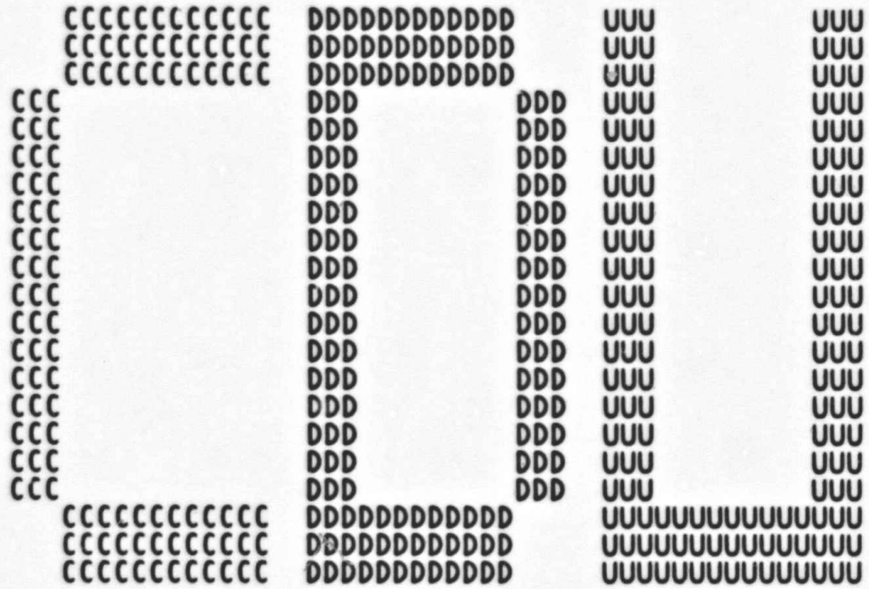
CCCCCCCCCCC DDDDDDDDDDD UUU UUU
CCCCCCCCCCC DDDDDDDDDDD UUU UUU
CCCCCCCCCCC DDDDDDDDDDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCC DDD DDD UUU UUU
CCCCCCCCCCC DDDDDDDDDDD UUUUUUUUUUUUUUU
CCCCCCCCCCC DDDDDDDDDDD UUUUUUUUUUUUUUU
CCCCCCCCCCC DDDDDDDDDDD UUUUUUUUUUUUUUU

FILEID**UPGRADE


```
UU      UU  PPPPPPPP     GGGGGGGG  RRRRRRRR     AAAAAA    DDDDDDDD    EEEEEEEEEE
UU      UU  PPPPPPPP     GGGGGGGG  RRRRRRRR     AAAAAA    DDDDDDDD    EEEEEEEEEE
UU      UU  PP      PP  GG        RR      RR   AA    AA  DD      DD  EE
UU      UU  PP      PP  GG        RR      RR   AA    AA  DD      DD  EE
UU      UU  PP      PP  GG        RR      RR   AA    AA  DD      DD  EE
UU      UU  PP      PP  GG        RR      RR   AA    AA  DD      DD  EE
UU      UU  PPPPPPPP    GG        RRRRRRRR     AA    AA  DD      DD  EEEEEEEE
UU      UU  PPPPPPPP    GG        RRRRRRRR     AA    AA  DD      DD  EEEEEEEE
UU      UU  PP          GG  GGGGGG RR   RR      AAAAAAAAAA  DD      DD  EE
UU      UU  PP          GG  GGGGGG RR   RR      AAAAAAAAAA  DD      DD  EE
UU      UU  PP          GG      GG RR    RR     AA    AA  DD      DD  EE          ....
UU      UU  PP          GG      GG RR    RR     AA    AA  DD      DD  EE          ....
UUUUUUUUUU  PP            GGGGGG   RR      RR   AA    AA  DDDDDDDD    EEEEEEEEEE  ....
UUUUUUUUUU  PP            GGGGGG   RR      RR   AA    AA  DDDDDDDD    EEEEEEEEEE  ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE upgrade          (IDENT='V04-000',
;     2          0002  0                          ADDRESSING_MODE(EXTERNAL=GENERAL))
;     3          0003  1 = BEGIN
;     4          0004  1
;     5          0005  1 !*****************************************************************************
;     6          0006  1 !*                                                                           *
;     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     9          0009  1 !*  ALL RIGHTS RESERVED.                                                      *
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    16          0016  1 !*  TRANSFERRED.                                                              *
;    17          0017  1 !*                                                                           *
;    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    20          0020  1 !*  CORPORATION.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*****************************************************************************
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! Facility:     Command Definition Utility, CLI Table Upgrade Module
;    30          0030  1 !
;    31          0031  1 ! Abstract:     This module is solely responsible for the upgrading of
;    32          0032  1 !               old format CLI tables to the latest format level.  The
;    33          0033  1 !               module is included in both the CDU and the CLIs, and thus
;    34          0034  1 !               must be completely self-contained.
;    35          0035  1 !
;    36          0036  1 ! Environment:  No assumptions may be made about the environment.
;    37          0037  1 !               No own storage is allowed.
;    38          0038  1 !               No external references are allowed.
;    39          0039  1 !
;    40          0040  1 ! Author:       Paul C. Anagnostopoulos
;    41          0041  1 ! Creation:     8 March 1983
;    42          0042  1 !
;    43          0043  1 ! Modifications:
;    44          0044  1 !
;    45          0045  1 !       V04-003 BLS0285                 Benn Schreiber          9-MAR-1984
;    46          0046  1 !               If image name length is 0, then it's a routine address,
;    47          0047  1 !               which counts for 4 bytes.
;    48          0048  1 !
;    49          0049  1 !       V04-002 BLS0270                 Benn Schreiber          9-FEB-1984
;    50          0050  1 !               Correct errors in structure length computation.
;    51          0051  1 !
;    52          0052  1 !       V04-001 PCA1026                 Paul C. Anagnostopoulos 25-Jul-1983
;    53          0053  1 !               Add probe to check readability of command table to be
;    54          0054  1 !               converted.  Fix bug in creation of entity block, so that
;    55          0055  1 !               a label is always included.
;    56          0056  1 !--
;    57          0057  1
```

```
;    58          0058  1
;    59          0059  1 library 'sys$library:lib';
;    60          0060  1 require 'clitabdef';
;    61          0385  1 require 'cli5def';
;    62          0657  1 require 'cdureq';
```

```
;    64          1071  1 !       P S E C T   N A M E S
;    65          1072  1 !       ---------   ---------
;    66          1073  1
;    67          1074  1 ! The following psect names are chosen so DCL won't break when it links
;    68          1075  1 ! with this module.  The leading underscores cause the psect to appear
;    69          1076  1 ! at the end of the image.
;    70          1077  1
;    71          1078  1 psect plit = _cdu$plit(align(0),read,nowrite,noexecute);
;    72          1079  1 psect code = _cdu$code(align(0),read,nowrite,execute);
;    73          1080  1
;    74          1081  1
;    75          1082  1 !       T A B L E   O F   C O N T E N T S
;    76          1083  1 !       ---------   ---   -------------
;    77          1084  1
;    78          1085  1 forward routine
;    79          1086  1         cdu$upgrade_table,
;    80          1087  1         upgrade_5_to_6,
;    81          1088  1         upgrade_5_to_6_allocate,
;    82          1089  1         upgrade_5_to_6_vector: novalue,
;    83          1090  1         upgrade_5_to_6_command: novalue,
;    84          1091  1         upgrade_5_to_6_entity: novalue;
;    85          1092  1
;    86          1093  1
;    87          1094  1 !       M A C R O   D E F I N I T I O N S
;    88          1095  1 !       ---------   -------------------
;    89          1096  1
;    90          1097  1 ! The following macro will return the length of an ASCIC string which is
;    91          1098  1 ! present at a certain offset within a block.  If the offset is within the
;    92          1099  1 ! fixed portion of the block, then there is no string.  Or, the byte at the
;    93          1100  1 ! offset may not be readable.
;    94          1101  1
;    95          1102  1 macro
;    96        M 1103  1         ascic_length(the_block,the_offset) =
;    97        M 1104  1                 (builtin
;    98        M 1105  1                         prober;
;    99        M 1106  1
;   100        M 1107  1                 if the_offset lequ 8 then
;   101        M 1108  1                         0
;   102        M 1109  1                 else if prober(%ref(psl$c_user),%ref(1),the_block+the_offset) then
;   103        M 1110  1                         1+ch$rchar(the_block+the_offset)
;   104        M 1111  1                 else
;   105        M 1112  1                         0
;   106          1113  1                 ) %;
;   107          1114  1
;   108          1115  1 ! The following macro will translate an old block address into the
;   109          1116  1 ! corresponding new block address.  This is done by looking up the old
;   110          1117  1 ! address in the vector of old addresses and taking the corresponding entry
;   111          1118  1 ! from the vector of new addresses.  Note that the zeroth entry of the vectors
;   112          1119  1 ! contains the entry count.
;   113          1120  1
;   114          1121  1 macro
;   115        M 1122  1         new_block_address(old_block_address) =
;   116        M 1123  1                 (bind
;   117        M 1124  1                         old_block_address_bind = old_block_address: block[,byte];
;   118        M 1125  1
;   119        M 1126  1                 incr i from 1 to .old_vector[0] do
;   120        M 1127  1                         if old_block_address_bind eqla .old_vector[.i] then
```

```
;  121          M 1128  1                    exitloop .new_vector[.i]
;  122            1129  1            ) %;
```

```
;  124          1130  1 !++
;  125          1131  1 ! Description:  This routine is called whenever a CLI table is about to be
;  126          1132  1 !               used.  Its goal is to upgrade the CLI table to the latest
;  127          1133  1 !               format level, so that no other module need be concerned
;  128          1134  1 !               with any format but the latest.
;  129          1135  1 !
;  130          1136  1 ! Parameters:   table           By reference, the address of the CLI table
;  131          1137  1 !                               (its primary vector block).
;  132          1138  1 !               new_pointer     Optional, by reference, a longword which is
;  133          1139  1 !                               to receive the address of the upgraded table.
;  134          1140  1 !               get_vm          Optional, by reference, the address of a
;  135          1141  1 !                               routine with the same interface as LIB$GET_VM,
;  136          1142  1 !                               for obtaining virtual memory.
;  137          1143  1 !               free_vm         Optional, by reference, self-explanatory.
;  138          1144  1 !
;  139          1145  1 !
;  140          1146  1 ! Returns:      A status describing what happened.
;  141          1147  1 !
;  142          1148  1 ! Notes:
;  143          1149  1 !--
;  144          1150  1
;  145          1151  1 GLOBAL ROUTINE cdu$upgrade_table(table: pointer,
;  146          1152  1                                 new_pointer: ref vector[1,long],
;  147          1153  1                                 get_vm: pointer,
;  148          1154  1                                 free_vm: pointer)
;  149          1155  2 = BEGIN
;  150          1156  2
;  151          1157  2 local
;  152          1158  2         level: long;
;  153          1159  2
;  154          1160  2 builtin
;  155          1161  2         nullparameter,
;  156          1162  2         prober;
;  157          1163  2
;  158          1164  2
;  159          1165  2 ! The first thing to do is ensure that we can read the table.  If not,
;  160          1166  2 ! just return a bad status.  This is done because we may be called
;  161          1167  2 ! by DCL when there is no current CLI table.
;  162          1168  2
;  163          1169  2 if not prober(%ref(psl$c_user),%ref(1),.table) then
;  164          1170  2         return msg(cli$_invtab);
;  165          1171  2
;  166          1172  2 ! We need to do is determine the format level of the table.
;  167          1173  2 ! Prior to level 6, the primary vector block had a different format, so
;  168          1174  2 ! we have to determine the basic format and then the exact level.
;  169          1175  2
;  170          1176  3 level = (if .table[vec_w_size] eqlu vec_k_length and
;  171          1177  3           .table[vec_b_type] eqlu block_k_vector then
;  172          1178  3                   .table[vec_b_strlvl]                    ! Level 6 or later.
;  173          1179  3           else
;  174          1180  2                   .table[vec5_b_strlvl]);                 ! Level 5 or earlier.
;  175          1181  2
;  176          1182  2 ! Select on the format level of the table.
;  177          1183  2
;  178          1184  2 selectoneu .level of set
;  179          1185  2
;  180          1186  2 [5]:
```

```
;  181          1187  2
;  182          1188  2          ! It's a level 5 table, so we can upgrade it to the latest level.
;  183          1189  2          ! If we were called with only one argument, however, that means that
;  184          1190  2          ! the caller doesn't think we should upgrade an old table.  This is
;  185          1191  2          ! true of the process-permanent table, because the user should
;  186          1192  2          ! understand the implication of upgrading old CLDs, and is thus
;  187          1193  2          ! required to do it by hand.
;  188          1194  2
;  189          1195  2          (if nullparameter(2) then
;  190          1196  2                  return msg(cli$_oldtab);
;  191          1197  2
;  192          1198  2          ! Call a routine to upgrade the table.  It returns the final status.
