP), 23$
```

```
                                                 2C   A9  DD 00282          PUSHL   44(WORK_FAB)                          ; 2640
                                              7E 34   A9  9A 00285          MOVZBL  52(WORK_FAB), -(SP)
                                                 07   11  00289          BRB     24$                                   ; 2639
                                                 0C   AA  DD 0028B 23$:    PUSHL   12(WORK_NAM)                          ; 2643
                                              7E 0B   AA  9A 0028E          MOVZBL  11(WORK_NAM), -(SP)
                                                 57   DD  00292 24$:       PUSHL   FC                                    ; 2642
                                                 24   AE  DD 00294          PUSHL   WORK_FC
                                           F707  CF       04  FB 00297      CALLS   #4, COPY_FC
                                                 56   DD  0029C          PUSHL   NAM                                   ; 2647
                                                 04   AE  DD 0029E          PUSHL   PTR
                                           FD1E  CF       02  FB 002A1      CALLS   #2, BUILD_QUAL_DESC
                                             04  AE       6E  D0 002A6 25$:    MOVL    PTR, LAST_PTR                 ; 2655
                                                 9E   DD  002AA          PUSHL   @PTR                                  ; 2656
                                                 FDDB 31  002AC          BRW     3$                                    ; 2479
                                              21 0C   AE  E9 002AF 26$:    BLBC    12(SP), 27$                           ; 2664
00000000'  EF          01                    06 00000000' EF  F0 002B3      INSV    CHK_FLAGS+1, #6, #1, QUAL+15          ; 2663
           50 00000000' EF                   01           01  EF 002C0      EXTZV   #1, #1, CHK_FLAGS+1, R0               ; 2664
00000000'  EF          01                    07           50  F0 002C9      INSV    R0, #7, #1, QUAL+15
                                                 1F   11  002D2          BRB     28$                                   ; 2660
00000000'  EF          01                    04 00000000' EF  F0 002D4 27$:    INSV    CHK_FLAGS+1, #4, #1, QUAL+15          ; 2668
           50 00000000' EF                   01           01  EF 002E1      EXTZV   #1, #1, CHK_FLAGS+1, R0               ; 2669
00000000'  EF          01                    05           50  F0 002EA      INSV    R0, #5, #1, QUAL+15
                                              56 10   AA  D0 002F3 28$:    MOVL    16(WORK_NAM), NAM                     ; 2676
                                                 29   13  002F7 29$:    BEQL    31$                                   ; 2677
                                              6E 10   A6  D0 002F9          MOVL    16(NAM), PTR                          ; 2679
                                                 04   A6  D5 002FD          TSTL    4(NAM)                                ; 2680
                                                 0E   13  00300          BEQL    30$
                                                 04   A6  DD 00302          PUSHL   4(NAM)                                ; 2682
                                              7E 02   A6  9A 00305          MOVZBL  2(NAM), -(SP)
                                     00000000G 00         02  FB 00309      CALLS   #2, FREE_VM
                                                 56   DD  00310 30$:    PUSHL   NAM                                   ; 2683
                                              7E 60   8F  9A 00312          MOVZBL  #96, -(SP)
                                     00000000G 00         02  FB 00316      CALLS   #2, FREE_VM
                                                 56   6E  D0 0031D          MOVL    PTR, NAM                              ; 2684
                                                 D5   11  00320          BRB     29$                                   ; 2677
                                                 2C   A9  D5 00322 31$:    TSTL    44(WORK_FAB)                          ; 2690
                                                 0E   13  00325          BEQL    32$
                                                 2C   A9  DD 00327          PUSHL   44(WORK_FAB)                          ; 2692
                                              7E 34   A9  9A 0032A          MOVZBL  52(WORK_FAB), -(SP)
                                     00000000G 00         02  FB 0032E      CALLS   #2, FREE_VM
                                                 18   AE  DD 00335 32$:    PUSHL   WORK_FC                               ; 2693
                                              7E 0354 8F  3C 00338          MOVZWL  #852, -(SP)
                                     00000000G 00         02  FB 0033D      CALLS   #2, FREE_VM
                                                          04  00344          RET                                           ; 2695

; Routine Size:  837 bytes,    Routine Base:  CODE + 0755

; 1604          2696  1
```

```
;  1606          2697  1 %SBTTL 'COMMAND - main command parser'
;  1607          2698  1 GLOBAL ROUTINE COMMAND: NOVALUE=
;  1608          2699  1
;  1609          2700  1 !++
;  1610          2701  1 !
;  1611          2702  1 ! FUNCTIONAL DESCRIPTION:
;  1612          2703  1 !       This routine is the command parser.
;  1613          2704  1 !
;  1614          2705  1 ! INPUT PARAMETERS:
;  1615          2706  1 !       NONE
;  1616          2707  1 !
;  1617          2708  1 ! IMPLICIT INPUTS:
;  1618          2709  1 !       NONE
;  1619          2710  1 !
;  1620          2711  1 ! OUTPUT PARAMETERS:
;  1621          2712  1 !       NONE
;  1622          2713  1 !
;  1623          2714  1 ! IMPLICIT OUTPUTS:
;  1624          2715  1 !       NONE
;  1625          2716  1 !
;  1626          2717  1 ! ROUTINE VALUE:
;  1627          2718  1 !       NONE
;  1628          2719  1 !
;  1629          2720  1 ! SIDE EFFECTS:
;  1630          2721  1 !       NONE
;  1631          2722  1 !
;  1632          2723  1 !--
;  1633          2724  1
;  1634          2725  2 BEGIN
;  1635          2726  2 MACRO
;  1636          2727  2
;  1637          2728  2         ! Generate static string descriptors.
;  1638          2729  2         !
;  1639        M 2730  2         SD[A]=
;  1640          2731  2             BIND %NAME('SD_', A) = $DESCRIPTOR(A) %;
;  1641          2732  2
;  1642          2733  2
;  1643          2734  2         ! Generate the static descriptors.
;  1644          2735  2         !
;  1645        P 2736  2         SD(
;  1646        P 2737  2             '$LINE',
;  1647        P 2738  2             'ANALYZE',
;  1648        P 2739  2             'BACKUP',
;  1649        P 2740  2             'BEFORE',
;  1650        P 2741  2             'BLOCK_SIZE',
;  1651        P 2742  2             'BRIEF',
;  1652        P 2743  2             'BUFFER_COUNT',
;  1653        P 2744  2             'COMMENT',
;  1654        P 2745  2             'COMPARE',
;  1655        P 2746  2             'CONFIRM',
;  1656        P 2747  2             'CRC',
;  1657        P 2748  2             'CREATED',
;  1658        P 2749  2             'DELETE',
;  1659        P 2750  2             'DENSITY',
;  1660        P 2751  2             'ENCRYPT',
;  1661        P 2752  2             'EXCLUDE',
;  1662        P 2753  2             'EXPIRED',
```

```
;  1663        P 2754  2                'FAST',
;  1664        P 2755  2                'FULL',
;  1665        P 2756  2                'GROUP_SIZE',
;  1666        P 2757  2                'IGNORE',
;  1667        P 2758  2                'IMAGE',
;  1668        P 2759  2                'INCREMENTAL',
;  1669        P 2760  2                'INITIALIZE',
;  1670        P 2761  2                'INPUTS',
;  1671        P 2762  2                'INTERCHANGE',
;  1672        P 2763  2                'JOURNAL',
;  1673        P 2764  2                'LABEL',
;  1674        P 2765  2                'LIST',
;  1675        P 2766  2                'LOG',
;  1676        P 2767  2                'MODIFIED',
;  1677        P 2768  2                'NEW_VERSION',
;  1678        P 2769  2                'OUTPUTS',
;  1679        P 2770  2                'OVERLAY',
;  1680        P 2771  2                'OWNER_UIC',
;  1681        P 2772  2                'PHYSICAL',
;  1682        P 2773  2                'PROTECTION',
;  1683        P 2774  2                'RECORD',
;  1684        P 2775  2                'REPLACE',
;  1685        P 2776  2                'REWIND',
;  1686        P 2777  2                'SAVE_SET',
;  1687        P 2778  2                'SELECT',
;  1688        P 2779  2                'SINCE',
;  1689        P 2780  2                'TRUNCATE',
;  1690        P 2781  2                'VERIFY',
;  1691          2782  2                'VOLUME');
;  1692          2783  2
;  1693          2784  2 MACRO
;  1694          2785  2         SD_ENCRYPT_NAME = $DESCRIPTOR('ENCRYPT.NAME') %,
;  1695          2786  2         SD_ENCRYPT_ALGO = $DESCRIPTOR('ENCRYPT.ALGORITHM') %,
;  1696          2787  2         SD_ENCRYPT_VALU = $DESCRIPTOR('ENCRYPT.VALUE') %;
;  1697          2788  2
;  1698          2789  2 LITERAL
;  1699          2790  2         MAX_STRING=     1024;           ! Maximum string value
;  1700          2791  2 LOCAL
;  1701          2792  2         BUFFER:         VOLATILE BBLOCK[MAX_STRING+2],
;  1702          2793  2                                         ! Buffer for string value
;  1703          2794  2         STATUS,                         ! Status variable
;  1704          2795  2         INIT_STATUS,                    ! CLI status for INITIALIZE qualifier
;  1705          2796  2         VALUE,                          ! Buffer for converted numerics
;  1706          2797  2         DESC:           BBLOCK[8],      ! Local descriptor
;  1707          2798  2         DESC2:          BBLOCK[8],      ! Second local descriptor
;  1708          2799  2         LAST:           REF BBLOCK;     ! Pointer to last value block
;  1709          2800  2
;  1710          2801  2
;  1711          2802  2 ! Execute services to get information needed by the utility.
;  1712          2803  2 !
;  1713          2804  2 IF
;  1714          2805  3     BEGIN
;  1715          2806  3     IF .COM_FLAGS[COM_STANDALONE]
;  1716          2807  3     THEN TRUE
;  1717        P 2808  3     ELSE IF $GETDVI(
;  1718        P 2809  3         DEVNAM=$DESCRIPTOR('SYS$COMMAND'),
;  1719        P 2810  3         ITMLST=
```

```
;  1720        P 2811  3              UPLIT(
;  1721        P 2812  3                  WORD(4, DVI$_DEVCLASS), DEVCLASS, 0,
;  1722          2813  4                  0))
;  1723          2814  3      THEN
;  1724          2815  4          BEGIN
;  1725          2816  4          $WAITFR(EFN=0);
;  1726          2817  4          .DEVCLASS EQL DC$_TERM
;  1727          2818  4          OR .DEVCLASS EQL DC$_MAILBOX
;  1728          2819  4          END
;  1729          2820  3      ELSE
;  1730          2821  3          FALSE
;  1731          2822  3      END
;  1732          2823  2  THEN
;  1733          2824  2      COM_FLAGS[COM_INTERACT] = TRUE;
;  1734          2825  2
;  1735          2826  2
;  1736        P 2827  2  $GETJPI(ITMLST=
;  1737        P 2828  2      UPLIT(
;  1738        P 2829  2          WORD(4, JPI$_UIC), JPI_UIC, 0,
;  1739        P 2830  2          WORD(12, JPI$_USERNAME), JPI_USERNAME, 0,
;  1740        P 2831  2          WORD(8, JPI$_CURPRIV), JPI_CURPRIV, 0,
;  1741          2832  2          0));
;  1742        P 2833  2  $GETSYI(ITMLST=
;  1743        P 2834  2      UPLIT(
;  1744        P 2835  2          WORD(4, SYI$_VERSION), SYI_VERSION, 0,
;  1745        P 2836  2          WORD(4, SYI$_SID), SYI_SID, 0,
;  1746          2837  2          0));
;  1747          2838  2  $GETTIM(TIMADR=JPI_DATE);
;  1748          2839  2  DESC[DSC$W_LENGTH] = MAX_STRING;
;  1749          2840  2  DESC[DSC$B_DTYPE] = 0;
;  1750          2841  2  DESC[DSC$B_CLASS] = 0;
;  1751          2842  2  DESC[DSC$A_POINTER] = BUFFER[VAR_BODY];
;  1752        P 2843  2  STATUS = $TRNLOG(
;  1753        P 2844  2      LOGNAM=$DESCRIPTOR('SYS$NODE'),
;  1754        P 2845  2      RSLLEN=BUFFER[VAR_LENGTH],
;  1755          2846  2      RSLBUF=DESC);
;  1756          2847  2  IF .STATUS AND .STATUS NEQ SS$_NOTRAN
;  1757          2848  2      THEN PRESERVE_STRING(BUFFER, JPI_NODE_DESC);
;  1758          2849  2
;  1759          2850  2
;  1760          2851  2  ! Initialize descriptor to describe the string buffer as a PL/I varying string.
;  1761          2852  2  ! Initialize second descriptor to describe the body portion -- the length will
;  1762          2853  2  ! be filled in as necessary.
;  1763          2854  2  !
;  1764          2855  2  DESC[DSC$W_MAXSTRLEN] = MAX_STRING;
;  1765          2856  2  DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  1766          2857  2  DESC[DSC$B_CLASS] = DSC$K_CLASS_VS;
;  1767          2858  2  DESC[DSC$A_POINTER] = BUFFER;
;  1768          2859  2  DESC2[DSC$W_LENGTH] = 0;
;  1769          2860  2  DESC2[DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  1770          2861  2  DESC2[DSC$B_CLASS] = DSC$K_CLASS_S;
;  1771          2862  2  DESC2[DSC$A_POINTER] = BUFFER[VAR_BODY];
;  1772          2863  2
;  1773          2864  2
;  1774          2865  2  ! Get the command.
;  1775          2866  2  !
;  1776          2867  2  CLI$GET_VALUE(SD_$LINE, DESC);
```

```
;  1777          2868  2 PRESERVE_STRING(BUFFER, QUAL[QUAL_CMD_DESC]);
;  1778          2869  2
;  1779          2870  2
;  1780          2871  2 ! Get the /ANALYZE qualifier.
;  1781          2872  2 !
;  1782          2873  2 QUAL[QUAL_ANAL] = CLI$PRESENT(SD_ANALYZE);
;  1783          2874  2
;  1784          2875  2
;  1785          2876  2 ! Get the /BACKUP qualifier.
;  1786          2877  2 !
;  1787          2878  2 QUAL[QUAL_BACK] = CLI$PRESENT(SD_BACKUP);
;  1788          2879  2
;  1789          2880  2
;  1790          2881  2 ! Get the /BEFORE qualifier.
;  1791          2882  2 !
;  1792          2883  2 IF CLI$GET_VALUE(SD_BEFORE, DESC)
;  1793          2884  2 THEN
;  1794          2885  3     BEGIN
;  1795          2886  3     QUAL[QUAL_BEFO] = TRUE;
;  1796          2887  3     IF CALL_TPARSE(BUFFER, BAK_STATES, BAK_KEYS)
;  1797          2888  3     THEN
;  1798          2889  3         QUAL[QUAL_BEFO_BACK] = TRUE
;  1799          2890  3     ELSE
;  1800          2891  4         BEGIN
;  1801          2892  4         DESC2[DSC$W_LENGTH] = .BUFFER[VAR_LENGTH];
;  1802          2893  4         IF NOT LIB$CVT_TIME(DESC2, QUAL[QUAL_BEFO_VALUE])
;  1803          2894  4             THEN SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_BEFORE);
;  1804          2895  3         END;
;  1805          2896  2     END;
;  1806          2897  2
;  1807          2898  2
;  1808          2899  2 ! Get the /BLOCK_SIZE qualifier.
;  1809          2900  2 !
;  1810          2901  2 IF CLI$GET_VALUE(SD_BLOCK_SIZE, DESC)
;  1811          2902  2 THEN
;  1812          2903  3     BEGIN
;  1813          2904  3     IF
;  1814          2905  4         BEGIN
;  1815          2906  4         IF NOT LIB$CVT_DTB(.BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], VALUE)
;  1816          2907  4         THEN
;  1817          2908  4             TRUE
;  1818          2909  4         ELSE
;  1819          2910  4             .VALUE - 2048 GTRU 65535 - 2048 ! 2048 <= N <= 65535
;  1820          2911  4         END
;  1821          2912  3     THEN
;  1822          2913  3         SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_BLOCK_SIZE);
;  1823          2914  3     QUAL[QUAL_BLOC] = TRUE;
;  1824          2915  3     QUAL[QUAL_BLOC_VALUE] = .VALUE;
;  1825          2916  2     END;
;  1826          2917  2
;  1827          2918  2
;  1828          2919  2 ! Get the /BRIEF qualifier.
;  1829          2920  2 !
;  1830          2921  2 CLI$PRESENT(SD_BRIEF);
;  1831          2922  2
;  1832          2923  2
;  1833          2924  2 ! Get the /BUFFER_COUNT qualifier.
```

```
;  1834          2925  2 !
;  1835          2926  2 IF CLI$GET_VALUE(SD_BUFFER_COUNT, DESC)
;  1836          2927  2 THEN
;  1837          2928  3     BEGIN
;  1838          2929  3     IF
;  1839          2930  4         BEGIN
;  1840          2931  4         IF NOT LIB$CVT_DTB(.BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], VALUE)
;  1841          2932  4         THEN
;  1842          2933  4             TRUE
;  1843          2934  4         ELSE
;  1844          2935  4             .VALUE - 2 GTRU 5 - 2 ! 2 <= N <= 5
;  1845          2936  4         END
;  1846          2937  3     THEN
;  1847          2938  3         SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_BUFFER_COUNT);
;  1848          2939  3     QUAL[QUAL_BUFF] = TRUE;
;  1849          2940  3     QUAL[QUAL_BUFF_VALUE] = .VALUE;
;  1850          2941  2     END;
;  1851          2942  2
;  1852          2943  2
;  1853          2944  2 ! Get the /COMMENT qualifier.
;  1854          2945  2 !
;  1855          2946  2 QUAL[QUAL_COMM] = CLI$GET_VALUE(SD_COMMENT, DESC);
;  1856          2947  2 PRESERVE_STRING(BUFFER, QUAL[QUAL_COMM_DESC]);
;  1857          2948  2
;  1858          2949  2
;  1859          2950  2 ! Get the /COMPARE qualifier.
;  1860          2951  2 !
;  1861          2952  2 QUAL[QUAL_COMP] = CLI$PRESENT(SD_COMPARE);
;  1862          2953  2
;  1863          2954  2
;  1864          2955  2 ! Get the /CONFIRM qualifier.
;  1865          2956  2 !
;  1866          2957  2 QUAL[QUAL_CONF] = CLI$PRESENT(SD_CONFIRM);
;  1867          2958  2
;  1868          2959  2
;  1869          2960  2 ! Get the /CRC qualifier.
;  1870          2961  2 !
;  1871          2962  2 QUAL[QUAL_CRC] = CLI$PRESENT(SD_CRC);
;  1872          2963  2
;  1873          2964  2
;  1874          2965  2 ! Get the /CREATED qualifier.
;  1875          2966  2 !
;  1876          2967  2 QUAL[QUAL_CREA] = CLI$PRESENT(SD_CREATED);
;  1877          2968  2
;  1878          2969  2
;  1879          2970  2 ! Get the /DELETE qualifier.
;  1880          2971  2 !
;  1881          2972  2 QUAL[QUAL_DELE] = CLI$PRESENT(SD_DELETE);
;  1882          2973  2
;  1883          2974  2
;  1884          2975  2 ! Get the /DENSITY qualifier.
;  1885          2976  2 !
;  1886          2977  2 IF CLI$GET_VALUE(SD_DENSITY, DESC)
;  1887          2978  2 THEN
;  1888          2979  3     BEGIN
;  1889          2980  3     IF
;  1890          2981  4         BEGIN
```

