```
   CCCCCCCCCCC  RRRRRRRRRRR     FFFFFFFFFFFFFFF
   CCCCCCCCCCC  RRRRRRRRRRR     FFFFFFFFFFFFFFF
   CCCCCCCCCCC  RRRRRRRRRRR     FFFFFFFFFFFFFFF
CCC             RRR         RRR  FFF
CCC             RRR         RRR  FFF
CCC             RRR         RRR  FFF
CCC             RRR         RRR  FFF
CCC             RRR         RRR  FFF
CCC             RRR         RRR  FFF
CCC             RRRRRRRRRRR      FFFFFFFFFFFF
CCC             RRRRRRRRRRR      FFFFFFFFFFFF
CCC             RRRRRRRRRRR      FFFFFFFFFFFF
CCC             RRR   RRR        FFF
CCC             RRR   RRR        FFF
CCC             RRR   RRR        FFF
CCC             RRR      RRR     FFF
CCC             RRR      RRR     FFF
CCC             RRR      RRR     FFF
   CCCCCCCCCCC  RRR         RRR  FFF
   CCCCCCCCCCC  RRR         RRR  FFF
   CCCCCCCCCCC  RRR         RRR  FFF
```

```
  CCCCCCCC  RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
  CCCCCCCC  RRRRRRRR    EEEEEEEEEE  FFFFFFFFFF
CC          RR      RR  EE          FF
CC          RR      RR  EE          FF
CC          RR      RR  EE          FF
CC          RR      RR  EE          FF
CC          RRRRRRRR    EEEEEEEE    FFFFFFFF
CC          RRRRRRRR    EEEEEEEE    FFFFFFFF
CC          RR  RR      EE          FF
CC          RR  RR      EE          FF
CC          RR    RR    EE          FF              ....
CC          RR    RR    EE          FF              ....
  CCCCCCCC  RR      RR  EEEEEEEEEE  FF              ....
  CCCCCCCC  RR      RR  EEEEEEEEEE  FF              ....


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 crf_cref (                     ! Cross reference facility
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1 %TITLE  'Cross reference facility';
;     7          0007  1
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 !
;    34          0034  1 ! FACILITY:  Cross reference
;    35          0035  1 !
;    36          0036  1 ! ABSTRACT:
;    37          0037  1 !
;    38          0038  1 !       The cross reference facility is a set of routines to input the cross
;    39          0039  1 !       reference data and output the cross reference on demand.
;    40          0040  1 !
;    41          0041  1 ! ENVIRONMENT:
;    42          0042  1 !
;    43          0043  1 !       VAX native, user mode.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Benn Schreiber,       CREATION DATE:  3-Dec-1979
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-002         BLS0124         Benn Schreiber          1-Jan-1982
;    53          0053  1 !               Ensure that refindent and refsperline are initialized before
;    54          0054  1 !               being used.  Better checking of output line too long.
;    55          0055  1 !
;    56          0056  1 !       V03-001         BLS0059         Benn Schreiber          1-Jul-1981
;    57          0057  1 !               If just printing keys and values, go through delete logic.
```

```
;   58          0058  1 !--
```

```
;    60          0059  1 %SBTTL  'Declarations';
;    61          0060  1 LIBRARY
;    62          0061  1         'SYS$LIBRARY:STARLET.L32';
;    63          0062  1 REQUIRE
;    64          0063  1         'CRFMDL';                       !Cross reference definitions
;    65          0132  1 REQUIRE
;    66          0133  1         'CRFERRMSG';                    !Cross reference messages
;    67          0200  1 !
;    68          0201  1 ! Define VMS block structures
;    69          0202  1 !
;    70          0203  1 STRUCTURE
;    71          0204  1         BBLOCK [O, P, S, E; N] =
;    72          0205  1                 [N]
;    73          0206  1                 (BBLOCK + O) <P, S, E>;
;    74          0207  1 !
;    75          0208  1 ! Macro to execute a given sequence of commands and return if any error
;    76          0209  1 !
;    77          0210  1 MACRO
;    78        M 0211  1     perform (command) =
;    79        M 0212  1         BEGIN
;    80        M 0213  1         LOCAL
;    81        M 0214  1             status;
;    82        M 0215  1         status = command;
;    83        M 0216  1         IF NOT .status                  ! If error detected,
;    84        M 0217  1         THEN RETURN creferror (.status); ! then return with error (after calling user routine if there is on
;    85          0218  1         END%;
;    86          0219  1
;    87          0220  1 LINKAGE
;    88          0221  1     JSB_0 = JSB : GLOBAL (controlreg = 11),
;    89          0222  1     JSB_1 = JSB (REGISTER = 0) : GLOBAL (controlreg = 11)
;    90          0223  1                                 PRESERVE (2, 3, 4) NOTUSED (5, 6, 7, 8, 9, 10),
;    91          0224  1     JSB_2 = JSB (REGISTER = 0, REGISTER = 1) : GLOBAL (controlreg = 11)
;    92          0225  1                                             PRESERVE (2, 3, 4) NOTUSED (5, 6, 7, 8, 9, 10),
;    93          0226  1     CALL_2 = CALL (STANDARD, STANDARD) : GLOBAL (controlreg = 11),
;    94          0227  1     CALL_3 = CALL (STANDARD, STANDARD, STANDARD) : GLOBAL (controlreg = 11),
;    95          0228  1     CALL_4 = CALL (STANDARD, STANDARD, STANDARD, STANDARD) : GLOBAL (controlreg = 11),
;    96          0229  1     CALL_SYS = CALL : PRESERVE (2, 3, 4, 5, 6, 7, 8, 9, 10, 11);
;    97          0230  1
;    98          0231  1 EXTERNAL ROUTINE
;    99          0232  1     sort_hash_table,                    !Sort hash table into linked list
;   100          0233  1     get_mem : JSB_2,                    !Allocate virtual memory
;   101          0234  1     get_zmem : JSB_2,                   !Allocate zeroed virtual memory
;   102          0235  1     free_mem : JSB_2,                   ! and deallocate it
;   103          0236  1     SYS$FAO : ADDRESSING_MODE (GENERAL) CALL_SYS; !Formatted ascii output
;   104          0237  1
;   105          0238  1 GLOBAL LITERAL
;   106          0239  1     crf$c_maxcol = 64,                  !Max cols on cref of symbol/value
;   107          0240  1     crf$c_maxlinwid = 132;              !Maximum line width
;   108          0241  1
;   109          0242  1 EXTERNAL LITERAL
;   110          0243  1     crf$c_hashsize;                     !Size of hash table
;   111          0244  1
;   112          0245  1 LITERAL
;   113          0246  1     crf$c_linextra = 10,                !Extra padding in line buffer
;   114          0247  1     true = 1,                           !True
;   115          0248  1     false = 0;                          ! and false
;   116          0249  1
```

```
;  117          0250  1 FORWARD ROUTINE
;  118          0251  1     add_key : CALL_2,                  !Add/lookup key in hash table
;  119          0252  1     creferror : JSB_1;                 .Call user routine on error
```

```
;   121          0253  1 %SBTTL  'add_key -- add key to hash table';
;   122          0254  1 ROUTINE add_key (keyadr, entryadr) : CALL_2 =
;   123          0255  2 BEGIN
;   124          0256  2
;   125          0257  2 !++
;   126          0258  2 !       Lookup/insert a key into the hash table
;   127          0259  2 !
;   128          0260  2 ! Inputs:
;   129          0261  2 !
;   130          0262  2 !       keyadr          Address of ascic key or binary value
;   131          0263  2 !       entryadr        Address of longword to store entry address
;   132          0264  2 !
;   133          0265  2 ! Outputs:
;   134          0266  2 !
;   135          0267  2 !       Address of key block is returned in .entryadr
;   136          0268  2 !
;   137          0269  2 !--
;   138          0270  2
;   139          0271  2 ROUTINE compute_hash (ascbin, keyadr) : JSB_2 =
;   140          0272  3 BEGIN
;   141          0273  3
;   142          0274  3 !++
;   143          0275  3 !       Compute the hash value for a key
;   144          0276  3 !
;   145          0277  3 ! Inputs:
;   146          0278  3 !
;   147          0279  3 !       ascbin          0 if ascic, 1 if binary keys
;   148          0280  3 !       keyadr          address of ascic string or binary value
;   149          0281  3 !
;   150          0282  3 ! Outputs:
;   151          0283  3 !
;   152          0284  3 !       Routine value is hash value for key
;   153          0285  3 !
;   154          0286  3 !--
;   155          0287  3
;   156          0288  3 LOCAL
;   157          0289  3     hashval;
;   158          0290  3
;   159          0291  3 IF NOT .ascbin
;   160          0292  4 THEN BEGIN
;   161          0293  4     MAP
;   162          0294  4         keyadr : REF VECTOR [,BYTE];
;   163          0295  4
;   164          0296  4 !
;   165          0297  4 ! Hash value for ascic keys is the number of letters in the key
;   166          0298  4 ! plus the sum of all the letters.
;   167          0299  4 !
;   168          0300  4     hashval = 0;
;   169          0301  4     INCRU i FROM 0 TO .keyadr [0]
;   170          0302  4         DO hashval = .hashval + .keyadr [.i];
;   171          0303  5     RETURN ((.hashval AND %X'7FFFFFFF') MOD crf$c_hashsize)
;   172          0304  4     END
;   173          0305  4 !
;   174          0306  4 ! Hash value for binary keys is low 31 bits of value
;   175          0307  4 !
;   176          0308  3 ELSE RETURN ((..keyadr AND %X'7FFFFFFF') MOD crf$c_hashsize);
;   177          0309  2 END;
```

```
                                                        .TITLE  CRF_CREF Cross reference facility
                                                        .IDENT  \V04-000\

                                                CRF$C_MAXCOL==      64
                                                CRF$C_MAXLINWID==   132
                                                        .EXTRN  SORT_HASH_TABLE
                                                        .EXTRN  GET_MEM, GET_ZMEM
                                                        .EXTRN  FREE_MEM, SYS$FAO
                                                        .EXTRN  CRF$C_HASHSIZE

                                                        .PSECT  $CODE$,NOWRT,2

                                          1C  BB 00000 COMPUTE_HASH:
                                                        PUSHR   #^M<R2,R3,R4>                                   ; 0271
                                    1C          50  E8 00002        BLBS    ASCBIN, 3$                          ; 0291
                                                53  D4 00005        CLRL    HASHVAL                             ; 0300
                                    52          61  9A 00007        MOVZBL  (KEYADR), R2                        ; 0301
                                                50  D4 0000A        CLRL    I
                                                07  11 0000C        BRB     2$
                                    54        8041  9A 0000E 1$:     MOVZBL  (I)+[KEYADR], R4                    ; 0302
                                    53          54  C0 00012        ADDL2   R4, HASHVAL
                                    52          50  D1 00015 2$:     CMPL    I, R2
                                                F4  1B 00018        BLEQU   1$
        50          53              1F          00  EF 0001A        EXTZV   #0, #31, HASHVAL, R0                ; 0303
                                                05  11 0001F        BRB     4$
        50          61              1F          00  EF 00021 3$:     EXTZV   #0, #32, (KEYADR), R0               ; 0308
        7E          00              50          01  7A 00026 4$:     EMUL    #1, R0, #0, -(SP)
        50          50      8E 00000000G        8F  7B 0002B        EDIV    #CRF$C_HASHSIZE, (SP)+, R0, R0
                                                1C  BA 00034        POPR    #^M<R2,R3,R4>                       ; 0309
                                                    05 00036        RSB

; Routine Size:  55 bytes,    Routine Base:  $CODE$ + 0000

;  178          0310  2
;  179          0311  2 !
;  180          0312  2 ! Main body of add_key
;  181          0313  2 !
;  182          0314  2 MAP
;  183          0315  2     keyadr : REF VECTOR [,BYTE];
;  184          0316  2
;  185          0317  2 LOCAL
;  186          0318  2     ch_result,
;  187          0319  2     ascbin,
;  188          0320  2     thisval,
;  189          0321  2     newbinarykey,
;  190          0322  2     keystring : REF VECTOR [,BYTE],
;  191          0323  2     prevkey : REF BBLOCK,
;  192          0324  2     thiskey : REF BBLOCK;
;  193          0325  2
;  194          0326  2 EXTERNAL REGISTER
;  195          0327  2     controlreg = 11 : REF BBLOCK;
;  196          0328  2
;  197          0329  2 BIND
;  198          0330  2     hashtable = .controlreg [crf$l_hasht] : VECTOR [,LONG];
```

```
;  199          0331  2
;  200          0332  2 .entryadr = 0;
;  201          0333  2 ascbin = .controlreg [crf$v_binary];
;  202          0334  2 IF .ascbin
;  203          0335  2     THEN newbinarykey = ..keyadr;
;  204          0336  2 !
;  205          0337  2 ! See if we just dealt with this key last call
;  206          0338  2 !
;  207          0339  2 IF (thiskey = .controlreg [crf$l_lastkey]) NEQ 0
;  208          0340  2 THEN
;  209          0341  3     IF (
;  210          0342  3         IF .ascbin
;  211          0343  3             THEN .newbinarykey EQL .thiskey [key$l_keyadr]
;  212          0344  4             ELSE BEGIN
;  213          0345  4                 keystring = .thiskey [key$l_keyadr];
;  214          0346  4                 CH$EQL (.keyadr [0], keyadr [1],
;  215          0347  4                         .keystring [0], keystring [1])
;  216          0348  4                 END
;  217          0349  3         )
;  218          0350  3         THEN BEGIN
;  219          0351  3             controlreg [crf$l_lastkey] = 0;             !Assume we won't hit it again
;  220          0352  3             .entryadr = .thiskey;
;  221          0353  3             RETURN crf$_keyprventr
;  222          0354  2             END;
;  223          0355  2
;  224          0356  2 !
;  225          0357  2 ! Look through the bucket for the key
;  226          0358  2 !
;  227          0359  2 controlreg [crf$l_lastkey] = 0;                         !No last key