;  193          1199  2
;  194          1200  2          return upgrade_5_to_6(.table,new_pointer[0],.get_vm,.free_vm););
;  195          1201  2
;  196          1202  2  [6]:
;  197          1203  2
;  198          1204  2          ! Level 6 is the current level.
;  199          1205  2
;  200          1206  2          (if not nullparameter(2) then
;  201          1207  2                  new_pointer[0] = .table;
;  202          1208  2          return msg(cli$_oktab););
;  203          1209  2
;  204          1210  2  [otherwise]:
;  205          1211  2
;  206          1212  2          ! God knows what this table is.
;  207          1213  2
;  208          1214  2          return msg(cli$_invtab);
;  209          1215  2  tes;
;  210          1216  2
;  211          1217  1  END;


                                                .TITLE  UPGRADE
                                                .IDENT  \V04-000\

                                                .EXTRN  CLI$_INVTAB, CLI$_OLDTAB
                                                .EXTRN  CLI$_OKTAB

                                                .PSECT  _CDU$CODE,NOWRT,0

                                    0004 00000          .ENTRY  CDU$UPGRADE_TABLE, Save R2                      ;  1151
                    52 00000000G    8F   D0 00002          MOVL    #CLI$_INVTAB, R2
                    50          04  AC   D0 00009          MOVL    TABLE, R0                                   ;  1169
          60        01              03   0C 0000D          PROBER  #3, #1, (R0)
                                    04   12 00011          BNEQ    1$
                    50              52   D0 00013          MOVL    R2, R0                                      ;  1170
                                    04   00016          RET
                    14              60   B1 00017 1$:      CMPW    (R0), #20                                   ;  1176
                                    0C   12 0001A          BNEQ    2$
                    01          02  A0   91 0001C          CMPB    2(R0), #1                                   ;  1177
                                    06   12 00020          BNEQ    2$
                    51          04  A0   9A 00022          MOVZBL  4(R0), LEVEL                                ;  1178
                                    04   11 00026          BRB     3$
                    51          28  A0   9A 00028 2$:      MOVZBL  40(R0), LEVEL                               ;  1180
                    05              51   D1 0002C 3$:      CMPL    LEVEL, #5                                   ;  1186
                                    21   12 0002F          BNEQ    6$
```

```
                         02              6C  91 00031          CMPB    (AP), #2                                ; 1195
                                         05  1F 00034          BLSSU   4$                                      ;
                               08        AC  D5 00036          TSTL    8(AP)                                   ;
                                         08  12 00039          BNEQ    5$                                      ;
                         50 00000000G    8F  DC 0003B 4$:      MOVL    #CLI$_OLDTAB, R0                        ; 1196
                                             04 00042          RET                                             ;
                         7E        0C    AC  7D 00043 5$:      MOVQ    GET_VM, -(SP)                           ; 1200
                                   08    AC  DD 00047          PUSHL   NEW_POINTER                             ;
                                         50  DD 0004A          PUSHL   R0                                      ;
                   0000V CF              04  FB 0004C          CALLS   #4, UPGRADE_5_TO_6                      ;
                                             04 00051          RET                                             ;
                         06              51  D1 00052 6$:      CMPL    LEVEL, #6                               ; 1202
                                         16  12 00055          BNEQ    8$                                      ;
                         02              6C  91 00057          CMPB    (AP), #2                                ; 1206
                                         09  1F 0005A          BLSSU   7$                                      ;
                               08        AC  D5 0005C          TSTL    8(AP)                                   ;
                                         04  13 0005F          BEQL    7$                                      ;
                      08 BC              50  D0 00061          MOVL    R0, @NEW_POINTER                        ; 1207
                         50 00000000G    8F  D0 00065 7$:      MOVL    #CLI$_OKTAB, R0                         ; 1208
                                             04 0006C          RET                                             ;
                         50              52  D0 0006D 8$:      MOVL    R2, R0                                  ; 1214
                                             04 00070          RET                                             ; 1217

; Routine Size:  113 bytes,    Routine Base:  _CDU$CODE + 0000
```

```
;  213          1218  1 !++
;  214          1219  1 ! Description:  This is the main routine for upgrading a level 5 (VMS V3)
;  215          1220  1 !               CLI table to level 6 (VMS V4).
;  216          1221  1 !
;  217          1222  1 ! Parameters:   table           By reference, the address of the CLI table
;  218          1223  1 !                               (its primary vector block).
;  219          1224  1 !               new_pointer     By reference, a longword in which to return
;  220          1225  1 !                               the address of the new table.
;  221          1226  1 !               get_vm          By reference, see above.
;  222          1227  1 !               free_vm         By reference, see above.
;  223          1228  1 !
;  224          1229  1 ! Returns:      By reference, the new primary vector block.
;  225          1230  1 !
;  226          1231  1 ! Notes:
;  227          1232  1 !--
;  228          1233  1
;  229          1234  1 ROUTINE upgrade_5_to_6(table: pointer,
;  230          1235  1                        new_pointer: ref vector[1,long],
;  231          1236  1                        get_vm: pointer,
;  232          1237  1                        free_vm: pointer)
;  233          1238  2 = BEGIN
;  234          1239  2
;  235          1240  2 local
;  236          1241  2         status: long,
;  237          1242  2         old_vector: ref vector[,long],
;  238          1243  2         new_vector: ref vector[,long],
;  239          1244  2         block_count: long initial(0),
;  240          1245  2         old_block: pointer;
;  241          1246  2
;  242          1247  2
;  243          1248  2 ! First we must allocate space for two vectors, with an entry for each of
;  244          1249  2 ! the blocks in the old CLI table.  The OLD_VECTOR will contain the
;  245          1250  2 ! addresses of the old CLI table blocks, while the NEW_VECTOR will contain
;  246          1251  2 ! the address of the corresponding new block.
;  247          1252  2
;  248          1253  2 status = (.get_vm)(%ref(.table[vec5_l_free]/12*4), old_vector);
;  249          1254  2 status = (.get_vm)(%ref(.table[vec5_l_free]/12*4), new_vector);
;  250          1255  2
;  251          1256  2 ! Now we can allocate space for new blocks, one for each of the old
;  252          1257  2 ! blocks.  This is done by scanning the old CLI table from beginning to
;  253          1258  2 ! end, and calling the allocation routine for each one.  As we go, the
;  254          1259  2 ! old and new block address vectors will be filled in.  Note that the first
;  255          1260  2 ! entry in the vectors will reference the primary vector block.
;  256          1261  2
;  257          1262  2 old_block = .table;
;  258          1263  3 while .old_block lssa .table + .table[vec5_l_free] do (
;  259          1264  3         increment(block_count);
;  260          1265  3         old_vector[.block_count] = .old_block;
;  261          1266  3         old_block = .old_block +
;  262          1267  3                     upgrade_5_to_6_allocate(.old_block,new_vector[.block_count],.table,.get_vm);
;  263          1268  3         if .new_vector[.block_count] eqla 0 then
;  264          1269  3                 return msg(cli$_invtab);
;  265          1270  2 );
;  266          1271  2
;  267          1272  2 ! Store the block count as the zeroth entry in both vectors, so that the
;  268          1273  2 ! vectors are self-describing.
;  269          1274  2
```

```
;  270          1275  2 old_vector[0] = new_vector[0] = .block_count;
;  271          1276  2
;  272          1277  2 ! Once the new blocks are allocated, we can fill them in.  We make a pass
;  273          1278  2 ! over the address vectors, calling a routine for each of the possible
;  274          1279  2 ! cases.
;  275          1280  2
;  276          1281  3 incru i from 1 to .block_count do (
;  277          1282  3
;  278          1283  3         bind
;  279          1284  3                 new_block = .new_vector[.i]: block[,byte];
;  280          1285  3
;  281          1286  4         (selectoneu .new_block[vec_b_type] of set
;  282          1287  4         [block_k_vector]:       upgrade_5_to_6_vector;
;  283          1288  4         [block_k_command]:      upgrade_5_to_6_command;
;  284          1289  4         [block_k_entity]:       upgrade_5_to_6_entity;
;  285          1290  3         tes) (new_block,.old_vector[.i],.new_vector,.old_vector,.get_vm);
;  286          1291  2 );
;  287          1292  2
;  288          1293  2 ! Store the address of the new table in the requested place.
;  289          1294  2
;  290          1295  2 new_pointer[0] = .new_vector[1];
;  291          1296  2
;  292          1297  2 ! Free up the memory that was allocated for the address vectors.
;  293          1298  2
;  294          1299  2 status = (.free_vm)(%ref(.table[vec5_l_free]/12*4), old_vector);
;  295          1300  2 status = (.free_vm)(%ref(.table[vec5_l_free]/12*4), new_vector);
;  296          1301  2
;  297          1302  2 return msg(cli$_upgtab);
;  298          1303  2
;  299          1304  1 END;


                                                .EXTRN  CLI$_UPGTAB

                                00FC 00000 UPGRADE_5_TO_6:
                                                .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1234
                                   5E           0C   C2 00002           SUBL2   #12, SP                 ;
                                                53   D4 00005           CLRL    BLOCK_COUNT             ; 1238
                                      04        AE   9F 00007           PUSHAB  OLD_VECTOR              ; 1253
                                   55   04      AC   D0 0000A           MOVL    TABLE, R5               ;
                              52        24      A5   0C C7 0000E       DIVL3   #12, 36(R5), R2         ;
                                                52   04 C4 00013       MULL2   #4, R2                  ;
                                      04        AE   52 D0 00016       MOVL    R2, 4(SP)               ;
                                      04        AE   9F 0001A          PUSHAB  4(SP)                   ;
                                      0C        BC   02 FB 0001D       CALLS   #2, @GET_VM             ;
                                   57           50   D0 00021          MOVL    R0, STATUS              ;
                                      08        AE   9F 00024          PUSHAB  NEW_VECTOR              ; 1254
                                      04        AE   52 D0 00027       MOVL    R2, 4(SP)               ;
                                      04        AE   9F 0002B          PUSHAB  4(SP)                   ;
                                      0C        BC   02 FB 0002E       CALLS   #2, @GET_VM             ;
                                   57           50   D0 00032          MOVL    R0, STATUS              ;
                                   56           55   D0 00035          MOVL    R5, OLD_BLOCK           ; 1262
                                   54      08   AE   D0 00038          MOVL    NEW_VECTOR, R4          ; 1267
                              52   55      24   A5   C1 0003C 1$:      ADDL3   36(R5), R5, R2          ; 1263
                                   52           56   D1 00041          CMPL    OLD_BLOCK, R2           ;
                                                26   1E 00044          BGEQU   2$                      ;
```

```
                                53   D6 00046          INCL    BLOCK_COUNT                                     ; 1264
                          04 BE43    56   D0 00048          MOVL    OLD_BLOCK, @OLD_VECTOR[BLOCK_COUNT]             ; 1265
                                0C   AC   DD 0004D          PUSHL   GET_VM                                          ; 1267
                                55   DD 00050          PUSHL   R5
                              6443   DF 00052          PUSHAL  (R4)[BLOCK_COUNT]
                                56   DD 00055          PUSHL   OLD_BLOCK
                     0000V      CF   04   FB 00057          CALLS   #4, UPGRADE_5_TO_6_ALLOCATE
                           56        50   C0 0005C          ADDL2   R0, OLD_BLOCK
                              6443   D5 0005F          TSTL    (R4)[BLOCK_COUNT]                               ; 1268
                                D8   12 00062          BNEQ    1$
                           50 00000000G  8F   D0 00064          MOVL    #CLI$_INVTAB, R0                        ; 1269
                                04 0006B          RET                                                     ;
                           64        53   D0 0006C 2$:     MOVL    BLOCK_COUNT, (R4)                       ; 1275
                     04    BE        53   D0 0006F          MOVL    BLOCK_COUNT, @OLD_VECTOR                ;
                           52        01   D0 00073          MOVL    #1, I                                   ; 1281
                                42   11 00076          BRB     8$
                           51        6442 D0 00078 3$:     MOVL    (R4)[I], R1                             ; 1284
                           50     02   A1   9A 0007C          MOVZBL  2(R1), R0                               ; 1286
                           01        50   91 00080          CMPB    R0, #1                                  ; 1287
                                07   12 00083          BNEQ    4$
                           50   0000V CF   9E 00085          MOVAB   UPGRADE_5_TO_6_VECTOR, R0               ;
                                1B   11 0008A          BRB     7$
                           02        50   91 0008C 4$:     CMPB    R0, #2                                  ; 1288
                                07   12 0008F          BNEQ    5$
                           50   0000V CF   9E 00091          MOVAB   UPGRADE_5_TO_6_COMMAND, R0              ;
                                0F   11 00096          BRB     7$
                           04        50   91 00098 5$:     CMPB    R0, #4                                  ; 1289
                                05   13 0009B          BEQL    6$
                           50        01   CE 0009D          MNEGL   #1, R0                                  ;
                                05   11 000A0          BRB     7$
                           50   0000V CF   9E 000A2 6$:     MOVAB   UPGRADE_5_TO_6_ENTITY, R0               ;
                              0C   AC   DD 000A7 7$:     PUSHL   GET_VM                                  ; 1290
                              08   AE   DD 000AA          PUSHL   OLD_VECTOR                              ;
                                54   DD 000AD          PUSHL   R4
                           10 BE42   DD 000AF          PUSHL   @OLD_VECTOR[I]
                                51   DD 000B3          PUSHL   R1
                           60        05   FB 000B5          CALLS   #5, (R0)                                ;
                                52   D6 000B8          INCL    I                                       ; 1281
                           53        52   D1 000BA 8$:     CMPL    I, BLOCK_COUNT                          ;
                                B9   1B 000BD          BLEQU   3$
                     08    BC   04   A4   D0 000BF          MOVL    4(R4), @NEW_POINTER                     ; 1295
                              04   AE   9F 000C4          PUSHAB  OLD_VECTOR                              ; 1299
                52   24    A5        0C   C7 000C7          DIVL3   #12, 36(R5), R2                         ;
                           52        04   C4 000CC          MULL2   #4, R2
                     04    AE        52   D0 000CF          MOVL    R2, 4(SP)
                              04   AE   9F 000D3          PUSHAB  4(SP)
                     10    BC        02   FB 000D6          CALLS   #2, @FREE_VM
                           57        50   D0 000DA          MOVL    R0, STATUS
                              08   AE   9F 000DD          PUSHAB  NEW_VECTOR                              ; 1300
                     04    AE        52   D0 000E0          MOVL    R2, 4(SP)
                              04   AE   9F 000E4          PUSHAB  4(SP)
                     10    BC        02   FB 000E7          CALLS   #2, @FREE_VM
                           57        50   D0 000EB          MOVL    R0, STATUS
                           50 00000000G  8F   D0 000EE          MOVL    #CLI$_UPGTAB, R0                        ; 1302
                                04 000F5          RET                                                     ; 1304

; Routine Size:  246 bytes,    Routine Base:  _CDU$CODE + 0071
```

```
;   301          1305  1 !++
;   302          1306  1 ! Description:  This routine is called to allocate a level 6 block
;   303          1307  1 !               corresponding to an old level 5 block.