```
;  1891     2982  4          IF NOT LIB$CVT_DTB(.BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], VALUE)
;  1892     2983  4          THEN
;  1893     2984  4              TRUE
;  1894     2985  4          ELSE
;  1895     2986  4              SELECTONE .VALUE OF
;  1896     2987  4                  SET
;  1897     2988  4                  [ 800]: (QUAL[QUAL_DENS_VALUE] = MT$K_NRZI_800; FALSE);
;  1898     2989  4                  [1600]: (QUAL[QUAL_DENS_VALUE] = MT$K_PE_1600;  FALSE);
;  1899     2990  4                  [6250]: (QUAL[QUAL_DENS_VALUE] = MT$K_GCR_6250; FALSE);
;  1900     2991  4                  [OTHERWISE]: TRUE;
;  1901     2992  4                  TES
;  1902     2993  4          END
;  1903     2994  3      THEN
;  1904     2995  3          SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_DENSITY);
;  1905     2996  3      QUAL[QUAL_DENS] = TRUE;
;  1906     2997  2      END;
;  1907     2998  2
;  1908     2999  2
;  1909     3000  2  ! Get the /ENCRYPT qualifier.
;  1910     3001  2  !
;  1911     3002  2  QUAL[QUAL_SS_ENCRYP] = CLI$PRESENT(SD_ENCRYPT);
;  1912     3003  2  IF .QUAL[QUAL_SS_ENCRYP]
;  1913     3004  2  THEN
;  1914     3005  3  BEGIN
;  1915     3006  3      IF CRYPTO_INIT EQLA 0
;  1916     3007  3      THEN
;  1917     3008  3          SIGNAL(BACKUP$_ENCNOTSUP, 3, .BUFFER[VAR_LENGTH],
;  1918     3009  3                  BUFFER[VAR_BODY], SD_ENCRYPT);
;  1919     3010  3      !
;  1920     3011  3      ! Obtain the key name if present
;  1921     3012  3      !
;  1922     3013  3      IF CLI$GET_VALUE(SD_ENCRYPT_NAME,DESC)
;  1923     3014  3      THEN
;  1924     3015  3          PRESERVE_STRING(BUFFER,QUAL[QUAL_CRYP_USERKEY]);
;  1925     3016  3      !
;  1926     3017  3      ! Obtain the algorithm name.
;  1927     3018  3      !
;  1928     3019  3      IF CLI$GET_VALUE(SD_ENCRYPT_ALGO, DESC)
;  1929     3020  3      THEN
;  1930     3021  3          PRESERVE_STRING(BUFFER,QUAL[QUAL_CRYP_USERALG])
;  1931     3022  3      ELSE
;  1932     3023  3          !
;  1933     3024  3          ! Cram the default algorithm name string
;  1934     3025  3          !
;  1935     3026  4          BEGIN
;  1936     3027  4          BBLOCK [ QUAL[QUAL_CRYP_USERALG ] ,DSC$B_DTYPE] = DSC$K_DTYPE_T ;
;  1937     3028  4          BBLOCK [ QUAL[QUAL_CRYP_USERALG ] ,DSC$B_CLASS] = DSC$K_CLASS_S ;
;  1938     3029  4          BBLOCK [ QUAL[QUAL_CRYP_USERALG ] ,DSC$W_LENGTH] = 6 ;
;  1939     3030  4          BBLOCK [ QUAL[QUAL_CRYP_USERALG ] ,DSC$A_POINTER] = GET_VM(6) ;
;  1940     3031  4          CH$MOVE(6,PLIT('DESCBC'),
;  1941     3032  4                  .BBLOCK[ QUAL[QUAL_CRYP_USERALG], DSC$A_POINTER ])
;  1942     3033  3          END;
;  1943     3034  3      !
;  1944     3035  3      ! Obtain one or more key values if present
;  1945     3036  3      !
;  1946     3037  3      WHILE CLI$GET_VALUE(SD_ENCRYPT_VALU, DESC) DO
;  1947     3038  4          BEGIN
```

```
;  1948          3039  4          LOCAL
;  1949          3040  4              VALU: REF BBLOCK;
;  1950          3041  4          VALU = GET_ZERO_VM(QUAL_S_ENCV) ;
;  1951          3042  4          PRESERVE_STRING(BUFFER,VALU[QUAL_ENVA_DESC]) ;
;  1952          3043  4          VALU[QUAL_ENVA_LINK] = .QUAL[QUAL_CRYP_VLIST] ;
;  1953          3044  4          QUAL[QUAL_CRYP_VLIST] = .VALU ;
;  1954          3045  3          END;
;  1955          3046  2  END;
;  1956          3047  2  ! Get the /EXCLUDE qualifier.
;  1957          3048  2  !
;  1958          3049  2  LAST = 0;
;  1959          3050  2  WHILE CLI$GET_VALUE(SD_EXCLUDE, DESC) DO
;  1960          3051  3      BEGIN
;  1961          3052  3      LOCAL
;  1962          3053  3          EXCL:   REF BBLOCK;
;  1963          3054  3
;  1964          3055  3      QUAL[QUAL_EXCL] = TRUE;
;  1965          3056  3      EXCL = GET_ZERO_VM(QUAL_S_EXCL);
;  1966          3057  3      IF .LAST EQL 0
;  1967          3058  3          THEN QUAL[QUAL_EXCL_LIST] = .EXCL
;  1968          3059  3          ELSE LAST[QUAL_NEXT] = .EXCL;
;  1969          3060  3      LAST = .EXCL;
;  1970          3061  3      PARSE_EXC_SEL(BUFFER, SD_EXCLUDE, EXCL[QUAL_EXCL_DESC]);
;  1971          3062  2      END;
;  1972          3063  2
;  1973          3064  2
;  1974          3065  2  ! Get the /EXPIRED qualifier.
;  1975          3066  2  !
;  1976          3067  2  QUAL[QUAL_EXPI] = CLI$PRESENT(SD_EXPIRED);
;  1977          3068  2
;  1978          3069  2
;  1979          3070  2  ! Get the /FAST qualifier.
;  1980          3071  2  !
;  1981          3072  2  QUAL[QUAL_FAST] = CLI$PRESENT(SD_FAST);
;  1982          3073  2
;  1983          3074  2
;  1984          3075  2  ! Get the /FULL qualifier.
;  1985          3076  2  !
;  1986          3077  2  QUAL[QUAL_FULL] = CLI$PRESENT(SD_FULL);
;  1987          3078  2
;  1988          3079  2
;  1989          3080  2  ! Get the /GROUP_SIZE qualifier.
;  1990          3081  2  !
;  1991          3082  2  IF CLI$GET_VALUE(SD_GROUP_SIZE, DESC)
;  1992          3083  2  THEN
;  1993          3084  3      BEGIN
;  1994          3085  3      IF
;  1995          3086  4          BEGIN
;  1996          3087  4          IF NOT LIB$CVT_DTB(.BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], VALUE)
;  1997          3088  4          THEN
;  1998          3089  4              TRUE
;  1999          3090  4          ELSE
;  2000          3091  4              .VALUE GTRU 100
;  2001          3092  4          END
;  2002          3093  3      THEN
;  2003          3094  3          SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_GROUP_SIZE);
;  2004          3095  3      QUAL[QUAL_GROU] = TRUE;
```

```
;  2005          3096  3     QUAL[QUAL_GROU_VALUE] = .VALUE;
;  2006          3097  2     END;
;  2007          3098  2
;  2008          3099  2
;  2009          3100  2 ! Get the /IGNORE qualifier.
;  2010          3101  2 !
;  2011          3102  2 WHILE CLI$GET_VALUE(SD_IGNORE, DESC) DO
;  2012          3103  3     BEGIN
;  2013          3104  3     TPA_FLAGS = 0;
;  2014          3105  3     IF NOT CALL_TPARSE(BUFFER, IGN_STATES, IGN_KEYS)
;  2015          3106  3     THEN
;  2016          3107  3         SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_IGNORE);
;  2017          3108  3     IF .TPA_FLAGS[TPA_NOBACKUP] THEN QUAL[QUAL_IGNO_NOBA] = TRUE;
;  2018          3109  3     IF .TPA_FLAGS[TPA_INTERLOCK] THEN QUAL[QUAL_IGNO_INTE] = TRUE;
;  2019          3110  2     END;
;  2020          3111  2
;  2021          3112  2
;  2022          3113  2 ! Get the /IMAGE qualifier.
;  2023          3114  2 !
;  2024          3115  2 QUAL[QUAL_IMAG] = CLI$PRESENT(SD_IMAGE);
;  2025          3116  2
;  2026          3117  2
;  2027          3118  2 ! Get the /INCREMENTAL qualifier.
;  2028          3119  2 !
;  2029          3120  2 QUAL[QUAL_INCR] = CLI$PRESENT(SD_INCREMENTAL);
;  2030          3121  2
;  2031          3122  2
;  2032          3123  2 ! Get the /INITIALIZE qualifier. It defaults to TRUE if the output
;  2033          3124  2 ! is not a save set.
;  2034          3125  2 !
;  2035          3126  2 INIT_STATUS = CLI$PRESENT(SD_INITIALIZE);
;  2036          3127  2 QUAL[QUAL_INIT] = .INIT_STATUS;
;  2037          3128  2
;  2038          3129  2
;  2039          3130  2
;  2040          3131  2 ! Get the /INTERCHANGE qualifier.
;  2041          3132  2 !
;  2042          3133  2 QUAL[QUAL_INTE] = CLI$PRESENT(SD_INTERCHANGE);
;  2043          3134  2
;  2044          3135  2
;  2045          3136  2 ! Get the /JOURNAL qualifier.
;  2046          3137  2 !
;  2047          3138  2 IF CLI$PRESENT(SD_JOURNAL)
;  2048          3139  2 THEN
;  2049          3140  3     BEGIN
;  2050          3141  3     QUAL[QUAL_JOUR] = TRUE;
;  2051          3142  3     CLI$GET_VALUE(SD_JOURNAL, DESC);
;  2052          3143  3     QUAL[QUAL_JOUR_FC] = GET_FC(BUFFER);
;  2053          3144  2     END;
;  2054          3145  2
;  2055          3146  2
;  2056          3147  2 ! Get the /LABEL qualifier.
;  2057          3148  2 !
;  2058          3149  2 LAST = 0;
;  2059          3150  2 WHILE CLI$GET_VALUE(SD_LABEL, DESC) DO
;  2060          3151  3     BEGIN
;  2061          3152  3     LOCAL
```

```
; 2062          3153  3          LABE:   REF BBLOCK;
; 2063          3154  3
; 2064          3155  3      IF .BUFFER[VAR_LENGTH] GTRU 12
; 2065          3156  3      THEN
; 2066          3157  3          SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_LABEL);
; 2067          3158  3      QUAL[QUAL_LABE] = TRUE;
; 2068          3159  3      LABE = GET_ZERO_VM(QUAL_S_LABE);
; 2069          3160  3      IF .LAST EQL 0
; 2070          3161  3          THEN QUAL[QUAL_LABE_LIST] = .LABE
; 2071          3162  3          ELSE LAST[QUAL_NEXT] = .LABE;
; 2072          3163  3      LAST = .LABE;
; 2073          3164  3      CH$COPY(
; 2074          3165  3          .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY],
; 2075          3166  3          %C' ',
; 2076          3167  3          12, LABE[QUAL_LABE_VALUE]);
; 2077          3168  3      END;
; 2078          3169  2
; 2079          3170  2
; 2080          3171  2  ! Get the /LIST qualifier.
; 2081          3172  2  !
; 2082          3173  2  IF CLI$PRESENT(SD_LIST)
; 2083          3174  2  THEN
; 2084          3175  3      BEGIN
; 2085          3176  3      QUAL[QUAL_LIST] = TRUE;
; 2086          3177  3      CLI$GET_VALUE(SD_LIST, DESC);
; 2087          3178  3      QUAL[QUAL_LIST_FC] = GET_FC(BUFFER);
; 2088          3179  2      END;
; 2089          3180  2
; 2090          3181  2
; 2091          3182  2  ! Get the /LOG qualifier.
; 2092          3183  2  !
; 2093          3184  2  QUAL[QUAL_LOG] = CLI$PRESENT(SD_LOG);
; 2094          3185  2
; 2095          3186  2
; 2096          3187  2  ! Get the /MODIFIED qualifier.
; 2097          3188  2  !
; 2098          3189  2  QUAL[QUAL_MODI] = CLI$PRESENT(SD_MODIFIED);
; 2099          3190  2
; 2100          3191  2
; 2101          3192  2  ! Get the /NEW_VERSION qualifier.
; 2102          3193  2  !
; 2103          3194  2  QUAL[QUAL_NEWV] = CLI$PRESENT(SD_NEW_VERSION);
; 2104          3195  2
; 2105          3196  2
; 2106          3197  2  ! Get the /OVERLAY qualifier.
; 2107          3198  2  !
; 2108          3199  2  QUAL[QUAL_OVER] = CLI$PRESENT(SD_OVERLAY);
; 2109          3200  2
; 2110          3201  2
; 2111          3202  2  ! Get the /PHYSICAL qualifier.
; 2112          3203  2  !
; 2113          3204  2  QUAL[QUAL_PHYS] = CLI$PRESENT(SD_PHYSICAL);
; 2114          3205  2
; 2115          3206  2
; 2116          3207  2  ! Get the /RECORD qualifier.
; 2117          3208  2  !
; 2118          3209  2  QUAL[QUAL_RECO] = CLI$PRESENT(SD_RECORD);
```

```
;  2119          3210  2
;  2120          3211  2
;  2121          3212  2 ! Get the /REPLACE qualifier.
;  2122          3213  2 !
;  2123          3214  2 QUAL[QUAL_REPL] = CLI$PRESENT(SD_REPLACE);
;  2124          3215  2
;  2125          3216  2
;  2126          3217  2 ! Get the /REWIND qualifier.
;  2127          3218  2 !
;  2128          3219  2 QUAL[QUAL_REWI] = CLI$PRESENT(SD_REWIND);
;  2129          3220  2
;  2130          3221  2
;  2131          3222  2 ! Get the /SELECT qualifier.
;  2132          3223  2 !
;  2133          3224  2 LAST = 0;
;  2134          3225  2 WHILE CLI$GET_VALUE(SD_SELECT, DESC) DO
;  2135          3226  3     BEGIN
;  2136          3227  3     LOCAL
;  2137          3228  3         SELE:   REF BBLOCK;
;  2138          3229  3
;  2139          3230  3     QUAL[QUAL_SELE] = TRUE;
;  2140          3231  3     SELE = GET_ZERO_VM(QUAL_S_SELE);
;  2141          3232  3     IF .LAST EQL 0
;  2142          3233  3         THEN QUAL[QUAL_SELE_LIST] = .SELE
;  2143          3234  3         ELSE LAST[QUAL_NEXT] = .SELE;
;  2144          3235  3     LAST = .SELE;
;  2145          3236  3     PARSE_EXC_SEL(BUFFER, SD_SELECT, SELE[QUAL_SELE_DESC]);
;  2146          3237  2     END;
;  2147          3238  2
;  2148          3239  2
;  2149          3240  2 ! Get the /SINCE qualifier.
;  2150          3241  2 !
;  2151          3242  2 IF CLI$GET_VALUE(SD_SINCE, DESC)
;  2152          3243  2 THEN
;  2153          3244  3     BEGIN
;  2154          3245  3     QUAL[QUAL_SINC] = TRUE;
;  2155          3246  3     IF CALL_TPARSE(BUFFER, BAK_STATES, BAK_KEYS)
;  2156          3247  3     THEN
;  2157          3248  3         QUAL[QUAL_SINC_BACK] = TRUE
;  2158          3249  3     ELSE
;  2159          3250  4         BEGIN
;  2160          3251  4         DESC2[DSC$W_LENGTH] = .BUFFER[VAR_LENGTH];
;  2161          3252  4         IF NOT LIB$CVT_TIME(DESC2, QUAL[QUAL_SINC_VALUE])
;  2162          3253  4             THEN SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_SINCE);
;  2163          3254  3         END;
;  2164          3255  2     END;
;  2165          3256  2
;  2166          3257  2
;  2167          3258  2 ! Get the /TRUNCATE qualifier.
;  2168          3259  2 !
;  2169          3260  2 QUAL[QUAL_TRUN] = CLI$PRESENT(SD_TRUNCATE);
;  2170          3261  2
;  2171          3262  2
;  2172          3263  2 ! Get the /VERIFY qualifier.
;  2173          3264  2 !
;  2174          3265  2 QUAL[QUAL_VERI] = CLI$PRESENT(SD_VERIFY);
;  2175          3266  2
```

```
;  2176          3267  2
;  2177          3268  2 ! Get the /VOLUME qualifier.
;  2178          3269  2 !
;  2179          3270  2 IF CLI$GET_VALUE(SD_VOLUME, DESC)
;  2180          3271  2 THEN
;  2181          3272  3     BEGIN
;  2182          3273  3     IF
;  2183          3274  4         BEGIN
;  2184          3275  4         IF NOT LIB$CVT_DTB(.BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], VALUE)
;  2185          3276  4         THEN
;  2186          3277  4             TRUE
;  2187          3278  4         ELSE
;  2188          3279  4             .VALUE - 1 GTRU 255 - 1 ! 1 <= N <= 255
;  2189          3280  4         END
;  2190          3281  3     THEN
;  2191          3282  3         SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_VOLUME);
;  2192          3283  3     QUAL[QUAL_VOLU] = TRUE;
;  2193          3284  3     QUAL[QUAL_VOLU_VALUE] = .VALUE;
;  2194          3285  2     END;
;  2195          3286  2
;  2196          3287  2
;  2197          3288  2 ! Get the input parameter, and its local qualifiers.
;  2198          3289  2 !
;  2199          3290  2 LAST = 0;
;  2200          3291  2 WHILE CLI$GET_VALUE(SD_INPUTS, DESC) DO
;  2201          3292  3     BEGIN
;  2202          3293  3     LOCAL
;  2203          3294  3         INPU:   REF BBLOCK;
;  2204          3295  3
;  2205          3296  3
;  2206          3297  3     INPU = GET_ZERO_VM(QUAL_S_INPU);
;  2207          3298  3     IF .LAST EQL 0
;  2208          3299  3         THEN QUAL[QUAL_INPU_LIST] = .INPU
;  2209          3300  3         ELSE LAST[QUAL_NEXT] = .INPU;
;  2210          3301  3     LAST = .INPU;
;  2211          3302  3
;  2212          3303  3
;  2213          3304  3     INPU[QUAL_PARA_FC] = GET_FC(BUFFER);
;  2214          3305  3
;  2215          3306  3
;  2216          3307  3     ! Get the /OWNER_UIC qualifier.
;  2217          3308  3     !
;  2218          3309  3     IF CLI$PRESENT(SD_OWNER_UIC)
;  2219          3310  3     THEN
;  2220          3311  4         BEGIN
;  2221          3312  4         QUAL[QUAL_I_OWNE] = TRUE;
;  2222          3313  4         IF NOT CLI$GET_VALUE(SD_OWNER_UIC, DESC)
;  2223          3314  4         THEN
;  2224          3315  5             BEGIN
;  2225          3316  5             QUAL[QUAL_I_OWN_VALU] = .JPI_UIC;
;  2226          3317  5             QUAL[QUAL_I_OWN_WGRP] = FALSE;
;  2227          3318  5             QUAL[QUAL_I_OWN_WMEM] = FALSE;
;  2228          3319  5             END
;  2229          3320  4         ELSE
;  2230          3321  5             BEGIN
;  2231          3322  5             TPA_FLAGS = 0;
;  2232          3323  5             IF NOT CALL_TPARSE(BUFFER, UIC_STATES, UIC_KEYS)
```

```
; 2233          3324  5              THEN
; 2234          3325  5                  SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_OWNER_UIC);
; 2235          3326  5              IF .CONVERTED_UIC<16,16> EQL UIC$K_WILD_GROUP
; 2236          3327  5              THEN
; 2237          3328  6                  BEGIN
; 2238          3329  6                  TPA_FLAGS[TPA_WGROUP] = TRUE;
; 2239          3330  6                  CONVERTED_UIC<16,16> = 0;
; 2240          3331  5                  END;
; 2241          3332  5              IF .CONVERTED_UIC<0,16> EQL UIC$K_WILD_MEMBER
; 2242          3333  5              THEN
; 2243          3334  6                  BEGIN
; 2244          3335  6                  TPA_FLAGS[TPA_WMEMBER] = TRUE;
; 2245          3336  6                  CONVERTED_UIC<0,16> = 0;
; 2246          3337  5                  END;
; 2247          3338  5              QUAL[QUAL_I_OWN_VALU] = .CONVERTED_UIC;
; 2248          3339  5              QUAL[QUAL_I_OWN_WGRP] = .TPA_FLAGS[TPA_WGROUP];
; 2249          3340  5              QUAL[QUAL_I_OWN_WMEM] = .TPA_FLAGS[TPA_WMEMBER];
; 2250          3341  4              END;
; 2251          3342  3          END;
; 2252          3343  3
; 2253          3344  3
; 2254          3345  3      ! Get the /SAVE_SET qualifier.  The rightmost explicit specification
; 2255          3346  3      ! applies.
; 2256          3347  3      !
; 2257          3348  3      STATUS = CLI$PRESENT(SD_SAVE_SET);
; 2258          3349  3      IF .STATUS EQL CLI$_LOCPRES OR .STATUS EQL CLI$_LOCNEG
; 2259          3350  3          THEN QUAL[QUAL_I_SAVE] = .STATUS;
; 2260          3351  2      END;
; 2261          3352  2
; 2262          3353  2
; 2263          3354  2 ! Get the output parameter, and its local qualifiers.
; 2264          3355  2 !
; 2265          3356  2 LAST = 0;
; 2266          3357  2 WHILE CLI$GET_VALUE(SD_OUTPUTS, DESC) DO
; 2267          3358  3     BEGIN
; 2268          3359  3     LOCAL
; 2269          3360  3         OUTP:   REF BBLOCK;
; 2270          3361  3
; 2271          3362  3
; 2272          3363  3     OUTP = GET_ZERO_VM(QUAL_S_OUTP);
; 2273          3364  3     IF .LAST EQL 0
; 2274          3365  3         THEN QUAL[QUAL_OUTP_LIST] = .OUTP
; 2275          3366  3         ELSE LAST[QUAL_NEXT] = .OUTP;
; 2276          3367  3     LAST = .OUTP;
; 2277          3368  3
; 2278          3369  3
; 2279          3370  3     OUTP[QUAL_PARA_FC] = GET_FC(BUFFER);
; 2280          3371  3
; 2281          3372  3
; 2282          3373  3     ! Get the /OWNER_UIC qualifier.
; 2283          3374  3     !
; 2284          3375  3     IF CLI$PRESENT(SD_OWNER_UIC)
; 2285          3376  3     THEN
; 2286          3377  4         BEGIN
; 2287          3378  4         IF NOT CLI$GET_VALUE(SD_OWNER_UIC, DESC)
; 2288          3379  4         THEN
; 2289          3380  5             BEGIN
```