;  228          0360  2 prevkey = hashtable [compute_hash (.ascbin, .keyadr)];
;  229          0361  2 thiskey = .prevkey [key$l_next];
;  230          0362  2 WHILE .thiskey NEQ 0
;  231          0363  3 DO BEGIN
;  232          0364  4     IF (
;  233          0365  4         IF .ascbin
;  234          0366  5         THEN BEGIN
;  235          0367  5             thisval = .thiskey [key$l_keyadr];
;  236          0368  5             .newbinarykey EQL .thisval
;  237          0369  5             END
;  238          0370  5         ELSE BEGIN
;  239          0371  5             keystring = .thiskey [key$l_keyadr];
;  240          0372  6             (ch_result = CH$COMPARE (.keyadr [0], keyadr [1],
;  241          0373  5                                      .keystring [0], keystring [1])) EQL 0
;  242          0374  5         END
;  243          0375  4         )
;  244          0376  4
;  245          0377  4     THEN BEGIN
;  246          0378  4 !
;  247          0379  4 ! Found in the table
;  248          0380  4 !
;  249          0381  4
;  250          0382  4         .entryadr = .thiskey;
;  251          0383  4         RETURN crf$_keyprventr
;  252          0384  4         END
;  253          0385  4
;  254          0386  4     ELSE BEGIN
;  255          0387  5         IF (
```

```
;  256          0388  5              IF .ascbin
;  257          0389  5                  THEN .newbinarykey LSSU .thisval
;  258          0390  6                  ELSE (.ch_result LSS 0)
;  259          0391  5              )
;  260          0392  4          THEN EXITLOOP;
;  261          0393  4          prevkey = .thiskey;
;  262          0394  4          thiskey = .thiskey [key$l_next];
;  263          0395  3          END;
;  264          0396  2      END;
;  265          0397  2 !
;  266          0398  2 ! Not in the table, so insert it
;  267          0399  2 !
;  268          0400  2 perform (get_mem (key$c_length, thiskey));              !Allocate it
;  269          0401  2 thiskey [key$l_reflist] = 0;
;  270          0402  2 thiskey [key$w_defflg] = 0;
;  271          0403  2 thiskey [key$l_defnam] = 0;
;  272          0404  2 thiskey [key$l_next] = .prevkey [key$l_next];
;  273          0405  2 prevkey [key$l_next] = .thiskey;
;  274          0406  3 thiskey [key$l_keyadr] = (IF .ascbin
;  275          0407  3                                THEN .newbinarykey
;  276          0408  2                                ELSE .keyadr);
;  277          0409  2 .entryadr = .thiskey;
;  278          0410  2 controlreg [crf$l_lastkey] = .thiskey;
;  279          0411  2 controlreg [crf$l_entries] = .controlreg [crf$l_entries] + 1;
;  280          0412  2 RETURN crf$_keyistentr
;  281          0413  1 END;                                                   !Of add_key

                                                07FC 00000 ADD_KEY:.WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10        ; 0254
                                             5E         0C  C2 00002        SUBL2   #12, SP                         ;
                                             54         6B  D0 00005        MOVL    (CONTROLREG), R4                ; 0330
                                                     08  BC  D4 00008        CLRL    @ENTRYADR                       ; 0332
                                04   AE     08   AB     01  00  EF 0000B    EXTZV   #0, #1, 8(CONTROLREG), ASCBIN   ; 0333
                                             04         04  AE  E9 00012    BLBC    ASCBIN, 1$                      ; 0334
                                             6E         04  BC  D0 00016    MOVL    @KEYADR, NEWBINARYKEY           ; 0335
                                          08 AE         04  AB  D0 0001A 1$: MOVL    4(CONTROLREG), THISKEY          ; 0339
                                                        30  13  0001F      BEQL    4$                              ;
                                             50         08  AE  D0 00021    MOVL    THISKEY, R0                     ; 0343
                                             06         04  AE  E9 00025    BLBC    ASCBIN, 2$                      ;
                                          08 A0                 6E  D1 00029    CMPL    NEWBINARYKEY, 8(R0)             ;
                                                                16  11 0002D    BRB     3$                              ;
                                             56         08  A0  D0 0002F 2$: MOVL    8(R0), KEYSTRING                ; 0345
                                             50         04  AC  D0 00033    MOVL    KEYADR, R0                      ; 0346
                                             52                 60  9A 00037    MOVZBL  (R0), R2                        ;
                                             51                 66  9A 0003A    MOVZBL  (KEYSTRING), R1                 ; 0347
                                   51             00    01  A0  52  2D 0003D    CMPC5   R2, 1(R0), #0, R1, 1(KEYSTRING) ;
                                                     01  A6      00043        ;
                                                                0A  12 00045 3$: BNEQ    4$                              ;
                                                     04  AB  D4 00047        CLRL    4(CONTROLREG)                   ; 0351
                                                  08 BC  08  AE  D0 0004A    MOVL    THISKEY, @ENTRYADR              ; 0352
                                                                4F  11 0004F    BRB     9$                              ; 0353
                                                     04  AB  D4 00051 4$: CLRL    4(CONTROLREG)                   ; 0359
                                             57         04  AC  D0 00054    MOVL    KEYADR, R7                      ; 0360
                                             51                 57  D0 00058    MOVL    R7, R1                          ;
```

```
                                50        04    AE  D0 0005B          MOVL    ASCBIN, R0                                      ;
                                        FF67    30  0005F          BSBW    COMPUTE_HASH                                    ;
                                5A              64  D0 00062          MOVAL   (R4)[R0], PREVKEY                               ;
                             08 AE               6A  D0 00066          MOVL    (PREVKEY), THISKEY                              ; 0361
                                54        08    AE  D0 0006A 5$:      MOVL    THISKEY, R4                                     ; 0362
                                          50    13  0006E              BEQL    13$                                             ;
                                09        04    AE  E9 00070          BLBC    ASCBIN, 6$                                      ; 0367
                                58        08    A4  D0 00074          MOVL    8(R4), THISVAL                                  ;
                                58              6E  D1 00078          CMPL    NEWBINARYKEY, THISVAL                           ; 0368
                                          1D    11  0007B              BRB     8$                                              ;
                                56        08    A4  D0 0007D 6$:      MOVL    8(R4), KEYSTRING                                ; 0371
                                51              67  9A 00081          MOVZBL  (R7), R1                                        ; 0372
                                50              66  9A 00084          MOVZBL  (KEYSTRING), R0                                 ; 0373
                                55              01  D0 00087          MOVL    #1, R5                                          ;
                50              00        01    A7  51  2D 0008A      CMPC5   R1, 1(R7), #0, R0, 1(KEYSTRING)                 ;
                                        01    A6      00090                                                                   ;
                                          03    1A  00092              BGTRU   7$                                              ;
                                55              01  D9 00094          SBWC    #1, R5                                          ;
                                59              55  D0 00097 7$:      MOVL    R5, CH_RESULT                                   ;
                                          0C    12  0009A 8$:          BNEQ    10$                                             ;
                             08 BC              54  D0 0009C          MOVL    R4, @ENTRYADR                                   ; 0382
                                50 00658009    8F  D0 000A0 9$:      MOVL    #6651913, R0                                    ; 0383
                                                04  000A7              RET                                                     ;
                                07        04    AE  E9 000A8 10$:     BLBC    ASCBIN, 11$                                     ; 0388
                                58              6E  D1 000AC          CMPL    NEWBINARYKEY, THISVAL                           ; 0389
                                          06    1E  000AF              BGEQU   12$                                             ;
                                          0D    11  000B1              BRB     13$                                             ;
                                          59    D5  000B3 11$:         TSTL    CH_RESULT                                       ; 0390
                                          09    19  000B5              BLSS    13$                                             ;
                                5A              54  D0 000B7 12$:     MOVL    R4, PREVKEY                                     ; 0393
                             08 AE              64  D0 000BA          MOVL    (R4), THISKEY                                   ; 0394
                                          AA    11  000BE              BRB     5$                                              ; 0362
                                51        08    AE  9E 000C0 13$:     MOVAB   THISKEY, R1                                     ; 0400
                                50              18  D0 000C4          MOVL    #24, R0                                         ;
                                          0000G 30  000C7              BSBW    GET_MEM                                         ;
                                04              50  E8 000CA          BLBS    STATUS, 14$                                     ;
                                          0000V 30  000CD              BSBW    CREFERROR                                       ;
                                                04  000D0              RET                                                     ;
                                50        08    AE  D0 000D1 14$:     MOVL    THISKEY, R0                                     ; 0401
                                          04    A0  D4 000D5          CLRL    4(R0)                                           ;
                                          12    A0  B4 000D8          CLRW    18(R0)                                          ; 0402
                                          14    A0  D4 000DB          CLRL    20(R0)                                          ; 0403
                                60              6A  D0 000DE          MOVL    (PREVKEY), (R0)                                 ; 0404
                                6A              50  D0 000E1          MOVL    R0, (PREVKEY)                                   ; 0405
                                05        04    AE  E9 000E4          BLBC    ASCBIN, 15$                                     ; 0406
                                51              6E  D0 000E8          MOVL    NEWBINARYKEY, R1                                ; 0407
                                          03    11  000EB              BRB     16$                                             ;
                                51              57  D0 000ED 15$:     MOVL    R7, R1                                          ; 0408
                             08 A0              51  D0 000F0 16$:     MOVL    R1, 8(R0)                                       ; 0406
                             08 BC              50  D0 000F4          MOVL    R0, @ENTRYADR                                   ; 0409
                             04 AB              50  D0 000F8          MOVL    R0, 4(CONTROLREG)                               ; 0410
                                      0C        AB  D6 000FC          INCL    12(CONTROLREG)                                  ; 0411
                                50 00658011    8F  D0 000FF          MOVL    #6651921, R0                                    ; 0412
                                                04  00106              RET                                                     ; 0413

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0037
```

```
;  283          0414  1 %SBTTL  'get_hash -- Allocate hash table if needed'
;  284          0415  1 ROUTINE get_hash : JSB_0 =
;  285          0416  2 BEGIN
;  286          0417  2
;  287          0418  2 !++
;  288          0419  2 !       Allocate a hash table if none allocated
;  289          0420  2 !
;  290          0421  2 ! Inputs:
;  291          0422  2 !
;  292          0423  2 !       None
;  293          0424  2 !
;  294          0425  2 ! Outputs:
;  295          0426  2 !
;  296          0427  2 !       hash table allocated if not done so
;  297          0428  2 !
;  298          0429  2 !--
;  299          0430  2
;  300          0431  2 EXTERNAL REGISTER
;  301          0432  2     controlreg = 11 : REF BBLOCK;
;  302          0433  2
;  303          0434  2 IF .controlreg [crf$l_hasht] EQL 0
;  304          0435  3 THEN BEGIN
;  305          0436  3     perform (get_zmem (crf$c_hashsize *4, controlreg [crf$l_hasht]));
;  306          0437  3     controlreg [crf$l_lastkey] = 0;
;  307          0438  3     controlreg [crf$l_entries] = 0;
;  308          0439  2     END;
;  309          0440  2
;  310          0441  2 RETURN true