;   304          1308  1 !
;   305          1309  1 ! Parameters:   old_block       By reference, the old block for which a new
;   306          1310  1 !                               one is to be allocated.
;   307          1311  1 !               new_pointer     By reference, a longword to receive the
;   308          1312  1 !                               address of the new block.
;   309          1313  1 !               table           By reference, the address of the old CLI
;   310          1314  1 !                               table.
;   311          1315  1 !               get_vm          By reference, see above.
;   312          1316  1 !
;   313          1317  1 ! Returns:      Length of the old block.
;   314          1318  1 !
;   315          1319  1 ! Notes:
;   316          1320  1 !--
;   317          1321  1
;   318          1322  1 ROUTINE upgrade_5_to_6_allocate(old_block: pointer,
;   319          1323  1                                 new_pointer: ref vector[1,long],
;   320          1324  1                                 table: pointer,
;   321          1325  1                                 get_vm: pointer)
;   322          1326  2 = BEGIN
;   323          1327  2
;   324          1328  2 local
;   325          1329  2         status: long,
;   326          1330  2         new_block: pointer,
;   327          1331  2         old_length: long;
;   328          1332  2
;   329          1333  2
;   330          1334  2 ! To allocate a new block, we must determine the type of the old block.
;   331          1335  2 ! This is not easy.  Once determined, we can allocate space for the new
;   332          1336  2 ! block and set up its type and subtype.  We must also determine the length
;   333          1337  2 ! of the old block so we can return it.
;   334          1338  2
;   335          1339  2 ! First we will determine whether the old block is a vector block.
;   336          1340  2 ! This is done by comparing its address to the addresses of the primary
;   337          1341  2 ! vector block, the verb name table, and the command block pointer table.
;   338          1342  2
;   339          1343  3 if .old_block eqla .table then (
;   340          1344  3
;   341          1345  3         ! It's the primary vector block.
;   342          1346  3
;   343          1347  3         old_length = vec5_k_length;
;   344          1348  3         status = (.get_vm)(%ref(vec_k_length), new_block);
;   345          1349  3         new_block[vec_b_type] = block_k_vector;
;   346          1350  3         new_block[vec_b_subtype] = .old_block[vec5_b_cli]+1;
;   347          1351  3
;   348          1352  3 ) else if .old_block eqla .table + .table[vec5_l_verbtbl] then (
;   349          1353  3
;   350          1354  3         ! It's the verb name table.
;   351          1355  3
;   352          1356  3         old_length = .table[vec5_l_verbend] - .table[vec5_l_verbtbl];
;   353          1357  3         status = (.get_vm)(%ref(vec_k_header_length + .old_length), new_block);
;   354          1358  3         new_block[vec_b_type] = block_k_vector;
;   355          1359  3         new_block[vec_b_subtype] = vec_k_verb;
;   356          1360  3
;   357          1361  3 ) else if .old_block eqla .table + .table[vec5_l_comdptr] then (
```

```
;  358          1362  3
;  359          1363  3          ! It's the command block pointer table.
;  360          1364  3
;  361          1365  3          old_length = .table[vec5_l_verbend] - .table[vec5_l_verbtbl];
;  362          1366  3          status = (.get_vm)(%ref(vec_k_header_length + .old_length), new_block);
;  363          1367  3          new_block[vec_b_type] = block_k_vector;
;  364          1368  3          new_block[vec_b_subtype] = vec_k_command;
;  365          1369  3
;  366          1370  3      ) else (
;  367          1371  3
;  368          1372  3          local
;  369          1373  3                  chg_length: long,
;  370          1374  3                  cmd_length: long,
;  371          1375  3                  cmdnam_length: long,
;  372          1376  3                  ent_length: long;
;  373          1377  3
;  374          1378  3          ! Because the level 5 table blocks are not self-identifying, it is
;  375          1379  3          ! difficult to determine what kind of block we have.  We will
;  376          1380  3          ! calculate the block length for each of the three other block types,
;  377          1381  3          ! and then decide which one we have.
;  378          1382  3
;  379          1383  3          chg_length =    chg5_k_length +
;  380          1384  3                          ascic_length(.old_block,.old_block[chg5_w_image]);
;  381          1385  3          cmdnam_length = ascic_length(.old_block,.old_block[cmd5_w_image]);
;  382          1386  3          !
;  383          1387  3          ! If the length is 0 (ascic_length adds 1 for the count byte) then
;  384          1388  3          ! there is no image name.  length will be 4 for routine address
;  385          1389  3          !
;  386          1390  3          if .cmdnam_length eql 1
;  387          1391  3              then cmdnam_length = 4;
;  388          1392  3          cmd_length =    cmd5_k_length + .cmdnam_length +
;  389          1393  3                          ascic_length(.old_block,.old_block[cmd5_w_outputs]);
;  390          1394  3          ent_length =    ent5_k_length +
;  391          1395  3                          ascic_length(.old_block,.old_block[ent5_w_name]) +
;  392          1396  3                          ascic_length(.old_block,.old_block[ent5_w_label]) +
;  393          1397  3                          ascic_length(.old_block,.old_block[ent5_w_defval]) +
;  394          1398  3                          ascic_length(.old_block,.old_block[ent5_w_prompt]);
;  395          1399  3
;  396          1400  4          if .chg_length eqlu .old_block[chg5_b_size] then (
;  397          1401  4
;  398          1402  4                  ! We have a change block.  This becomes a command block in
;  399          1403  4                  ! the new table.
;  400          1404  4
;  401          1405  4                  old_length = .chg_length;
;  402          1406  4                  status = (.get_vm)(%ref(cmd_k_length + 4+1), new_block);
;  403          1407  4                  new_block[cmd_b_type] = block_k_command;
;  404          1408  4                  new_block[cmd_b_subtype] = cmd_k_syntax;
;  405          1409  4
;  406          1410  4          ) else if .cmd_length eqlu .old_block[cmd5_b_size] then (
;  407          1411  4
;  408          1412  4                  ! We have a command block.
;  409          1413  4
;  410          1414  4                  old_length = .cmd_length;
;  411          1415  4                  status = (.get_vm)(%ref(cmd_k_length + 4+1 + 12), new_block);
;  412          1416  4                  new_block[cmd_b_type] = block_k_command;
;  413          1417  4                  new_block[cmd_b_subtype] = cmd_k_verb;
;  414          1418  4
```

```
;  415     1419  4          ) else if .ent_length eqlu .old_block[ent5_b_size] then (
;  416     1420  4
;  417     1421  4                  ! We have an entity block.
;  418     1422  4
;  419     1423  4                  old_length = .ent_length;
;  420     1424  4                  status = (.get_vm)(%ref(ent_k_length + 16 + 16 + 64 + 16), new_block);
;  421     1425  4                  new_block[ent_b_type] = block_k_entity;
;  422     1426  4
;  423     1427  4          ) else (
;  424     1428  4
;  425     1429  4                  ! Oh God, who knows what this block is?
;  426     1430  4
;  427     1431  4                  old_length = 0;
;  428     1432  4                  new_block = 0;
;  429     1433  3          );
;  430     1434  2 );
;  431     1435  2
;  432     1436  2 ! Store the address of the new block where requested, and return the length
;  433     1437  2 ! of the old block;
;  434     1438  2
;  435     1439  2 new_pointer[0] = .new_block;
;  436     1440  2 return .old_length;
;  437     1441  2
;  438     1442  1 END;


                                01FC 00000 UPGRADE_5_TO_6_ALLOCATE:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                ; 1322
                     5E         08   C2 00002      SUBL2   #8, SP                           ;
                     54   04    AC   D0 00005      MOVL    OLD_BLOCK, R4                    ; 1343
                     53   0C    AC   D0 00009      MOVL    TABLE, R3                        ;
                     53         54   D1 0000D      CMPL    R4, R3                           ;
                                21   12 00010      BNEQ    1$                               ;
                     52         3C   D0 00012      MOVL    #60, OLD_LENGTH                  ; 1347
                          04    AE   9F 00015      PUSHAB  NEW_BLOCK                        ; 1348
                04   AE         14   D0 00018      MOVL    #20, 4(SP)                       ;
                          04    AE   9F 0001C      PUSHAB  4(SP)                            ;
                10   BC         02   FB 0001F      CALLS   #2, @GET_VM                      ;
                     51   04    AE   D0 00023      MOVL    NEW_BLOCK, R1                    ; 1349
                02   A1         01   90 00027      MOVB    #1, 2(R1)                        ;
03   A1         2A   A4         01   81 0002B      ADDB3   #1, 42(R4), 3(R1)                ; 1350
                                54   11 00031      BRB     3$                               ; 1343
     51              53   0C    A3   C1 00033 1$:  ADDL3   12(R3), R3, R1                  ; 1352
                     51         54   D1 00038      CMPL    R4, R1                           ;
                                21   12 0003B      BNEQ    2$                               ;
     52         10   A3         0C   A3   C3 0003D      SUBL3   12(R3), 16(R3), OLD_LENGTH   ; 1356
                          04    AE   9F 00043      PUSHAB  NEW_BLOCK                        ; 1357
                04   AE         08   A2   9E 00046      MOVAB   8(R2), 4(SP)             ;
                          04    AE   9F 0004B      PUSHAB  4(SP)                            ;
                10   BC         02   FB 0004E      CALLS   #2, @GET_VM                      ;
                     51   04    AE   D0 00052      MOVL    NEW_BLOCK, R1                    ; 1358
                02   A1   0301 8F   B0 00056      MOVW    #769, 2(R1)                      ;
                                29   11 0005C      BRB     3$                               ; 1352
     51              53   1C    A3   C1 0005E 2$:  ADDL3   28(R3), R3, R1                  ; 1361
```

```
                        51              54  D1 00063          CMPL    R4, R1                                          ;
                                        22  12 00066          BNEQ    4$                                              ;
                52      10  A3      0C  A3  C3 00068          SUBL3   12(R3), 16(R3), OLD_LENGTH                       ; 1365
                                    04  AE  9F 0006E          PUSHAB  NEW_BLOCK                                       ; 1366
                        04  AE      08  A2  9E 00071          MOVAB   8(R2), 4(SP)                                    ;
                                    04  AE  9F 00076          PUSHAB  4(SP)                                           ;
                        10  BC          02  FB 00079          CALLS   #2, @GET_VM                                     ;
                        51          04  AE  D0 0007D          MOVL    NEW_BLOCK, R1                                   ; 1367
                        02  A1  0401        8F  B0 00081      MOVW    #1025, 2(R1)                                    ;
                                        0142 31  00087 3$:     BRW     24$                                             ; 1361
                        51          02  A4  32 0008A 4$:     CVTWL   2(R4), R1                                       ; 1384
                        08              51  B1 0008E          CMPW    R1, #8                                          ;
                                        0F  1B 00091          BLEQU   5$                                              ;
                6144    01              03  0C 00093          PROBER  #3, #1, (R1)[R4]                                ;
                                        08  13 00098          BEQL    5$                                              ;
                        51          6144    9A 0009A          MOVZBL  (R1)[R4], R1                                    ;
                                        51  D6 0009E          INCL    R1                                              ;
                                        02  11 000A0          BRB     6$                                              ;
                                        51  D4 000A2 5$:      CLRL    R1                                              ;
                        51              09  C0 000A4 6$:      ADDL2   #9, CHG_LENGTH                                  ; 1383
                        56          04  A4  32 000A7          CVTWL   4(R4), R6                                       ; 1385
                                        57  D4 000AB          CLRL    R7                                              ;
                        08              56  B1 000AD          CMPW    R6, #8                                          ;
                                        04  1A 000B0          BGTRU   7$                                              ;
                                        57  D6 000B2          INCL    R7                                              ;
                                        0F  11 000B4          BRB     8$                                              ;
                6644    01              03  0C 000B6 7$:      PROBER  #3, #1, (R6)[R4]                                ;
                                        08  13 000BB          BEQL    8$                                              ;
                        53          6644    9A 000BD          MOVZBL  (R6)[R4], CMDNAM_LENGTH                         ;
                                        53  D6 000C1          INCL    CMDNAM_LENGTH                                   ;
                                        02  11 000C3          BRB     9$                                              ;