```
;  2290          3381  5              QUAL[QUAL_O_OWN_VALU] = 0;
;  2291          3382  5              QUAL[QUAL_O_OWN_DEFA] = FALSE;
;  2292          3383  5              QUAL[QUAL_O_OWN_ORIG] = TRUE;
;  2293          3384  5              QUAL[QUAL_O_OWN_PARE] = FALSE;
;  2294          3385  5              QUAL[QUAL_O_OWN_UIC] = FALSE;
;  2295          3386  5              END
;  2296          3387  4          ELSE
;  2297          3388  5              BEGIN
;  2298          3389  5              TPA_FLAGS = 0;
;  2299          3390  5              IF
;  2300          3391  6                  BEGIN
;  2301          3392  6                  IF NOT CALL_TPARSE(BUFFER, OWN_STATES, OWN_KEYS)
;  2302          3393  6                  THEN
;  2303          3394  6                      TRUE
;  2304          3395  6                  ELSE
;  2305          3396  6                      .CONVERTED_UIC<16,16> EQL UIC$K_WILD_GROUP OR
;  2306          3397  6                      .CONVERTED_UIC<0,16> EQL UIC$K_WILD_MEMBER
;  2307          3398  6                  END
;  2308          3399  5              THEN
;  2309          3400  5                  SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_OWNER_UIC);
;  2310          3401  5              QUAL[QUAL_O_OWN_VALU] = .CONVERTED_UIC;
;  2311          3402  5              QUAL[QUAL_O_OWN_DEFA] = .TPA_FLAGS[TPA_DEFAULT];
;  2312          3403  5              QUAL[QUAL_O_OWN_ORIG] = .TPA_FLAGS[TPA_ORIGINAL];
;  2313          3404  5              QUAL[QUAL_O_OWN_PARE] = .TPA_FLAGS[TPA_PARENT];
;  2314          3405  5              QUAL[QUAL_O_OWN_UIC] = .TPA_FLAGS[TPA_UIC];
;  2315          3406  4              END;
;  2316          3407  3          END;
;  2317          3408  3
;  2318          3409  3
;  2319          3410  3      ! Get the /PROTECTION qualifier.
;  2320          3411  3      !
;  2321          3412  3      IF CLI$PRESENT(SD_PROTECTION)
;  2322          3413  3      THEN
;  2323          3414  4          BEGIN
;  2324          3415  4          LOCAL
;  2325          3416  4              DEFAULT_PROT;
;  2326          3417  4
;  2327          3418  4          QUAL[QUAL_PROT] = TRUE;
;  2328          3419  4          TPA_PROTECTION = 0;
;  2329          3420  4          WHILE CLI$GET_VALUE(SD_PROTECTION, DESC) DO
;  2330          3421  5              BEGIN
;  2331          3422  5              IF NOT CALL_TPARSE(BUFFER, PRO_STATES, PRO_KEYS)
;  2332          3423  5              THEN
;  2333          3424  5                  SIGNAL(BACKUP$_INVQUAVAL, 3, .BUFFER[VAR_LENGTH], BUFFER[VAR_BODY], SD_PROTECTION);
;  2334          3425  4              END;
;  2335          3426  4
;  2336          3427  4
;  2337          3428  4          DEFAULT_PROT = %X'FF00';          ! S:RWED,O:RWED,G,W
;  2338          3429  4          IF NOT .COM_FLAGS[COM_STANDALONE] THEN SYS$SETDFPROT(0, DEFAULT_PROT);
;  2339          3430  4          QUAL[QUAL_PROT_VALUE] =
;  2340          3431  4              (.DEFAULT_PROT AND NOT .TPA_PROTECTION[1]) OR
;  2341          3432  4              (.TPA_PROTECTION[1] AND NOT .TPA_PROTECTION[0]);
;  2342          3433  3          END;
;  2343          3434  3
;  2344          3435  3
;  2345          3436  3      ! Get the /SAVE_SET qualifier.  The rightmost explicit specification
;  2346          3437  3      ! applies.
```

```
; 2347          3438  3      !
; 2348          3439  3      STATUS = CLI$PRESENT(SD_SAVE_SET);
; 2349          3440  3      IF .STATUS EQL CLI$_LOCPRES OR .STATUS EQL CLI$_LOCNEG
; 2350          3441  3          THEN QUAL[QUAL_O_SAVE] = .STATUS;
; 2351          3442  2      END;
; 2352          3443  2
; 2353          3444  2
; 2354          3445  2  ! Indicate completion of parsing.
; 2355          3446  2  !
; 2356          3447  2  IF .COM_FLAGS[COM_STANDALONE] THEN CLI$END_PARSE();
; 2357          3448  2
; 2358          3449  2
; 2359          3450  2  ! Check the input and output lists.
; 2360          3451  2  !
; 2361          3452  2  IF .QUAL[QUAL_INPU_LIST] EQL 0 AND NOT .QUAL[QUAL_JOUR]
; 2362          3453  2      THEN SIGNAL(BACKUP$_LISINPSAV);
; 2363          3454  2  CHECK( QUAL[QUAL_INPU_LIST], FALSE );
; 2364          3455  2  CHECK( QUAL[QUAL_OUTP_LIST], TRUE );
; 2365          3456  2
; 2366          3457  2
; 2367          3458  2  ! Get a verification file context area if required.
; 2368          3459  2  !
; 2369          3460  3  IF .QUAL[QUAL_VERI] AND (.QUAL[QUAL_OSAV] OR .QUAL[QUAL_ISAV])
; 2370          3461  2      THEN VERIFY_FAB = GET_FC(UPLIT WORD(0));
; 2371          3462  2
; 2372          3463  2
; 2373          3464  2  ! Postprocess the command qualifiers.
; 2374          3465  2  !
; 2375          3466  2  IF NOT .QUAL[QUAL_OSAV]
; 2376          3467  2  AND .INIT_STATUS EQL CLI$_ABSENT
; 2377          3468  2  THEN QUAL[QUAL_INIT] = TRUE;
; 2378          3469  2
; 2379          3470  2  IF .COM_FLAGS[COM_STANDALONE]
; 2380          3471  2  THEN
; 2381          3472  3      BEGIN
; 2382          3473  3      QUAL[QUAL_IMAG] = TRUE;
; 2383          3474  3      QUAL[QUAL_JOUR] = FALSE;
; 2384          3475  2      END;
; 2385          3476  2  IF .QUAL[QUAL_PHYS]
; 2386          3477  2  THEN
; 2387          3478  3      BEGIN
; 2388          3479  3      QUAL[QUAL_DELE] = FALSE;
; 2389          3480  3      QUAL[QUAL_IMAG] = FALSE;
; 2390          3481  3      QUAL[QUAL_INCR] = FALSE;
; 2391          3482  3      QUAL[QUAL_JOUR] = FALSE;
; 2392          3483  3      QUAL[QUAL_RECO] = FALSE;
; 2393          3484  2      END;
; 2394          3485  2  IF .QUAL[QUAL_IMAG]
; 2395          3486  2  THEN
; 2396          3487  3      BEGIN
; 2397          3488  3      QUAL[QUAL_FAST] = TRUE;
; 2398          3489  3      QUAL[QUAL_INCR] = FALSE;
; 2399          3490  3          END
; 2400          3491  2  ELSE
; 2401          3492  2      QUAL[QUAL_VOLU] = FALSE;
; 2402          3493  2  !
; 2403          3494  2  ! Don't support encryption of disk image copy
```

```
;  2404          3495  2
;  2405          3496  2 IF .QUAL[QUAL_OF11] AND .QUAL[QUAL_IF11] AND .QUAL[QUAL_SS_ENCRYP]
;  2406          3497  2 THEN
;  2407          3498  3     BEGIN
;  2408          3499  3     QUAL[QUAL_SS_ENCRYP] = FALSE;       ! No saveset encryption for image copy
;  2409          3500  3     SIGNAL(BACKUP$_ENCQUAIGN);          ! Tell the user what we've done
;  2410          3501  2     END;
;  2411          3502  2
;  2412          3503  2 IF .QUAL[QUAL_COMP] THEN QUAL[QUAL_VERI] = FALSE;
;  2413          3504  2 IF .QUAL[QUAL_OF11] AND .QUAL[QUAL_VOLU] AND NOT .QUAL[QUAL_INIT]
;  2414          3505  2 THEN
;  2415          3506  2     SIGNAL(BACKUP$_CONFQUAL);
;  2416          3507  2 IF NOT .QUAL[QUAL_BUFF] THEN QUAL[QUAL_BUFF_VALUE] = 3;
;  2417          3508  2 IF NOT .QUAL[QUAL_GROU] THEN QUAL[QUAL_GROU_VALUE] = 10;
;  2418          3509  2 IF .QUAL[QUAL_OSAV]
;  2419          3510  2 THEN
;  2420          3511  3     BEGIN
;  2421          3512  3     LOCAL
;  2422          3513  3         FAB:            REF BBLOCK,     ! Pointer to first output FAB
;  2423          3514  3         BLOC_VALUE;                     ! Temporary to compute actual blocksize
;  2424          3515  3
;  2425          3516  3     FAB = .BBLOCK[.QUAL[QUAL_OUTP_LIST], QUAL_PARA_FC];
;  2426          3517  3     BLOC_VALUE = .QUAL[QUAL_BLOC_VALUE];
;  2427          3518  3     IF NOT .QUAL[QUAL_BLOC]
;  2428          3519  3     THEN
;  2429          3520  4         BEGIN
;  2430          3521  4         IF .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
;  2431          3522  4             THEN BLOC_VALUE = 8192
;  2432          3523  4             ELSE BLOC_VALUE = 512*63;
;  2433          3524  3         END;
;  2434          3525  3     BLOC_VALUE = .BLOC_VALUE + 511 AND NOT 511;
;  2435          3526  3     IF .QUAL[QUAL_SS_FILE]
;  2436          3527  3     OR NOT .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD]
;  2437          3528  3     THEN
;  2438          3529  4         BEGIN
;  2439          3530  4         IF .BLOC_VALUE GTRU 63*512 THEN BLOC_VALUE = 63*512;
;  2440          3531  4         END
;  2441          3532  3     ELSE
;  2442          3533  4         BEGIN
;  2443          3534  4         IF .BLOC_VALUE GTRU 127*512 THEN BLOC_VALUE = 127*512;
;  2444          3535  3         END;
;  2445          3536  3
;  2446          3537  3     IF
;  2447          3538  3         .QUAL[QUAL_INTE] AND
;  2448          3539  3         .BBLOCK[FAB[FAB$L_DEV], DEV$V_SQD] AND
;  2449          3540  3         .BLOC_VALUE GTRU 8192
;  2450          3541  3     THEN
;  2451          3542  3         BLOC_VALUE = 8192;
;  2452          3543  3
;  2453          3544  3     QUAL[QUAL_BLOC_VALUE] = .BLOC_VALUE;
;  2454          3545  2     END;
;  2455          3546  1 END;


                                45  4E  49  4C  24  00A9A P.AAJ:  .ASCII  \$LINE\                              ;
                                                      00A9F        .BLKB   1
```

```
                                                00000005  00AA0 P.AAI:  .LONG   5                                ;
                                                00000000' 00AA4         .ADDRESS P.AAJ                          ;
                                    45  5A  59  4C  41  4E  41  00AA8 P.AAL:  .ASCII  \ANALYZE\                  ;
                                                            00AAF       .BLKB   1                               ;
                                                00000007  00AB0 P.AAK:  .LONG   7                               ;
                                                00000000' 00AB4         .ADDRESS P.AAL                          ;
                                        50  55  4B  43  41  42  00AB8 P.AAN:  .ASCII  \BACKUP\                  ;
                                                            00ABE       .BLKB   2                               ;
                                                00000006  00AC0 P.AAM:  .LONG   6                               ;
                                                00000000' 00AC4         .ADDRESS P.AAN                          ;
                                        45  52  4F  46  45  42  00AC8 P.AAP:  .ASCII  \BEFORE\                  ;
                                                            00ACE       .BLKB   2                               ;
                                                00000006  00AD0 P.AAO:  .LONG   6                               ;
                                                00000000' 00AD4         .ADDRESS P.AAP                          ;
                                45  5A  49  53  5F  4B  43  4F  4C  42  00AD8 P.AAR:  .ASCII  \BLOCK_SIZE\      ;
                                                            00AE2       .BLKB   2                               ;
                                                0000000A  00AE4 P.AAQ:  .LONG   10                              ;
                                                00000000' 00AE8         .ADDRESS P.AAR                          ;
                                            46  45  49  52  42  00AEC P.AAT:  .ASCII  \BRIEF\                   ;
                                                            00AF1       .BLKB   3                               ;
                                                00000005  00AF4 P.AAS:  .LONG   5                               ;
                                                00000000' 00AF8         .ADDRESS P.AAT                          ;
                        54  4E  55  4F  43  5F  52  45  46  46  55  42  00AFC P.AAV:  .ASCII  \BUFFER_COUNT\      ;
                                                0000000C  00B08 P.AAU:  .LONG   12                              ;
                                                00000000' 00B0C         .ADDRESS P.AAV                          ;
                                    54  4E  45  4D  4D  4F  43  00B10 P.AAX:  .ASCII  \COMMENT\                  ;
                                                            00B17       .BLKB   1                               ;
                                                00000007  00B18 P.AAW:  .LONG   7                               ;
                                                00000000' 00B1C         .ADDRESS P.AAX                          ;
                    45  52  41  50  4D  4F  43  00B20 P.AAZ:  .ASCII  \COMPARE\                                  ;
                                                            00B27       .BLKB   1                               ;
                                                00000007  00B28 P.AAY:  .LONG   7                               ;
                                                00000000' 00B2C         .ADDRESS P.AAZ                          ;
                    4D  52  49  46  4E  4F  43  00B30 P.ABB:  .ASCII  \CONFIRM\                                  ;
                                                            00B37       .BLKB   1                               ;
                                                00000007  00B38 P.ABA:  .LONG   7                               ;
                                                00000000' 00B3C         .ADDRESS P.ABB                          ;
                                            43  52  43  00B40 P.ABD:  .ASCII  \CRC\                             ;
                                                            00B43       .BLKB   1                               ;
                                                00000003  00B44 P.ABC:  .LONG   3                               ;
                                                00000000' 00B48         .ADDRESS P.ABD                          ;
                    44  45  54  41  45  52  43  00B4C P.ABF:  .ASCII  \CREATED\                                  ;
                                                            00B53       .BLKB   1                               ;
                                                00000007  00B54 P.ABE:  .LONG   7                               ;
                                                00000000' 00B58         .ADDRESS P.ABF                          ;
                        45  54  45  4C  45  44  00B5C P.ABH:  .ASCII  \DELETE\                                   ;
                                                            00B62       .BLKB   2                               ;
                                                00000006  00B64 P.ABG:  .LONG   6                               ;
                                                00000000' 00B68         .ADDRESS P.ABH                          ;
                    59  54  49  53  4E  45  44  00B6C P.ABJ:  .ASCII  \DENSITY\                                  ;
                                                            00B73       .BLKB   1                               ;
                                                00000007  00B74 P.ABI:  .LONG   7                               ;
                                                00000000' 00B78         .ADDRESS P.ABJ                          ;
                    54  50  59  52  43  4E  45  00B7C P.ABL:  .ASCII  \ENCRYPT\                                  ;
                                                            00B83       .BLKB   1                               ;
                                                00000007  00B84 P.ABK:  .LONG   7                               ;
                                                00000000' 00B88         .ADDRESS P.ABL                          ;
```

```
                                45  44  55  4C  43  58  45  00B8C P.ABN:   .ASCII  \EXCLUDE\                ;
                                                            00B93         .BLKB   1
                                                00000007    00B94 P.ABM:  .LONG   7                       ;
                                                00000000'   00B98         .ADDRESS P.ABN                  ;
                                44  45  52  49  50  58  45  00B9C P.ABP:   .ASCII  \EXPIRED\              ;
                                                            00BA3         .BLKB   1
                                                00000007    00BA4 P.ABO:  .LONG   7                       ;
                                                00000000'   00BA8         .ADDRESS P.ABP                  ;
                                            54  53  41  46  00BAC P.ABR:   .ASCII  \FAST\                 ;
                                                00000004    00BB0 P.ABQ:  .LONG   4                       ;
                                                00000000'   00BB4         .ADDRESS P.ABR                  ;
                                            4C  4C  55  46  00BB8 P.ABT:   .ASCII  \FULL\                 ;
                                                00000004    00BBC P.ABS:  .LONG   4                       ;
                                                00000000'   00BC0         .ADDRESS P.ABT                  ;
                        45  5A  49  53  5F  50  55  4F  52  47  00BC4 P.ABV:   .ASCII  \GROUP_SIZE\          ;
                                                            00BCE         .BLKB   2
                                                0000000A    00BD0 P.ABU:  .LONG   10                      ;
                                                00000000'   00BD4         .ADDRESS P.ABV                  ;
                                    45  52  4F  4E  47  49  00BD8 P.ABX:   .ASCII  \IGNORE\               ;
                                                            00BDE         .BLKB   2
                                                00000006    00BE0 P.ABW:  .LONG   6                       ;
                                                00000000'   00BE4         .ADDRESS P.ABX                  ;
                                        45  47  41  4D  49  00BE8 P.ABZ:   .ASCII  \IMAGE\                ;
                                                            00BED         .BLKB   3
                                                00000005    00BF0 P.ABY:  .LONG   5                       ;
                                                00000000'   00BF4         .ADDRESS P.ABZ                  ;
                4C  41  54  4E  45  4D  45  52  43  4E  49  00BF8 P.ACB:   .ASCII  \INCREMENTAL\           ;
                                                            00C03         .BLKB   1
                                                0000000B    00C04 P.ACA:  .LONG   11                      ;
                                                00000000'   00C08         .ADDRESS P.ACB                  ;
                        45  5A  49  4C  41  49  54  49  4E  49  00C0C P.ACD:   .ASCII  \INITIALIZE\         ;
                                                            00C16         .BLKB   2
                                                0000000A    00C18 P.ACC:  .LONG   10                      ;
                                                00000000'   00C1C         .ADDRESS P.ACD                  ;
                                    53  54  55  50  4E  49  00C20 P.ACF:   .ASCII  \INPUTS\               ;
                                                            00C26         .BLKB   2
                                                00000006    00C28 P.ACE:  .LONG   6                       ;
                                                00000000'   00C2C         .ADDRESS P.ACF                  ;
        45  47  4E  41  48  43  52  45  54  4E  49  00C30 P.ACH:   .ASCII  \INTERCHANGE\                   ;
                                                            00C3B         .BLKB   1
                                                0000000B    00C3C P.ACG:  .LONG   11                      ;
                                                00000000'   00C40         .ADDRESS P.ACH                  ;
                                4C  41  4E  52  55  4F  4A  00C44 P.ACJ:   .ASCII  \JOURNAL\               ;
                                                            00C4B         .BLKB   1
                                                00000007    00C4C P.ACI:  .LONG   7                       ;
                                                00000000'   00C50         .ADDRESS P.ACJ                  ;
                                        4C  45  42  41  4C  00C54 P.ACL:   .ASCII  \LABEL\                ;
                                                            00C59         .BLKB   3
                                                00000005    00C5C P.ACK:  .LONG   5                       ;
                                                00000000'   00C60         .ADDRESS P.ACL                  ;
                                            54  53  49  4C  00C64 P.ACN:   .ASCII  \LIST\                 ;
                                                00000004    00C68 P.ACM:  .LONG   4                       ;
                                                00000000'   00C6C         .ADDRESS P.ACN                  ;
                                                47  4F  4C  00C70 P.ACP:   .ASCII  \LOG\                  ;
                                                            00C73         .BLKB   1
                                                00000003    00C74 P.ACO:  .LONG   3                       ;
                                                00000000'   00C78         .ADDRESS P.ACP                  ;
```