;  311          0442  2
;  312          0443  1 END;                                !Of get_hash


                                    6B  D5 00000 GET_HASH:
                                                 TSTL    (CONTROLREG)                    ; 0434
                                    19  12 00002          BNEQ    2$                      ;
                              51    5B  D0 00004          MOVL    CONTROLREG, R1          ; 0436
                              50 00000000*  8F  D0 00007  MOVL    #<CRF$C_HASHSIZE*4>, R0 ;
                                  0000G 30 0000E          BSBW    GET_ZMEM                ;
                              03    50  E8 00011          BLBS    STATUS, 1$              ;
                                  0000V 31 00014          BRW     CREFERROR               ;
                                 04   AB  D4 00017 1$:     CLRL    4(CONTROLREG)           ; 0437
                                 0C   AB  D4 0001A         CLRL    12(CONTROLREG)          ; 0438
                              50    01  D0 0001D 2$:      MOVL    #1, R0                  ; 0441
                                        05 00020          RSB                             ; 0443

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 013E
```

```
;  314          0444  1 %SBTTL  'crf$insrtkey -- Insert key into hash table';
;  315          0445  1 GLOBAL ROUTINE crf$insrtkey (control, keyadr, valadr, valflg) =
;  316          0446  2 BEGIN
;  317          0447  2
;  318          0448  2 !++
;  319          0449  2 !       Insert a key into the table
;  320          0450  2 !
;  321          0451  2 ! Inputs:
;  322          0452  2 !
;  323          0453  2 !       control         Address of cross-reference control block
;  324          0454  2 !       keyadr          Address of ASCIC key or binary value
;  325          0455  2 !       valadr          Address of key value
;  326          0456  2 !       valflg          flags word
;  327          0457  2 !
;  328          0458  2 ! Outputs:
;  329          0459  2 !
;  330          0460  2 !       key is entered into table
;  331          0461  2 !
;  332          0462  2 !--
;  333          0463  2
;  334          0464  2 MAP
;  335          0465  2     control : REF BBLOCK;
;  336          0466  2
;  337          0467  2 LOCAL
;  338          0468  2     cachentry : REF BBLOCK,
;  339          0469  2     status;
;  340          0470  2
;  341          0471  2 GLOBAL REGISTER
;  342          0472  2     controlreg = 11 : REF BBLOCK;
;  343          0473  2
;  344          0474  2 controlreg = .control;
;  345          0475  2
;  346          0476  3 IF NOT (status = get_hash ())
;  347          0477  2     THEN RETURN .status;
;  348          0478  2
;  349          0479  2 status = add_key (.keyadr, cachentry);          !Add/lookup key in hash table
;  350          0480  2 !
;  351          0481  2 ! Update entry in cache
;  352          0482  2 !
;  353          0483  2 IF .cachentry NEQ 0                            !If it was found/allocated
;  354          0484  3 THEN BEGIN
;  355          0485  3     cachentry [key$l_valadr] = .valadr;
;  356          0486  3     cachentry [key$w_valflg] = .valflg;
;  357          0487  2     END;
;  358          0488  2
;  359          0489  2 RETURN .status
;  360          0490  1 END;                                           !Of crf$insrtkey


                                        0FFC 00000          .ENTRY  CRF$INSRTKEY, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0445
                                                                    R10,R11                                       ;
                                5E      04   C2 00002          SUBL2   #4, SP                                    ;
                                5B  04  AC   D0 00005          MOVL    CONTROL, CONTROLREG                       ; 0474
                                        D4   10 00009          BSBB    GET_HASH                                  ; 0476
```

```
                                 19             50  E9 0000B          BLBC    STATUS, 1$                                      ;
                                                5E  DD 0000E          PUSHL   SP                                              ; 0479
                                          08    AC  DD 00010          PUSHL   KEYADR                                          ;
                                    FECO  CF    02  FB 00013          CALLS   #2, ADD_KEY                                     ;
                                          51    6E  D0 00018          MOVL    CACHENTRY, R1                                   ; 0483
                                                0A  13 0001B          BEQL    1$                                              ;
                                      0C  A1    0C  AC  D0 0001D      MOVL    VALADR, 12(R1)                                  ; 0485
                                      10  A1    10  AC  B0 00022      MOVW    VALFLG, 16(R1)                                  ; 0486
                                                    04 00027 1$:      RET                                                     ; 0490

; Routine Size:  40 bytes,    Routine Base:  $CODE$ + 015F
```

```
;  362          0491  1 %SBTTL  'crf$insrtref -- Insert reference to a key';
;  363          0492  1 GLOBAL ROUTINE crf$insrtref (control, keyadr, refadr, reflag, defind) =
;  364          0493  2 BEGIN
;  365          0494  2
;  366          0495  2 !++
;  367          0496  2 !       Insert a reference to a key.  If the key is not in the table, it is
;  368          0497  2 !       inserted.
;  369          0498  2 !
;  370          0499  2 ! Inputs:
;  371          0500  2 !
;  372          0501  2 !       control         Address of cross-reference control table
;  373          0502  2 !       keyadr          Address of ASCIC key or binary value
;  374          0503  2 !       refadr          Address of ASCIC string for referencer
;  375          0504  2 !       reflag          referencer flags
;  376          0505  2 !       defind          0 = reference, 1 = definition
;  377          0506  2 !
;  378          0507  2 !--
;  379          0508  2
;  380          0509  2 MAP
;  381          0510  2     control : REF BBLOCK,
;  382          0511  2     refadr : REF VECTOR [,BYTE];
;  383          0512  2
;  384          0513  2 LOCAL
;  385          0514  2     cachentry : REF BBLOCK,
;  386          0515  2     defnam : REF VECTOR [,BYTE],
;  387          0516  2     refblk : REF BBLOCK,
;  388          0517  2     lastref : REF BBLOCK,
;  389          0518  2     ch_result,
;  390          0519  2     status;
;  391          0520  2
;  392          0521  2 GLOBAL REGISTER
;  393          0522  2     controlreg = 11 : REF BBLOCK;
;  394          0523  2
;  395          0524  2 controlreg = .control;
;  396          0525  2
;  397          0526  3 IF NOT (status = get_hash ())
;  398          0527  2     THEN RETURN .status;
;  399          0528  2
;  400          0529  2 status = add_key (.keyadr, cachentry);                   !Add/lookup key
;  401          0530  2 IF .cachentry EQL 0                                      !If no memory today
;  402          0531  2     THEN RETURN .status;
;  403          0532  2 !
;  404          0533  2 ! If defining reference, then put definer name info in, else add to
;  405          0534  2 ! the reference list
;  406          0535  2 !
;  407          0536  2 IF .defind
;  408          0537  2
;  409          0538  3 THEN BEGIN
;  410          0539  3     IF (defnam = .cachentry [key$l_defnam]) NEQ 0
;  411          0540  3         THEN free_mem (.defnam [0] + 1, defnam [0]);        !Deallocate old name
;  412          0541  3
;  413          0542  3     perform (get_mem (.refadr [0] + 1, defnam));            !Allocate space for definer name
;  414          0543  3     CH$MOVE (.refadr [0] + 1, .refadr, .defnam);            ! and fill it in
;  415          0544  3     cachentry [key$l_defnam] = .defnam;
;  416          0545  3     cachentry [key$w_defflg] = .reflag;
;  417          0546  3     RETURN crf$_success
;  418          0547  3     END
```

```
;  419          0548  3
;  420          0549  3 ELSE BEGIN
;  421          0550  3 !
;  422          0551  3 ! Not the defining reference, look through the reference list, find the insertion point and insert
;  423          0552  3 ! this reference if not already there
;  424          0553  3 !
;  425          0554  3     refblk = cachentry [key$l_reflist];
;  426          0555  3     lastref = .refblk;
;  427          0556  3
;  428          0557  3     WHILE (refblk = .refblk [ref$l_next]) NEQ 0
;  429          0558  4     DO BEGIN
;  430          0559  4         ch_result = CH$COMPARE (.refadr [0], refadr [1],
;  431          0560  4                                 .refblk [ref$b_reflng], refblk [ref$t_refnam]);
;  432          0561  4         IF .ch_result EQL 0
;  433          0562  4             THEN RETURN crf$_refprventr
;  434          0563  4             ELSE IF .ch_result LSS 0                                    !Found the point
;  435          0564  4                 THEN EXITLOOP;
;  436          0565  4         lastref = .refblk;                                              !Remember last ref
;  437          0566  3         END;
;  438          0567  3 !
;  439          0568  3 ! Insert the new reference
;  440          0569  3 !
;  441          0570  3     perform (get_mem (ref$c_fixedsize + .refadr [0], refblk));          !Allocate memory for it
;  442          0571  3     refblk [ref$l_next] = .lastref [ref$l_next];
;  443          0572  3     lastref [ref$l_next] = .refblk;
;  444          0573  3     refblk [ref$w_refflg] = .reflag;
;  445          0574  3     CH$MOVE (.refadr [0] + 1, refadr [0], refblk [ref$b_reflng]);       !Move in reference name
;  446          0575  3     RETURN crf$_success;
;  447          0576  2     END;
;  448          0577  1 END;                                                                   !Of crf$insrtref
```

```
                                0FFC 00000          .ENTRY  CRF$INSRTREF, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 0492
                                                            R10,R11                                     ;
            5E                  0C   C2 00002       SUBL2   #12, SP                                     ;
            5B        04        AC   D0 00005       MOVL    CONTROL, CONTROLREG                         ; 0524
                                AC   10 00009       BSBB    GET_HASH                                    ; 0526
            01                  50   E8 0000B       BLBS    STATUS, 1$                                  ;
                                04   0000E          RET                                                 ;
                                5E   DD 0000F 1$:    PUSHL   SP                                          ; 0529
                          08    AC   DD 00011       PUSHL   KEYADR                                      ;
          FE97    CF            02   FB 00014       CALLS   #2, ADD_KEY                                 ;
            56                  6E   D0 00019       MOVL    CACHENTRY, R6                               ; 0530
                                01   12 0001C       BNEQ    2$                                          ;
                                04   0001E          RET                                                 ;
            57        0C        AC   D0 0001F 2$:    MOVL    REFADR, R7                                  ; 0542
            37        14        AC   E9 00023       BLBC    DEFIND, 4$                                  ; 0536
          04    AE    14        A6   D0 00027       MOVL    20(R6), DEFNAM                              ; 0539
                                0D   13 0002C       BEQL    3$                                          ;
            50        04        BE   9A 0002E       MOVZBL  @DEFNAM, R0                                 ; 0540
                                50   D6 00032       INCL    R0                                          ;
            51        04        AE   D0 00034       MOVL    DEFNAM, R1                                  ;
                          00000G  30   00038       BSBW    FREE_MEM                                    ;
            51        04        AE   9E 0003B 3$:    MOVAB   DEFNAM, R1                                  ; 0542
```

```
                                52              67  9A 0003F          MOVZBL  (R7), R2                                        ;
                                                52  D6 00042          INCL    R2                                              ;
                                50              52  D0 00044          MOVL    R2, R0                                          ;
                                              0000G 30 00047          BSBW    GET_MEM                                         ;
                                63              50  E9 0004A          BLBC    STATUS, 9$                                      ;
                          04    BE              67  52 28 0004D          MOVC3   R2, (R7), @DEFNAM                           ; 0543
                                          14    A6  04 AE D0 00052       MOVL    DEFNAM, 20(R6)                              ; 0544
                                          12    A6  10 AC B0 00057       MOVW    REFLAG, 18(R6)                              ; 0545
                                                6F  11 0005C             BRB     11$                                         ; 0549
                                          08    AE  04 A6 9E 0005E 4$:   MOVAB   4(R6), REFBLK                               ; 0554
                                                58  08 AE D0 00063       MOVL    REFBLK, LASTREF                             ; 0555
                                                54  08 AE D0 00067       MOVL    REFBLK, R4                                  ; 0557
                                          08    AE      64 D0 0006B 5$:  MOVL    (R4), REFBLK                                ;