                                        53  D4 000C5 8$:      CLRL    CMDNAM_LENGTH                                   ;
                        01              53  D1 000C7 9$:      CMPL    CMDNAM_LENGTH, #1                               ; 1390
                                        03  12 000CA          BNEQ    10$                                             ;
                        53              04  D0 000CC          MOVL    #4, CMDNAM_LENGTH                               ; 1391
                        55      0A      A4  32 000CF 10$:     CVTWL   10(R4), R5                                      ; 1393
                        08              55  B1 000D3          CMPW    R5, #8                                          ;
                                        0F  1B 000D6          BLEQU   11$                                             ;
                6544    01              03  0C 000D8          PROBER  #3, #1, (R5)[R4]                                ;
                                        08  13 000DD          BEQL    11$                                             ;
                        55          6544    9A 000DF          MOVZBL  (R5)[R4], R5                                    ;
                                        55  D6 000E3          INCL    R5                                              ;
                                        02  11 000E5          BRB     12$                                             ;
                                        55  D4 000E7 11$:      CLRL    R5                                              ;
                        58      10  A543    9E 000E9 12$:     MOVAB   16(R5)[CMDNAM_LENGTH], CMD_LENGTH               ; 1392
                        0F              57  E8 000EE          BLBS    R7, 13$                                         ; 1395
                6644    01              03  0C 000F1          PROBER  #3, #1, (R6)[R4]                                ;
                                        08  13 000F6          BEQL    13$                                             ;
                        53          6644    9A 000F8          MOVZBL  (R6)[R4], R3                                    ;
                                        53  D6 000FC          INCL    R3                                              ;
                                        02  11 000FE          BRB     14$                                             ;
                                        53  D4 00100 13$:     CLRL    R3                                              ;
                        55      06      A4  32 00102 14$:     CVTWL   6(R4), R5                                       ; 1396
                        08              55  B1 00106          CMPW    R5, #8                                          ;
                                        0F  1B 00109          BLEQU   15$                                             ;
                6544    01              03  0C 0010B          PROBER  #3, #1, (R5)[R4]                                ;
                                        08  13 00110          BEQL    15$                                             ;
```

```
                     55           6544  9A 00112       MOVZBL  (R5)[R4], R5                     ;
                                    55  D6 00116       INCL    R5                               ;
                                    02  11 00118       BRB     16$                              ;
                                    55  D4 0011A 15$:  CLRL    R5                               ;
                     53             55  C0 0011C 16$:  ADDL2   R5, R3                           ; 1395
                     55        08   A4  32 0011F       CVTWL   8(R4), R5                        ; 1397
                     08             55  B1 00123       CMPW    R5, #8                           ;
                                    0F  1B 00126       BLEQU   17$                              ;
          6544       01             03  0C 00128       PROBER  #3, #1, (R5)[R4]                 ;
                                    08  13 0012D       BEQL    17$                              ;
                     55           6544  9A 0012F       MOVZBL  (R5)[R4], R5                     ;
                                    55  D6 00133       INCL    R5                               ;
                                    02  11 00135       BRB     18$                              ;
                                    55  D4 00137 17$:  CLRL    R5                               ;
                     55             53  C0 00139 18$:  ADDL2   R3, R5                           ; 1396
                     53        0E   A4  32 0013C       CVTWL   14(R4), R3                       ; 1398
                     08             53  B1 00140       CMPW    R3, #8                           ;
                                    0F  1B 00143       BLEQU   19$                              ;
          6344       01             03  0C 00145       PROBER  #3, #1, (R3)[R4]                 ;
                                    08  13 0014A       BEQL    19$                              ;
                     53           6344  9A 0014C       MOVZBL  (R3)[R4], R3                     ;
                                    53  D6 00150       INCL    R3                               ;
                                    02  11 00152       BRB     20$                              ;
                                    53  D4 00154 19$:  CLRL    R3                               ;
                     53        14 A345  9E 00156 20$:  MOVAB   20(R3)[R5], ENT_LENGTH           ; 1397
 51           64               08   00  ED 0015B       CMPZV   #0, #8, (R4), CHG_LENGTH         ; 1400
                                    1D  12 00160       BNEQ    21$                              ;
                     52             51  D0 00162       MOVL    CHG_LENGTH, OLD_LENGTH           ; 1405
                               04   AE  9F 00165       PUSHAB  NEW_BLOCK                        ; 1406
               04    AE             25  D0 00168       MOVL    #37, 4(SP)                       ;
                               04   AE  9F 0016C       PUSHAB  4(SP)                            ;
               10    BC             02  FB 0016F       CALLS   #2, @GET_VM                      ;
                     51        04   AE  D0 00173       MOVL    NEW_BLOCK, R1                    ; 1407
               02    A1      0202   8F  B0 00177       MOVW    #514, 2(R1)                      ;
                                    4D  11 0017D       BRB     24$                              ; 1400
 58           64               08   00  ED 0017F 21$:  CMPZV   #0, #8, (R4), CMD_LENGTH         ; 1410
                                    1D  12 00184       BNEQ    22$                              ;
                     52             58  D0 00186       MOVL    CMD_LENGTH, OLD_LENGTH           ; 1414
                               04   AE  9F 00189       PUSHAB  NEW_BLOCK                        ; 1415
               04    AE             31  D0 0018C       MOVL    #49, 4(SP)                       ;
                               04   AE  9F 00190       PUSHAB  4(SP)                            ;
               10    BC             02  FB 00193       CALLS   #2, @GET_VM                      ;
                     51        04   AE  D0 00197       MOVL    NEW_BLOCK, R1                    ; 1416
               02    A1      0102   8F  B0 0019B       MOVW    #258, 2(R1)                      ;
                                    29  11 001A1       BRB     24$                              ; 1410
 53      01   A4               08   00  ED 001A3 22$:  CMPZV   #0, #8, 1(R4), ENT_LENGTH        ; 1419
                                    1C  12 001A9       BNEQ    23$                              ;
                     52             53  D0 001AB       MOVL    ENT_LENGTH, OLD_LENGTH           ; 1423
                               04   AE  9F 001AE       PUSHAB  NEW_BLOCK                        ; 1424
               04    AE        8E   8F  9A 001B1       MOVZBL  #142, 4(SP)                      ;
                               04   AE  9F 001B6       PUSHAB  4(SP)                            ;
               10    BC             02  FB 001B9       CALLS   #2, @GET_VM                      ;
                     50        04   AE  D0 001BD       MOVL    NEW_BLOCK, R0                    ; 1425
               02    A0             04  90 001C1       MOVB    #4, 2(R0)                        ;
                                    05  11 001C5       BRB     24$                              ; 1419
                                    52  D4 001C7 23$:  CLRL    OLD_LENGTH                       ; 1431
                               04   AE  D4 001C9       CLRL    NEW_BLOCK                        ; 1432
```

```
                08  BC      04  AE  DO 001CC 24$:    MOVL    NEW_BLOCK, @NEW_POINTER                 ; 1439
                    50          52  DO 001D1         MOVL    OLD_LENGTH, R0                          ; 1440
                                    04 001D4         RET                                             ; 1442

; Routine Size:  469 bytes,    Routine Base:  _CDU$CODE + 0167
```

```
;   440          1443  1 !++
;   441          1444  1 ! Description:  This routine is called to fill in a new vector block from
;   442          1445  1 !               the corresponding old block.
;   443          1446  1 !
;   444          1447  1 ! Parameters:   new_block       By reference, the new block to be filled in.
;   445          1448  1 !                               The type and subtype are already present.
;   446          1449  1 !               old_block       By reference, the corresponding old block.
;   447          1450  1 !               new_vector      By reference, the vector of new block
;   448          1451  1 !                               addresses.  Zeroth entry is block count.
;   449          1452  1 !               old_vector      By reference, the vector of old block
;   450          1453  1 !                               addresses.
;   451          1454  1 !               get_vm          By reference, see above.
;   452          1455  1 !
;   453          1456  1 ! Returns:      Nothing.
;   454          1457  1 !
;   455          1458  1 ! Notes:
;   456          1459  1 !--
;   457          1460  1
;   458          1461  1 ROUTINE upgrade_5_to_6_vector(new_block: pointer,
;   459          1462  1                               old_block: pointer,
;   460          1463  1                               new_vector: ref vector[,long],
;   461          1464  1                               old_vector: ref vector[,long],
;   462          1465  1                               get_vm: pointer)                : novalue
;   463          1466  1
;   464          1467  2 = BEGIN
;   465          1468  2
;   466          1469  2 local
;   467          1470  2         entry_count: long;
;   468          1471  2
;   469          1472  2
;   470          1473  2 ! Select on the subtype of the vector block.
;   471          1474  2
;   472          1475  2 selectoneu .new_block[vec_b_subtype] of set
;   473          1476  2
;   474          1477  2 [vec_k_dcl,
;   475          1478  2  vec_k_mcr]:
;   476          1479  2
;   477          1480  2         ! We have the primary vector block.  Fill in each field from the
;   478          1481  2         ! old primary vector block.  We cannot determine the overall table
;   479          1482  2         ! size.
;   480          1483  2
;   481          1484  3         (new_block[vec_w_size] = vec_k_length;
;   482          1485  3         new_block[vec_w_flags] = 0;
;   483          1486  3         new_block[vec_b_strlvl] = 6;
;   484          1487  3         new_block[vec_w_tro_count] = 2;
;   485          1488  3         new_block[vec_l_verbtbl] = new_block_address(.old_block+.old_block[vec5_l_verbtbl]) - .new_block;
;   486          1489  3         new_block[vec_l_comdptr] = new_block_address(.old_block+.old_block[vec5_l_comdptr]) - .new_block;
;   487          1490  2         new_block[vec_l_table_size] = 0;);
;   488          1491  2
;   489          1492  2 [vec_k_verb]:
;   490          1493  2
;   491          1494  2         ! We have the verb name table.  Initialize the new block header.
;   492          1495  2         ! Then copy the verb name entries, converting them from blank
;   493          1496  2         ! padded with bit 7 set to zero padded with bit 7 clear.
;   494          1497  2
;   495          1498  3         (bind
;   496          1499  3                 new_verb_names = .new_block + vec_k_header_length: vector[,long],
```

```
;  497          1500  3                 old_primary = .old_vector[1]: block[,byte],
;  498          1501  3                 old_verb_names = .old_block: vector[,long];
;  499          1502  3
;  500          1503  3             entry_count = (.old_primary[vec5_l_verbend] - .old_primary[vec5_l_verbtbl]) / 4;
;  501          1504  3             new_block[vec_w_size] = vec_k_header_length + .entry_count*4;
;  502          1505  3             new_block[vec_w_flags] = 0;
;  503          1506  3             new_block[vec_w_tro_count] = 0;
;  504          1507  3
;  505          1508  4             incr i from 0 to .entry_count-1 do (
;  506          1509  4
;  507          1510  4                 bind
;  508          1511  4                         old_name = old_verb_names[.i]: vector[4,byte],
;  509          1512  4                         new_name = new_verb_names[.i]: vector[4,byte];
;  510          1513  4
;  511          1514  4                 new_name[0] = .old_name[0] and %x'7f';
;  512          1515  4                 new_name[1] = (if .old_name[1] eqlu ' ' then %x'00' else .old_name[1]);
;  513          1516  4                 new_name[2] = (if .old_name[2] eqlu ' ' then %x'00' else .old_name[2]);
;  514          1517  4                 new_name[3] = (if .old_name[3] eqlu ' ' then %x'00' else .old_name[3]);
;  515          1518  2             ););
;  516          1519  2
;  517          1520  2 [vec_k_command]:
;  518          1521  2
;  519          1522  2             ! We have the command block pointer table.  Initialize the new
;  520          1523  2             ! block header.  Then copy the pointers, translating them to point
;  521          1524  2             ! at the new command blocks.