```
            44  45  49  46  49  44  4F  4D  00C7C P.ACR:   .ASCII  \MODIFIED\
                                  00000008  00C84 P.ACQ:   .LONG   8
                                  00000000' 00C88          .ADDRESS P.ACR
    4E  4F  49  53  52  45  56  5F  57  45  4E  00C8C P.ACT:   .ASCII  \NEW_VERSION\
                                            00C97          .BLKB   1
                                  0000000B  00C98 P.ACS:   .LONG   11
                                  00000000' 00C9C          .ADDRESS P.ACT
                53  54  55  50  54  55  4F  00CA0 P.ACV:   .ASCII  \OUTPUTS\
                                            00CA7          .BLKB   1
                                  00000007  00CA8 P.ACU:   .LONG   7
                                  00000000' 00CAC          .ADDRESS P.ACV
                59  41  4C  52  45  56  4F  00CB0 P.ACX:   .ASCII  \OVERLAY\
                                            00CB7          .BLKB   1
                                  00000007  00CB8 P.ACW:   .LONG   7
                                  00000000' 00CBC          .ADDRESS P.ACX
        43  49  55  5F  52  45  4E  57  4F  00CC0 P.ACZ:   .ASCII  \OWNER_UIC\
                                            00CC9          .BLKB   3
                                  00000009  00CCC P.ACY:   .LONG   9
                                  00000000' 00CD0          .ADDRESS P.ACZ
            4C  41  43  49  53  59  48  50  00CD4 P.ADB:   .ASCII  \PHYSICAL\
                                  00000008  00CDC P.ADA:   .LONG   8
                                  00000000' 00CE0          .ADDRESS P.ADB
      4E  4F  49  54  43  45  54  4F  52  50  00CE4 P.ADD:   .ASCII  \PROTECTION\
                                            00CEE          .BLKB   2
                                  0000000A  00CF0 P.ADC:   .LONG   10
                                  00000000' 00CF4          .ADDRESS P.ADD
                    44  52  4F  43  45  52  00CF8 P.ADF:   .ASCII  \RECORD\
                                            00CFE          .BLKB   2
                                  00000006  00D00 P.ADE:   .LONG   6
                                  00000000' 00D04          .ADDRESS P.ADF
                45  43  41  4C  50  45  52  00D08 P.ADH:   .ASCII  \REPLACE\
                                            00D0F          .BLKB   1
                                  00000007  00D10 P.ADG:   .LONG   7
                                  00000000' 00D14          .ADDRESS P.ADH
                    44  4E  49  57  45  52  00D18 P.ADJ:   .ASCII  \REWIND\
                                            00D1E          .BLKB   2
                                  00000006  00D20 P.ADI:   .LONG   6
                                  00000000' 00D24          .ADDRESS P.ADJ
            54  45  53  5F  45  56  41  53  00D28 P.ADL:   .ASCII  \SAVE_SET\
                                  00000008  00D30 P.ADK:   .LONG   8
                                  00000000' 00D34          .ADDRESS P.ADL
                    54  43  45  4C  45  53  00D38 P.ADN:   .ASCII  \SELECT\
                                            00D3E          .BLKB   2
                                  00000006  00D40 P.ADM:   .LONG   6
                                  00000000' 00D44          .ADDRESS P.ADN
                        45  43  4E  49  53  00D48 P.ADP:   .ASCII  \SINCE\
                                            00D4D          .BLKB   3
                                  00000005  00D50 P.ADO:   .LONG   5
                                  00000000' 00D54          .ADDRESS P.ADP
            45  54  41  43  4E  55  52  54  00D58 P.ADR:   .ASCII  \TRUNCATE\
                                  00000008  00D60 P.ADQ:   .LONG   8
                                  00000000' 00D64          .ADDRESS P.ADR
                    59  46  49  52  45  56  00D68 P.ADT:   .ASCII  \VERIFY\
                                            00D6E          .BLKB   2
                                  00000006  00D70 P.ADS:   .LONG   6
                                  00000000' 00D74          .ADDRESS P.ADT
                    45  4D  55  4C  4F  56  00D78 P.ADV:   .ASCII  \VOLUME\
```

```
                                                 00D7E          .BLKB   2                       :
                                       00000006  00D80 P.ADU:   .LONG   6                       :
                                      00000000'  00D84          .ADDRESS P.ADV                  :
          44  4E  41  4D  4D  4F  43  24  53  59  53  00D88 P.ADX:   .ASCII  \SYS$COMMAND\       :
                                                 00D93          .BLKB   1
                                       0000000B  00D94 P.ADW:   .LONG   11                      :
                                      00000000'  00D98          .ADDRESS P.ADX                  :
                                     0004 0004   00D9C P.ADY:   .WORD   4, 4                    :
                                      00000000'  00DA0          .ADDRESS DEVCLASS               :
                             00000000  00000000  00DA4          .LONG   0, 0                    :
                                     0304 0004   00DAC P.ADZ:   .WORD   4, 772                  :
                                      00000000'  00DB0          .ADDRESS JPI_UIC                :
                                       00000000  00DB4          .LONG   0                       :
                                     0202 000C   00DB8          .WORD   12, 514                 :
                                      00000000'  00DBC          .ADDRESS JPI_USERNAME           :
                                       00000000  00DC0          .LONG   0                       :
                                     0400 0008   00DC4          .WORD   8, 1024                 :
                                      00000000'  00DC8          .ADDRESS JPI_CURPRIV            :
                             00000000  00000000  00DCC          .LONG   0, 0                    :
                                     1000 0004   00DD4 P.AEA:   .WORD   4, 4096                 :
                                      00000000'  00DD8          .ADDRESS SYI_VERSION            :
                                       00000000  00DDC          .LONG   0                       :
                                     1001 0004   00DE0          .WORD   4, 4097                 :
                                      00000000'  00DE4          .ADDRESS SYI_SID                :
                             00000000  00000000  00DE8          .LONG   0, 0                    :
                      45  44  4F  4E  24  53  59  53  00DF0 P.AEC:   .ASCII  \SYS$NODE\          :
                                       00000008  00DF8 P.AEB:   .LONG   8                       :
                                      00000000'  00DFC          .ADDRESS P.AEC                  :
      45  4D  41  4E  2E  54  50  59  52  43  4E  45  00E00 P.AEE:   .ASCII  \ENCRYPT.NAME\      :
                                       0000000C  00E0C P.AED:   .LONG   12                      :
                                      00000000'  00E10          .ADDRESS P.AEE                  :
54  49  52  4F  47  4C  41  2E  54  50  59  52  43  4E  45  00E14 P.AEG:   .ASCII  \ENCRYPT.ALGORITHM\ :
                                         4D  48  00E23                                          :
                                                 00E25          .BLKB   3
                                       00000011  00E28 P.AEF:   .LONG   17                      :
                                      00000000'  00E2C          .ADDRESS P.AEG                  :
                                       00000002  00E30          .LONG   2                       :
                  00  00  43  42  43  53  45  44  00E34 P.AEH:   .ASCII  \DESCBC\<0><0>      :
  45  55  4C  41  56  2E  54  50  59  52  43  4E  45  00E3C P.AEJ:   .ASCII  \ENCRYPT.VALUE\     :
                                                 00E49          .BLKB   3
                                       0000000D  00E4C P.AEI:   .LONG   13                      :
                                      00000000'  00E50          .ADDRESS P.AEJ                  :
                                           0000  00E54 P.AEK:   .WORD   0                       :

                                                       SD_$LINE=               P.AAI
                                                       SD_ANALYZE=             P.AAK
                                                       SD_BACKUP=              P.AAM
                                                       SD_BEFORE=              P.AAO
                                                       SD_BLOCK_SIZE=          P.AAQ
                                                       SD_BRIEF=               P.AAS
                                                       SD_BUFFER_COUNT=        P.AAU
                                                       SD_COMMENT=             P.AAW
                                                       SD_COMPARE=             P.AAY
                                                       SD_CONFIRM=             P.ABA
                                                       SD_CRC=                 P.ABC
                                                       SD_CREATED=             P.ABE
                                                       SD_DELETE=              P.ABG
```

```
                                                SD_DENSITY=         P.ABI
                                                SD_ENCRYPT=         P.ABK
                                                SD_EXCLUDE=         P.ABM
                                                SD_EXPIRED=         P.ABO
                                                SD_FAST=            P.ABQ
                                                SD_FULL=            P.ABS
                                                SD_GROUP_SIZE=      P.ABU
                                                SD_IGNORE=          P.ABW
                                                SD_IMAGE=           P.ABY
                                                SD_INCREMENTAL=     P.ACA
                                                SD_INITIALIZE=      P.ACC
                                                SD_INPUTS=          P.ACE
                                                SD_INTERCHANGE=     P.ACG
                                                SD_JOURNAL=         P.ACI
                                                SD_LABEL=           P.ACK
                                                SD_LIST=            P.ACM
                                                SD_LOG=             P.ACO
                                                SD_MODIFIED=        P.ACQ
                                                SD_NEW_VERSION=     P.ACS
                                                SD_OUTPUTS=         P.ACU
                                                SD_OVERLAY=         P.ACW
                                                SD_OWNER_UIC=       P.ACY
                                                SD_PHYSICAL=        P.ADA
                                                SD_PROTECTION=      P.ADC
                                                SD_RECORD=          P.ADE
                                                SD_REPLACE=         P.ADG
                                                SD_REWIND=          P.ADI
                                                SD_SAVE_SET=        P.ADK
                                                SD_SELECT=          P.ADM
                                                SD_SINCE=           P.ADO
                                                SD_TRUNCATE=        P.ADQ
                                                SD_VERIFY=          P.ADS
                                                SD_VOLUME=          P.ADU
                                                        .EXTRN  SYS$GETDVI, SYS$WAITFR
                                                        .EXTRN  SYS$GETJPI, SYS$GETSYI
                                                        .EXTRN  SYS$GETTIM, SYS$TRNLOG

                                        0FFC 00000      .ENTRY  COMMAND, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 2698
                                                                R11                                              ;
                                5B 00000000G    00  9E 00002      MOVAB   CLI$PRESENT, R11                     ;
                                5A      F1C9    CF  9E 00009      MOVAB   PRESERVE_STRING, R10                 ;
                                59 00000000'    EF  9E 0000E      MOVAB   QUAL+8, R9                           ;
                                5E      FBE4    CE  9E 00015      MOVAB   -1052(SP), SP                        ;
37          72          A9                  01  E0 0001A      BBS     #1, COM_FLAGS, 1$                    ; 2806
                                                7E  7C 0001F      CLRQ    -(SP)                                ; 2813
                                                7E  7C 00021      CLRQ    -(SP)                                ;
                                        FF1F    CF  9F 00023      PUSHAB  P.ADY                                ;
                                        FF13    CF  9F 00027      PUSHAB  P.ADW                                ;
                                                7E  7C 0002B      CLRQ    -(SP)                                ;
    00000000G   00                          08  FB 0002D      CALLS   #8, SYS$GETDVI                       ;
                23                          50  E9 00034      BLBC    R0, 2$                               ;
                                                7E  D4 00037      CLRL    -(SP)                                ; 2816
    00000000G   00                          01  FB 00039      CALLS   #1, SYS$WAITFR                       ;
    00000042    8F      0180                C9  D1 00040      CMPL    DEVCLASS, #66                        ; 2817
                                                0B  13 00049      BEQL    1$                                   ;
    000000A0    8F      0180                C9  D1 0004B      CMPL    DEVCLASS, #160                       ; 2818
                                                04  12 00054      BNEQ    2$                                   ;
```

```
                        73   A9              01  88 00056 1$:    BISB2   #1, COM_FLAGS+1                 ; 2824
                                             7E  7C 0005A 2$:    CLRQ    -(SP)                           ; 2832
                                             7E  D4 0005C        CLRL    -(SP)                           ;
                                    FEF4     CF  9F 0005E        PUSHAB  P.ADZ                           ;
                                             7E  7C 00062        CLRQ    -(SP)                           ;
                                             7E  D4 00064        CLRL    -(SP)                           ;
                00000000G    00              07  FB 00066        CALLS   #7, SYS$GETJPI                  ;
                                             7E  7C 0006D        CLRQ    -(SP)                           ; 2837
                                             7E  D4 0006F        CLRL    -(SP)                           ;
                                    FF09     CF  9F 00071        PUSHAB  P.AEA                           ;
                                             7E  7C 00075        CLRQ    -(SP)                           ;
                                             7E  D4 00077        CLRL    -(SP)                           ;
                00000000G    00              07  FB 00079        CALLS   #7, SYS$GETSYI                  ;
                                    FEDC     C9  9F 00080        PUSHAB  JPI_DATE                        ; 2838
                00000000G    00              01  FB 00084        CALLS   #1, SYS$GETTIM                  ;
                      10     AE       0400    8F  3C 00088        MOVZWL  #1024, DESC                     ; 2839
                      14     AE         1A    AE  9E 00091        MOVAB   BUFFER+2, DESC+4                ; 2842
                                             7E  7C 00096        CLRQ    -(SP)                           ; 2846
                                             7E  D4 00098        CLRL    -(SP)                           ;
                                      1C     AE  9F 0009A        PUSHAB  DESC                            ;
                                      28     AE  9F 0009D        PUSHAB  BUFFER                          ;
                                    FEFE     CF  9F 000A0        PUSHAB  P.AEB                           ;
                00000000G    00              06  FB 000A4        CALLS   #6, SYS$TRNLOG                  ;
                             57              50  D0 000AB        MOVL    R0, STATUS                      ;
                             13              57  E9 000AE        BLBC    STATUS, 3$                      ; 2847
                00000629     8F              57  D1 000B1        CMPL    STATUS, #1577                   ;
                                             0A  13 000B8        BEQL    3$                              ;
                                    FEE4     C9  9F 000BA        PUSHAB  JPI_NODE_DESC                   ; 2848
                                      1C     AE  9F 000BE        PUSHAB  BUFFER                          ;
                             6A              02  FB 000C1        CALLS   #2, PRESERVE_STRING             ;
                      10     AE  0B0E0400    8F  D0 000C4 3$:    MOVL    #185467904, DESC                ; 2855
                      14     AE         18   AE  9E 000CC        MOVAB   BUFFER, DESC+4                  ; 2858
                      08     AE  010E0000    8F  D0 000D1        MOVL    #17694720, DESC2                ; 2859
                      0C     AE         1A   AE  9E 000D9        MOVAB   BUFFER+2, DESC2+4               ; 2862
                                      10     AE  9F 000DE        PUSHAB  DESC                            ; 2867
                                    FB65     CF  9F 000E1        PUSHAB  SD_$LINE                        ;
                    00000000G    00              02  FB 000E5        CALLS   #2, CLI$GET_VALUE           ;
                                      20     A9  9F 000EC        PUSHAB  QUAL+40                         ; 2868
                                      1C     AE  9F 000EF        PUSHAB  BUFFER                          ;
                             6A              02  FB 000F2        CALLS   #2, PRESERVE_STRING             ;
                                    FB61     CF  9F 000F5        PUSHAB  SD_ANALYZE                      ; 2873
                             6B              01  FB 000F9        CALLS   #1, CLI$PRESENT                 ;
          69         01              00      50  F0 000FC        INSV    R0, #0, #1, QUAL+8              ;
                                    FB65     CF  9F 00101        PUSHAB  SD_BACKUP                       ; 2878
                             6B              01  FB 00105        CALLS   #1, CLI$PRESENT                 ;
          69         01              01      50  F0 00108        INSV    R0, #1, #1, QUAL+8              ;
                                      10     AE  9F 0010D        PUSHAB  DESC                            ; 2883
                                    FB66     CF  9F 00110        PUSHAB  SD_BEFORE                       ;
                    00000000G    00              02  FB 00114        CALLS   #2, CLI$GET_VALUE           ;
                             4D              50  E9 0011B        BLBC    R0, 5$                          ;
                             69              04  88 0011E        BISB2   #4, QUAL+8                      ; 2886
                                00000000V    EF  9F 00121        PUSHAB  BAK_KEYS                        ; 2887
                                00000000V    EF  9F 00127        PUSHAB  BAK_STATES                      ;
                                      20     AE  9F 0012D        PUSHAB  BUFFER                          ;
                      D4     AA              03  FB 00130        CALLS   #3, CALL_TPARSE                 ;
                             05              50  E9 00134        BLBC    R0, 4$                          ;
                             69              08  88 00137        BISB2   #8, QUAL+8                      ; 2889
```

```
                                    2F  11 0013A        BRB      5$                                 ;
             08   AE          18    AE  B0 0013C 4$:    MOVW     BUFFER, DESC2                      ; 2892
                              08    A9  9F 00141        PUSHAB   QUAL+16                            ; 2893
                              0C    AE  9F 00144        PUSHAB   DESC2                              ;
       00000000G  00                02  FB 00147        CALLS    #2, LIB$CVT_TIME                   ;
                  1A                50  E8 0014E        BLBS     R0, 5$                             ;
                            FB25    CF  9F 00151        PUSHAB   SD_BEFORE                          ; 2894
                              1E    AE  9F 00155        PUSHAB   BUFFER+2                           ;
                  7E          20    AE  3C 00158        MOVZWL   BUFFER, -(SP)                      ;
                                    03  DD 0015C        PUSHL    #3                                 ;
                       00000000G    8F  DD 0015E        PUSHL    #BACKUP$_INVQUAVAL                 ;
       00000000G  00                05  FB 00164        CALLS    #5, LIB$SIGNAL                     ;
                              10    AE  9F 0016B 5$:    PUSHAB   DESC                               ; 2901
                            FB1C    CF  9F 0016E        PUSHAB   SD_BLOCK_SIZE                      ;
       00000000G  00                02  FB 00172        CALLS    #2, CLI$GET_VALUE                  ;
                  45                50  E9 00179        BLBC     R0, 8$                             ;
                                    5E  DD 0017C        PUSHL    SP                                 ; 2906
                              1E    AE  9F 0017E        PUSHAB   BUFFER+2                           ;
                  7E          20    AE  3C 00181        MOVZWL   BUFFER, -(SP)                      ;
       00000000G  00                03  FB 00185        CALLS    #3, LIB$CVT_DTB                    ;
                  11                50  E9 0018C        BLBC     R0, 6$                             ;
    50            6E 00000800       8F  C3 0018F        SUBL3    #2048, VALUE, R0                   ; 2910
       0000F7FF   8F                50  D1 00197        CMPL     R0, #63487                         ;
                                    1A  1B 0019E        BLEQU    7$                                 ;
                            FAEA    CF  9F 001A0 6$:    PUSHAB   SD_BLOCK_SIZE                      ; 2913
                              1E    AE  9F 001A4        PUSHAB   BUFFER+2                           ;
                  7E          20    AE  3C 001A7        MOVZWL   BUFFER, -(SP)                      ;
                                    03  DD 001AB        PUSHL    #3                                 ;
                       00000000G    8F  DD 001AD        PUSHL    #BACKUP$_INVQUAVAL                 ;
       00000000G  00                05  FB 001B3        CALLS    #5, LIB$SIGNAL                     ;
                  69                10  88 001BA 7$:    BISB2    #16, QUAL+8                        ; 2914
             40   A9                6E  B0 001BD        MOVW     VALUE, QUAL+72                     ; 2915
                            FAD9    CF  9F 001C1 8$:    PUSHAB   SD_BRIEF                           ; 2921
                  6B                01  FB 001C5        CALLS    #1, CLI$PRESENT                    ;
                              10    AE  9F 001C8        PUSHAB   DESC                               ; 2926
                            FAE3    CF  9F 001CB        PUSHAB   SD_BUFFER_COUNT                    ;
       00000000G  00                02  FB 001CF        CALLS    #2, CLI$GET_VALUE                  ;
                  3D                50  E9 001D6        BLBC     R0, 11$                            ;
                                    5E  DD 001D9        PUSHL    SP                                 ; 2931
                              1E    AE  9F 001DB        PUSHAB   BUFFER+2                           ;
                  7E          20    AE  3C 001DE        MOVZWL   BUFFER, -(SP)                      ;
       00000000G  00                03  FB 001E2        CALLS    #3, LIB$CVT_DTB                    ;
                  09                50  E9 001E9        BLBC     R0, 9$                             ;
    50            6E                02  C3 001EC        SUBL3    #2, VALUE, R0                      ; 2935
                  03                50  D1 001F0        CMPL     R0, #3                             ;
                                    1A  1B 001F3        BLEQU    10$                                ;
                            FAB9    CF  9F 001F5 9$:    PUSHAB   SD_BUFFER_COUNT                    ; 2938
                              1E    AE  9F 001F9        PUSHAB   BUFFER+2                           ;
                  7E          20    AE  3C 001FC        MOVZWL   BUFFER, -(SP)                      ;
                                    03  DD 00200        PUSHL    #3                                 ;
                       00000000G    8F  DD 00202        PUSHL    #BACKUP$_INVQUAVAL                 ;
       00000000G  00                05  FB 00208        CALLS    #5, LIB$SIGNAL                     ;
                  69                20  88 0020F 10$:   BISB2    #32, QUAL+8                        ; 2939
             44   A9                6E  90 00212        MOVB     VALUE, QUAL+76                     ; 2940
                              10    AE  9F 00216 11$:   PUSHAB   DESC                               ; 2946
                            FAA5    CF  9F 00219        PUSHAB   SD_COMMENT                         ;
       00000000G  00                02  FB 0021D        CALLS    #2, CLI$GET_VALUE                  ;
```