                                                      2F 13 0006F        BEQL    8$                                          ;
                                                51      67 9A 00071      MOVZBL  (R7), R1                                    ; 0559
                                                54  08 AE D0 00074       MOVL    REFBLK, R4                                  ; 0560
                                                50  06 A4 9A 00078       MOVZBL  6(R4), R0                                   ;
                                                55      01 D0 0007C      MOVL    #1, R5                                      ;
                    50          00      01    A7          51 2D 0007F    CMPC5   R1, 1(R7), #0, R0, 7(R4)                    ;
                                                  07    A4      00085    ;                                                   ;
                                                          03 1A 00087    BGTRU   6$                                          ;
                                                55        01 D9 00089    SBWC    #1, R5                                      ;
                                                56        55 D0 0008C 6$:  MOVL    R5, CH_RESULT                           ;
                                                          08 12 0008F    BNEQ    7$                                          ; 0561
                                                50 00658019 8F D0 00091  MOVL    #6651929, R0                                ; 0562
                                                          04 00098       RET                                                 ;
                                                          05 19 00099 7$:  BLSS    8$                                          ; 0563
                                                58        54 D0 0009B    MOVL    R4, LASTREF                                 ; 0565
                                                          CB 11 0009E    BRB     5$                                          ; 0557
                                                51  08 AE 9E 000A0 8$:  MOVAB   REFBLK, R1                                 ; 0570
                                                50      67 9A 000A4      MOVZBL  (R7), R0                                    ;
                                                50      07 C0 000A7      ADDL2   #7, R0                                      ;
                                                      0000G 30 000AA     BSBW    GET_MEM                                     ;
                                                04        50 E8 000AD    BLBS    STATUS, 10$                                 ;
                                                        0000V 30 000B0 9$:  BSBW    CREFERROR                                ;
                                                          04 000B3       RET                                                 ;
                                                50  08 AE D0 000B4 10$:  MOVL    REFBLK, R0                                  ; 0571
                                                60      68 D0 000B8      MOVL    (LASTREF), (R0)                             ;
                                                68      50 D0 000BB      MOVL    R0, (LASTREF)                               ; 0572
                                          04    A0  10 AC B0 000BE       MOVW    REFLAG, 4(R0)                               ; 0573
                                                51      67 9A 000C3      MOVZBL  (R7), R1                                    ; 0574
                                                        51 D6 000C6      INCL    R1                                          ;
                          06    A0              67  51 28 000C8          MOVC3   R1, (R7), 6(R0)                             ;
                                                50 00658001 8F D0 000CD 11$:  MOVL    #6651905, R0                                ; 0575
                                                          04 000D4       RET                                                 ; 0577

; Routine Size:  213 bytes,    Routine Base:  $CODE$ + 0187
```

```
;  450          0578  1 %SBTTL  'creferror -- call user error routine';
;  451          0579  1 ROUTINE creferror (errorcode) : JSB_1 =
;  452          0580  2 BEGIN
;  453          0581  2
;  454          0582  2 !++
;  455          0583  2 !       Call user error routine on error
;  456          0584  2 !
;  457          0585  2 ! Inputs:
;  458          0586  2 !
;  459          0587  2 !       errorcode       the error code to pass to user
;  460          0588  2 !
;  461          0589  2 !--
;  462          0590  2
;  463          0591  2 EXTERNAL REGISTER
;  464          0592  2     controlreg = 11 : REF BBLOCK;
;  465          0593  2
;  466          0594  2 BIND ROUTINE
;  467          0595  2     user_error_routine = controlreg [crf$l_error];
;  468          0596  2
;  469          0597  2 IF .user_error_routine NEQ 0                          !If user supplied routine
;  470          0598  2     THEN (.user_error_routine) (.errorcode);
;  471          0599  2
;  472          0600  2 RETURN .errorcode
;  473          0601  1 END;                                                  !Of creferror


                                        52  DD 00000 CREFERROR:
                                                     PUSHL   R2                                      ; 0579
                                    52  50  DD 00002         MOVL    R0, R2                          ;
                                40  AB  D5 00005         TSTL    64(CONTROLREG)                  ; 0597
                                    06  13 00008         BEQL    1$                              ;
                                    52  DD 0000A         PUSHL   ERRORCODE                       ; 0598
                                40  BB  01  FB 0000C         CALLS   #1, @64(CONTROLREG)             ;
                                    50  52  D0 00010 1$:      MOVL    ERRORCODE, R0                   ; 0600
                                        04  BA 00013         POPR    #^M<R2>                         ; 0601
                                            05 00015         RSB                                     ;

; Routine Size:  22 bytes,    Routine Base:  $CODE$ + 025C
```

```
;  475          0602  1 %SBTTL  'crf$out -- output the cross reference';
;  476          0603  1 GLOBAL ROUTINE crf$out (control, linwid, linespage1, linesucpag, printind, saveind) =
;  477          0604  2 BEGIN
;  478          0605  2
;  479          0606  2 !++
;  480          0607  2 !       Output the cross reference
;  481          0608  2 !
;  482          0609  2 ! Inputs:
;  483          0610  2 !
;  484          0611  2 !       control         Address of cross-reference control table
;  485          0612  2 !       linwid          Width of listing line
;  486          0613  2 !       linespage1      Number of lines on first page
;  487          0614  2 !       linesucpag      Number of lines on succeeding pages
;  488          0615  2 !       printind        print indicator (crf$k_values, crf$k_vals_refs, crf$k_defs_refs)
;  489          0616  2 !       saveind         save tree indicator (crf$k_save, crf$k_delete)
;  490          0617  2 !
;  491          0618  2 !--
;  492          0619  2
;  493          0620  2 ROUTINE maxfieldwidth (tableaddr) : JSB_1 =
;  494          0621  3 BEGIN
;  495          0622  3
;  496          0623  3 !++
;  497          0624  3 !       Search through the field descriptors for the one with the
;  498          0625  3 !       maximum length and return it.
;  499          0626  3 !
;  500          0627  3 ! Inputs:
;  501          0628  3 !
;  502          0629  3 !       tableaddr               Address of start of field descriptors
;  503          0630  3 !
;  504          0631  3 ! Routine value:
;  505          0632  3 !
;  506          0633  3 !       width of biggest field descriptor
;  507          0634  3 !
;  508          0635  3 !--
;  509          0636  3
;  510          0637  3 LOCAL
;  511          0638  3     wid,
;  512          0639  3     fldptr : REF BBLOCK;
;  513          0640  3
;  514          0641  3 IF .tableaddr EQL 0                             !If no table, return 0
;  515          0642  3  THEN RETURN 0;
;  516          0643  3
;  517          0644  3 fldptr = .tableaddr;
;  518          0645  3 wid = 0;                                        !Set max width to 0
;  519          0646  3
;  520          0647  3 WHILE .fldptr [fld$w_mask] NEQ 0                !Loop through all looking for larger width
;  521          0648  4 DO BEGIN
;  522          0649  4     IF .wid LSSU .fldptr [fld$b_maxlng]         ! and if one is found
;  523          0650  4         THEN wid = .fldptr [fld$b_maxlng];      ! then set max width to that
;  524          0651  4     fldptr = fldptr [fld$t_nxtdesc];            !Link to next field descriptor
;  525          0652  3     END;
;  526          0653  3
;  527          0654  3 RETURN .wid                                     !Return maximum width
;  528          0655  3
;  529          0656  2 END;                                            !Of maxfieldwidth
```

```
                                                50  D5 00000 MAXFIELDWIDTH:
                                                             TSTL    TABLEADDR                                  ; 0641
                                                1B  13 00002         BEQL    4$                                 ;
                                                51  D4 00004         CLRL    WID                                ; 0645
                                                60  B5 00006 1$:      TSTW    (FLDPTR)                           ; 0647
                                                11  13 00008         BEQL    3$                                 ;
        51      03  A0          08             00  ED 0000A         CMPZV   #0, #8, 3(FLDPTR), WID             ; 0649
                                                04  1B 00010         BLEQU   2$                                 ;
                                    51      03  A0  9A 00012         MOVZBL  3(FLDPTR), WID                     ; 0650
                                    50          0C  C0 00016 2$:      ADDL2   #12, FLDPTR                        ; 0651
                                                EB  11 00019         BRB     1$                                 ; 0647
                                    50          51  D0 0001B 3$:      MOVL    WID, R0                            ; 0654
                                                    05 0001E         RSB                                        ;
                                                50  D4 0001F 4$:      CLRL    R0                                 ; 0656
                                                    05 00021         RSB                                        ;

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0272
```

```
;   531          0657  2 ROUTINE formatfield (fieldwidth, outdesc, faocontrol, faoarg) : CALL_4 =
;   532          0658  3 BEGIN
;   533          0659  3
;   534          0660  3 !++
;   535          0661  3 !       Format a field
;   536          0662  3 !
;   537          0663  3 ! Inputs:
;   538          0664  3 !
;   539          0665  3 !       fieldwidth      width of the field
;   540          0666  3 !       outdesc         address of string descriptor for buffer
;   541          0667  3 !       faocontrol      address of fao control string
;   542          0668  3 !       faoarg          argument for fao
;   543          0669  3 !
;   544          0670  3 ! Outputs:
;   545          0671  3 !
;   546          0672  3 !       string is formatted, outdesc is updated.
;   547          0673  3 !
;   548          0674  3 !--
;   549          0675  3
;   550          0676  3 MAP
;   551          0677  3     outdesc : REF BBLOCK;
;   552          0678  3
;   553          0679  3 LOCAL
;   554          0680  3     length : WORD;
;   555          0681  3
;   556          0682  3 EXTERNAL REGISTER
;   557          0683  3     controlreg = 11 : REF BBLOCK;
;   558          0684  3
;   559          0685  3 perform (SYS$FAO (.faocontrol, length, .outdesc, .faoarg));          !Format the field
;   560          0686  3 outdesc [dsc$w_length] = .outdesc [dsc$w_length] - .fieldwidth;      !Adjust the descriptor
;   561          0687  3 outdesc [dsc$a_pointer] = .outdesc [dsc$a_pointer] + .fieldwidth;
;   562          0688  3
;   563          0689  3 RETURN true
;   564          0690  3
;   565          0691  2 END;                                    !Of formatfield


                                001C 00000 FORMATFIELD:
                                           .WORD    Save R2,R3,R4                          ; 0657
                          5E        04 C2 00002      SUBL2    #4, SP                        ;
                       10 AC        DD 00005         PUSHL    FAOARG                        ; 0685
                          52 08 AC  D0 00008         MOVL     OUTDESC, R2                   ;
                             52     DD 0000C         PUSHL    R2                            ;
                          08 AE     9F 0000E         PUSHAB   LENGTH                        ;
                          0C AC     DD 00011         PUSHL    FAOCONTROL                    ;
                00000000G 00        04 FB 00014      CALLS    #4, SYS$FAO                   ;
                          03        50 E8 0001B      BLBS     STATUS, 1$                    ;
                                    A8 10 0001E      BSBB     CREFERROR                     ;
                                    04 00020         RET                                    ;
                          62 04 AC  A2 00021 1$:     SUBW2    FIELDWIDTH, (R2)              ; 0686
                       04 A2 04 AC  C0 00025         ADDL2    FIELDWIDTH, 4(R2)             ; 0687
                          50        01 D0 0002A      MOVL     #1, R0                        ; 0689
                                    04 0002D         RET                                    ; 0691
```