;  522          1525  2
;  523          1526  3             (bind
;  524          1527  3                     new_command_block_pointers = .new_block + vec_k_header_length: vector[,long],
;  525          1528  3                     old_primary = .old_vector[1]: block[,byte],
;  526          1529  3                     old_command_block_pointers = .old_block: vector[,long];
;  527          1530  3
;  528          1531  3             entry_count = (.old_primary[vec5_l_verbend] - .old_primary[vec5_l_verbtbl]) / 4;
;  529          1532  3             new_block[vec_w_size] = vec_k_header_length + .entry_count*4;
;  530          1533  3             new_block[vec_w_flags] = 0;
;  531          1534  3             new_block[vec_w_tro_count] = .entry_count;
;  532          1535  3
;  533          1536  3             incr i from 0 to .entry_count-1 do
;  534          1537  4                 new_command_block_pointers[.i] = (if .old_command_block_pointers[.i] eqlu 0 then 0 else
;  535          1538  4                         new_block_address(old_command_block_pointers[.i+1]+.old_command_block_pointers[.i])
;  536          1539  3                                 .new_vector[1]);
;  537          1540  2             );
;  538          1541  2
;  539          1542  2 tes;
;  540          1543  2
;  541          1544  2 return;
;  542          1545  2
;  543          1546  1 END;


                                00FC 00000 UPGRADE_5_TO_6_VECTOR:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1461
                        50      04   AC  D0 00002  MOVL    NEW_BLOCK, R0                   ; 1475
                        51      03   A0  9A 00006  MOVZBL  3(R0), R1                      ;
                                     63  13 0000A  BEQL    7$                              ; 1477
```

```
                                02              51  91 0000C            CMPB    R1, #2                                          ;
                                                5E  1A 0000F            BGTRU   7$                                              ;
                                60              14  B0 00011            MOVW    #20, (R0)                                       ; 1484
                            04  A0 00020000     8F  D0 00014            MOVL    #131072, 4(R0)                                  ; 1485
                            04  A0              06  90 0001C            MOVB    #6, 4(R0)                                       ; 1486
                                52      08      AC  D0 00020            MOVL    OLD_BLOCK, R2                                   ; 1488
                        53      52      0C      A2  C1 00024            ADDL3   12(R2), R2, R3                                  ;
                                                51  D4 00029            CLRL    I                                               ;
                                                0E  11 0002B            BRB     2$                                              ;
                            10  BC41            53  D1 0002D 1$:        CMPL    R3, @OLD_VECTOR[I]                              ;
                                                07  12 00032            BNEQ    2$                                              ;
                                51      0C  BC41    D0 00034            MOVL    @NEW_VECTOR[I], R1                              ;
                                                08  11 00039            BRB     3$                                              ;
                        ED      51      10      BC  F3 0003B 2$:        AOBLEQ  @OLD_VECTOR, I, 1$                              ;
                                51              01  CE 00040            MNEGL   #1, R1                                          ;
                    08  A0      51              50  C3 00043 3$:        SUBL3   R0, R1, 8(R0)                                   ;
                                52      1C      A2  C0 00048            ADDL2   28(R2), R2                                      ; 1489
                                                51  D4 0004C            CLRL    I                                               ;
                                                0E  11 0004E            BRB     5$                                              ;
                            10  BC41            52  D1 00050 4$:        CMPL    R2, @OLD_VECTOR[I]                              ;
                                                07  12 00055            BNEQ    5$                                              ;
                                51      0C  BC41    D0 00057            MOVL    @NEW_VECTOR[I], R1                              ;
                                                08  11 0005C            BRB     6$                                              ;
                        ED      51      10      BC  F3 0005E 5$:        AOBLEQ  @OLD_VECTOR, I, 4$                              ;
                                51              01  CE 00063            MNEGL   #1, R1                                          ;
                    0C  A0      51              50  C3 00066 6$:        SUBL3   R0, R1, 12(R0)                                  ;
                                        10      A0  D4 0006B            CLRL    16(R0)                                          ; 1490
                                                04  0006E            RET                                                     ; 1475
                                03              51  91 0006F 7$:        CMPB    R1, #3                                          ; 1492
                                                6F  12 00072            BNEQ    16$                                             ;
                                51      10      AC  D0 00074            MOVL    OLD_VECTOR, R1                                  ; 1500
                                51      04      A1  D0 00078            MOVL    4(R1), R1                                       ;
                        51      10      A1      0C  A1  C3 0007C        SUBL3   12(R1), 16(R1), R1                              ; 1503
                        56      51              04  C7 00082            DIVL3   #4, R1, ENTRY_COUNT                             ;
                        51      56              02  78 00086            ASHL    #2, ENTRY_COUNT, R1                             ; 1504
                        60      51              08  A1 0008A            ADDW3   #8, R1, (R0)                                    ;
                                        04      A0  D4 0008E            CLRL    4(R0)                                           ; 1505
                                52              01  CE 00091            MNEGL   #1, I                                           ; 1508
                                                48  11 00094            BRB     15$                                             ;
                                51      08  BC42  DE 00096 8$:         MOVAL   @OLD_BLOCK[I], R1                               ; 1511
                                53      08  A042  DE 0009B            MOVAL   8(R0)[I], R3                                    ; 1512
                54      61              07          00  EF 000A0            EXTZV   #0, #7, (R1), R4                                ; 1514
                                63              54  90 000A5            MOVB    R4, (R3)                                        ;
                                20      01      A1  91 000A8            CMPB    1(R1), #32                                      ; 1515
                                                04  12 000AC            BNEQ    9$                                              ;
                                                54  D4 000AE            CLRL    R4                                              ;
                                                04  11 000B0            BRB     10$                                             ;
                                54      01      A1  9A 000B2 9$:        MOVZBL  1(R1), R4                                       ;
                            01  A3              54  90 000B6 10$:       MOVB    R4, 1(R3)                                       ;
                                20      02      A1  91 000BA            CMPB    2(R1), #32                                      ; 1516
                                                04  12 000BE            BNEQ    11$                                             ;
                                                54  D4 000C0            CLRL    R4                                              ;
                                                04  11 000C2            BRB     12$                                             ;
                                54      02      A1  9A 000C4 11$:       MOVZBL  2(R1), R4                                       ;
                            02  A3              54  90 000C8 12$:       MOVB    R4, 2(R3)                                       ;
                                20      03      A1  91 000CC            CMPB    3(R1), #32                                      ; 1517
                                                04  12 000D0            BNEQ    13$                                             ;
```

```
                                 51   D4 000D2          CLRL    R1                                                      ;
                                 04   11 000D4          BRB     14$                                                     ;
                           51    03   A1 9A 000D6 13$:  MOVZBL  3(R1), R1                                               ;
                        03 A3          51 90 000DA 14$:  MOVB    R1, 3(R3)                                               ;
                     B4    52          56 F2 000DE 15$:  AOBLSS  ENTRY_COUNT, I, 8$                                      ; 1508
                                       04 000E2          RET                                                             ; 1475
                              04       51 91 000E3 16$:  CMPB    R1, #4                                                  ; 1520
                                       64 12 000E6       BNEQ    24$                                                     ;
                              55 10    AC D0 000E8       MOVL    OLD_VECTOR, R5                                          ; 1528
                              51 04    A5 D0 000EC       MOVL    4(R5), R1                                               ;
                     51       10 A1    0C A1 C3 000F0    SUBL3   12(R1), 16(R1), R1                                      ; 1531
                     56          51    04 C7 000F6       DIVL3   #4, R1, ENTRY_COUNT                                     ;
                     51          56    02 78 000FA       ASHL    #2, ENTRY_COUNT, R1                                     ; 1532
                     60          51    08 A1 000FE       ADDW3   #8, R1, (R0)                                            ;
                                    04 A0 B4 00102       CLRW    4(R0)                                                   ; 1533
                                 06 A0    56 B0 00105    MOVW    ENTRY_COUNT, 6(R0)                                      ; 1534
                                 52       01 CE 00109    MNEGL   #1, I                                                   ; 1536
                                          3A 11 0010C    BRB     23$                                                     ;
                                 54 08 BC42  D0 0010E 17$:  MOVL   @OLD_BLOCK[I], R4                                    ; 1537
                                          04 12 00113    BNEQ    18$                                                     ;
                                          51 D4 00115    CLRL    R1                                                      ;
                                          2A 11 00117    BRB     22$                                                     ;
                                 53 08 BC42  DE 00119 18$:  MOVAL  @OLD_BLOCK[I], R3                                    ; 1538
                                          51 D4 0011E    CLRL    I                                                       ;
                                          12 11 00120    BRB     20$                                                     ;
                                    57 04 A443  9E 00122 19$:  MOVAB   4(R4)[R3], R7                                    ;
                                 6541       57 D1 00127    CMPL    R7, (R5)[I]                                             ;
                                          07 12 0012B    BNEQ    20$                                                     ;
                                 51 0C BC41  D0 0012D    MOVL    @NEW_VECTOR[I], R1                                      ;
                                          07 11 00132    BRB     21$                                                     ;
                              EA          51 65 F3 00134 20$:  AOBLEQ  (R5), I, 19$                                    ;
                                          51 01 CE 00138    MNEGL   #1, R1                                               ;
                                 54 0C    AC D0 0013B 21$:  MOVL    NEW_VECTOR, R4                                       ; 1539
                                 51 04    A4 C2 0013F       SUBL2   4(R4), R1                                            ;
                              08 A042        51 D0 00143 22$:  MOVL    R1, 8(R0)[I]                                    ; 1537
                              C2             52 F2 00148 23$:  AOBLSS  ENTRY_COUNT, I, 17$                             ;
                                             04 0014C 24$:  RET                                                          ; 1546

; Routine Size:  333 bytes,    Routine Base:  _CDU$CODE + 033C
```

```
;  545          1547  1 !++
;  546          1548  1 ! Description:  This routine is called to fill in a new command block from
;  547          1549  1 !               the corresponding old block.
;  548          1550  1 !
;  549          1551  1 ! Parameters:   new_block       By reference, the new block to be filled in.
;  550          1552  1 !                               The type and subtype are already present.
;  551          1553  1 !               old_block       By reference, the corresponding old block.
;  552          1554  1 !               new_vector      By reference, the vector of new block
;  553          1555  1 !                               addresses.  Zeroth entry is block count.
;  554          1556  1 !               old_vector      By reference, the vector of old block
;  555          1557  1 !                               addresses.
;  556          1558  1 !               get_vm          By reference, see above.
;  557          1559  1 !
;  558          1560  1 ! Returns:      Nothing.
;  559          1561  1 !
;  560          1562  1 ! Notes:
;  561          1563  1 !--
;  562          1564  1
;  563          1565  1 ROUTINE upgrade_5_to_6_command(new_block: pointer,
;  564          1566  1                                old_block: pointer,
;  565          1567  1                                new_vector: ref vector[,long],
;  566          1568  1                                old_vector: ref vector[,long],
;  567          1569  1                                get_vm: pointer)                : novalue
;  568          1570  1
;  569          1571  2 = BEGIN
;  570          1572  2
;  571          1573  2 local
;  572          1574  2         variable_ptr: pointer;
;  573          1575  2
;  574          1576  2
;  575          1577  2 ! Set up to add information to the variable portion of the new block.
;  576          1578  2
;  577          1579  2 variable_ptr = new_block[cmd_z_variable];
;  578          1580  2
;  579          1581  2 ! Split up depending upon whether we are to build a verb command block
;  580          1582  2 ! or a syntax change command block.
;  581          1583  2
;  582          1584  3 if .new_block[cmd_b_subtype] eqlu cmd_k_verb then (
;  583          1585  3
;  584          1586  3         ! We are building a verb command block.  Fill in the new block from
;  585          1587  3         ! the old one.