```
                69              01              06              50  F0 00224        INSV    R0, #6, #1, QUAL+8
                                                        10      A9  9F 00229        PUSHAB  QUAL+24                         ; 2947
                                                        1C      AE  9F 0022C        PUSHAB  BUFFER
                                                                02  FB 0022F        CALLS   #2, PRESERVE_STRING
                                                    FA9C        CF  9F 00232        PUSHAB  SD_COMPARE                      ; 2952
                                                                01  FB 00236        CALLS   #1, CLI$PRESENT
                69              01              07              50  F0 00239        INSV    R0, #7, #1, QUAL+8
                                                    FAA0        CF  9F 0023E        PUSHAB  SD_CONFIRM                      ; 2957
                                                                01  FB 00242        CALLS   #1, CLI$PRESENT
            01  A9              01              00              50  F0 00245        INSV    R0, #0, #1, QUAL+9
                                                    FA9F        CF  9F 0024B        PUSHAB  SD_CRC                          ; 2962
                                                                01  FB 0024F        CALLS   #1, CLI$PRESENT
            01  A9              01              01              50  F0 00252        INSV    R0, #1, #1, QUAL+9
                                                    FAA2        CF  9F 00258        PUSHAB  SD_CREATED                      ; 2967
                                                                01  FB 0025C        CALLS   #1, CLI$PRESENT
            01  A9              01              02              50  F0 0025F        INSV    R0, #2, #1, QUAL+9
                                                    FAA5        CF  9F 00265        PUSHAB  SD_DELETE                       ; 2972
                                                                01  FB 00269        CALLS   #1, CLI$PRESENT
            06  A9              01              02              50  F0 0026C        INSV    R0, #2, #1, QUAL+14
                                                        10      AE  9F 00272        PUSHAB  DESC                            ; 2977
                                                    FAA5        CF  9F 00275        PUSHAB  SD_DENSITY
                                    00000000G       00              02  FB 00279    CALLS   #2, CLI$GET_VALUE
                                                    61              50  E9 00280    BLBC    R0, 16$
                                                                    5E  DD 00283    PUSHL   SP                               ; 2982
                                                            1E      AE  9F 00285    PUSHAB  BUFFER+2
                                                    7E      20      AE  3C 00288    MOVZWL  BUFFER, -(SP)
                                    00000000G       00              03  FB 0028C    CALLS   #3, LIB$CVT_DTB
                                                    30              50  E9 00293    BLBC    R0, 14$
                                                    50              6E  D0 00296    MOVL    VALUE, R0                       ; 2986
                                    00000320        8F              50  D1 00299    CMPL    R0, #800                        ; 2988
                                                                    06  12 002A0    BNEQ    12$
                                          45        A9              03  90 002A2    MOVB    #3, QUAL+77
                                                                    38  11 002A6    BRB     15$
                                    00000640        8F              50  D1 002A8 12$: CMPL    R0, #1600                       ; 2989
                                                                    06  12 002AF    BNEQ    13$
                                          45        A9              04  90 002B1    MOVB    #4, QUAL+77
                                                                    29  11 002B5    BRB     15$
                                    0000186A        8F              50  D1 002B7 13$: CMPL    R0, #6250                       ; 2990
                                                                    06  12 002BE    BNEQ    14$
                                          45        A9              05  90 002C0    MOVB    #5, QUAL+77
                                                                    1A  11 002C4    BRB     15$
                                                        FA54        CF  9F 002C6 14$: PUSHAB  SD_DENSITY                      ; 2995
                                                            1E      AE  9F 002CA    PUSHAB  BUFFER+2
                                                    7E      20      AE  3C 002CD    MOVZWL  BUFFER, -(SP)
                                                                    03  DD 002D1    PUSHL   #3
                                                00000000G           8F  DD 002D3    PUSHL   #BACKUP$_INVQUAVAL
                                    00000000G       00              05  FB 002D9    CALLS   #5, LIB$SIGNAL
                                              01    A9              08  88 002E0 15$: BISB2   #8, QUAL+9                      ; 2996
                                                        FA46        CF  9F 002E4 16$: PUSHAB  SD_ENCRYPT                      ; 3002
                                                    6B              01  FB 002E8    CALLS   #1, CLI$PRESENT
        06  A9                  01                  04              50  F0 002EB    INSV    R0, #4, #1, QUAL+14
                                03          06      A9              04  E0 002F1    BBS     #4, QUAL+14, 17$                ; 3003
                                                                00A5    31 002F6    BRW     22$
                                                    50  00000000G   00  9E 002F9 17$: MOVAB   CRYPTO_INIT, R0              ; 3006
                                                                    1A  12 00300    BNEQ    18$
                                                        FA28        CF  9F 00302    PUSHAB  SD_ENCRYPT                      ; 3009
                                                            1E      AE  9F 00306    PUSHAB  BUFFER+2
```

```
                 7E        20  AE  3C 00309       MOVZWL  BUFFER, -(SP)                        ;
                               03  DD 0030D       PUSHL   #3                                   ;
                    00000000G  8F  DD 0030F       PUSHL   #BACKUP$_ENCNOTSUP                   ;
         00000000G  00         05  FB 00315       CALLS   #5, LIB$SIGNAL                       ;
                           10  AE  9F 0031C 18$:  PUSHAB  DESC                                 ; 3013
                         FC93  CF  9F 0031F       PUSHAB  P.AED                                ;
         00000000G  00         02  FB 00323       CALLS   #2, CLI$GET_VALUE                    ;
                    09         50  E9 0032A       BLBC    R0, 19$                              ;
                           50  A9  9F 0032D       PUSHAB  QUAL+88                              ; 3015
                           1C  AE  9F 00330       PUSHAB  BUFFER                               ;
                    6A         02  FB 00333       CALLS   #2, PRESERVE_STRING                  ;
                           10  AE  9F 00336 19$:  PUSHAB  DESC                                 ; 3019
                         FC95  CF  9F 00339       PUSHAB  P.AEF                                ;
         00000000G  0C         02  FB 0033D       CALLS   #2, CLI$GET_VALUE                    ;
                    0B         50  E9 00344       BLBC    R0, 20$                              ;
                           58  A9  9F 00347       PUSHAB  QUAL+96                              ; 3021
                           1C  AE  9F 0034A       PUSHAB  BUFFER                               ;
                    6A         02  FB 0034D       CALLS   #2, PRESERVE_STRING                  ;
                               1C  11 00350       BRB     21$                                  ;
                58  A9 010E0006  8F  D0 00352 20$:  MOVL    #17694726, QUAL+96                   ; 3029
                               06  DD 0035A       PUSHL   #6                                   ; 3030
         00000000G  00         01  FB 0035C       CALLS   #1, GET_VM                           ;
                5C  A9         50  D0 00363       MOVL    R0, QUAL+100                         ;
    5C    B9  FC72  CF         06  28 00367       MOVC3   #6, P.AEH, @QUAL+100                 ; 3032
                           10  AE  9F 0036E 21$:  PUSHAB  DESC                                 ; 3037
                         FC81  CF  9F 00371       PUSHAB  P.AEI                                ;
         00000000G  00         02  FB 00375       CALLS   #2, CLI$GET_VALUE                    ;
                    1F         50  E9 0037C       BLBC    R0, 22$                              ;
                               0C  DD 0037F       PUSHL   #12                                  ; 3041
         00000000G  00         01  FB 00381       CALLS   #1, GET_ZERO_VM                      ;
                    52         50  D0 00388       MOVL    R0, VALU                             ;
                           04  A2  9F 0038B       PUSHAB  4(VALU)                              ; 3042
                           1C  AE  9F 0038E       PUSHAB  BUFFER                               ;
                    6A         02  FB 00391       CALLS   #2, PRESERVE_STRING                  ;
                    62     60  A9  D0 00394       MOVL    QUAL+104, (VALU)                     ; 3043
                60  A9         52  D0 00398       MOVL    VALU, QUAL+104                       ; 3044
                               D0  11 0039C       BRB     21$                                  ; 3037
                               56  D4 0039E 22$:  CLRL    LAST                                 ; 3049
                           10  AE  9F 003A0 23$:  PUSHAB  DESC                                 ; 3050
                         F997  CF  9F 003A3       PUSHAB  SD_EXCLUDE                           ;
         00000000G  00         02  FB 003A7       CALLS   #2, CLI$GET_VALUE                    ;
                    2E         50  E9 003AE       BLBC    R0, 26$                              ;
                01  A9         10  88 003B1       BISB2   #16, QUAL+9                          ; 3055
                               0C  DD 003B5       PUSHL   #12                                  ; 3056
         00000000G  00         01  FB 003B7       CALLS   #1, GET_ZERO_VM                      ;
                               56  D5 003BE       TSTL    LAST                                 ; 3057
                               06  12 003C0       BNEQ    24$                                  ;
                28  A9         50  D0 003C2       MOVL    EXCL, QUAL+48                        ; 3058
                               03  11 003C6       BRB     25$                                  ;
                    66         50  D0 003C8 24$:  MOVL    EXCL, (LAST)                         ; 3059
                    56         50  D0 003CB 25$:  MOVL    EXCL, LAST                           ; 3060
                           04  A0  9F 003CE       PUSHAB  4(EXCL)                              ; 3061
                         F969  CF  9F 003D1       PUSHAB  SD_EXCLUDE                           ;
                           20  AE  9F 003D5       PUSHAB  BUFFER                               ;
              0335  CA         03  FB 003D8       CALLS   #3, PARSE_EXC_SEL                    ;
                               C1  11 003DD       BRB     23$                                  ; 3050
                         F96B  CF  9F 003DF 26$:  PUSHAB  SD_EXPIRED                           ; 3067
```

```
                                6B                      01  FB 003E3           CALLS   #1, CLI$PRESENT                         ;
      01    A9                    01                    05  50 F0 003E6           INSV    R0, #5, #1, QUAL+9                      ;
                                                  F96A  CF  9F 003EC           PUSHAB  SD_FAST                                 ; 3072
                                6B                      01  FB 003F0           CALLS   #1, CLI$PRESENT                         ;
      01    A9                    01                    06  50 F0 003F3           INSV    R0, #6, #1, QUAL+9                      ;
                                                  F969  CF  9F 003F9           PUSHAB  SD_FULL                                 ; 3077
                                6B                      01  FB 003FD           CALLS   #1, CLI$PRESENT                         ;
      01    A9                    01                    07  50 F0 00400           INSV    R0, #7, #1, QUAL+9                      ;
                                                    10  AE  9F 00406           PUSHAB  DESC                                    ; 3082
                                                  F96D  CF  9F 00409           PUSHAB  SD_GROUP_SIZE                           ;
                                    00000000G   00          02  FB 0040D           CALLS   #2, CLI$GET_VALUE                       ;
                                3E                      50  E9 00414           BLBC    R0, 29$                                 ;
                                                        5E  DD 00417           PUSHL   SP                                      ; 3087
                                                    1E  AE  9F 00419           PUSHAB  BUFFER+2                                ;
                                7E                  20  AE  3C 0041C           MOVZWL  BUFFER, -(SP)                           ;
                                    00000000G   00          03  FB 00420           CALLS   #3, LIB$CVT_DTB                         ;
                                09                      50  E9 00427           BLBC    R0, 27$                                 ;
                                    00000064    8F          6E  D1 0042A           CMPL    VALUE, #100                             ; 3091
                                                        1A  1B 00431           BLEQU   28$                                     ;
                                                  F943  CF  9F 00433 27$:      PUSHAB  SD_GROUP_SIZE                           ; 3094
                                                    1E  AE  9F 00437           PUSHAB  BUFFER+2                                ;
                                7E                  20  AE  3C 0043A           MOVZWL  BUFFER, -(SP)                           ;
                                                        03  DD 0043E           PUSHL   #3                                      ;
                                                00000000G   8F  DD 00440           PUSHL   #BACKUP$_INVQUAVAL                      ;
                                    00000000G   00          05  FB 00446           CALLS   #5, LIB$SIGNAL                          ;
                                          02    A9                01  88 0044D 28$:      BISB2   #1, QUAL+10                     ; 3095
                                          46    A9                6E  90 00451           MOVB    VALUE, QUAL+78                  ; 3096
                                                    10  AE  9F 00455 29$:      PUSHAB  DESC                                    ; 3102
                                                  F92E  CF  9F 00458           PUSHAB  SD_IGNORE                               ;
                                    00000000G   00          02  FB 0045C           CALLS   #2, CLI$GET_VALUE                       ;
                                4A                      50  E9 00463           BLBC    R0, 32$                                 ;
                                                  0188  C9  D4 00466           CLRL    TPA_FLAGS                               ; 3104
                                                00000000V   EF  9F 0046A           PUSHAB  IGN_KEYS                                ; 3105
                                                00000000V   EF  9F 00470           PUSHAB  IGN_STATES                              ;
                                                    20  AE  9F 00476           PUSHAB  BUFFER                                  ;
                                          D4    AA                03  FB 00479           CALLS   #3, CALL_TPARSE                 ;
                                1A                      50  E8 0047D           BLBS    R0, 30$                                 ;
                                                  F906  CF  9F 00480           PUSHAB  SD_IGNORE                               ; 3107
                                                    1E  AE  9F 00484           PUSHAB  BUFFER+2                                ;
                                7E                  20  AE  3C 00487           MOVZWL  BUFFER, -(SP)                           ;
                                                        03  DD 0048B           PUSHL   #3                                      ;
                                                00000000G   8F  DD 0048D           PUSHL   #BACKUP$_INVQUAVAL                      ;
                                    00000000G   00          05  FB 00493           CALLS   #5, LIB$SIGNAL                          ;
                                04      0188    C9                06  E1 0049A 30$:      BBC     #6, TPA_FLAGS, 31$              ; 3108
                                          02    A9                02  88 004A0           BISB2   #2, QUAL+10                     ;
                                                  0188  C9  95 004A4 31$:      TSTB    TPA_FLAGS                               ; 3109
                                                        AB  18 004A8           BGEQ    29$                                     ;
                                          02    A9                04  88 004AA           BISB2   #4, QUAL+10                     ;
                                                        A5  11 004AE           BRB     29$                                     ; 3102
                                                  F8E6  CF  9F 004B0 32$:      PUSHAB  SD_IMAGE                                ; 3115
                                6B                      01  FB 004B4           CALLS   #1, CLI$PRESENT                         ;
      02    A9                    01                    03  50 F0 004B7           INSV    R0, #3, #1, QUAL+10                     ;
                                                  F8ED  CF  9F 004BD           PUSHAB  SD_INCREMENTAL                          ; 3120
                                6B                      01  FB 004C1           CALLS   #1, CLI$PRESENT                         ;
      02    A9                    01                    04  50 F0 004C4           INSV    R0, #4, #1, QUAL+10                     ;
                                                  F8F4  CF  9F 004CA           PUSHAB  SD_INITIALIZE                           ; 3126
                                6B                      01  FB 004CE           CALLS   #1, CLI$PRESENT                         ;
```

```
                                 58          50  D0 004D1          MOVL    R0, INIT_STATUS                              ;
     02   A9         01          05          58  F0 004D4          INSV    INIT_STATUS, #5, #1, QUAL+10                 ; 3127
                                       F908  CF  9F 004DA          PUSHAB  SD_INTERCHANGE                               ; 3133
                                 6B          01  FB 004DE          CALLS   #1, CLI$PRESENT                              ;
     06   A9         01          01          50  F0 004E1          INSV    R0, #1, #1, QUAL+14                          ;
                                       F90B  CF  9F 004E7          PUSHAB  SD_JOURNAL                                   ; 3138
                                 6B          01  FB 004EB          CALLS   #1, CLI$PRESENT                              ;
                                 1E          50  E9 004EE          BLBC    R0, 33$                                      ;
                         02      A9    40    8F  88 004F1          BISB2   #64, QUAL+10                                 ; 3141
                                       10    AE  9F 004F6          PUSHAB  DESC                                         ; 3142
                                       F8F9  CF  9F 004F9          PUSHAB  SD_JOURNAL                                   ;
                  00000000G      00          02  FB 004FD          CALLS   #2, CLI$GET_VALUE                            ;
                                       18    AE  9F 00504          PUSHAB  BUFFER                                       ; 3143
                         2C      AA          01  FB 00507          CALLS   #1, GET_FC                                   ;
                         2C      A9          50  D0 0050B          MOVL    R0, QUAL+52                                  ;
                                             56  D4 0050F 33$:     CLRL    LAST                                         ; 3149
                                       10    AE  9F 00511 34$:     PUSHAB  DESC                                         ; 3150
                                       F8EE  CF  9F 00514          PUSHAB  SD_LABEL                                     ;
                  00000000G      00          02  FB 00518          CALLS   #2, CLI$GET_VALUE                            ;
                                 49          50  E9 0051F          BLBC    R0, 38$                                      ;
                                 0C    18    AE  B1 00522          CMPW    BUFFER, #12                                  ; 3155
                                             1A  1B 00526          BLEQU   35$                                          ;
                                       F8DA  CF  9F 00528          PUSHAB  SD_LABEL                                     ; 3157
                                       1E    AE  9F 0052C          PUSHAB  BUFFER+2                                     ;
                                 7E    20    AE  3C 0052F          MOVZWL  BUFFER, -(SP)                                ;
                                             03  DD 00533          PUSHL   #3                                           ;
                                  00000000G  8F  DD 00535          PUSHL   #BACKUP$_INVQUAVAL                           ;
                  00000000G      00          05  FB 0053B          CALLS   #5, LIB$SIGNAL                               ;
                         02      A9    80    8F  88 00542 35$:     BISB2   #128, QUAL+10                                ; 3158
                                             10  DD 00547          PUSHL   #16                                          ; 3159
                  00000000G      00          01  FB 00549          CALLS   #1, GET_ZERO_VM                              ;
                                             56  D5 00550          TSTL    LAST                                         ; 3160
                                             06  12 00552          BNEQ    36$                                          ;
                         48      A9          50  D0 00554          MOVL    LABE, QUAL+80                                ; 3161
                                             03  11 00558          BRB     37$                                          ;
                                 66          50  D0 0055A 36$:     MOVL    LABE, (LAST)                                 ; 3162
                                 56          50  D0 0055D 37$:     MOVL    LABE, LAST                                   ; 3163
          0C         20    1A    AE    18    AE  2C 00560          MOVC5   BUFFER, BUFFER+2, #32, #12, 4(LABE)          ; 3167
                                       04    A0     00567                                                               ;
                                             A6  11 00569          BRB     34$                                          ; 3150
                                       F8A3  CF  9F 0056B 38$:     PUSHAB  SD_LIST                                      ; 3173
                                 6B          01  FB 0056F          CALLS   #1, CLI$PRESENT                              ;
                                 1D          50  E9 00572          BLBC    R0, 39$                                      ;
                         03      A9          01  88 00575          BISB2   #1, QUAL+11                                  ; 3176
                                       10    AE  9F 00579          PUSHAB  DESC                                         ; 3177
                                       F892  CF  9F 0057C          PUSHAB  SD_LIST                                      ;
                  00000000G      00          02  FB 00580          CALLS   #2, CLI$GET_VALUE                            ;
                                       18    AE  9F 00587          PUSHAB  BUFFER                                       ; 3178
                         2C      AA          01  FB 0058A          CALLS   #1, GET_FC                                   ;
                         30      A9          50  D0 0058E          MOVL    R0, QUAL+56                                  ;
                                       F888  CF  9F 00592 39$:     PUSHAB  SD_LOG                                       ; 3184
                                 6B          01  FB 00596          CALLS   #1, CLI$PRESENT                              ;
     03   A9         01          01          50  F0 00599          INSV    R0, #1, #1, QUAL+11                          ;
                                       F88B  CF  9F 0059F          PUSHAB  SD_MODIFIED                                  ; 3189
                                 6B          01  FB 005A3          CALLS   #1, CLI$PRESENT                              ;
     03   A9         01          02          50  F0 005A6          INSV    R0, #2, #1, QUAL+11                          ;
                                       F892  CF  9F 005AC          PUSHAB  SD_NEW_VERSION                               ; 3194
```