; Routine Size: 46 bytes, Routine Base: $CODE$ + 0294

```
;  567          0692  2 ROUTINE findformat (outdesc, keyflag, formatdesc) : CALL_3 =
;  568          0693  2 BEGIN
;  569          0694  2
;  570          0695  2 !++
;  571          0696  2 !       Find format associated with a bit mask and the format the field
;  572          0697  2 !
;  573          0698  2 ! Inputs:
;  574          0699  2 !
;  575          0700  2 !       outdesc         address of string descriptor for outpt buffer
;  576          0701  2 !       keyflag         flags to find format for
;  577          0702  2 !       formatdesc      address of start of cref field descriptors
;  578          0703  2 !
;  579          0704  2 ! Outputs:
;  580          0705  2 !
;  581          0706  2 !       field is formatted, outdesc is updated.
;  582          0707  2 !
;  583          0708  2 !--
;  584          0709  2
;  585          0710  2 LOCAL
;  586          0711  2     fldptr : REF BBLOCK,
;  587          0712  2     curfld : REF BBLOCK,
;  588          0713  2     mask,
;  589          0714  2     comflags;
;  590          0715  2
;  591          0716  2 EXTERNAL REGISTER
;  592          0717  2     controlreg = 11 : REF BBLOCK;
;  593          0718  2
;  594          0719  2 comflags = NOT .keyflag;                          !Get complement of flags
;  595          0720  2 comflags = .comflags AND %X'FFFF';                ! trimmed to a word
;  596          0721  2 fldptr = .formatdesc;                            !Point to the format descriptors
;  597          0722  2 curfld = .fldptr;                                 ! ...
;  598          0723  2
;  599          0724  3 WHILE (mask = .curfld [fld$w_mask]) NEQ 0         !Look at all field descriptors
;  600          0725  4 DO BEGIN
;  601          0726  4     mask = .mask AND fld$m_usrbits;               !Get mask this descriptor
;  602          0727  5     IF (IF NOT .curfld [fld$b_set_clr]            !Check if set or clear bits
;  603          0728  5                 THEN (.mask AND (NOT .comflags)) EQL 0   ! and compare
;  604          0729  5                 ELSE (.mask  AND (NOT .keyflag)) EQL 0
;  605          0730  5         )
;  606          0731  4         THEN fldptr = .curfld;                    !If it matches then remember it
;  607          0732  4     curfld = curfld [fld$t_nxtdesc];              !Link to next field
;  608          0733  3     END;
;  609          0734  3
;  610          0735  3 RETURN formatfield (.fldptr [fld$b_maxlng], .outdesc,   !Format the field and return
;  611          0736  3                     fldptr [fld$b_faodsc], .keyflag)
;  612          0737  3
;  613          0738  2 END;                                              !Of findformat


                                003C 00000 FINDFORMAT:
                                                .WORD   Save R2,R3,R4,R5                                        ; 0692
                        55      08   AC  D2 00002        MCOML   KEYFLAG, R5                                     ; 0719
                        50           55  D0 00006        MOVL    R5, COMFLAGS                                    ;
                        50           50  3C 00009        MOVZWL  COMFLAGS, COMFLAGS                              ; 0720
```

```
                                53      0C   AC  D0 0000C          MOVL    FORMATDESC, FLDPTR                      ; 0721
                                51           53  D0 00010          MOVL    FLDPTR, CURFLD                          ; 0722
                                54           61  3C 00013 1$:      MOVZWL  (CURFLD), MASK                          ; 0724
                                             1E  13 00016          BEQL    5$                                      ;
            54          54      0F           00  EF 00018          EXTZV   #0, #15, MASK, MASK                     ; 0726
                                08      02   A1  E8 0001D          BLBS    2(CURFLD), 2$                           ; 0727
                                52           50  D2 00021          MCOML   COMFLAGS, R2                            ; 0728
                                52           54  D3 00024          BITL    MASK, R2                                ;
                                             03  11 00027          BRB     3$                                      ;
                                55           54  D3 00029 2$:      BITL    MASK, R5                                ; 0729
                                             03  12 0002C 3$:      BNEQ    4$                                      ;
                                53           51  D0 0002E          MOVL    CURFLD, FLDPTR                          ; 0731
                                51           0C  C0 00031 4$:      ADDL2   #12, CURFLD                             ; 0732
                                             DD  11 00034          BRB     1$                                      ; 0724
                                        08   AC  DD 00036 5$:      PUSHL   KEYFLAG                                 ; 0736
                                        04   A3  9F 00039          PUSHAB  4(FLDPTR)                               ;
                                        04   AC  DD 0003C          PUSHL   OUTDESC                                 ;
                                7E      03   A3  9A 0003F          MOVZBL  3(FLDPTR), -(SP)                        ;
                            8B  AF              04  FB 00043          CALLS   #4, FORMATFIELD                         ;
                                                    04 00047          RET                                     ; 0738

; Routine Size:  72 bytes,    Routine Base:  $CODE$ + 02C2
```

```
;  615          0739  2 ROUTINE outputline (linedesc, protodesc)  : CALL_2 =
;  616          0740    BEGIN
;  617          0741
;  618          0742    !++
;  619          0743    !       Output line by calling user routine
;  620          0744    !
;  621          0745    ! Inputs:
;  622          0746    !
;  623          0747    !       linedesc                address of string descriptor for line
;  624          0748    !       protodesc               address of original string descriptor for line
;  625          0749    !
;  626          0750    ! Outputs:
;  627          0751    !
;  628          0752    !       linedesc                reset to protodesc
;  629          0753    !       buffer pointed to by protodesc is blank-filled.
;  630          0754    !
;  631          0755    !       user output routine is called.
;  632          0756    !
;  633          0757    !--
;  634          0758
;  635          0759    MAP
;  636          0760        linedesc : REF BBLOCK,
;  637          0761        protodesc : REF BBLOCK;
;  638          0762
;  639          0763    LOCAL
;  640          0764        desc : BBLOCK [dsc$c_s_bln];
;  641          0765
;  642          0766    EXTERNAL REGISTER
;  643          0767        controlreg = 11 : REF BBLOCK;
;  644          0768
;  645          0769    desc [dsc$w_length] = .protodesc [dsc$w_length] -       !Set descriptor to length of line to print
;  646          0770                          .linedesc [dsc$w_length];
;  647          0771    desc [dsc$a_pointer] = .protodesc [dsc$a_pointer];
;  648          0772    CH$MOVE (dsc$c_s_bln, .protodesc, .linedesc);           !Reset the descriptor
;  649          0773
;  650          0774    (.controlreg [crf$l_output]) (desc);                    !Call user routine to output line
;  651          0775
;  652          0776    CH$FILL (%ASCII ' ', .desc [dsc$w_length],              !Blank fill used part of line
;  653          0777                         .desc [dsc$a_pointer]);
;  654          0778
;  655          0779    RETURN true
;  656          0780
;  657          0781  2 END;                                          !Of outputline
```

```
                                003C 00000 OUTPUTLINE:
                                           .WORD   Save R2,R3,R4,R5                ; 0739
                          5E    08  C2 00002       SUBL2   #8, SP                  ;
                          50 08 AC  D0 00005       MOVL    PROTODESC, R0           ; 0769
                       6E 60 04 BC  A3 00009       SUBW3   @LINEDESC, (R0), DESC   ; 0770
                    04 AE    04 A0  D0 0000E       MOVL    4(R0), DESC+4           ; 0771
                04  BC 60        08  28 00013       MOVC3   #8, (R0), @LINEDESC     ; 0772
                                5E  DD 00018       PUSHL   SP                      ; 0774
                    44  BB      01  FB 0001A       CALLS   #1, @68(CONTROLREG)     ;
```

```
            6E          20          6E          00 2C 0001E         MOVC5   #0, (SP), #32, DESC, @DESC+4            ; 0777
                                    04          BE    00023         .                                               .
                                    50          01 D0 00025         MOVL    #1, R0                                  ; 0779
                                                   04 00028         RET                                             ; 0781

; Routine Size:  41 bytes,    Routine Base:  $CODE$ + 030A
```

```
;  659     0782  2 ROUTINE formatkeyval (keyadr, linedesc) : CALL_2 =
;  660     0783  2 BEGIN
;  661     0784  2
;  662     0785  2 !++
;  663     0786  2 !       Format key1/key2/val1/val2 fields for a key
;  664     0787  2 !
;  665     0788  2 ! Inputs:
;  666     0789  2 !
;  667     0790  2 !       keyadr          Address of ASCIC string or value of binary key
;  668     0791  2 !       linedesc        Address of string descriptor for output line
;  669     0792  2 !
;  670     0793  2 ! Outputs:
;  671     0794  2 !
;  672     0795  2 !       fields are formatted into output line.
;  673     0796  2 !
;  674     0797  2 !--
;  675     0798  2
;  676     0799  2 MAP
;  677     0800  2     keyadr : REF BBLOCK,
;  678     0801  2     linedesc : REF BBLOCK;
;  679     0802  2
;  680     0803  2 LOCAL
;  681     0804  2     fieldwidth,
;  682     0805  2     curfld : REF BBLOCK;
;  683     0806  2
;  684     0807  2 EXTERNAL REGISTER
;  685     0808  2     controlreg = 11 : REF BBLOCK;
;  686     0809  2
;  687     0810  2 IF (curfld = .controlreg [crf$l_k1fmtbl]) NEQ 0                  !Get field descriptor for key 1
;  688     0811  2     THEN formatfield (.curfld [fld$b_maxlng], .linedesc,        ! and format the field
;  689     0812  2                       curfld [fld$b_faodsc], .keyadr [key$l_keyadr]);
;  690     0813  2
;  691     0814  2 IF (curfld = .controlreg [crf$l_k2fmtbl]) NEQ 0                  !If key2 field descriptor present
;  692     0815  2 THEN findformat (.linedesc, .keyadr [key$w_valflg], .curfld);    ! then find format and output it
;  693     0816  2
;  694     0817  2 IF (curfld = .controlreg [crf$l_v1fmtbl]) NEQ 0                  !If val1 field descriptor present
;  695     0818  2 THEN IF .keyadr [key$l_valadr] EQL 0                            ! then handle 0-address
;  696     0819  3     THEN BEGIN
;  697     0820  3         fieldwidth = .curfld [fld$b_maxlng];                    ! by getting the field width
;  698     0821  3         linedesc [dsc$w_length] = .linedesc [dsc$w_length] - .fieldwidth; ! and just skipping the field
;  699     0822  3         linedesc [dsc$a_pointer] = .linedesc [dsc$a_pointer] + .fieldwidth;
;  700     0823  3         END
;  701     0824  2     ELSE formatfield (.curfld [fld$b_maxlng], .linedesc, curfld [fld$b_faodsc], !otherwise format val1
;  702     0825  2                                  .(.keyadr [key$l_valadr]));
;  703     0826  2
;  704     0827  2 IF (curfld = .controlreg [crf$l_v2fmtbl]) NEQ 0                  !Format val2 field if present
;  705     0828  2 THEN findformat (.linedesc, .keyadr [key$w_valflg], .curfld);
;  706     0829  2
;  707     0830  2 RETURN true
;  708     0831  2
;  709     0832  2 END;                                            !Of formatkeyval


                                        000C 00000 FORMATKEYVAL:
```

```
                                                                .WORD   Save R2,R3                                      ; 0782
                53      FF5B    CF  9E 00002                    MOVAB   FORMATFIELD, R3
                52        48    AB  D0 00007                    MOVL    72(CONTROLREG), CURFLD                          ; 0810
                                14  13 0000B                    BEQL    1$
                50        04    AC  D0 0000D                    MOVL    KEYADR, R0                                      ; 0812
                          08    A0  DD 00011                    PUSHL   8(R0)
                          04    A2  9F 00014                    PUSHAB  4(CURFLD)
                          08    AC  DD 00017                    PUSHL   LINEDESC
                7E        03    A2  9A 0001A                    MOVZBL  3(CURFLD), -(SP)
                63              04  FB 0001E                    CALLS   #4, FORMATFIELD
                52        4C    AB  D0 00021 1$:                MOVL    76(CONTROLREG), CURFLD                          ; 0814
                                11  13 00025                    BEQL    2$
                                52  DD 00027                    PUSHL   CURFLD                                          ; 0815
                50        04    AC  D0 00029                    MOVL    KEYADR, R0
                7E        10    A0  3C 0002D                    MOVZWL  16(R0), -(SP)
                          08    AC  DD 00031                    PUSHL   LINEDESC
            2E  A3              03  FB 00034                    CALLS   #3, FINDFORMAT
                52        50    AB  D0 00038 2$:                MOVL    80(CONTROLREG), CURFLD                          ; 0817
                                25  13 0003C                    BEQL    4$
                50        04    AC  7D 0003E                    MOVQ    KEYADR, R0                                      ; 0818
                          0C    A0  D5 00042                    TSTL    12(R0)
                                0D  12 00045                    BNEQ    3$
                50        03    A2  9A 00047                    MOVZBL  3(CURFLD), FIELDWIDTH                           ; 0820
                61              5C  A2 0004B                    SUBW2   FIELDWIDTH, (R1)                                ; 0821
            04  A1              50  C0 0004E                    ADDL2   FIELDWIDTH, 4(R1)                               ; 0822
                                0F  11 00052                    BRB     4$                                              ; 0818
                          0C    B0  DD 00054 3$:                PUSHL   @12(R0)                                         ; 0825
                          04    A2  9F 00057                    PUSHAB  4(CURFLD)                                       ; 0824
                                51  DD 0005A                    PUSHL   R1
                7E        03    A2  9A 0005C                    MOVZBL  3(CURFLD), -(SP)
                63              04  FB 00060                    CALLS   #4, FORMATFIELD
                52        54    AB  D0 00063 4$:                MOVL    84(CONTROLREG), CURFLD                          ; 0827
                                11  13 00067                    BEQL    5$
                                52  DD 00069                    PUSHL   CURFLD                                          ; 0828
                50        04    AC  D0 0006B                    MOVL    KEYADR, R0
                7E        10    A0  3C 0006F                    MOVZWL  16(R0), -(SP)
                          08    AC  DD 00073                    PUSHL   LINEDESC
            2E  A3              03  FB 00076                    CALLS   #3, FINDFORMAT
                50              01  D0 0007A 5$:                MOVL    #1, R0                                          ; 0830