;  586          1588  3
;  587          1589  3         new_block[cmd_w_flags] = 0;
;  588          1590  3         new_block[cmd_v_abbrev] = .old_block[cmd5_v_abrev];
;  589          1591  3         new_block[cmd_v_nostat] = .old_block[cmd5_v_nostat];
;  590          1592  3         new_block[cmd_v_foreign] = .old_block[cmd5_v_foreign];
;  591          1593  3         new_block[cmd_v_immed] = .old_block[cmd5_v_immed];
;  592          1594  3         new_block[cmd_v_mcrparse] = .old_block[cmd5_v_mcrparse];
;  593          1595  3         new_block[cmd_v_parms] = new_block[cmd_v_quals] = new_block[cmd_v_disallows] = true;
;  594          1596  3         new_block[cmd_w_tro_count] = 3;
;  595          1597  4         new_block[cmd_l_parms] = (if .old_block[cmd5_w_parms] eqlu 0 then 0 else
;  596          1598  3                 new_block_address(.old_block+.old_block[cmd5_w_parms]) - .new_vector[1]);
;  597          1599  4         new_block[cmd_l_quals] = (if .old_block[cmd5_w_quals] eqlu 0 then 0 else
;  598          1600  3                 new_block_address(.old_block+.old_block[cmd5_w_quals]) - .new_vector[1]);
;  599          1601  3         new_block[cmd_l_disallow] = 0;
;  600          1602  3         new_block[cmd_b_handler] = (if .old_block[cmd5_w_image] eqlu 0 then 0 else cmd_k_user);
;  601          1603  3         new_block[cmd_v_minparm] = .old_block[cmd5_v_minparm];
```

```
;  602          1604  3          new_block[cmd_v_maxparm] = .old_block[cmd5_v_maxparm];
;  603          1605  3          new_block[cmd_b_verbtyp] = 0;
;  604          1606  3          new_block[cmd_w_name] = 0;
;  605          1607  3          if .old_block[cmd5_w_image] eglu 0 then
;  606          1608  3                  new_block[cmd_w_image] = 0
;  607          1609  4          else (
;  608          1610  4                  bind
;  609          1611  4                          routine_longword = .old_block + .old_block[cmd5_w_image]: long;
;  610          1612  4
;  611          1613  4                  new_block[cmd_w_image] = .variable_ptr - .new_block;
;  612          1614  4                  variable_ptr[0,0,32,0] = .routine_longword;
;  613          1615  4                  variable_ptr[4,0,8,0] = 0;
;  614          1616  4                  variable_ptr = .variable_ptr + 4+1;
;  615          1617  3          );
;  616          1618  3          if .old_block[cmd5_w_outputs] eglu 0 then
;  617          1619  3                  new_block[cmd_w_outputs] = 0
;  618          1620  4          else (
;  619          1621  4                  bind
;  620          1622  4                          outputs_list = .old_block + .old_block[cmd5_w_outputs]: vector[,byte];
;  621          1623  4
;  622          1624  4                  new_block[cmd_w_outputs] = .variable_ptr - .new_block;
;  623          1625  4                  ch$move(1+.outputs_list[0],outputs_list[0], .variable_ptr);
;  624          1626  4                  variable_ptr = .variable_ptr + 1+.outputs_list[0];
;  625          1627  3          );
;  626          1628  3          new_block[cmd_w_prefix] = 0;
;  627          1629  3
;  628          1630  3  ) else (
;  629          1631  3
;  630          1632  3          ! We are building a syntax change command block.  Fill in the new
;  631          1633  3          ! block from the old one as much as possible.
;  632          1634  3
;  633          1635  3          new_block[cmd_w_flags] = 0;
;  634          1636  3          new_block[cmd_v_parms] = .old_block[chg5_v_parms];
;  635          1637  3          new_block[cmd_v_quals] = .old_block[chg5_v_quals];
;  636          1638  3          new_block[cmd_v_disallows] = true;
;  637          1639  3          new_block[cmd_w_tro_count] = 3;
;  638          1640  4          new_block[cmd_l_parms] = (if .old_block[chg5_w_parms] eglu 0 then 0 else
;  639          1641  3                  new_block_address(.old_block+.old_block[chg5_w_parms]) - .new_vector[1]);
;  640          1642  4          new_block[cmd_l_quals] = (if .old_block[chg5_w_quals] eglu 0 then 0 else
;  641          1643  3                  new_block_address(.old_block+.old_block[chg5_w_quals]) - .new_vector[1]);
;  642          1644  3          new_block[cmd_l_disallow] = 0;
;  643          1645  4          new_block[cmd_b_handler] = (if not .old_block[chg5_v_image] then cmd_k_same
;  644          1646  4                                      else if .old_block[chg5_w_image] eglu 0 then cmd_k_none
;  645          1647  3                                      else cmd_k_user);
;  646          1648  3          new_block[cmd_v_minparm] = .old_block[chg5_v_minparm];
;  647          1649  3          new_block[cmd_v_maxparm] = .old_block[chg5_v_maxparm];
;  648          1650  3          new_block[cmd_b_verbtyp] = 0;
;  649          1651  3          new_block[cmd_w_name] = 0;
;  650          1652  3          if .old_block[chg5_w_image] eglu 0 then
;  651          1653  3                  new_block[cmd_w_image] = 0
;  652          1654  4          else (
;  653          1655  4                  bind
;  654          1656  4                          routine_longword = .old_block + .old_block[chg5_w_image]: long;
;  655          1657  4
;  656          1658  4                  new_block[cmd_w_image] = .variable_ptr - .new_block;
;  657          1659  4                  variable_ptr[0,0,32,0] = .routine_longword;
;  658          1660  4                  variable_ptr[4,0,8,0] = 0;
```

```
;  659        1661  4                     variable_ptr = .variable_ptr + 4+1;
;  660        1662  3              );
;  661        1663  3              new_block[cmd_w_outputs] = 0;
;  662        1664  3              new_block[cmd_w_prefix] = 0;
;  663        1665  2  );
;  664        1666  2
;  665        1667  2  ! Now we can fill in the final size of the new block.
;  666        1668  2
;  667        1669  2  new_block[cmd_w_size] = .variable_ptr - .new_block;
;  668        1670  2
;  669        1671  2  return;
;  670        1672  2
;  671        1673  1  END;



                                  01FC 00000 UPGRADE_5_TO_6_COMMAND:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8              ; 1565
                  57       04   AC   D0 00002      MOVL    NEW_BLOCK, R7                          ; 1579
                  56       20   A7   9E 00006      MOVAB   32(R7), VARIABLE_PTR                   ;
                  51       04   A7   9E 0000A      MOVAB   4(R7), R1                              ; 1589
                  50       08   AC   D0 0000E      MOVL    OLD_BLOCK, R0                          ; 1590
                  54       04   A0   9E 00012      MOVAB   4(R0), R4                              ; 1602
                  01       03   A7   91 00016      CMPB    3(R7), #1                              ; 1584
                                03   13 0001A      BEQL    1$                                     ;
                              010D   31 0001C      BRW     17$                                    ;
                                61   B4 0001F 1$:  CLRW    (R1)                                   ; 1589
                  52       03   A0   9E 00021      MOVAB   3(R0), R2                              ; 1590
   61     01      00            62   F0 00025      INSV    (R2), #0, #1, (R1)                     ;
   53     62      01            01   EF 0002A      EXTZV   #1, #1, (R2), R3                       ; 1591
   61     01      01            53   F0 0002F      INSV    R3, #1, #1, (R1)                       ;
   53     62      01            02   EF 00034      EXTZV   #2, #1, (R2), R3                       ; 1592
   61     01      02            53   F0 00039      INSV    R3, #2, #1, (R1)                       ;
   53     62      01            03   EF 0003E      EXTZV   #3, #1, (R2), R3                       ; 1593
   61     01      03            53   F0 00043      INSV    R3, #3, #1, (R1)                       ;
   53     62      01            04   EF 00048      EXTZV   #4, #1, (R2), R3                       ; 1594
   61     01      04            53   F0 0004D      INSV    R3, #4, #1, (R1)                       ;
                  61       E0   8F   88 00052      BISB2   #224, (R1)                             ; 1595
            06    A7            03   B0 00056      MOVW    #3, 6(R7)                              ; 1596
                  53       08   A0   32 0005A      CVTWL   8(R0), R3                              ; 1597
                                04   12 0005E      BNEQ    2$                                     ;
                                51   D4 00060      CLRL    R1                                     ;
                                26   11 00062      BRB     6$                                     ;
                                51   D4 00064 2$:  CLRL    I                                      ; 1598
                                12   11 00066      BRB     4$                                     ;
          52      50            53   C1 00068 3$:  ADDL3   R3, R0, R2                             ;
            10 BC41             52   D1 0006C      CMPL    R2, @OLD_VECTOR[I]                     ;
                                07   12 00071      BNEQ    4$                                     ;
                  51      0C BC41    D0 00073      MOVL    @NEW_VECTOR[I], R1                     ;
                                08   11 00078      BRB     5$                                     ;
          E9      51       10   BC   F3 0007A 4$:  AOBLEQ  @OLD_VECTOR, I, 3$                     ;
                  51            01   CE 0007F      MNEGL   #1, R1                                 ;
                  52       0C   AC   D0 00082 5$:  MOVL    NEW_VECTOR, R2                         ;
                  51       04   A2   C2 00086      SUBL2   4(R2), R1                              ;
            08    A7            51   D0 0008A 6$:  MOVL    R1, 8(R7)                              ; 1597
```

```
                          53       06   A0   32 0008E          CVTWL    6(R0), R3                                  ; 1599
                                        04   12 00092          BNEQ     7$                                         ;
                                        51   D4 00094          CLRL     R1                                         ;
                                        26   11 00096          BRB      11$                                        ;
                                        51   D4 00098 7$:      CLRL     I                                          ; 1600
                                        12   11 0009A          BRB      9$                                         ;
                     52           50    53   C1 0009C 8$:      ADDL3    R3, R0, R2                                 ;
                          10 BC41       52   D1 000A0          CMPL     R2, @OLD_VECTOR[I]                         ;
                                        07   12 000A5          BNEQ     9$                                         ;
                                  51  0C BC41  D0 000A7        MOVL     @NEW_VECTOR[I], R1                         ;
                                        08   11 000AC          BRB      10$                                        ;
                     E9           51   10   BC   F3 000AE 9$:  AOBLEQ   @OLD_VECTOR, I, 8$                         ;
                                  51        01   CE 000B3      MNEGL    #1, R1                                     ;
                                  52   0C   AC   D0 000B6 10$: MOVL     NEW_VECTOR, R2                             ;
                                  51   04   A2   C2 000BA      SUBL2    4(R2), R1                                  ;
                          0C      A7        51   D0 000BE 11$: MOVL     R1, 12(R7)                                 ; 1599
                                       10   A7   D4 000C2      CLRL     16(R7)                                     ; 1601
                                  52        64   32 000C5      CVTWL    (R4), R2                                   ; 1602
                                            53   D4 000C8      CLRL     R3                                         ;
                                            52   D5 000CA      TSTL     R2                                         ;
                                            06   12 000CC      BNEQ     12$                                        ;
                                            53   D6 000CE      INCL     R3                                         ;
                                            51   D4 000D0      CLRL     R1                                         ;
                                            03   11 000D2      BRB      13$                                        ;
                                  51        02   D0 000D4 12$: MOVL     #2, R1                                     ;
                          14      A7        51   90 000D7 13$: MOVB     R1, 20(R7)                                 ;
    15   A7          04           00   02   A0   F0 000DB      INSV     2(R0), #0, #4, 21(R7)                      ; 1603
         51     02   A0           04        04   EF 000E2      EXTZV    #4, #4, 2(R0), R1                          ; 1604
    15   A7          04           04        51   F0 000E8      INSV     R1, #4, #4, 21(R7)                         ;
                                       16   A7   94 000EE      CLRB     22(R7)                                     ; 1605
                                       18   A7   B4 000F1      CLRW     24(R7)                                     ; 1606
                                  05        53   E9 000F4      BLBC     R3, 14$                                    ; 1607
                                       1A   A7   B4 000F7      CLRW     26(R7)                                     ; 1608
                                            0D   11 000FA      BRB      15$                                        ;
                1A   A7           56        57   A3 000FC 14$: SUBW3    R7, VARIABLE_PTR, 26(R7)                   ; 1613
                                          6240   9F 00101      PUSHAB   (R2)[R0]                                   ; 1614
                                  86        9E   D0 00104      MOVL     @(SP)+, (VARIABLE_PTR)+                    ;
                                            86   94 00107      CLRB     (VARIABLE_PTR)+                            ; 1615
                                  51   0A   A0   32 00109 15$: CVTWL    10(R0), R1                                 ; 1618
                                            03   12 0010D      BNEQ     16$                                        ;
                                          00ED   31 0010F      BRW      32$                                        ;
                1C   A7           56        57   A3 00112 16$: SUBW3    R7, VARIABLE_PTR, 28(R7)                   ; 1624
                                  58      6140   9A 00117      MOVZBL   (R1)[R0], R8                               ; 1625
                                  52   01   A8   9E 0011B      MOVAB    1(R8), R2                                  ;
                     66         6140        52   28 0011F      MOVC3    R2, (R1)[R0], (VARIABLE_PTR)               ;
                                  56   01 A846   9E 00124      MOVAB    1(R8)[VARIABLE_PTR], VARIABLE_PTR          ; 1626