```
                                      6B      01 FB 005B0          CALLS   #1, CLI$PRESENT                 ;
  03   A9           01                03      50 F0 005B3          INSV    R0, #3, #1, QUAL+11             ;
                                            F8A5   CF 9F 005B9          PUSHAB  SD_OVERLAY                      ; 3199
                                      6B      01 FB 005BD          CALLS   #1, CLI$PRESENT                 ;
  03   A9           01                04      50 F0 005C0          INSV    R0, #4, #1, QUAL+11             ;
                                            F8BC   CF 9F 005C6          PUSHAB  SD_PHYSICAL                     ; 3204
                                      6B      01 FB 005CA          CALLS   #1, CLI$PRESENT                 ;
  04   A9           01                05      50 F0 005CD          INSV    R0, #5, #1, QUAL+12             ;
                                            F8D3   CF 9F 005D3          PUSHAB  SD_RECORD                       ; 3209
                                      6B      01 FB 005D7          CALLS   #1, CLI$PRESENT                 ;
  04   A9           01                06      50 F0 005DA          INSV    R0, #6, #1, QUAL+12             ;
                                            F8D6   CF 9F 005E0          PUSHAB  SD_REPLACE                      ; 3214
                                      6B      01 FB 005E4          CALLS   #1, CLI$PRESENT                 ;
  04   A9           01                07      50 F0 005E7          INSV    R0, #7, #1, QUAL+12             ;
                                            F8D9   CF 9F 005ED          PUSHAB  SD_REWIND                       ; 3219
                                      6B      01 FB 005F1          CALLS   #1, CLI$PRESENT                 ;
  05   A9           01                00      50 F0 005F4          INSV    R0, #0, #1, QUAL+13             ;
                                              56 D4 005FA          CLRL    LAST                            ; 3224
                                              10   AE 9F 005FC 40$:     PUSHAB  DESC                            ; 3225
                                            F8E7   CF 9F 005FF          PUSHAB  SD_SELECT                       ;
                      00000000G       00      02 FB 00603          CALLS   #2, CLI$GET_VALUE               ;
                                      2E      50 E9 0060A          BLBC    R0, 43$                         ;
                             05       A9      08 88 0060D          BISB2   #8, QUAL+13                     ; 3230
                                              0C DD 00611          PUSHL   #12                             ; 3231
                      00000000G       00      01 FB 00613          CALLS   #1, GET_ZERO_VM                 ;
                                              56 D5 0061A          TSTL    LAST                            ; 3232
                                              06 12 0061C          BNEQ    41$                             ;
                             3C       A9      50 D0 0061E          MOVL    SELE, QUAL+68                   ; 3233
                                              03 11 00622          BRB     42$                             ;
                                      66      50 D0 00624 41$:     MOVL    SELE, (LAST)                    ; 3234
                                      56      50 D0 00627 42$:     MOVL    SELE, LAST                      ; 3235
                                              04   A0 9F 0062A          PUSHAB  4(SELE)                         ; 3236
                                            F8B9   CF 9F 0062D          PUSHAB  SD_SELECT                       ;
                                              20   AE 9F 00631          PUSHAB  BUFFER                          ;
                           0335       CA      03 FB 00634          CALLS   #3, PARSE_EXC_SEL               ;
                                              C1 11 00639          BRB     40$                             ; 3225
                                              10   AE 9F 0063B 43$:     PUSHAB  DESC                            ; 3242
                                            F8B8   CF 9F 0063E          PUSHAB  SD_SINCE                        ;
                      00000000G       00      02 FB 00642          CALLS   #2, CLI$GET_VALUE               ;
                                      4F      50 E9 00649          BLBC    R0, 45$                         ;
                             05       A9      10 88 0064C          BISB2   #16, QUAL+13                    ; 3245
                                       00000000V   EF 9F 00650          PUSHAB  BAK_KEYS                        ; 3246
                                       00000000V   EF 9F 00656          PUSHAB  BAK_STATES                      ;
                                              20   AE 9F 0065C          PUSHAB  BUFFER                          ;
                             D4       AA      03 FB 0065F          CALLS   #3, CALL_TPARSE                 ;
                                      06      50 E9 00663          BLBC    R0, 44$                         ;
                             05       A9      20 88 00666          BISB2   #32, QUAL+13                    ; 3248
                                              2F 11 0066A          BRB     45$                             ;
                             08       AE      18   AE B0 0066C 44$:     MOVW    BUFFER, DESC2                   ; 3251
                                              18   A9 9F 00671          PUSHAB  QUAL+32                         ; 3252
                                              0C   AE 9F 00674          PUSHAB  DESC2                           ;
                      00000000G       00      02 FB 00677          CALLS   #2, LIB$CVT_TIME                ;
                                      1A      50 E8 0067E          BLBS    R0, 45$                         ;
                                            F875   CF 9F 00681          PUSHAB  SD_SINCE                        ; 3253
                                              1E   AE 9F 00685          PUSHAB  BUFFER+2                        ;
                                      7E      20   AE 3C 00688          MOVZWL  BUFFER, -(SP)                   ;
                                              03 DD 0068C          PUSHL   #3                              ;
```

```
                              00000000G  8F  DD 0068E       PUSHL   #BACKUP$_INVQUAVAL            ;
                    00000000G  00        05  FB 00694       CALLS   #5, LIB$SIGNAL                ;
                                  F86B   CF  9F 0069B 45$:  PUSHAB  SD_TRUNCATE                   ; 3260
                              6B         01  FB 0069F       CALLS   #1, CLI$PRESENT               ;
   05   A9            01      06         50  F0 006A2       INSV    R0, #6, #1, QUAL+13           ;
                                  F86E   CF  9F 006A8       PUSHAB  SD_VERIFY                     ; 3265
                              6B         01  FB 006AC       CALLS   #1, CLI$PRESENT               ;
   05   A9            01      07         50  F0 006AF       INSV    R0, #7, #1, QUAL+13           ;
                                    10   AE  9F 006B5       PUSHAB  DESC                          ; 3270
                                  F86E   CF  9F 006B8       PUSHAB  SD_VOLUME                     ;
                    00000000G  00        02  FB 006BC       CALLS   #2, CLI$GET_VALUE             ;
                              42         50  E9 006C3       BLBC    R0, 48$                       ;
                                         5E  DD 006C6       PUSHL   SP                            ; 3275
                                    1E   AE  9F 006C8       PUSHAB  BUFFER+2                      ;
                              7E    20   AE  3C 006CB       MOVZWL  BUFFER, -(SP)                 ;
                    00000000G  00        03  FB 006CF       CALLS   #3, LIB$CVT_DTB               ;
                              0D         50  E9 006D6       BLBC    R0, 46$                       ;
                      50      6E         01  C3 006D9       SUBL3   #1, VALUE, R0                 ; 3279
                    000000FE  8F         50  D1 006DD       CMPL    R0, #254                      ;
                                         1A  1B 006E4       BLEQU   47$                           ;
                                  F840   CF  9F 006E6 46$:  PUSHAB  SD_VOLUME                     ; 3282
                                    1E   AE  9F 006EA       PUSHAB  BUFFER+2                      ;
                              7E    20   AE  3C 006ED       MOVZWL  BUFFER, -(SP)                 ;
                                         03  DD 006F1       PUSHL   #3                            ;
                              00000000G  8F  DD 006F3       PUSHL   #BACKUP$_INVQUAVAL            ;
                    00000000G  00        05  FB 006F9       CALLS   #5, LIB$SIGNAL                ;
                           06  A9        01  88 00700 47$:  BISB2   #1, QUAL+14                   ; 3283
                           47  A9        6E  90 00704       MOVB    VALUE, QUAL+79                ; 3284
                                         56  D4 00708 48$:  CLRL    LAST                          ; 3290
                                    10   AE  9F 0070A 49$:  PUSHAB  DESC                          ; 3291
                                  F6C1   CF  9F 0070D       PUSHAB  SD_INPUTS                     ;
                    00000000G  00        02  FB 00711       CALLS   #2, CLI$GET_VALUE             ;
                              03         50  E8 00718       BLBS    R0, 50$                       ;
                                       00EB  31 0071B       BRW     61$                           ;
                                         28  DD 0071E 50$:  PUSHL   #40                           ; 3297
                    00000000G  00        01  FB 00720       CALLS   #1, GET_ZERO_VM               ;
                              52         50  D0 00727       MOVL    R0, INPU                      ;
                                         56  D5 0072A       TSTL    LAST                          ; 3298
                              06         12  0072C          BNEQ    51$                           ;
                           F8  A9        52  D0 0072E       MOVL    INPU, QUAL                    ; 3299
                                         03  11 00732       BRB     52$                           ;
                              66         52  D0 00734 51$:  MOVL    INPU, (LAST)                  ; 3300
                              56         52  D0 00737 52$:  MOVL    INPU, LAST                    ; 3301
                                    18   AE  9F 0073A       PUSHAB  BUFFER                        ; 3304
                           2C  AA        01  FB 0073D       CALLS   #1, GET_FC                    ;
                           04  A2        50  D0 00741       MOVL    R0, 4(INPU)                   ;
                                  F72D   CF  9F 00745       PUSHAB  SD_OWNER_UIC                  ; 3309
                              6B         01  FB 00749       CALLS   #1, CLI$PRESENT               ;
                              20         50  E9 0074C       BLBC    R0, 53$                       ;
                           03  A9        20  88 0074F       BISB2   #32, QUAL+11                  ; 3312
                                    10   AE  9F 00753       PUSHAB  DESC                          ; 3313
                                  F71C   CF  9F 00756       PUSHAB  SD_OWNER_UIC                  ;
                    00000000G  00        02  FB 0075A       CALLS   #2, CLI$GET_VALUE             ;
                              0D         50  E8 00761       BLBS    R0, 54$                       ;
                           34  A9  FECC  C9  D0 00764       MOVL    JPI_UIC, QUAL+60              ; 3316
                           03  A9    C0  8F  8A 0076A       BICB2   #192, QUAL+11                 ; 3318
                                         73  11 0076F 53$:  BRB     58$                           ; 3313
```

```
                                  0188  C9  D4 00771 54$:   CLRL    TPA_FLAGS                          ; 3322
                             00000000V  EF  9F 00775        PUSHAB  UIC_KEYS                           ; 3323
                             00000000V  EF  9F 0077B        PUSHAB  UIC_STATES                         ;
                                    20  AE  9F 00781        PUSHAB  BUFFER                             ;
                      D4    AA          03  FB 00784        CALLS   #3, CALL_TPARSE                    ;
                            1A          50  E8 00788        BLBS    R0, 55$                            ;
                                  F6E7  CF  9F 0078B        PUSHAB  SD_OWNER_UIC                       ; 3325
                                    1E  AE  9F 0078F        PUSHAB  BUFFER+2                           ;
                            7E      20  AE  3C 00792        MOVZWL  BUFFER, -(SP)                      ;
                                        03  DD 00796        PUSHL   #3                                 ;
                             00000000G  8F  DD 00798        PUSHL   #BACKUP$_INVQUAVAL                 ;
               00000000G    00          05  FB 0079E        CALLS   #5, LIB$SIGNAL                     ;
                    3FFF    8F    0182  C9  B1 007A5 55$:   CMPW    CONVERTED_UIC+2, #16383            ; 3326
                                        09  12 007AC        BNEQ    56$                                ;
                    0188    C9          01  88 007AE        BISB2   #1, TPA_FLAGS                      ; 3329
                                  0182  C9  B4 007B3        CLRW    CONVERTED_UIC+2                    ; 3330
                    FFFF    8F    0180  C9  B1 007B7 56$:   CMPW    CONVERTED_UIC, #65535              ; 3332
                                        09  12 007BE        BNEQ    57$                                ;
                    0188    C9          02  88 007C0        BISB2   #2, TPA_FLAGS                      ; 3335
                                  0180  C9  B4 007C5        CLRW    CONVERTED_UIC                      ; 3336
                      34    A9    0180  C9  D0 007C9 57$:   MOVL    CONVERTED_UIC, QUAL+60             ; 3338
  03   A9          01       06    0188  C9  F0 007CF        INSV    TPA_FLAGS, #6, #1, QUAL+11         ; 3339
       50   0188   C9       01          01  EF 007D7        EXTZV   #1, #1, TPA_FLAGS, R0              ; 3340
  03   A9          01       07          50  F0 007DE        INSV    R0, #7, #1, QUAL+11                ;
                                  F6F2  CF  9F 007E4 58$:   PUSHAB  SD_SAVE_SET                        ; 3348
                            6B          01  FB 007E8        CALLS   #1, CLI$PRESENT                    ;
                            57          50  D0 007EB        MOVL    R0, STATUS                         ;
               00000000G    8F          57  D1 007EE        CMPL    STATUS, #CLI$_LOCPRES              ; 3349
                            09          13 007F5            BEQL    59$                                ;
               00000000G    8F          57  D1 007F7        CMPL    STATUS, #CLI$_LOCNEG               ;
                            06          12 007FE            BNEQ    60$                                ;
  05   A9          01       01          57  F0 00800 59$:   INSV    STATUS, #1, #1, QUAL+13            ; 3350
                                      FF01  31 00806 60$:   BRW     49$                                ; 3291
                                        56  D4 00809 61$:   CLRL    LAST                               ; 3356
                                    10  AE  9F 0080B 62$:   PUSHAB  DESC                               ; 3357
                                  F640  CF  9F 0080E        PUSHAB  SD_OUTPUTS                         ;
               00000000G    00          02  FB 00812        CALLS   #2, CLI$GET_VALUE                  ;
                            03          50  E8 00819        BLBS    R0, 63$                            ;
                                      0181  31 0081C        BRW     77$                                ;
                                        20  DD 0081F 63$:   PUSHL   #32                                ; 3363
               00000000G    00          01  FB 00821        CALLS   #1, GET_ZERO_VM                    ;
                            52          50  D0 00828        MOVL    R0, OUTP                           ;
                                        56  D5 0082B        TSTL    LAST                               ; 3364
                                        06  12 0082D        BNEQ    64$                                ;
                      FC    A9          52  D0 0082F        MOVL    OUTP, QUAL+4                       ; 3365
                                        03  11 00833        BRB     65$                                ;
                            66          52  D0 00835 64$:   MOVL    OUTP, (LAST)                       ; 3366
                            56          52  D0 00838 65$:   MOVL    OUTP, LAST                         ; 3367
                                    18  AE  9F 0083B        PUSHAB  BUFFER                             ; 3370
                      2C    AA          01  FB 0083E        CALLS   #1, GET_FC                         ;
                      04    A2          50  D0 00842        MOVL    R0, 4(OUTP)                        ;
                                  F62C  CF  9F 00846        PUSHAB  SD_OWNER_UIC                       ; 3375
                            6B          01  FB 0084A        CALLS   #1, CLI$PRESENT                    ;
                            20          50  E9 0084D        BLBC    R0, 66$                            ;
                                    10  AE  9F 00850        PUSHAB  DESC                               ; 3378
                                  F61F  CF  9F 00853        PUSHAB  SD_OWNER_UIC                       ;
               00000000G    00          02  FB 00857        CALLS   #2, CLI$GET_VALUE                  ;
```

```
                              12           50  E8 0085E         BLBS    R0, 67$                              ;
                                      38   A9  D4 00861         CLRL    QUAL+64                              ; 3381
                         04   A9           02  8A 00864         BICB2   #2, QUAL+12                          ; 3382
                         04   A9           04  88 00868         BISB2   #4, QUAL+12                          ; 3383
                         04   A9           18  8A 0086C         BICB2   #24, QUAL+12                         ; 3385
                                         0080  31 00870 66$:    BRW     70$                                  ; 3378
                                   0188    C9  D4 00873 67$:    CLRL    TPA_FLAGS                            ; 3389
                              00000000V    EF  9F 00877         PUSHAB  OWN_KEYS                             ; 3392
                              00000000V    EF  9F 0087D         PUSHAB  OWN_ST  ES                           ;
                                      20   AE  9F 00883         PUSHAB  BUFFER                               ;
                         D4   AA           03  FB 00886         CALLS   #3, CALL_TPARSE                      ;
                              12           50  E9 0088A         BLBC    R0, 68$                              ;
                       3FFF   8F     0182  C9  B1 0088D         CMPW    CONVERTED_UIC+2, #16383              ; 3396
                                           09  13 00894         BEQL    68$                                  ;
                       FFFF   8F     0180  C9  B1 00896         CMPW    CONVERTED_UIC, #65535                ; 3397
                                           1A  12 0089D         BNEQ    69$                                  ;
                                     F5D3  CF  9F 0089F 68$:    PUSHAB  SD_OWNER_UIC                         ; 3400
                                       1E  AE  9F 008A3         PUSHAB  BUFFER+2                             ;
                              7E       20  AE  3C 008A6         MOVZWL  BUFFER, -(SP)                        ;
                                           03  DD 008AA         PUSHL   #3                                   ;
                                00000000G  8F  DD 008AC         PUSHL   #BACKUP$_INVQUAVAL                   ;
                    00000000G  00          05  FB 008B2         CALLS   #5, LIB$SIGNAL                       ;
                         38   A9     0180  C9  D0 008B9 69$:    MOVL    CONVERTED_UIC, QUAL+64               ; 3401
        50    0188   C9       01           02  EF 008BF         EXTZV   #2, #1, TPA_FLAGS, R0                ; 3402
   04   A9           01       01           50  F0 008C6         INSV    R0, #1, #1, QUAL+12                  ;
        50    0188   C9       01           03  EF 008CC         EXTZV   #3, #1, TPA_FLAGS, R0                ; 3403
   04   A9           01       02           50  F0 008D3         INSV    R0, #2, #1, QUAL+12                  ;
        50    0188   C9       01           04  EF 008D9         EXTZV   #4, #1, TPA_FLAGS, R0                ; 3404
   04   A9           01       03           50  F0 008E0         INSV    R0, #3, #1, QUAL+12                  ;
        50    0188   C9       01           05  EF 008E6         EXTZV   #5, #1, TPA_FLAGS, R0                ; 3405
   04   A9           01       04           50  F0 008ED         INSV    R0, #4, #1, QUAL+12                  ;
                                     F5A3  CF  9F 008F3 70$:    PUSHAB  SD_PROTECTION                        ; 3412
                              6B           01  FB 008F7         CALLS   #1, CLI$PRESENT                      ;
                              7E           50  E9 008FA         BLBC    R0, 74$                              ;
                         06   A9           08  88 008FD         BISB2   #8, QUAL+14                          ; 3418
                                     0180  C9  D4 00901         CLRL    TPA_PROTECTION                       ; 3419
                                       10  AE  9F 00905 71$:    PUSHAB  DESC                                 ; 3420
                                     F58E  CF  9F 00908         PUSHAB  SD_PROTECTION                        ;
                    00000000G  00          02  FB 0090C         CALLS   #2, CLI$GET_VALUE                    ;
                              32           50  E9 00913         BLBC    R0, 72$                              ;
                              00000000V    EF  9F 00916         PUSHAB  PRO_KEYS                             ; 3422
                              00000000V    EF  9F 0091C         PUSHAB  PRO_STATES                           ;
                                      20   AE  9F 00922         PUSHAB  BUFFER                               ;
                         D4   AA           03  FB 00925         CALLS   #3, CALL_TPARSE                      ;
                              D9           50  E8 00929         BLBS    R0, 71$                              ;
                                     F56A  CF  9F 0092C         PUSHAB  SD_PROTECTION                        ; 3424
                                       1E  AE  9F 00930         PUSHAB  BUFFER+2                             ;
                              7E       20  AE  3C 00933         MOVZWL  BUFFER, -(SP)                        ;
                                           03  DD 00937         PUSHL   #3                                   ;
                                00000000G  8F  DD 00939         PUSHL   #BACKUP$_INVQUAVAL                   ;
                    00000000G  00          05  FB 0093F         CALLS   #5, LIB$SIGNAL                       ;
                                           BD  11 00946         BRB     71$                                  ; 3420
                         04   AE     FF00  8F  3C 00948 72$:    MOVZWL  #65280, DEFAULT_PROT                 ; 3428
                    0C   72   A9           01  E0 0094E         BBS     #1, COM_FLAGS, 73$                   ; 3429
                                       04  AE  9F 00953         PUSHAB  DEFAULT_PROT                         ;
                                           7E  D4 00956         CLRL    -(SP)                                ;
                    00000000G  00          02  FB 00958         CALLS   #2, SYS$SETDFPROT                    ;
```