                                    04 0007D                    RET                                                     ; 0832

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 0333
```

```
;  711          0833  2 !
;  712          0834  2 ! Main body of CRF$OUT
;  713          0835  2 !
;  714          0836  2
;  715          0837  2 MAP
;  716          0838  2     control : REF BBLOCK;
;  717          0839  2
;  718          0840  2 LOCAL
;  719          0841  2     linebuf : BBLOCK [crf$c_maxlinwid + crf$c_linextra],
;  720          0842  2     sortedkeys : REF VECTOR [,LONG],
;  721          0843  2     refsperline,
;  722          0844  2     refindent,
;  723          0845  2     fldesc : REF BBLOCK,
;  724          0846  2     curkey : REF BBLOCK,
;  725          0847  2     refdesc : REF BBLOCK,
;  726          0848  2     refcount : REF BBLOCK,
;  727          0849  2     refsize,
;  728          0850  2     keyvalsiz,
;  729          0851  2     linesthispage,
;  730          0852  2     linecount,
;  731          0853  2     curlinedesc : BBLOCK [dsc$c_s_bln],
;  732          0854  2     linedesc : BBLOCK [dsc$c_s_bln];
;  733          0855  2
;  734          0856  2 GLOBAL REGISTER
;  735          0857  2     controlreg = 11 : REF BBLOCK;
;  736          0858  2
;  737          0859  2 !
;  738          0860  2 ! If nothing has been entered into the cross ref then quit now
;  739          0861  2 !
;  740          0862  2
;  741          0863  2 controlreg = .control;
;  742          0864  2 IF .controlreg [crf$l_hasht] EQL 0
;  743          0865  2     OR .controlreg [crf$l_entries] EQL 0
;  744          0866  2     THEN RETURN true;
;  745          0867  2
;  746          0868  2 !
;  747          0869  2 ! Make sure output routine is present
;  748          0870  2 !
;  749          0871  2
;  750          0872  2 IF .controlreg [crf$l_output] EQL 0
;  751          0873  2     THEN RETURN creferror (crf$_no_outrtn);
;  752          0874  2
;  753          0875  2 !
;  754          0876  2 ! Initialize output line to blanks, set up descriptors for line
;  755          0877  2 !
;  756          0878  2
;  757          0879  2 CH$FILL (%ASCII ' ', crf$c_maxlinwid + crf$c_linextra, linebuf);
;  758          0880  2 linedesc [dsc$w_length] = .linwid;
;  759          0881  2 linedesc [dsc$a_pointer] = linebuf;
;  760          0882  2 CH$MOVE (dsc$c_s_bln, linedesc, curlinedesc);
;  761          0883  2 !
;  762          0884  2 ! Sort the hash table
;  763          0885  2 !
;  764          0886  2 perform (get_mem (.controlreg [crf$l_entries] * 4, sortedkeys));
;  765          0887  2 sort_hash_table (crf$c_hashsize, .sortedkeys);
;  766          0888  2 !
;  767          0889  2 ! Determine widths of various parts of line
```

```
;   768         0890  2 !
;   769         0891  2 refindent = 0;
;   770         0892  2
;   771         0893  2 IF (fldesc = .controlreg [crf$l_k1fmtbl]) NEQ 0
;   772         0894  2     THEN refindent = .fldesc [fld$b_maxlng];
;   773         0895  2
;   774         0896  2 IF (fldesc = .controlreg [crf$l_k2fmtbl]) NEQ 0
;   775         0897  2     THEN refindent = .refindent + maxfieldwidth (.fldesc);
;   776         0898  2
;   777         0899  2 IF (fldesc = .controlreg [crf$l_v1fmtbl]) NEQ 0
;   778         0900  2     THEN refindent = .refindent + .fldesc [fld$b_maxlng];
;   779         0901  2
;   780         0902  2 IF (fldesc = .controlreg [crf$l_v2fmtbl]) NEQ 0
;   781         0903  2     THEN refindent = .refindent + maxfieldwidth (.fldesc);
;   782         0904  2 !
;   783         0905  2 ! Make sure that key1/key2/val1/val2 can fit on line
;   784         0906  2 !
;   785         0907  2 IF .refindent GTRU .linwid OR .linwid GTRU crf$c_maxlinwid
;   786         0908  2     THEN RETURN crf$_linelng;
;   787         0909  2
;   788         0910  2 keyvalsiz = .refindent;                                 !Remember width of key1/key2/val1/val2
;   789         0911  2
;   790         0912  2 linesthispage = .linespage1;                            !Set number of lines on first page
;   791         0913  2
;   792         0914  2 !
;   793         0915  2 ! If just printing keys and values, then do it all now
;   794         0916  2 !
;   795         0917  2
;   796         0918  2 IF .printind EQL crf$k_values
;   797         0919  3 THEN BEGIN
;   798         0920  3     LOCAL
;   799         0921  3         keypointer : VECTOR [crf$c_maxcol, LONG],
;   800         0922  3         lwidth : VECTOR [2],
;   801         0923  3         keysperline,
;   802         0924  3         thispointer : REF BBLOCK,
;   803         0925  3         blanksbtwncol,
;   804         0926  3         columnsleft;
;   805         0927  3
;   806         0928  3     BUILTIN EDIV;
;   807         0929  3
;   808         0930  3 !
;   809         0931  3 ! Figure the number of keys per line and the number of spaces
;   810         0932  3 ! between columns.
;   811         0933  3 !
;   812         0934  3
;   813         0935  3     lwidth [0] = .linwid;                               !Put width into a quadword
;   814         0936  3     lwidth [1] = 0;
;   815         0937  3     EDIV (keyvalsiz, lwidth, keysperline, columnsleft); !divide to get number of columns, and spaces
;   816         0938  3     keysperline = .keysperline - 1;                     !One less column than keys
;   817         0939  3     IF (blanksbtwncol = .columnsleft / .keysperline) EQL 0  !figure blanks between columns
;   818         0940  4     THEN BEGIN
;   819         0941  4         keysperline = .keysperline - 1;                 !if none, then one less key per line
;   820         0942  4         columnsleft = .columnsleft + .keyvalsiz;        !adjust columns left
;   821         0943  4         blanksbtwncol = .columnsleft / .keysperline;    !Recompute blanks between columns
;   822         0944  4         END;
;   823         0945  3     keysperline = .keysperline + 1;                     !readjust number of keys per line
;   824         0946  3
```

```
;   825     0947  3     CH$FILL (0, (.keysperline - 1) * 4, keypointer [1]);
;   826     0948  3     keypointer [0] = 1;
;   827     0949  3 !
;   828     0950  3 ! Loop until all are printed
;   829     0951  3 !
;   830     0952  3     WHILE .keypointer [0] LEQ .controlreg [crf$l_entries]
;   831     0953  4     DO BEGIN
;   832     0954  4         !
;   833     0955  4         !  Find key that goes in each column
;   834     0956  4         !
;   835     0957  4         INCRU col FROM 1 TO .keysperline - 1
;   836     0958  4         DO keypointer [.col] = .keypointer [.col - 1]  + .linesthispage;
;   837     0959  4         !
;   838     0960  4         ! Now loop over all lines on page
;   839     0961  4         !
;   840     0962  4         INCRU currentline FROM 1 TO .linesthispage
;   841     0963  5         DO BEGIN
;   842     0964  5
;   843     0965  5             !
;   844     0966  5             ! Loop over each column on line
;   845     0967  5             !
;   846     0968  5             INCRU col FROM 0 TO .keysperline - 1
;   847     0969  5             DO IF .keypointer [.col] LEQ .controlreg [crf$l_entries]
;   848     0970  5                 AND (thispointer = .sortedkeys [.keypointer [.col] - 1]) NEQ 0
;   849     0971  6             THEN BEGIN
;   850     0972  6
;   851     0973  6                 formatkeyval (.thispointer, curlinedesc);           !format this key
;   852     0974  6                 IF .col NEQ .keysperline - 1                        !If not last key on line, add spaces
;   853     0975  7                 THEN BEGIN
;   854     0976  7                     curlinedesc [dsc$w_length] = .curlinedesc [dsc$w_length] -
;   855     0977  7                                                  .blanksbtwncol;
;   856     0978  7                     curlinedesc [dsc$a_pointer] = .curlinedesc [dsc$a_pointer] +
;   857     0979  7                                                   .blanksbtwncol;
;   858     0980  6                     END;
;   859     0981  6                 keypointer [.col] = .keypointer [.col] + 1;
;   860     0982  5                 END;
;   861     0983  5             outputline (curlinedesc, linedesc);                     !Output the line
;   862     0984  5             IF .keypointer [0] GTR .controlreg [crf$l_entries] ! If no more entries
;   863     0985  5                 THEN EXITLOOP;
;   864     0986  4             END;                                                    .Of loop on lines per page
;   865     0987  4
;   866     0988  4         keypointer [0] = .keypointer [.keysperline - 1];            !Set index to next key
;   867     0989  4         CH$FILL (0, (.keysperline - 1) * 4, keypointer [1]);        !Zero rest of array
;   868     0990  4         linesthispage = .linesucpag;                                !Set count of lines per page
;   869     0991  3         END;                                                        !of WHILE loop
;   870     0992  3
;   871     0993  3     END
;   872     0994  3
;   873     0995  3 !
;   874     0996  3 ! Printing at least the refs, so only 1 key per line
;   875     0997  3 !
;   876     0998  3
;   877     0999  3 ELSE BEGIN
;   878     1000  3
;   879     1001  3 !
;   880     1002  3 ! Determine width of a reference
;   881     1003  3 !
```

```
;  882          1004  3
;  883          1005  3      refsize = 0;
;  884          1006  3
;  885          1007  3      IF (fldesc = .controlreg [crf$l_r1fmtbl]) NEQ 0
;  886          1008  3          THEN refsize = maxfieldwidth (.fldesc);
;  887          1009  3
;  888          1010  3      IF (fldesc = .controlreg [crf$l_r2fmtbl]) NEQ 0
;  889          1011  3          THEN refsize = .refsize + .fldesc [fld$b_maxlng];
;  890          1012  3
;  891          1013  3  !
;  892          1014  3  ! If printing defs, then update indent to start of references
;  893          1015  3  !
;  894          1016  3
;  895          1017  3      IF .printind EQL crf$k_defs_refs
;  896          1018  3          THEN refindent = .refindent + .refsize;
;  897          1019  3
;  898          1020  3      IF .refsize NEQ 0
;  899          1021  4      THEN BEGIN
;  900          1022  4          refsperline = (.linwid - .refindent) / .refsize;        !Compute no. refs. per line
;  901          1023  4          IF .refsperline LEQ 0
;  902          1024  4              THEN RETURN crf$_linelng;
;  903          1025  4          END
;  904          1026  3      ELSE refsperline = 1;
;  905          1027  3
;  906          1028  3  !
;  907          1029  3  ! Now loop over all the keys
;  908          1030  3  !
;  909          1031  3
;  910          1032  3      INCRU i FROM 0 TO .controlreg [crf$l_entries] - 1
;  911          1033  4      DO BEGIN
;  912          1034  4
;  913          1035  4          curkey = .sortedkeys [.i];
;  914          1036  4          formatkeyval (.curkey, curlinedesc);                    !Format key1/key2/val1/val2
;  915          1037  4
;  916          1038  4          IF .printind EQL crf$k_defs_refs                        !Printing defining reference?
;  917          1039  5          THEN BEGIN
;  918          1040  5              IF (fldesc = .controlreg [crf$l_r1fmtbl]) NEQ 0
;  919          1041  5                  THEN findformat (curlinedesc, .curkey [key$w_defflg], .fldesc);
;  920          1042  5              IF (fldesc = .controlreg [crf$l_r2fmtbl]) NEQ 0
;  921          1043  5                  THEN formatfield (.fldesc [fld$b_maxlng], curlinedesc,
;  922          1044  5                                    fldesc [fld$b_faodsc],
;  923          1045  6                                    (IF (refdesc = .curkey [key$l_defnam]) NEQ 0    !use definer name if there
;  924          1046  6                                        THEN .refdesc
;  925          1047  5                                        ELSE curkey [key$l_defnam]));   !otherwise point to a zero
;  926          1048  4              END;
;  927          1049  4  !
;  928          1050  4  ! format the references
;  929          1051  4  !
;  930          1052  4          refdesc = .curkey [key$l_reflist];
;  931          1053  4          refcount = .refsperline;
;  932          1054  4          curlinedesc [dsc$w_length] = .linwid - .refindent;
;  933          1055  4          curlinedesc [dsc$a_pointer] = linebuf + .refindent;
;  934          1056  4
;  935          1057  4          WHILE .refdesc NEQ 0                                    !Loop over all references
;  936          1058  5          DO BEGIN
;  937          1059  5              IF (fldesc = .controlreg [crf$l_r1fmtbl]) NEQ 0
;  938          1060  5                  THEN findformat (curlinedesc, .refdesc [ref$w_refflg], .fldesc);
```

```
;  939          1061  5              IF (fldesc = .controlreg [crf$l_r2fmtbl]) NEQ 0
;  940          1062  5                  THEN formatfield (.fldesc [fld$b_maxlng], curlinedesc,
;  941          1063  5                                    fldesc [fld$b_faodsc], refdesc [ref$b_refing]);
;  942          1064  5              refcount = .refcount - 1;
;  943          1065  5              refdesc = .refdesc [ref$l_next];
;  944          1066  5              IF .refcount EQL 0
;  945          1067  5                  AND .refdesc NEQ 0
;  946          1068  6                  THEN BEGIN
;  947          1069  6                      outputline (curlinedesc, linedesc);
;  948          1070  6                      curlinedesc [dsc$w_length] = .curlinedesc [dsc$w_length] -
;  949          1071  6                                                   .refindent;
;  950          1072  6                      curlinedesc [dsc$a_pointer] = .curlinedesc [dsc$a_pointer] +
;  951          1073  6                                                   .refindent;
;  952          1074  6                      refcount = .refsperline;
;  953          1075  5                      END;
;  954          1076  4              END;
;  955          1077  4 !
;  956          1078  4 ! Output the line we've done
;  957          1079  4 !
;  958          1080  4          outputline (curlinedesc, linedesc);
;  959          1081  4          linecount = .linecount - 1;
;  960          1082  4          IF .linecount EQL 0
;  961          1083  5          THEN BEGIN
;  962          1084  5              linecount = .linesucpag;
;  963          1085  5              linesthispage = .linesucpag;
;  964          1086  4              END;
;  965          1087  4
;  966          1088  3          END;                                    !Of WHILE loop
;  967          1089  2      END;                                        !Of printing at least refs
;  968          1090  2
;  969          1091  2 !
;  970          1092  2 ! If requested, deallocate the memory we used for the cross-reference
;  971          1093  2 !
;  972          1094  2
;  973          1095  2 IF .saveind EQL crf$k_delete
;  974          1096  3 THEN BEGIN
;  975          1097  3     LOCAL
;  976          1098  3         defnam : REF VECTOR [,BYTE],
;  977          1099  3         thisref : REF BBLOCK,
;  978          1100  3         lastref : REF BBLOCK;
;  979          1101  3
;  980          1102  3 !
;  981          1103  3 ! Get the addresses of the key def. blocks out of the sorted list
;  982          1104  3 !
;  983          1105  3     INCRU i FROM 0 TO .controlreg [crf$l_entries] - 1
;  984          1106  4     DO BEGIN
;  985          1107  4         curkey = .sortedkeys [.i];
;  986          1108  4         IF (defnam = .curkey [key$l_defnam]) NEQ 0          !Deallocate definer name if present
;  987          1109  4             THEN free_mem (.defnam [0]+1, defnam [0]);
;  988          1110  4 !
;  989          1111  4 ! Deallocate reference list
;  990          1112  4 !
;  991          1113  4         thisref = .curkey [key$l_reflist];
;  992          1114  4         WHILE .thisref NEQ 0
;  993          1115  5         DO BEGIN
;  994          1116  5             lastref = .thisref;