                                          00D6   31 00129      BRW      33$                                        ; 1628
                                            61   B4 0012C 17$: CLRW     (R1)                                       ; 1635
         52     01   A0           01        01   EF 0012E      EXTZV    #1, #1, 1(R0), R2                          ; 1636
         61          01           05        52   F0 00134      INSV     R2, #5, #1, (R1)                           ;
         52     01   A0           01        02   EF 00139      EXTZV    #2, #1, 1(R0), R2                          ; 1637
         61          01           06        52   F0 0013F      INSV     R2, #6, #1, (R1)                           ;
                                  61   80   8F   88 00144      BISB2    #128, (R1)                                 ; 1638
                          06      A7        03   B0 00148      MOVW     #3, 6(R7)                                  ; 1639
                                  53   05   A0   32 0014C      CVTWL    5(R0), R3                                  ; 1640
                                            04   12 00150      BNEQ     18$                                        ;
                                            51   D4 00152      CLRL     R1                                         ;
```

```
                                         26 11 00154        BRB     22$                                 ;
                                         51 D4 00156 18$:   CLRL    I                                   ; 1641
                                         12 11 00158        BRB     20$                                 ;
               52         50             53 C1 0015A 19$:   ADDL3   R3, R0, R2                          ;
                     10 BC41             52 D1 0015E        CMPL    R2, @OLD_VECTOR[I]                  ;
                                         07 12 00163        BNEQ    20$                                 ;
                          51   0C BC41      D0 00165        MOVL    @NEW_VECTOR[I], R1                  ;
                                         08 11 0016A        BRB     21$                                 ;
               E9         51   10     BC    F3 0016C 20$:   AOBLEQ  @OLD_VECTOR, I, 19$                 ;
                          51          01    CE 00171        MNEGL   #1, R1                              ;
                          52   0C     AC    D0 00174 21$:   MOVL    NEW_VECTOR, R2                      ;
                          51   04     A2    C2 00178        SUBL2   4(R2), R1                           ;
                     08   A7          51    D0 0017C 22$:   MOVL    R1, 8(R7)                           ; 1640
                          53   07     A0    32 00180        CVTWL   7(R0), R3                           ; 1642
                                         04 12 00184        BNEQ    23$                                 ;
                                         51 D4 00186        CLRL    R1                                  ;
                                         26 11 00188        BRB     27$                                 ;
                                         51 D4 0018A 23$:   CLRL    I                                   ; 1643
                                         12 11 0018C        BRB     25$                                 ;
               52         50             53 C1 0018E 24$:   ADDL3   R3, R0, R2                          ;
                     10 BC41             52 D1 00192        CMPL    R2, @OLD_VECTOR[I]                  ;
                                         07 12 00197        BNEQ    25$                                 ;
                          51   0C BC41      D0 00199        MOVL    @NEW_VECTOR[I], R1                  ;
                                         08 11 0019E        BRB     26$                                 ;
               E9         51   10     BC    F3 001A0 25$:   AOBLEQ  @OLD_VECTOR, I, 24$                 ;
                          51          01    CE 001A5        MNEGL   #1, R1                              ;
                          52   0C     AC    D0 001A8 26$:   MOVL    NEW_VECTOR, R2                      ;
                          51   04     A2    C2 001AC        SUBL2   4(R2), R1                           ;
                     0C   A7          51    D0 001B0 27$:   MOVL    R1, 12(R7)                          ; 1642
                               10     A7    D4 001B4        CLRL    16(R7)                              ; 1644
                          05   01     A0    E8 001B7        BLBS    1(R0), 28$                          ; 1645
                          51          04    D0 001BB        MOVL    #4, R1                              ;
                                         0C 11 001BE        BRB     30$                                 ;
                               02     A0    B5 001C0 28$:   TSTW    2(R0)                               ; 1646
                                         04 12 001C3        BNEQ    29$                                 ;
                                         51 D4 001C5        CLRL    R1                                  ;
                                         03 11 001C7        BRB     30$                                 ;
                          51          02    D0 001C9 29$:   MOVL    #2, R1                              ;
                     14   A7          51    90 001CC 30$:   MOVB    R1, 20(R7)                          ; 1645
  15   A7      04         00          64    F0 001D0        INSV    (R4), #0, #4, 21(R7)                ; 1648
       51      64         04          04    EF 001D6        EXTZV   #4, #4, (R4), R1                    ; 1649
  15   A7      04         04          51    F0 001DB        INSV    R1, #4, #4, 21(R7)                  ;
                               16     A7    94 001E1        CLRB    22(R7)                              ; 1650
                               18     A7    B4 001E4        CLRW    24(R7)                              ; 1651
                          51   02     A0    32 001E7        CVTWL   2(R0), R1                           ; 1652
                                         05 12 001EB        BNEQ    31$                                 ;
                               1A     A7    B4 001ED        CLRW    26(R7)                              ; 1653
                                         0D 11 001F0        BRB     32$                                 ;
          1A   A7         56          57    A3 001F2 31$:   SUBW3   R7, VARIABLE_PTR, 26(R7)            ; 1658
                                      6140  9F 001F7        PUSHAB  (R1)[R0]                            ; 1659
                          86          9E    D0 001FA        MOVL    @(SP)+, (VARIABLE_PTR)+             ;
                                         86 94 001FD        CLRB    (VARIABLE_PTR)+                     ; 1660
                               1C     A7    B4 001FF 32$:   CLRW    28(R7)                              ; 1663
                               1E     A7    B4 00202 33$:   CLRW    30(R7)                              ; 1664
               67         56          57    A3 00205        SUBW3   R7, VARIABLE_PTR, (R7)              ; 1669
                                         04 00209           RET                                         ; 1673
```

```
; Routine Size:  522 bytes,    Routine Base:  _CDU$CODE + 0489
```

```
;  673          1674  1 !++
;  674          1675  1 ! Description:  This routine is called to fill in a new entity block from
;  675          1676  1 !               the corresponding old block.
;  676          1677  1 !
;  677          1678  1 ! Parameters:   new_block       By reference, the new block to be filled in.
;  678          1679  1 !                               The type and subtype are already present.
;  679          1680  1 !               old_block       By reference, the corresponding old block.
;  680          1681  1 !               new_vector      By reference, the vector of new block
;  681          1682  1 !                               addresses.  Zeroth entry is block count.
;  682          1683  1 !               old_vector      By reference, the vector of old block
;  683          1684  1 !                               addresses.
;  684          1685  1 !               get_vm          By reference, see above.
;  685          1686  1 !
;  686          1687  1 ! Returns:      Nothing.
;  687          1688  1 !
;  688          1689  1 ! Notes:
;  689          1690  1 !--
;  690          1691  1
;  691          1692  1 ROUTINE upgrade_5_to_6_entity(new_block: pointer,
;  692          1693  1                               old_block: pointer,
;  693          1694  1                               new_vector: ref vector[,long],
;  694          1695  1                               old_vector: ref vector[,long],
;  695          1696  1                               get_vm: pointer)                : novalue
;  696          1697  1
;  697          1698  2 = BEGIN
;  698          1699  2
;  699          1700  2 local
;  700          1701  2         status: long,
;  701          1702  2         variable_ptr: pointer;
;  702          1703  2
;  703          1704  2
;  704          1705  2 ! Set up to add information to the variable portion of the new block.
;  705          1706  2
;  706          1707  2 variable_ptr = new_block[ent_z_variable];
;  707          1708  2
;  708          1709  2 ! Now fill in the new entity block from the old one.  Note that we cannot
;  709          1710  2 ! differentiate between qualifiers and keywords.
;  710          1711  2
;  711          1712  2 new_block[ent_b_subtype] =
;  712          1713  2         (if .old_block[ent5_w_number] lequ 8 then ent_k_parameter else ent_k_qualifier);
;  713          1714  2 new_block[ent_w_flags] = 0;
;  714          1715  2 new_block[ent_v_val] = .old_block[ent5_v_val];
;  715          1716  2 new_block[ent_v_neg] = .old_block[ent5_v_neg];
;  716          1717  2 new_block[ent_v_deftrue] = .old_block[ent5_v_deftrue];
;  717          1718  2 new_block[ent_v_batdef] = .old_block[ent5_v_batdef];
;  718          1719  2 new_block[ent_v_valreq] = .old_block[ent5_v_valreq];
;  719          1720  2 new_block[ent_v_list] = .old_block[ent5_v_list];
;  720          1721  2 new_block[ent_v_concat] = .old_block[ent5_v_concat];
;  721          1722  2 new_block[ent_v_impcat] = .old_block[ent5_v_impcat];
;  722          1723  2 new_block[ent_v_verb] = .old_block[ent5_v_verb];
;  723          1724  2 new_block[ent_v_parm] = .old_block[ent5_v_parm];
;  724          1725  2 new_block[ent_v_mcroptdelim] = .old_block[ent5_v_mcroptdlm];
;  725          1726  2 new_block[ent_v_mcrignore] = .old_block[ent5_v_mcrignore];
;  726          1727  2 new_block[ent_w_tro_count] = 3;
;  727          1728  3 new_block[ent_l_next] = (if .old_block[ent5_b_next] eqlu 0 then 0 else
;  728          1729  2         new_block_address(.old_block+.old_block[ent5_b_next]) - .new_vector[1]);
;  729          1730  3 new_block[ent_l_syntax] = (if .old_block[ent5_w_syntax] eqlu 0 then 0 else
```

```
;  730          1731  2          new_block_address(.old_block+.old_block[ent5_w_syntax]) - .new_vector[1]);
;  731          1732  2
;  732          1733  2 ! For the user type definition, we have to create a skeleton type block as
;  733          1734  2 ! a header for the keyword entity blocks.
;  734          1735  2
;  735          1736  2 if .old_block[ent5_w_keywords] eqlu 0 then
;  736          1737  2          new_block[ent_l_user_type] = 0
;  737          1738  2 else (
;  738          1739  3          local
;  739          1740  3                  type_block: pointer;
;  740          1741  3
;  741          1742  3          status = (.get_vm)(%ref(type_k_length), type_block);
;  742          1743  3          type_block[type_w_size] = type_k_length;
;  743          1744  3          type_block[type_b_type] = block_k_type;
;  744          1745  3          type_block[type_b_subtype] = type_k_type;
;  745          1746  3          type_block[type_w_flags] = 0;
;  746          1747  3          type_block[type_w_tro_count] = 1;
;  747          1748  3          type_block[type_l_keywords] = new_block_address(.old_block+.old_block[ent5_w_keywords]) - .new_vecto
;  748          1749  3          type_block[type_w_name] = type_block[type_w_prefix] = 0;
;  749          1750  3          new_block[ent_l_user_type] = .type_block = .new_vector[1];
;  750          1751  2 );
;  751          1752  2
;  752          1753  2 ! Continue filling in the entity block.  Note that we can't get the entity
;  753          1754  2 ! number except for parameters.
;  754          1755  2
;  755          1756  2 new_block[ent_b_number] =
;  756          1757  2          (if .new_block[ent_b_subtype] eqlu ent_k_parameter then .old_block[ent5_w_number] else 0);
;  757          1758  2 new_block[ent_b_valtype] = .old_block[ent5_b_valtype];
;  758          1759  2 new_block[ent_w_name] = .variable_ptr - .new_block;
;  759          1760  2 if .old_block[ent5_w_name] lequ 8 then (
;  760          1761  3          variable_ptr[0,0,8,0] = 2;
;  761          1762  3          variable_ptr[1,0,8,0] = 'P';
;  762          1763  3          variable_ptr[2,0,8,0] = '0' + .old_block[ent5_w_name];
;  763          1764  3          variable_ptr = .variable_ptr + 1+2;
;  764          1765  2 ) else (
;  765          1766  3          bind
;  766          1767  3                  entity_name = .old_block + .old_block[ent5_w_name]: vector[,byte];
;  767          1768  3
;  768          1769  3          ch$move(1+.entity_name[0],entity_name[0], .variable_ptr);
;  769          1770  3          variable_ptr = .variable_ptr + 1+.entity_name[0];
;  770          1771  2 );
;  771          1772  2 if .old_block[ent5_w_label] eqlu 0 then
;  772          1773  2          new_block[ent_w_label] = .new_block[ent_w_name]
;  773          1774  2 else (
;  774          1775  3          bind
;  775          1776  3                  entity_label = .old_block + .old_block[ent5_w_label]: vector[,byte];
;  776          1777  3
;  777          1778  3          new_block[ent_w_label] = .variable_ptr - .new_block;
;  778          1779  3          ch$move(1+.entity_label[0],entity_label[0], .variable_ptr);
;  779          1780  3          variable_ptr = .variable_ptr + 1+.entity_label[0];
;  780          1781  2 );
;  781          1782  2 if .old_block[ent5_w_prompt] eqlu 0 then
;  782          1783  2          new_block[ent_w_prompt] = 0
;  783          1784  2 else (
;  784          1785  3          bind
;  785          1786  3                  entity_prompt = .old_block + .old_block[ent5_w_prompt]: vector[,byte];
;  786          1787  3
```

```
3           new_block[ent_w_prompt] = .variable_ptr - .new_block;
3           ch$move(1+.entity_prompt[0],entity_prompt[0], .variable_ptr);
3           variable_ptr = .variable_ptr + 1+.entity_prompt[0];
2   );
2   if .old_block[ent5_w_defval] eqlu 0 then
2           new_block[ent_w_defval] = 0
3   else (
3           bind
3                   entity_defval = .old_block + .old_block[ent5_w_defval]: vector[,byte];
3
3           new_block[ent_w_defval] = .variable_ptr - .new_block;
3           variable_ptr[0,0,8,0] = 1+.entity_defval[0];
3           ch$move(1+.entity_defval[0],entity_defval[0], .variable_ptr+1);
3           variable_ptr = .variable_ptr + 1 + 1+.entity_defval[0];
2   );
2
2   ! Now we can fill in the final size of the new block.