```
                        51      0182  C9  3C 0095F 73$:     MOVZWL  TPA_PROTECTION+2, R1                    ; 3431
            51      04  AE            51  CB 00964          BICL3   R1, DEFAULT_PROT, R1                    ;
                        50      0182  C9  3C 00969          MOVZWL  TPA_PROTECTION+2, R0                    ; 3432
                        52      0180  C9  3C 0096E          MOVZWL  TPA_PROTECTION, R2                      ;
                        50            52  CA 00973          BICL2   R2, R0                                  ;
       4C   A9          50            51  A9 00976          BISW3   R1, R0, QUAL+84                         ;
                              F55B    CF  9F 0097B 74$:     PUSHAB  SD_SAVE_SET                             ; 3439
                        6B            01  FB 0097F          CALLS   #1, CLI$PRESENT                         ;
                        57            50  D0 00982          MOVL    R0, STATUS                              ;
           00000000G    8F            57  D1 00985          CMPL    STATUS, #CLI$_LOCPRES                   ; 3440
                                      09  13 0098C          BEQL    75$                                     ;
           00000000G    8F            57  D1 0098E          CMPL    STATUS, #CLI$_LOCNEG                    ;
                                      06  12 00995          BNEQ    76$                                     ;
 05   A9    01          02            57  F0 00997 75$:     INSV    STATUS, #2, #1, QUAL+13                 ; 3441
                                    FE6B  31 0099D 76$:     BRW     62$                                     ; 3357
            07      72  A9            01  E1 009A0 77$:     BBC     #1, COM_FLAGS, 78$                      ; 3447
           00000000G    00            00  FB 009A5          CALLS   #0, CLI$END_PARSE                       ;
                                F8    A9  D5 009AC 78$:     TSTL    QUAL                                    ; 3452
                                      12  12 009AF          BNEQ    79$                                     ;
            0D      02  A9            06  E0 009B1          BBS     #6, QUAL+10, 79$                        ;
                         00000000G    8F  DD 009B6          PUSHL   #BACKUP$_LISINPSAV                      ; 3453
           00000000G    00            01  FB 009BC          CALLS   #1, LIB$SIGNAL                          ;
                                      7E  D4 009C3 79$:     CLRL    -(SP)                                   ; 3454
                                F8    A9  9F 009C5          PUSHAB  QUAL                                    ;
                  0729  CA            02  FB 009C8          CALLS   #2, CHECK                               ;
                                      01  DD 009CD          PUSHL   #1                                      ; 3455
                                FC    A9  9F 009CF          PUSHAB  QUAL+4                                  ;
                  0729  CA            02  FB 009D2          CALLS   #2, CHECK                               ;
                                05    A9  95 009D7          TSTB    QUAL+13                                 ; 3460
                                      17  18 009DA          BGEQ    81$                                     ;
                                07    A9  95 009DC          TSTB    QUAL+15                                 ;
                                      05  19 009DF          BLSS    80$                                     ;
            0D      07  A9            05  E1 009E1          BBC     #5, QUAL+15, 81$                        ;
                              F614    CF  9F 009E6 80$:     PUSHAB  P.AEK                                   ; 3461
                    2C  AA            01  FB 009EA          CALLS   #1, GET_FC                              ;
                  0218  C9            50  D0 009EE          MOVL    R0, VERIFY_FAB                          ;
                                07    A9  95 009F3 81$:     TSTB    QUAL+15                                 ; 3466
                                      0D  19 009F6          BLSS    82$                                     ;
           00000000G    8F            58  D1 009F8          CMPL    INIT_STATUS, #CLI$_ABSENT               ; 3467
                                      04  12 009FF          BNEQ    82$                                     ;
                    02  A9            20  88 00A01          BISB2   #32, QUAL+10                            ; 3468
            09      72  A9            01  E1 00A05 82$:     BBC     #1, COM_FLAGS, 83$                      ; 3470
                    02  A9            08  88 00A0A          BISB2   #8, QUAL+10                             ; 3473
                    02  A9      40    8F  8A 00A0E          BICB2   #64, QUAL+10                            ; 3474
            0C      04  A9            05  E1 00A13 83$:     BBC     #5, QUAL+12, 84$                        ; 3476
                    06  A9            04  8A 00A18          BICB2   #4, QUAL+14                             ; 3479
                    02  A9 00400058   8F  CA 00A1C          BICL2   #4194392, QUAL+10                       ; 3483
            0B      02  A9            03  E1 00A24 84$:     BBC     #3, QUAL+10, 85$                        ; 3485
                    01  A9      40    8F  88 00A29          BISB2   #64, QUAL+9                             ; 3488
                    02  A9            10  8A 00A2E          BICB2   #16, QUAL+10                            ; 3489
                                      04  11 00A32          BRB     86$                                     ; 3485
                    06  A9            01  8A 00A34 85$:     BICB2   #1, QUAL+14                             ; 3492
            1B      07  A9            06  E1 00A38 86$:     BBC     #6, QUAL+15, 87$                        ; 3496
            16      07  A9            04  E1 00A3D          BBC     #4, QUAL+15, 87$                        ;
            11      06  A9            04  E1 00A42          BBC     #4, QUAL+14, 87$                        ;
                    06  A9            10  8A 00A47          BICB2   #16, QUAL+14                            ; 3499
                         00000000G    8F  DD 00A4B          PUSHL   #BACKUP$_ENCQUAIGN                      ; 3500
```

```
                00000000G  00          01 FB 00A51          CALLS   #1, LIB$SIGNAL                          ;
                                       69 95 00A58 87$:     TSTB    QUAL+8                                  ; 3503
                                       05 18 00A5A          BGEQ    88$                                     ;
                     05    A9     80   8F 8A 00A5C          BICB2   #128, QUAL+13                           ;
                16   07    A9          06 E1 00A61 88$:     BBC     #6, QUAL+15, 89$                        ; 3504
                           12     06    A9 E9 00A66          BLBC    QUAL+14, 89$                            ;
                0D   02    A9          05 E0 00A6A          BBS     #5, QUAL+10, 89$                        ;
                           00000000G   8F DD 00A6F          PUSHL   #BACKUP$_CONFQUAL                       ; 3506
                00000000G  00          01 FB 00A75          CALLS   #1, LIB$SIGNAL                          ;
                04         69          05 E0 00A7C 89$:     BBS     #5, QUAL+8, 90$                         ; 3507
                     44    A9          03 90 00A80          MOVB    #3, QUAL+76                             ;
                           04     02    A9 E8 00A84 90$:     BLBS    QUAL+10, 91$                            ; 3508
                     46    A9          0A 90 00A88          MOVB    #10, QUAL+78                            ;
                           07     A9    95 00A8C 91$:        TSTB    QUAL+15                                 ; 3509
                                       72 18 00A8F          BGEQ    98$                                     ;
                           50     FC    A9 D0 00A91          MOVL    QUAL+4, R0                              ; 3516
                           51     04    A0 D0 00A95          MOVL    4(R0), FAB                              ;
                           50     40    A9 3C 00A99          MOVZWL  QUAL+72, BLOC_VALUE                     ; 3517
                11         69          04 E0 00A9D          BBS     #4, QUAL+8, 93$                         ; 3518
                07   40    A1          05 E1 00AA1          BBC     #5, 64(FAB), 92$                        ; 3521
                           50     2000  8F 3C 00AA6          MOVZWL  #8192, BLOC_VALUE                       ; 3522
                                       05 11 00AAB          BRB     93$                                     ;
                           50     7E00  8F 3C 00AAD 92$:     MOVZWL  #32256, BLOC_VALUE                      ; 3523
                           52     01FF  C0 9E 00AB2 93$:     MOVAB   511(R0), R2                             ; 3525
                50         52 000001FF 8F CB 00AB7          BICL3   #511, R2, BLOC_VALUE                    ;
                05   07    A9          03 E0 00ABF          BBS     #3, QUAL+15, 94$                        ; 3526
                10   40    A1          05 E0 00AC4          BBS     #5, 64(FAB), 95$                        ; 3527
                00007E00   8F          50 D1 00AC9 94$:     CMPL    BLOC_VALUE, #32256                      ; 3530
                                       15 1B 00AD0          BLEQU   96$                                     ;
                           50     7E00  8F 3C 00AD2          MOVZWL  #32256, BLOC_VALUE                      ;
                                       0E 11 00AD7          BRB     96$                                     ; 3526
                0000FE00   8F          50 D1 00AD9 95$:     CMPL    BLOC_VALUE, #65024                      ; 3534
                                       05 1B 00AE0          BLEQU   96$                                     ;
                           50     FE00' 8F 3C 00AE2          MOVZWL  #65024, BLOC_VALUE                      ;
                13         06    A9          01 E1 00AE7 96$:     BBC     #1, QUAL+14, 97$                ; 3538
                0E         40    A1          05 E1 00AEC          BBC     #5, 64(FAB), 97$                ; 3539
                00002000   8F          50 D1 00AF1          CMPL    BLOC_VALUE, #8192                       ; 3540
                                       05 1B 00AF8          BLEQU   97$                                     ;
                           50     2000  8F 3C 00AFA          MOVZWL  #8192, BLOC_VALUE                       ; 3542
                     40    A9          50 B0 00AFF 97$:     MOVW    BLOC_VALUE, QUAL+72                     ; 3544
                                       04 00B03 98$:        RET                                             ; 3546

; Routine Size:  2820 bytes,    Routine Base:  CODE + 0E56
```

```
;  2457          3547  1 %SBTTL 'TPARSE tables'
;  2458          3548  1 PSECT
;  2459          3549  1         GLOBAL= _LIB$KEY0$(NOWRITE,SHARE,PIC,EXECUTE,ALIGN(1),
;  2460          3550  1                 ADDRESSING_MODE(LONG_RELATIVE)),
;  2461          3551  1         GLOBAL= _LIB$STATE$(NOWRITE,SHARE,PIC,EXECUTE,ALIGN(0),
;  2462          3552  1                 ADDRESSING_MODE(LONG_RELATIVE));
;  2463          3553  1 SWITCHES
;  2464          3554  1         ADDRESSING_MODE(NONEXTERNAL=LONG_RELATIVE);
;  2465          3555  1
;  2466          3556  1
;  2467          3557  1 $INIT_STATE(UIC_STATES, UIC_KEYS);
;  2468        P 3558  1 $STATE(,
;  2469        P 3559  1     (TPA$_IDENT,,,,CONVERTED_UIC)
;  2470          3560  1     );
;  2471        P 3561  1 $STATE(,
;  2472        P 3562  1     (TPA$_EOS, TPA$_EXIT)
;  2473          3563  1     );
;  2474          3564  1
;  2475          3565  1
;  2476          3566  1 $INIT_STATE(OWN_STATES, OWN_KEYS);
;  2477        P 3567  1 $STATE(,
;  2478        P 3568  1     ('DEFAULT',OWN_END,,TPA_M_DEFAULT,TPA_FLAGS),
;  2479        P 3569  1     ('ORIGINAL',OWN_END,,TPA_M_ORIGINAL,TPA_FLAGS),
;  2480        P 3570  1     ('PARENT',OWN_END,,TPA_M_PARENT,TPA_FLAGS),
;  2481        P 3571  1     (TPA$_IDENT,OWN_SET_UIC,,,CONVERTED_UIC)
;  2482          3572  1     );
;  2483        P 3573  1 $STATE (OWN_SET_UIC,
;  2484        P 3574  1     (TPA$_LAMBDA,OWN_END,,TPA_M_UIC,TPA_FLAGS)
;  2485          3575  1     );
;  2486        P 3576  1 $STATE(OWN_END,
;  2487        P 3577  1     (TPA$_EOS, TPA$_EXIT)
;  2488          3578  1     );
;  2489          3579  1
;  2490          3580  1
;  2491          3581  1 $INIT_STATE(IGN_STATES, IGN_KEYS);
;  2492        P 3582  1 $STATE(,
;  2493        P 3583  1     ('NOBACKUP',,,TPA_M_NOBACKUP,TPA_FLAGS),
;  2494        P 3584  1     ('INTERLOCK',,,TPA_M_INTERLOCK,TPA_FLAGS)
;  2495          3585  1     );
;  2496        P 3586  1 $STATE(,
;  2497        P 3587  1     (TPA$_EOS, TPA$_EXIT)
;  2498          3588  1     );
;  2499          3589  1
;  2500          3590  1
;  2501          3591  1 $INIT_STATE(BAK_STATES, BAK_KEYS);
;  2502        P 3592  1 $STATE(,
;  2503        P 3593  1     ('BACKUP',TPA$_EXIT)
;  2504          3594  1     );
;  2505        P 3595  1 $STATE(,
;  2506        P 3596  1     (TPA$_EOS, TPA$_EXIT)
;  2507          3597  1     );
;  2508          3598  1
;  2509          3599  1
;  2510          3600  1 $INIT_STATE(PRO_STATES, PRO_KEYS);
;  2511        P 3601  1 $STATE(NEXTPRO,
;  2512        P 3602  1     ('SYSTEM',SYPR,,%X'000F0000',TPA_PROTECTION),
;  2513        P 3603  1     ('OWNER', OWPR,,%X'00F00000',TPA_PROTECTION),
```

```
;  2514        P 3604  1      ('GROUP', GRPR,,%X'0F000000',TPA_PROTECTION),
;  2515        P 3605  1      ('WORLD', WOPR,,%X'F0000000',TPA_PROTECTION)
;  2516          3606  1      );
;  2517        P 3607  1  $STATE(SYPR,
;  2518        P 3608  1      (':'),
;  2519        P 3609  1      ('='),
;  2520        P 3610  1      (TPA$_LAMBDA,ENDPRO)
;  2521          3611  1      );
;  2522        P 3612  1  $STATE(SYPRO,
;  2523        P 3613  1      ('R',SYPRO,,%X'0001',TPA_PROTECTION),
;  2524        P 3614  1      ('W',SYPRO,,%X'0002',TPA_PROTECTION),
;  2525        P 3615  1      ('E',SYPRO,,%X'0004',TPA_PROTECTION),
;  2526        P 3616  1      ('D',SYPRO,,%X'0008',TPA_PROTECTION),
;  2527        P 3617  1      (TPA$_LAMBDA,ENDPRO)
;  2528          3618  1      );
;  2529        P 3619  1  $STATE(OWPR,
;  2530        P 3620  1      (':'),
;  2531        P 3621  1      ('='),
;  2532        P 3622  1      (TPA$_LAMBDA,ENDPRO)
;  2533          3623  1      );
;  2534        P 3624  1  $STATE(OWPRO,
;  2535        P 3625  1      ('R',OWPRO,,%X'0010',TPA_PROTECTION),
;  2536        P 3626  1      ('W',OWPRO,,%X'0020',TPA_PROTECTION),
;  2537        P 3627  1      ('E',OWPRO,,%X'0040',TPA_PROTECTION),
;  2538        P 3628  1      ('D',OWPRO,,%X'0080',TPA_PROTECTION),
;  2539        P 3629  1      (TPA$_LAMBDA,ENDPRO)
;  2540          3630  1      );
;  2541        P 3631  1  $STATE(GRPR,
;  2542        P 3632  1      (':'),
;  2543        P 3633  1      ('='),
;  2544        P 3634  1      (TPA$_LAMBDA,ENDPRO)
;  2545          3635  1      );
;  2546        P 3636  1  $STATE(GRPRO,
;  2547        P 3637  1      ('R',GRPRO,,%X'0100',TPA_PROTECTION),
;  2548        P 3638  1      ('W',GRPRO,,%X'0200',TPA_PROTECTION),
;  2549        P 3639  1      ('E',GRPRO,,%X'0400',TPA_PROTECTION),
;  2550        P 3640  1      ('D',GRPRO,,%X'0800',TPA_PROTECTION),
;  2551        P 3641  1      (TPA$_LAMBDA,ENDPRO)
;  2552          3642  1      );
;  2553        P 3643  1  $STATE(WOPR,
;  2554        P 3644  1      (':'),
;  2555        P 3645  1      ('='),
;  2556        P 3646  1      (TPA$_LAMBDA,ENDPRO)
;  2557          3647  1      );
;  2558        P 3648  1  $STATE(WOPRO,
;  2559        P 3649  1      ('R',WOPRO,,%X'1000',TPA_PROTECTION),
;  2560        P 3650  1      ('W',WOPRO,,%X'2000',TPA_PROTECTION),
;  2561        P 3651  1      ('E',WOPRO,,%X'4000',TPA_PROTECTION),
;  2562        P 3652  1      ('D',WOPRO,,%X'8000',TPA_PROTECTION),
;  2563        P 3653  1      (TPA$_LAMBDA,ENDPRO)
;  2564          3654  1      );
;  2565        P 3655  1  $STATE(ENDPRO,
;  2566        P 3656  1      (',',NEXTPRO),
;  2567        P 3657  1      (TPA$_EOS,TPA$_EXIT)
;  2568          3658  1      );
;  2569          3659  1
;  2570          3660  1
```

```
; 2571          3661  1 END
; 2572          3662  0 ELUDOM


                                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                                        00000 ;TPA$KEYSTO
                                                              U.7:      .BLKB   0
                                    54  4C  55  41  46  45  44   00000 ;TPA$KEYST
                                                              U.9:      .ASCII  \DEFAULT\                       ;
                                                        FF    00007             .BYTE   -1                      ;
                                                        00008 ;TPA$KEYSTO
                                                              U.15:     .BLKB   0
                                4C  41  4E  49  47  49  52  4F   00008 ;TPA$KEYST
                                                              U.17:     .ASCII  \ORIGINAL\                      ;
                                                        FF    00010             .BYTE   -1                      ;
                                                        00011 ;TPA$KEYSTO
                                                              U.22:     .BLKB   0
                                    54  4E  45  52  41  50   00011 ;TPA$KEYST
                                                              U.24:     .ASCII  \PARENT\                        ;
                                                        FF    00017             .BYTE   -1                      ;
                                                        FF    00018 ;TPA$KEYFILL
                                                              U.33:     .BYTE   -1                      ;
                                                        00019 ;TPA$KEYSTO
                                                              U.41:     .BLKB   0
                                50  55  4B  43  41  42  4F  4E   00019 ;TPA$KEYST
                                                              U.43:     .ASCII  \NOBACKUP\                      ;
                                                        FF    00021             .BYTE   -1                      ;
                                                        00022 ;TPA$KEYSTO
                                                              U.47:     .BLKB   0
                            4B  43  4F  4C  52  45  54  4E  49   00022 ;TPA$KEYST
                                                              U.49:     .ASCII  \INTERLOCK\                     ;
                                                        FF    0002B             .BYTE   -1                      ;
                                                        FF    0002C ;TPA$KEYFILL
                                                              U.53:     .BYTE   -1                      ;
                                                        0002D ;TPA$KEYSTO
                                                              U.57:     .BLKB   0
                                    50  55  4B  43  41  42   0002D ;TPA$KEYST
                                                              U.59:     .ASCII  \BACKUP\                        ;
                                                        FF    00033             .BYTE   -1                      ;
                                                        FF    00034 ;TPA$KEYFILL
                                                              U.62:     .BYTE   -1                      ;
                                                        00035 ;TPA$KEYSTO
                                                              U.66:     .BLKB   0
                                    4D  45  54  53  59  53   00035 ;TPA$KEYST
                                                              U.68:     .ASCII  \SYSTEM\                        ;
                                                        FF    0003B             .BYTE   -1                      ;
                                                        0003C ;TPA$KEYSTO
                                                              U.74:     .BLKB   0
                                        52  45  4E  57  4F   0003C ;TPA$KEYST
                                                              U.76:     .ASCII  \OWNER\                         ;
                                                        FF    00041             .BYTE   -1                      ;
                                                        00042 ;TPA$KEYSTO
                                                              U.82:     .BLKB   0
                                        50  55  4F  52  47   00042 ;TPA$KEYST
                                                              U.84:     .ASCII  \GROUP\                         ;
```