;  995          1117  5             thisref = .thisref [ref$l_next];
```

```
;  996          1118  5              free_mem (ref$c_fixedsize + .lastref [ref$b_refln], .lastref);
;  997          1119  4              END;
;  998          1120  4 !
;  999          1121  4 ! Deallocate key def. entry
; 1000          1122  4 !
; 1001          1123  4          free_mem (key$c_length, .curkey);
; 1002          1124  3          END;
; 1003          1125  3 !
; 1004          1126  3 ! Deallocate the hash table
; 1005          1127  3 !
; 1006          1128  3      free_mem (crf$c_hashsize*4, .controlreg [crf$l_hasht]);
; 1007          1129  3      controlreg [crf$l_hasht] = 0;
; 1008          1130  3      controlreg [crf$l_entries] = 0;
; 1009          1131  3      controlreg [crf$l_lastkey] = 0;
; 1010          1132  2      END;
; 1011          1133  2
; 1012          1134  2 !
; 1013          1135  2 ! Deallocate the sorted list table
; 1014          1136  2 !
; 1015          1137  2
; 1016          1138  2 free_mem (.controlreg [crf$l_entries] * 4, .sortedkeys);
; 1017          1139  2
; 1018          1140  2 RETURN true
; 1019          1141  2
; 1020          1142  1 END;                                            !Of crf$out
; INFO#250                L1:1081
; Referenced LOCAL symbol LINECOUNT is probably not initialized
```

```
                                                 0FFC 00000          .ENTRY  CRF$OUT, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,- ; 0603
                                                                             R11
                                           5E    FE44    CE  9E 00002        MOVAB   -444(SP), SP                        ;
                                           5B      04    AC  D0 00007        MOVL    CONTROL, CONTROLREG                 ; 0863
                                                         6B  D5 0000B        TSTL    (CONTROLREG)                        ; 0864
                                                         03  13 0000D        BEQL    1$                                  ;
                                                       0C    AB  D5 0000F    TSTL    12(CONTROLREG)                      ; 0865
                                                         03  12 00012 1$:    BNEQ    2$                                  ;
                                                    031F     31  00014        BRW     50$                                 ;
                                                       44    AB  D5 00017 2$: TSTL    68(CONTROLREG)                      ; 0872
                                                         09  12 0001A        BNEQ    3$                                  ;
                                           50 0065808A    8F  D0 0001C        MOVL    #6652042, R0                        ; 0873
                                                         2E  11 00023        BRB     4$                                  ;
008E    8F          20                     6E          00  2C 00025 3$:     MOVC5   #0, (SP), #32, #142, LINEBUF        ; 0879
                                                 FF70    CD      0002C                                                   ;
                                  FF60   CD      08    AC  B0 0002F        MOVW    LINWID, LINEDESC                    ; 0880
                                  FF64   CD    FF70    CD  9E 00035        MOVAB   LINEBUF, LINEDESC+4                 ; 0881
                FF68    CD        FF60   CD          08  28 0003C        MOVC3   #8, LINEDESC, CURLINEDESC           ; 0882
                                           51      10    AE  9E 00044        MOVAB   SORTEDKEYS, R1                      ; 0886
                          50        0C   AB          02  78 00048        ASHL    #2, 12(CONTROLREG), R0              ;
                                                 0000G   30  0004D        BSBW    GET_MEM                             ;
                                           04          5U  E8 00050        BLBS    STATUS, 5$                          ;
                                                    FE55   30  00053 4$:     BSBW    CREFERROR                           ;
                                                           04  00056        RET                                         ;
                                                      10    AE  DD 00057 5$:     PUSHL   SORTEDKEYS                      ; 0887
```

```
                                00000000G  8F   DD 0005A          PUSHL   #CRF$C_HASHSIZE                                    ;
                          0000G CF         02   FB 00060          CALLS   #2, SORT_HASH_TABLE                               ;
                                           57   D4 00065          CLRL    REFINDENT                                         ; 0891
                                56       48   AB   D0 00067      MOVL    72(CONTROLREG), FLDESC                            ; 0893
                                           04   13 0006B          BEQL    6$                                                ;
                                57       03   A6   9A 0006D      MOVZBL  3(FLDESC), REFINDENT                              ; 0894
                                56       4C   AB   D0 00071 6$:   MOVL    76(CONTROLREG), FLDESC                            ; 0896
                                           09   13 00075          BEQL    7$                                                ;
                                5C               56   D0 00077      MOVL    FLDESC, R0                                    ; 0897
                                             FE44   30 0007A          BSBW    MAXFIELDWIDTH                                 ;
                                57               50   C0 0007D      ADDL2   R0, REFINDENT                                 ;
                                56       50   AB   D0 00080 7$:   MOVL    80(CONTROLREG), FLDESC                            ; 0899
                                           07   13 00084          BEQL    8$                                                ;
                                50       03   A6   9A 00086      MOVZBL  3(FLDESC), R0                                     ; 0900
                                57               50   C0 0008A      ADDL2   R0, REFINDENT                                 ;
                                56       54   AB   D0 0008D 8$:   MOVL    84(CONTROLREG), FLDESC                            ; 0902
                                           09   13 00091          BEQL    9$                                                ;
                                50               56   D0 00093      MOVL    FLDESC, R0                                    ; 0903
                                             FE28   30 00096          BSBW    MAXFIELDWIDTH                                 ;
                                57               50   C0 00099      ADDL2   R0, REFINDENT                                 ;
                             08 AC               57   D1 0009C 9$:   CMPL    REFINDENT, LINWID                             ; 0907
                                           08   1A 000A0          BGTRU   10$                                               ;
                       00000084 8F       08   AC   D1 000A2      CMPL    LINWID, #132                                      ;
                                           03   1B 000AA 10$:   BLEQU   11$                                               ;
                                         0116   31 000AC          BRW     29$                                               ;
                                52               57   D0 000AF 11$:   MOVL    REFINDENT, KEYVALSIZ                          ; 0910
                                6E       0C   AC   D0 000B2      MOVL    LINESPAGE1, LINESTHISPAGE                         ; 0912
                                         14   AC   D5 000B6      TSTL    PRINTIND                                          ; 0918
                                           03   13 000B9          BEQL    12$                                               ;
                                         00CB   31 000BB          BRW     25$                                               ;
                                14   AE       08   AC   D0 000BE 12$:   MOVL    LINWID, LWIDTH                                ; 0935
                                         18   AE   D4 000C3      CLRL    LWIDTH+4                                          ; 0936
                51              50       14   AE       52   7B 000C6      EDIV    KEYVALSIZ, LWIDTH, KEYSPERLINE, COLUMNSLEFT   ; 0937
                                           50   D7 000CC          DECL    KEYSPERLINE                                       ; 0938
                                5A               51   50   C7 000CE      DIVL3   KEYSPERLINE, COLUMNSLEFT, BLANKSBTWNCOL       ; 0939
                                           09   12 000D2          BNEQ    13$                                               ;
                                           50   D7 000D4          DECL    KEYSPERLINE                                       ; 0941
                                51               52   C0 000D6      ADDL2   KEYVALSIZ, COLUMNSLEFT                           ; 0942
                                5A               51   50   C7 000D9      DIVL3   KEYSPERLINE, COLUMNSLEFT, BLANKSBTWNCOL       ; 0943
                                           50   D6 000DD 13$:   INCL    KEYSPERLINE                                       ; 0945
                                59       FF   A0   9E 000DF      MOVAB   -1(R0), R9                                        ; 0947
                                58               59   02   78 000E3      ASHL    #2, R9, R8                                    ;
                58              00               6E       00   2C 000E7      MOVC5   #0, (SP), #0, R8, KEYPOINTER+4            ;
                                         20   AE           000EC
                                1C   AE           01   D0 000EE      MOVL    #1, KEYPOINTER                                    ; 0948
                                0C   AB       1C   AE   D1 000F2 14$:   CMPL    KEYPOINTER, 12(CONTROLREG)                    ; 0952
                                           03   15 000F7          BLEQ    15$                                               ;
                                         01CB   31 000F9          BRW     43$                                               ;
                                50               01   D0 000FC 15$:   MOVL    #1, COL                                       ; 0958
                                           0A   11 000FF          BRB     17$                                               ;
                      1C AE40       18 AE40           6E   C1 00101 16$:   ADDL3   LINESTHISPAGE, KEYPOINTER-4[COL], -       ;
                                                                                  KEYPOINTER[COL]                            ;
                                           50   D6 00109          INCL    COL                                               ;
                                59               50   D1 0010B 17$:   CMPL    COL, R9                                       ;
                                           F1   1B 0010E          BLEQU   16$                                               ;
                                53               01   D0 00110      MOVL    #1, CURRENTLINE                               ; 0962
                                           59   11 00113          BRB     23$                                               ;
```

```
                                                    52  D4 00115 18$:    CLRL    COL                                          ; 0968
                                                    3A  11 00117          BRB     22$
                                  0C    AB      1C AE42  D1 00119 19$:    CMPL    KEYPOINTER[COL], 12(CONTROLREG)              ; 0969
                                                    30  14 0011F          BGTR    21$
                                        50      1C AE42  D0 00121         MOVL    KEYPOINTER[COL], R0                          ; 0970
                                51      10      AE       04 C3 00126      SUBL3   #4, SORTEDKEYS, R1
                                        0C      AE     6140 D0 00128      MOVL    (R1)[R0], THISPOINTER
                                                    1F  13 00130          BEQL    21$
                                              FF68  CD  9F 00132          PUSHAB  CURLINEDESC                                  ; 0973
                                                10  AE  DD 00136         PUSHL   THISPOINTER
                                      FE44      CF       02 FB 00139      CALLS   #2, FORMATKEYVAL
                                        59              52 D1 0013E      CMPL    COL, R9                                      ; 0974
                                                        0A 13 00141      BEQL    20$
                                      FF68      CD      5A A2 00143      SUBW2   BLANKSBTWNCOL, CURLINEDESC                    ; 0977
                                      FF6C      CD      5A C0 00148      ADDL2   BLANKSBTWNCOL, CURLINEDESC+4                  ; 0979
                                                1C AE42  D6 0014D 20$:    INCL    KEYPOINTER[COL]                              ; 0981
                                                    52  D6 00151 21$:    INCL    COL                                          ; 0969
                                        59          52  D1 00153 22$:    CMPL    COL, R9
                                                    C1  1B 00156          BLEQU   19$
                                              FF60  CD  9F 00158          PUSHAB  LINEDESC                                     ; 0983
                                              FF68  CD  9F 0015C          PUSHAB  CURLINEDESC
                                      FDF4      CF       02 FB 00160      CALLS   #2, OUTPUTLINE
                                        0C      AB    1C AE  D1 00165      CMPL    KEYPOINTER, 12(CONTROLREG)                   ; 0984
                                                    07  14 0016A          BGTR    24$
                                                    53  D6 0016C          INCL    CURRENTLINE                                  ; 0962
                                        6E          53  D1 0016E 23$:    CMPL    CURRENTLINE, LINESTHISPAGE
                                                    A2  1B 00171          BLEQU   18$
                                                1C AE48  9F 00173 24$:    PUSHAB  KEYPOINTER[R8]                               ; 0988
                                          1C    AE       9E D0 00177      MOVL    @(SP)+, KEYPOINTER
                                58        00            6E 00 2C 0017B    MOVC5   #0, (SP), #0, R8, KEYPOINTER+4               ; 0989
                                                    20  AE    00180
                                        6E          10  AC  D0 00182      MOVL    LINESUCPAG, LINESTHISPAGE                    ; 0990
                                                  FF69  31 00186          BRW     14$                                          ; 0952
                                                    51  D4 00189 25$:    CLRL    REFSIZE                                      ; 1005
                                        56          58  AB  D0 0018B      MOVL    88(CONTROLREG), FLDESC                       ; 1007
                                                    09  13 0018F          BEQL    26$
                                        50              56 D0 00191      MOVL    FLDESC, R0                                   ; 1008
                                                  FD2A  30 00194          BSBW    MAXFIELDWIDTH
                                        51              50 D0 00197      MOVL    R0, REFSIZE