2
2   new_block[ent_w_size] = .variable_ptr - .new_block;
2
2   return;
2
1   END;
```

```
                                             03FC 00000 UPGRADE_5_TO_6_ENTITY:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 1692
                       5E          08     C2 00002      SUBL2   #8, SP
                       57      04  AC     7D 00005      MOVQ    NEW_BLOCK, R7                          ; 1707
                       56      1E  A7     9E 00009      MOVAB   30(R7), VARIABLE_PTR
                       08      04  A8     B1 0000D      CMPW    4(R8), #8                              ; 1713
                                   05     1A 00011      BGTRU   1$
                       50          01     D0 00013      MOVL    #1, R0                                 ;
                                   03     11 00016      BRB     2$                                     ;
                       50          02     D0 00018 1$:  MOVL    #2, R0                                 ;
                 03    A7          50     90 0001B 2$:  MOVB    R0, 3(R7)                              ;
                       50      04  A7     9E 0001F      MOVAB   4(R7), R0                              ; 1714
                                   60     B4 00023      CLRW    (R0)
                       51      10  A8     9E 00025      MOVAB   16(R8), R1                             ; 1715
  52         61        01          01     EF 00029      EXTZV   #1, #1, (R1), R2
  60         01        00          52     F0 0002E      INSV    R2, #0, #1, (R0)
  52         61        01          02     EF 00033      EXTZV   #2, #1, (R1), R2                       ; 1716
  60         01        01          52     F0 00038      INSV    R2, #1, #1, (R0)
  52         61        01          03     EF 0003D      EXTZV   #3, #1, (R1), R2                       ; 1717
  60         01        02          52     F0 00042      INSV    R2, #2, #1, (R0)
  52         61        01          04     EF 00047      EXTZV   #4, #1, (R1), R2                       ; 1718
  60         01        03          52     F0 0004C      INSV    R2, #3, #1, (R0)
  52         61        01          05     EF 00051      EXTZV   #5, #1, (R1), R2                       ; 1719
  60         01        04          52     F0 00056      INSV    R2, #4, #1, (R0)
  52         61        01          06     EF 0005B      EXTZV   #6, #1, (R1), R2                       ; 1720
  60         01        05          52     F0 00060      INSV    R2, #5, #1, (R0)
  52         61        01          07     EF 00065      EXTZV   #7, #1, (R1), R2                       ; 1721
  60         01        06          52     F0 0006A      INSV    R2, #6, #1, (R0)
  60         01        07      01  A1     F0 0006F      INSV    1(R1), #7, #1, (R0)                    ; 1722
```

```
                52              61              01              09  EF  00075          EXTZV   #9, #1, (R1), R2                        ; 1723
          01    A0              01              00              52  F0  00C7A          INSV    R2, #0, #1, 1(R0)
                52              61              01              0A  EF  00080          EXTZV   #10, #1, (R1), R2               ; 1724
                60              01              09              52  F0  00085          INSV    R2, #9, #1, (R0)
                52              61              01              0B  EF  0008A          EXTZV   #11, #1, (R1), R2               ; 1725
                60              01              0A              52  F0  0008F          INSV    R2, #10, #1, (R0)
                52              61              01              0C  EF  00094          EXTZV   #12, #1, (R1), R2               ; 1726
                60              01              0B              52  F0  00099          INSV    R2, #11, #1, (R0)
                                        06      A7              03  B0  0009E          MOVW    #3, 6(R7)                       ; 1727
                                                51              68  9A  000A2          MOVZBL  (R8), R1                        ; 1728
                                                                04  12  000A5          BNEQ    3$
                                                                50  D4  000A7          CLRL    R0
                                                                26  11  000A9          BRB     7$
                                                                50  D4  000AB  3$:     CLRL    I                               ; 1729
                                                                12  11  000AD          BRB     5$
                        52                      58              51  C1  000AF  4$:     ADDL3   R1, R8, R2
                                        10  BC40                52  D1  000B3          CMPL    R2, @OLD_VECTOR[I]
                                                                07  12  000B8          BNEQ    5$
                                                50      0C  BC40    D0  000BA          MOVL    @NEW_VECTOR[I], R0
                                                                08  11  000BF          BRB     6$
                        E9                      50      10      BC  F3  000C1  5$:     AOBLEQ  @OLD_VECTOR, I, 4$
                                                50              01  CE  000C6          MNEGL   #1, R0
                                                51      0C      AC  D0  000C9  6$:     MOVL    NEW_VECTOR, R1
                                                50      04      A1  C2  000CD          SUBL2   4(R1), R0
                                        08      A7              50  D0  000D1  7$:     MOVL    R0, 8(R7)                       ; 1728
                                                51      0A      A8  32  000D5          CVTWL   10(R8), R1                      ; 1730
                                                                04  12  000D9          BNEQ    8$
                                                                50  D4  000DB          CLRL    R0
                                                                26  11  000DD          BRB     12$
                                                                50  D4  000DF  8$:     CLRL    I                               ; 1731
                                                                12  11  000E1          BRB     10$
                52                      58                      51  C1  000E3  9$:     ADDL3   R1, R8, R2
                                        10  BC40                52  D1  000E7          CMPL    R2, @OLD_VECTOR[I]
                                                                07  12  000EC          BNEQ    10$
                                                50      0C  BC40    D0  000EE          MOVL    @NEW_VECTOR[I], R0
                                                                08  11  000F3          BRB     11$
                E9                      50      10      BC  F3  000F5  10$:    AOBLEQ  @OLD_VECTOR, I, 9$
                                                50              01  CE  000FA          MNEGL   #1, R0
                                                51      0C      AC  D0  000FD  11$:    MOVL    NEW_VECTOR, R1
                                                50      04      A1  C2  00101          SUBL2   4(R1), R0
                                        0C      A7              50  D0  00105  12$:    MOVL    R0, 12(R7)                      ; 1730
                                                53      0C      A8  32  00109          CVTWL   12(R8), R3                      ; 1736
                                                                05  12  0010D          BNEQ    13$
                                                        10      A7  D4  0010F          CLRL    16(R7)                          ; 1737
                                                                50  11  00112          BRB     17$
                                                        04      AE  9F  00114  13$:    PUSHAB  TYPE_BLOCK                      ; 1742
                                        04      AE                  10  D0  00117          MOVL    #16, 4(SP)
                                                        04      AE  9F  0011B          PUSHAB  4(SP)
                                        14      BC                  02  FB  0011E          CALLS   #2, @GET_VM
                                                50      04      AE  D0  00122          MOVL    TYPE_BLOCK, R0                  ; 1743
                                                60              10  B0  00126          MOVW    #16, (R0)
                                        02      A0      0103    8F  3C  00129          MOVZWL  #259, 2(R0)                     ; 1744
                                        06      A0              01  B0  0012F          MOVW    #1, 6(R0)                       ; 1747
                                                                51  D4  00133          CLRL    I                               ; 1748
                                                                12  11  00135          BRB     15$
                52                      58                      53  C1  00137  14$:    ADDL3   R3, R8, R2
                                        10  BC41                52  D1  0013B          CMPL    R2, @OLD_VECTOR[I]
```

```
                              07 12 00140        BNEQ    15$                                            ;
              52       0C BC41 D0 00142          MOVL    @NEW_VECTOR[I], R2                             ;
                              08 11 00147        BRB     16$                                            ;
        E9    51       10   BC F3 00149 15$:     AOBLEQ  @OLD_VECTOR, I, 14$                            ;
              52            01 CE 0014E          MNEGL   #1, R2                                         ;
              51       0C   AC D0 00151 16$:     MOVL    NEW_VECTOR, R1                                 ;
  08    A0    52       04   A1 C3 00155          SUBL3   4(R7), R2, 8(R0)                               ;
                       0C   A0 D4 0015B          CLRL    12(R0)                                         ; 1749
  10    A7    50       04   A1 C3 0015E          SUBL3   4(R1), R0, 16(R7)                              ; 1750
              01       03   A7 91 00164 17$:     CMPB    3(R7), #1                                      ; 1757
                            06 12 00168          BNEQ    18$                                            ;
              50       04   A8 3C 0016A          MOVZWL  4(R8), R0                                      ;
                            02 11 0016E          BRB     19$                                            ;
                            50 D4 00170 18$:     CLRL    R0                                             ;
           14    A7         50 90 00172 19$:     MOVB    R0, 20(R7)                                     ;
           15    A7    03   A8 90 00176          MOVB    3(R8), 21(R7)                                  ; 1758
  16    A7    56            57 A3 0017B          SUBW3   R7, VARIABLE_PTR, 22(R7)                       ; 1759
              50       04   A8 32 00180          CVTWL   4(R8), R0                                      ; 1760
              08            50 B1 00184          CMPW    R0, #8                                         ;
                            0B 1A 00187          BGTRU   20$                                            ;
              86     5002   8F B0 00189          MOVW    #20482, (VARIABLE_PTR)+                        ; 1761
        86    50            30 81 0018E          ADDB3   #48, R0, (VARIABLE_PTR)+                       ; 1763
                            12 11 00192          BRB     21$                                            ; 1760
              59          6048 9A 00194 20$:     MOVZBL  (R0)[R8], R9                                   ; 1769
              51       01   A9 9E 00198          MOVAB   1(R9), R1                                      ;
        66  6048            51 28 0019C          MOVC3   R1, (R0)[R8], (VARIABLE_PTR)                   ;
              56       01 A946 9E 001A1          MOVAB   1(R9)[VARIABLE_PTR], VARIABLE_PTR              ; 1770
              50       06   A8 32 001A6 21$:     CVTWL   6(R8), R0                                      ; 1772
                            07 12 001AA          BNEQ    22$                                            ;
           18    A7    16   A7 B0 001AC          MOVW    22(R7), 24(R7)                                 ; 1773
                            17 11 001B1          BRB     23$                                            ;
  18    A7    56            57 A3 001B3 22$:     SUBW3   R7, VARIABLE_PTR, 24(R7)                       ; 1778
              59          6048 9A 001B8          MOVZBL  (R0)[R8], R9                                   ; 1779
              51       01   A9 9E 001BC          MOVAB   1(R9), R1                                      ;
        66  6048            51 28 001C0          MOVC3   R1, (R0)[R8], (VARIABLE_PTR)                   ;
              56       01 A946 9E 001C5          MOVAB   1(R9)[VARIABLE_PTR], VARIABLE_PTR              ; 1780
              50       0E   A8 32 001CA 23$:     CVTWL   14(R8), R0                                     ; 1782
                            05 12 001CE          BNEQ    24$                                            ;
                       1A   A7 B4 001D0          CLRW    26(R7)                                         ; 1783
                            17 11 001D3          BRB     25$                                            ;
  1A    A7    56            57 A3 001D5 24$:     SUBW3   R7, VARIABLE_PTR, 26(R7)                       ; 1788
              59          6048 9A 001DA          MOVZBL  (R0)[R8], R9                                   ; 1789
              51       01   A9 9E 001DE          MOVAB   1(R9), R1                                      ;
        66  6048            51 28 001E2          MOVC3   R1, (R0)[R8], (VARIABLE_PTR)                   ;
              56       01 A946 9E 001E7          MOVAB   1(R9)[VARIABLE_PTR], VARIABLE_PTR              ; 1790
              51       08   A8 32 001EC 25$:     CVTWL   8(R8), R1                                      ; 1792
                            05 12 001F0          BNEQ    26$                                            ;
                       1C   A7 B4 001F2          CLRW    28(R7)                                         ; 1793
                            1B 11 001F5          BRB     27$                                            ;
  1C    A7    56            57 A3 001F7 26$:     SUBW3   R7, VARIABLE_PTR, 28(R7)                       ; 1798
              59          6148 9A 001FC          MOVZBL  (R1)[R8], R9                                   ; 1799
              50       01   A9 9E 00200          MOVAB   1(R9), R0                                      ;
              66            50 90 00204          MOVB    R0, (VARIABLE_PTR)                             ;
  01    A6  6148            50 28 00207          MOVC3   R0, (R1)[R8], 1(VARIABLE_PTR)                  ; 1800
              56       02 A946 9E 0020D          MOVAB   2(R9)[VARIABLE_PTR], VARIABLE_PTR              ; 1801
        67    56            57 A3 00212 27$:     SUBW3   R7, VARIABLE_PTR, (R7)                         ; 1806
                               04 00216          RET                                                    ; 1810
```

```
; Routine Size:  535 bytes,    Routine Base:  _CDU$CODE + 0693

;  810          1811  1
;  811          1812  1 END
;  812          1813  0 ELUDOM



;                               PSECT SUMMARY
;
;      Name                     Bytes                          Attributes
;
;  _CDU$CODE                     2218  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(0)


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



;                                    COMMAND QUALIFIERS

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

; Size:          2218 code + 0 data bytes
; Run Time:        00:48.9
; Elapsed Time:    01:45.2
; Lines/CPU Min:    2224
; Lexemes/CPU-Min: 25888
; Memory Used:  305 pages
; Compilation Complete
```

JNLBUSR
R32

JNLDEFINT
SDL

CJFV4.

JNLPREFIX
R32

CJFRUFMAC
SDL

RUFUSR
SDL

UPGRADE
LIS

JNLFILE
SDL

BOPTIONS
R32

JNLDEF
SDL