```
                        FF  00047          .BYTE   -1                                      ;
                            00048 ;TPA$KEYSTO
                            U.90:   .BLKB   0
                44  4C  52  4F  57  00048 ;TPA$KEYST
                            U.92:   .ASCII  \WORLD\                                         ;
                        FF  0004D          .BYTE   -1                                      ;
                        FF  0004E ;TPA$KEYFILL
                            U.98:   .BYTE   -1                                              ;

                                           .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                            00000 UIC_STATES::
                                           .BLKB   0
                      45EC  00000 ;TPA$TYPE
                            U.2:    .WORD   17900                                           ;
                00000000* 00002 ;TPA$ADDR
                            U.3:    .LONG   <<CONVERTED_UIC-U.3>-4>                         ;
                      15F7  00006 ;TPA$TYPE
                            U.4:    .WORD   5623                                            ;
                      FFFF  00008 ;TPA$TARGET
                            U.5:    .WORD   -1                                              ;
                            0000A          .BLKB   2
                            0000C OWN_STATES::
                                           .BLKB   0
                      7100  0000C ;TPA$TYPE
                            U.10:   .WORD   28928                                           ;
                00000000* 0000E ;TPA$ADDR
                            U.11:   .LONG   <<TPA_FLAGS-U.11>-4>                            ;
                00000004  00012 ;TPA$MASK
                            U.12:   .LONG   4                                               ;
                    0000* 00016 ;TPA$TARGET
                            U.14:   .WORD   <<U.13-U.14>-2>                                 ;
                      7101  00018 ;TPA$TYPE
                            U.18:   .WORD   28929                                           ;
                00000000* 0001A ;TPA$ADDR
                            U.19:   .LONG   <<TPA_FLAGS-U.19>-4>                            ;
                00000008  0001E ;TPA$MASK
                            U.20:   .LONG   8                                               ;
                    0000* 00022 ;TPA$TARGET
                            U.21:   .WORD   <<U.13-U.21>-2>                                 ;
                      7102  00024 ;TPA$TYPE
                            U.25:   .WORD   28930                                           ;
                00000000* 00026 ;TPA$ADDR
                            U.26:   .LONG   <<TPA_FLAGS-U.26>-4>                            ;
                00000010  0002A ;TPA$MASK
                            U.27:   .LONG   16                                              ;
                    0000* 0002E ;TPA$TARGET
                            U.28:   .WORD   <<U.13-U.28>-2>                                 ;
                      55EC  00030 ;TPA$TYPE
                            U.29:   .WORD   21996                                           ;
                00000000* 00032 ;TPA$ADDR
                            U.30:   .LONG   <<CONVERTED_UIC-U.30>-4>                        ;
                    0000* 00036 ;TPA$TARGET
                            U.32:   .WORD   <<U.31-U.32>-2>                                 ;
                            00038 ;OWN_SET_UIC
                            U.31:   .BLKB   0
                      75F6  00038 ;TPA$TYPE
```

```
                        U.34:   .WORD   30198                   ;
00000000*  0003A  ;TPA$ADDR
                        U.35:   .LONG   <<TPA_FLAGS-U.35>-4>    ;
00000020   0003E  ;TPA$MASK
                        U.36:   .LONG   32                      ;
    0000*  00042  ;TPA$TARGET
                        U.37:   .WORD   <<U.13-U.37>-2>         ;
           00044  ;OWN_END
                        U.13:   .BLKB   0
    15F7   00044  ;TPA$TYPE
                        U.38:   .WORD   5623                    ;
    FFFF   00046  ;TPA$TARGET
                        U.39:   .WORD   -1                      ;
           00048  IGN_STATES::
                                .BLKB   0
    6100   00048  ;TPA$TYPE
                        U.44:   .WORD   24832                   ;
00000000*  0004A  ;TPA$ADDR
                        U.45:   .LONG   <<TPA_FLAGS-U.45>-4>    ;
00000040   0004E  ;TPA$MASK
                        U.46:   .LONG   64                      ;
    6501   00052  ;TPA$TYPE
                        U.50:   .WORD   25857                   ;
00000000*  00054  ;TPA$ADDR
                        U.51:   .LONG   <<TPA_FLAGS-U.51>-4>    ;
00000080   00058  ;TPA$MASK
                        U.52:   .LONG   128                     ;
    15F7   0005C  ;TPA$TYPE
                        U.54:   .WORD   5623                    ;
    FFFF   0005E  ;TPA$TARGET
                        U.55:   .WORD   -1                      ;
           00060  BAK_STATES::
                                .BLKB   0
    1500   00060  ;TPA$TYPE
                        U.60:   .WORD   5376                    ;
    FFFF   00062  ;TPA$TARGET
                        U.61:   .WORD   -1                      ;
    15F7   00064  ;TPA$TYPE
                        U.63:   .WORD   5623                    ;
    FFFF   00066  ;TPA$TARGET
                        U.64:   .WORD   -1                      ;
           00068  PRO_STATES::
                                .BLKB   0
           00068  NEXTPRO:.BLKB   0
    7100   00068  ;TPA$TYPE
                        U.69:   .WORD   28928                   ;
00000000*  0006A  ;TPA$ADDR
                        U.70:   .LONG   <<TPA_PROTECTION-U.70>-4>;
000F0000   0006E  ;TPA$MASK
                        U.71:   .LONG   983040                  ;
    0000*  00072  ;TPA$TARGET
                        U.73:   .WORD   <<U.72-U.73>-2>         ;
    7101   00074  ;TPA$TYPE
                        U.77:   .WORD   28929                   ;
00000000*  00076  ;TPA$ADDR
                        U.78:   .LONG   <<TPA_PROTECTION-U.78>-4>;
00F00000   0007A  ;TPA$MASK
```

```
                        U.79:   .LONG   15728640                        ;
    0000* 0007E ;TPA$TARGET
                        U.81:   .WORD   <<U.80-U.81>-2>                 ;
    7102  00080 ;TPA$TYPE
                        U.85:   .WORD   28930                           ;
00000000* 00082 ;TPA$ADDR
                        U.86:   .LONG   <<TPA_PROTECTION-U.86>-4>       ;
0F000000  00086 ;TPA$MASK
                        U.87:   .LONG   251658240                       ;
    0000* 0008A ;TPA$TARGET
                        U.89:   .WORD   <<U.88-U.89>-2>                 ;
    7503  0008C ;TPA$TYPE
                        U.93:   .WORD   29955                           ;
00000000* 0008E ;TPA$ADDR
                        U.94:   .LONG   <<TPA_PROTECTION-U.94>-4>       ;
F0000000  00092 ;TPA$MASK
                        U.95:   .LONG   -268435456                      ;
    0000* 00096 ;TPA$TARGET
                        U.97:   .WORD   <<U.96-U.97>-2>                 ;
          00098 ;SYPR
                        U.72:   .BLKB   0
    003A  00098 ;TPA$TYPE
                        U.99:   .WORD   58                              ;
    003D  0009A ;TPA$TYPE
                        U.100:  .WORD   61                              ;
    15F6  0009C ;TPA$TYPE
                        U.101:  .WORD   5622                            ;
    0000* 0009E ;TPA$TARGET
                        U.103:  .WORD   <<U.102-U.103>-2>               ;
          000A0 SYPR0:  .BLKB   0
    7052  000A0 ;TPA$TYPE
                        U.104:  .WORD   28754                           ;
00000000* 000A2 ;TPA$ADDR
                        U.105:  .LONG   <<TPA_PROTECTION-U.105>-4>      ;
00000001  000A6 ;TPA$MASK
                        U.106:  .LONG   1                               ;
    0000* 000AA ;TPA$TARGET
                        U.107:  .WORD   <<SYPR0-U.107>-2>               ;
    7057  000AC ;TPA$TYPE
                        U.108:  .WORD   28759                           ;
00000000* 000AE ;TPA$ADDR
                        U.109:  .LONG   <<TPA_PROTECTION-U.109>-4>      ;
00000002  000B2 ;TPA$MASK
                        U.110:  .LONG   2                               ;
    0000* 000B6 ;TPA$TARGET
                        U.111:  .WORD   <<SYPR0-U.111>-2>               ;
    7045  000B8 ;TPA$TYPE
                        U.112:  .WORD   28741                           ;
00000000* 000BA ;TPA$ADDR
                        U.113:  .LONG   <<TPA_PROTECTION-U.113>-4>      ;
00000004  000BE ;TPA$MASK
                        U.114:  .LONG   4                               ;
    0000* 000C2 ;TPA$TARGET
                        U.115:  .WORD   <<SYPR0-U.115>-2>               ;
    7044  000C4 ;TPA$TYPE
                        U.116:  .WORD   28740                           ;
00000000* 000C6 ;TPA$ADDR
```

```
                     U.117:  .LONG   <<TPA_PROTECTION-U.117>-4>                ;
00000008  000CA ;TPA$MASK
                     U.118:  .LONG   8                                         ;
    0000* 000CE ;TPA$TARGET
                     U.119:  .WORD   <<SYPRO-U.119>-2>                         ;
    15F6  000D0 ;TPA$TYPE
                     U.120:  .WORD   5622                                      ;
    0000* 000D2 ;TPA$TARGET
                     U.121:  .WORD   <<U.102-U.121>-2>                         ;
          000D4 ;OWPR
                     U.80:   .BLKB   0
    003A  000D4 ;TPA$TYPE
                     U.122:  .WORD   58                                        ;
    003D  000D6 ;TPA$TYPE
                     U.123:  .WORD   61                                        ;
    15F6  000D8 ;TPA$TYPE
                     U.124:  .WORD   5622                                      ;
    0000* 000DA ;TPA$TARGET
                     U.125:  .WORD   <<U.102-U.125>-2>                         ;
          000DC OWPRO:  .BLKB   0
    7052  000DC ;TPA$TYPE
                     U.126:  .WORD   28754                                     ;
00000000* 000DE ;TPA$ADDR
                     U.127:  .LONG   <<TPA_PROTECTION-U.127>-4>                ;
00000010  000E2 ;TPA$MASK
                     U.128:  .LONG   16                                        ;
    0000* 000E6 ;TPA$TARGET
                     U.129:  .WORD   <<OWPRO-U.129>-2>                         ;
    7057  000E8 ;TPA$TYPE
                     U.130:  .WORD   28759                                     ;
00000000* 000EA ;TPA$ADDR
                     U.131:  .LONG   <<TPA_PROTECTION-U.131>-4>                ;
00000020  000EE ;TPA$MASK
                     U.132:  .LONG   32                                        ;
    0000* 000F2 ;TPA$TARGET
                     U.133:  .WORD   <<OWPRO-U.133>-2>                         ;
    7045  000F4 ;TPA$TYPE
                     U.134:  .WORD   28741                                     ;
00000000* 000F6 ;TPA$ADDR
                     U.135:  .LONG   <<TPA_PROTECTION-U.135>-4>                ;
00000040  000FA ;TPA$MASK
                     U.136:  .LONG   64                                        ;
    0000* 000FE ;TPA$TARGET
                     U.137:  .WORD   <<OWPRO-U.137>-2>                         ;
    7044  00100 ;TPA$TYPE
                     U.138:  .WORD   28740                                     ;
00000000* 00102 ;TPA$ADDR
                     U.139:  .LONG   <<TPA_PROTECTION-U.139>-4>                ;
00000080  00106 ;TPA$MASK
                     U.140:  .LONG   128                                       ;
    0000* 0010A ;TPA$TARGET
                     U.141:  .WORD   <<OWPRO-U.141>-2>                         ;
    15F6  0010C ;TPA$TYPE
                     U.142:  .WORD   5622                                      ;
    0000* 0010E ;TPA$TARGET
                     U.143:  .WORD   <<U.102-U.143>-2>                         ;
          00110 ;GRPR
```

```
                         U.88:    .BLKB    0
    003A  00110 ;TPA$TYPE
                         U.144:   .WORD    58                                  ;
    003D  00112 ;TPA$TYPE
                         U.145:   .WORD    61                                  ;
    15F6  00114 ;TPA$TYPE
                         U.146:   .WORD    5622                                ;
   0000*  00116 ;TPA$TARGET
                         U.147:   .WORD    <<U.102-U.147>-2>                   ;
          00118 GRPRO:   .BLKB    0
    7052  00118 ;TPA$TYPE
                         U.148:   .WORD    28754                               ;
00000000* 0011A ;TPA$ADDR
                         U.149:   .LONG    <<TPA_PROTECTION-U.149>-4>          ;
00000100  0011E ;TPA$MASK
                         U.150:   .LONG    256                                 ;
   0000*  00122 ;TPA$TARGET
                         U.151:   .WORD    <<GRPRO-U.151>-2>                   ;
    7057  00124 ;TPA$TYPE
                         U.152:   .WORD    28759                               ;
00000000* 00126 ;TPA$ADDR
                         U.153:   .LONG    <<TPA_PROTECTION-U.153>-4>          ;
00000200  0012A ;TPA$MASK
                         U.154:   .LONG    512                                 ;
   0000*  0012E ;TPA$TARGET
                         U.155:   .WORD    <<GRPRO-U.155>-2>                   ;
    7045  00130 ;TPA$TYPE
                         U.156:   .WORD    28741                               ;
00000000* 00132 ;TPA$ADDR
                         U.157:   .LONG    <<TPA_PROTECTION-U.157>-4>          ;
00000400  00136 ;TPA$MASK
                         U.158:   .LONG    1024                                ;
   0000*  0013A ;TPA$TARGET
                         U.159:   .WORD    <<GRPRO-U.159>-2>                   ;
    7044  0013C ;TPA$TYPE
                         U.160:   .WORD    28740                               ;
00000000* 0013E ;TPA$ADDR
                         U.161:   .LONG    <<TPA_PROTECTION-U.161>-4>          ;
00000800  00142 ;TPA$MASK
                         U.162:   .LONG    2048                                ;
   0000*  00146 ;TPA$TARGET
                         U.163:   .WORD    <<GRPRO-U.163>-2>                   ;
    15F6  00148 ;TPA$TYPE
                         U.164:   .WORD    5622                                ;
   0000*  0014A ;TPA$TARGET
                         U.165:   .WORD    <<U.102-U.165>-2>                   ;
          0014C ;WOPR
                         U.96:    .BLKB    0
    003A  0014C ;TPA$TYPE
                         U.166:   .WORD    58                                  ;
    003D  0014E ;TPA$TYPE
                         U.167:   .WORD    61                                  ;
    15F6  00150 ;TPA$TYPE
                         U.168:   .WORD    5622                                ;
   0000*  00152 ;TPA$TARGET
                         U.169:   .WORD    <<U.102-U.169>-2>                   ;
          00154 WOPRO:   .BLKB    0
```

```
                        7052  00154 ;TPA$TYPE
                              U.170:  .WORD   28754                                   ;
                00000000* 00156 ;TPA$ADDR
                              U.171:  .LONG   <<TPA_PROTECTION-U.171>-4>              ;
                00001000  0015A ;TPA$MASK
                              U.172:  .LONG   4096                                    ;
                    0000* 0015E ;TPA$TARGET
                              U.173:  .WORD   <<WOPRO-U.173>-2>                       ;
                        7057  00160 ;TPA$TYPE
                              U.174:  .WORD   28759                                   ;
                00000000* 00162 ;TPA$ADDR
                              U.175:  .LONG   <<TPA_PROTECTION-U.175>-4>              ;
                00002000  00166 ;TPA$MASK
                              U.176:  .LONG   8192                                    ;
                    0000* 0016A ;TPA$TARGET
                              U.177:  .WORD   <<WOPRO-U.177>-2>                       ;
                        7045  0016C ;TPA$TYPE
                              U.178:  .WORD   28741                                   ;
                00000000* 0016E ;TPA$ADDR
                              U.179:  .LONG   <<TPA_PROTECTION-U.179>-4>              ;
                00004000  00172 ;TPA$MASK
                              U.180:  .LONG   16384                                   ;
                    0000* 00176 ;TPA$TARGET
                              U.181:  .WORD   <<WOPRO-U.181>-2>                       ;
                        7044  00178 ;TPA$TYPE
                              U.182:  .WORD   28740                                   ;
                00000000* 0017A ;TPA$ADDR
                              U.183:  .LONG   <<TPA_PROTECTION-U.183>-4>              ;
                00008000  0017E ;TPA$MASK
                              U.184:  .LONG   32768                                   ;
                    0000* 00182 ;TPA$TARGET
                              U.185:  .WORD   <<WOPRO-U.185>-2>                       ;
                        15F6  00184 ;TPA$TYPE
                              U.186:  .WORD   5622                                    ;
                    0000* 00186 ;TPA$TARGET
                              U.187:  .WORD   <<U.102-U.187>-2>                       ;
                              00188 ;ENDPRO
                              U.102:  .BLKB   0
                        102C  00188 ;TPA$TYPE
                              U.188:  .WORD   4140                                    ;
                    0000* 0018A ;TPA$TARGET
                              U.189:  .WORD   <<NEXTPRO-U.189>-2>                     ;
                        15F7  0018C ;TPA$TYPE
                              U.190:  .WORD   5623                                    ;
                    FFFF  0018E ;TPA$TARGET
                              U.191:  .WORD   -1                                      ;

                                      .PSECT  _LIB$KEY0$,NOWRT,  SHR,  PIC,1

                              00000 UIC_KEYS::
                                      .BLKB   0
                              00000 ;TPA$KEY0
                                    U.1:      .BLKB   0
                              00000 OWN_KEYS::
                                      .BLKB   0
                              00000 ;TPA$KEY0
                                    U.6:      .BLKB   0
```

```
0000* 00000 ;TPA$KEY
            U.8:    .WORD   <U.7-U.6>                       ;
0000* 00002 ;TPA$KEY
            U.16:   .WORD   <U.15-U.6>                      ;
0000* 00004 ;TPA$KEY
            U.23:   .WORD   <U.22-U.6>                      ;
      00006         .BLKB   2
      00008 IGN_KEYS::
                    .BLKB   0
      00008 ;TPA$KEY0
            U.40:   .BLKB   0
0000* 00008 ;TPA$KEY
            U.42:   .WORD   <U.41-U.40>                     ;
0000* 0000A ;TPA$KEY
            U.48:   .WORD   <U.47-U.40>                     ;
      0000C BAK_KEYS::
                    .BLKB   0
      0000C ;TPA$KEY0
            U.56:   .BLKB   0
0000* 0000C ;TPA$KEY
            U.58:   .WORD   <U.57-U.56>                     ;
      0000E         .BLKB   2
      00010 PRO_KEYS::
                    .BLKB   0
      00010 ;TPA$KEY0
            U.65:   .BLKB   0
0000* 00010 ;TPA$KEY
            U.67:   .WORD   <U.66-U.65>                     ;
0000* 00012 ;TPA$KEY
            U.75:   .WORD   <U.74-U.65>                     ;
0000* 00014 ;TPA$KEY
            U.83:   .WORD   <U.82-U.65>                     ;
0000* 00016 ;TPA$KEY
            U.91:   .WORD   <U.90-U.65>                     ;

                    .EXTRN  LIB$SIGNAL
```

PSECT SUMMARY

Name	Bytes	Attributes
COMMON	2124	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, OVR,NOPIC,ALIGN(2)
DATA	4	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
CODE	6490	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
_LIB$KEY0$	24	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB$STATE$	400	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB$KEY1$	79	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time

```
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619       153         0       1000          00:02.2
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1          42        24        57         14          00:00.2



;                                       COMMAND QUALIFIERS

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

; Size:          5491 code + 3630 data bytes
; Run Time:        02:06.6
; Elapsed Time:    06:34.5
; Lines/CPU Min:    1735
; Lexemes/CPU-Min: 44120
; Memory Used:  845 pages
; Compilation Complete
```

BACKUPMSG
LIS

ANALYZE
LIS

BUFFERS
LIS

CREATEDIR
LIS

BADBLOCK
LIS

BACKUPCMD
LIS

COMMAND
LIS