                                        56          5C  AB  D0 0019A 26$:    MOVL    92(CONTROLREG), FLDESC                   ; 1010
                                                    07  13 0019E          BEQL    27$
                                        50          03  A6  9A 001A0      MOVZBL  3(FLDESC), R0                                ; 1011
                                        51              50  C0 001A4      ADDL2   R0, REFSIZE
                                                    0C  AE  D4 001A7 27$:    CLRL    12(SP)                                   ; 1017
                                        02          14  AC  D1 001AA      CMPL    PRINTIND, #2
                                                    06  12 001AE          BNEQ    28$
                                                    0C  AE  D6 001B0      INCL    12(SP)
                                        57              51  C0 001B3      ADDL2   REFSIZE, REFINDENT                           ; 1018
                                                        51  D5 001B6 28$:    TSTL    REFSIZE                                  ; 1020
                                                        13  13 001B8      BEQL    30$
                                  50        08    AC      57  C3 001BA      SUBL3   REFINDENT, LINWID, R0                        ; 1022
                                  55              50      51  C7 001BF      DIVL3   REFSIZE, R0, REFSPERLINE
                                                        0B  14 001C3      BGTR    31$                                          ; 1023
                                        50 00658092     8F  D0 001C5 29$:    MOVL    #6652050, R0                             ; 1024
                                                        04  001CC         RET
                                        55              01  D0 001CD 30$:    MOVL    #1, REFSPERLINE                          ; 1026
                            08    AE        0C    AB      01  C3 001D0 31$:    SUBL3   #1, 12(CONTROLREG), 8(SP)                ; 1032
```

```
                04   AE      08   AC                57   C3 001D6          SUBL3   REFINDENT, LINWID, 4(SP)                ; 1054
                                  5A          FF70  CD47 9E 001DC          MOVAB   LINEBUF[REFINDENT], R10                 ; 1055
                                                    54 D4 001E2          CLRL    I                                       ;
                                                  00D7 31 001E4          BRW     42$                                     ;
                                  53              10 BE44 D0 001E7 32$:    MOVL    @SORTEDKEYS[I], CURKEY                  ; 1035
                                              FF68   CD 9F 001EC          PUSHAB  CURLINEDESC                             ; 1036
                                                     53 DD 001F0          PUSHL   CURKEY                                  ;
                                      FD8B    CF       02 FB 001F2          CALLS   #2, FORMATKEYVAL                        ;
                                  3B              0C   AE E9 001F7          BLBC    12(SP), 36$                             ; 1038
                                  56              58   AB D0 001FB          MOVL    88(CONTROLREG), FLDESC                  ; 1040
                                                       0F 13 001FF          BEQL    33$                                     ;
                                                       56 DD 00201          PUSHL   FLDESC                                  ; 1041
                                  7E              12   A3 3C 00203          MOVZWL  18(CURKEY), -(SP)                       ;
                                              FF68   CD 9F 00207          PUSHAB  CURLINEDESC                             ;
                                      FD01    CF       03 FB 0020B          CALLS   #3, FINDFORMAT                          ;
                                  56              5C   AB D0 00210 33$:    MOVL    92(CONTROLREG), FLDESC                  ; 1042
                                                       20 13 00214          BEQL    36$                                     ;
                                  52              14   A3 D0 00216          MOVL    20(CURKEY), REFDESC                     ; 1045
                                                       04 13 0021A          BEQL    34$                                     ;
                                                       52 DD 0021C          PUSHL   REFDESC                                 ; 1046
                                                       06 11 0021E          BRB     35$                                     ;
                                  50              14   A3 9E 00220 34$:    MOVAB   20(CURKEY), R0                          ; 1047
                                                       50 DD 00224          PUSHL   R0                                      ;
                                                   04   A6 9F 00226 35$:    PUSHAB  4(FLDESC)                               ; 1044
                                              FF68   CD 9F 00229          PUSHAB  CURLINEDESC                             ; 1043
                                  7E              03   A6 9A 0022D          MOVZBL  3(FLDESC), -(SP)                        ; 1044
                                      FCAD    CF       04 FB 00231          CALLS   #4, FORMATFIELD                         ;
                                  52              04   A3 D0 00236 36$:    MOVL    4(CURKEY), REFDESC                      ; 1052
                                  58                   55 D0 0023A          MOVL    REFSPERLINE, REFCOUNT                   ; 1053
                                      FF68    CD       04 AE B0 0023D          MOVW    4(SP), CURLINEDESC                      ; 1054
                                      FF6C    CD          5A D0 00243          MOVL    R10, CURLINEDESC+4                      ; 1055
                                                          52 D5 00248 37$:    TSTL    REFDESC                                 ; 1057
                                                          57 13 0024A          BEQL    40$                                     ;
                                  56              58   AB D0 0024C          MOVL    88(CONTROLREG), FLDESC                  ; 1059
                                                       0F 13 00250          BEQL    38$                                     ;
                                                       56 DD 00252          PUSHL   FLDESC                                  ; 1060
                                  7E              04   A2 3C 00254          MOVZWL  4(REFDESC), -(SP)                       ;
                                              FF68   CD 9F 00258          PUSHAB  CURLINEDESC                             ;
                                      FCB0    CF       03 FB 0025C          CALLS   #3, FINDFORMAT                          ;
                                  56              5C   AB D0 00261 38$:    MOVL    92(CONTROLREG), FLDESC                  ; 1061
                                                       13 13 00265          BEQL    39$                                     ;
                                                   06   A2 9F 00267          PUSHAB  6(REFDESC)                              ; 1063
                                                   04   A6 9F 0026A          PUSHAB  4(FLDESC)                               ;
                                              FF68   CD 9F 0026D          PUSHAB  CURLINEDESC                             ; 1062
                                  7E              03   A6 9A 00271          MOVZBL  3(FLDESC), -(SP)                        ; 1063
                                      FC69    CF       04 FB 00275          CALLS   #4, FORMATFIELD                         ;
                                                       58 D7 0027A 39$:    DECL    REFCOUNT                                ; 1064
                                  52                   62 D0 0027C          MOVL    (REFDESC), REFDESC                      ; 1065
                                                       58 D5 0027F          TSTL    REFCOUNT                                ; 1066
                                                       C5 12 00281          BNEQ    37$                                     ;
                                                       52 D5 00283          TSTL    REFDESC                                 ; 1067
                                                       C1 13 00285          BEQL    37$                                     ;
                                              FF60   CD 9F 00287          PUSHAB  LINEDESC                                ; 1069
                                              FF68   CD 9F 0028B          PUSHAB  CURLINEDESC                             ;
                                      FCC5    CF       02 FB 0028F          CALLS   #2, OUTPUTLINE                          ;
                                      FF68    CD          57 A2 00294          SUBW2   REFINDENT, CURLINEDESC                  ; 1071
                                      FF6C    CD          57 C0 00299          ADDL2   REFINDENT, CURLINEDESC+4                ; 1073
```

```
                                58              55  D0 0029E          MOVL    REFSPERLINE, REFCOUNT                   ; 1074
                                                A5  11 002A1          BRB     37$                                     ; 1057
                                        FF60    CD  9F 002A3 40$:     PUSHAB  LINEDESC                                ; 1080
                                        FF68    CD  9F 002A7          PUSHAB  CURLINEDESC
                             FCA9       CF          02  FB 002AB      CALLS   #2, OUTPUTLINE
                                                59  D7 002B0          DECL    LINECOUNT                               ; 1081
                                                08  12 002B2          BNEQ    41$                                     ; 1082
                                59          10  AC  D0 002B4          MOVL    LINESUCPAG, LINECOUNT                   ; 1084
                                6E          10  AC  D0 002B8          MOVL    LINESUCPAG, LINESTHISPAGE               ; 1085
                                                54  D6 002BC 41$:     INCL    I                                       ; 1032
                                  08        AE        54  D1 002BE 42$:     CMPL    I, 8(SP)
                                                      03  1A 002C2          BGTRU   43$
                                                  FF20    31 002C4          BRW     32$
                                           18       AC  D5 002C7 43$:     TSTL    SAVEIND                             ; 1095
                                                5E  12 002CA          BNEQ    49$
                          55        0C      AB        01  C3 002CC          SUBL3   #1, 12(CONTROLREG), R5          ; 1105
                                                52  D4 002D1          CLRL    I
                                                3E  11 002D3          BRB     48$
                                           53       10 BE42 D0 002D5 44$:     MOVL    @SORTEDKEYS[I], CURKEY          ; 1107
                                           56       14  A3 D0 002DA          MOVL    20(CURKEY), DEFNAM              ; 1108
                                                    0B  13 002DE          BEQL    45$
                                           50           66  9A 002E0      MOVZBL  (DEFNAM), R0                    ; 1109
                                                        50  D6 002E3      INCL    R0
                                           51           56  D0 002E5      MOVL    DEFNAM, R1
                                                    0000G   30 002E8      BSBW    FREE_MEM
                                           57       04  A3 D0 002EB 45$:     MOVL    4(CURKEY), THISREF              ; 1113
                                                        57  D5 002EF 46$:     TSTL    THISREF                         ; 1114
                                                        15  13 002F1      BEQL    47$
                                           54           57  D0 002F3      MOVL    THISREF, LASTREF                ; 1116
                                           57           67  D0 002F6      MOVL    (THISREF), THISREF              ; 1117
                                           50       06  A4  9A 002F9      MOVZBL  6(LASTREF), R0                  ; 1118
                                           50           07  C0 002FD      ADDL2   #7, R0
                                           51           54  D0 00300      MOVL    LASTREF, R1
                                                    0000G   30 00303      BSBW    FREE_MEM
                                                        E7  11 00306      BRB     46$                             ; 1114
                                           51           53  D0 00308 47$:     MOVL    CURKEY, R1                      ; 1123
                                           50           18  D0 0030B      MOVL    #24, R0
                                                    0000G   30 0030E      BSBW    FREE_MEM
                                                        52  D6 00311      INCL    I                               ; 1105
                                           55           52  D1 00313 48$:     CMPL    I, R5
                                                        BD  1B 00316      BLEQU   44$
                                           51           6B  D0 00318      MOVL    (CONTROLREG), R1                ; 1128
                                           50 00000000*  8F  D0 0031B      MOVL    #<CRF$C_HASHSIZE*4>, R0
                                                    0000G   30 00322      BSBW    FREE_MEM
                                                   0C   AB  D4 00325      CLRL    12(CONTROLREG)                  ; 1130
                                                        6B  7C 00328      CLRQ    (CONTROLREG)                    ; 1129
                                  50        0C      AB        02  78 0032A 49$:     ASHL    #2, 12(CONTROLREG), R0  ; 1138
                                           51       10  AE  D0 0032F      MOVL    SORTEDKEYS, R1
                                                    0000G   30 00333      BSBW    FREE_MEM
                                           50           01  D0 00336 50$:     MOVL    #1, R0                          ; 1140
                                                            04 00339      RET                                     ; 1142

; Routine Size:  826 bytes,    Routine Base:  $CODE$ + 03B1

; 1021          1143  1 END
; 1022          1144  0 ELUDOM
```

```
;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                         1771  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  .  ABS  .                         0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)


;                                 Library Statistics
;                                          --------- Symbols ---------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1         9776        22         0         581         00:01.0


; Information:  1
; Warnings:     0
; Errors:       0


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

; Size:         1771 code + 0 data bytes
; Run Time:        00:39.8
; Elapsed Time:    01:50.5
; Lines/CPU Min:    1723
; Lexemes/CPU-Min: 17037
; Memory Used:  275 pages
; Compilation Complete
```

CRFOR
LIS

CRFMEM
LIS

CRFMACROS
MAR

CLITABDEF
SDL

FILINPUT
LIS

KEYS
LIS

CRFTFRVEC
LIS

DCLDEF
MDL

CRFMDL
MDL

CREF
LIS

CRFERRMSG
LIS

CRF
SDL

CRF

CRFSUB
LIS

FILOUTPUT
LIS

CRFSHR
MAP

CRFGLOBAL
LIS

CRFMAC
REQ

DCL

DCL
MAP