```
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
DDD         DDD  CCC              XXX         XXX
DDD         DDD  CCC              XXX         XXX
DDD         DDD  CCC              XXX         XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC                    XXX
DDD         DDD  CCC                    XXX
DDD         DDD  CCC                    XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC                 XXX   XXX
DDD         DDD  CCC              XXX         XXX
DDD         DDD  CCC              XXX         XXX
DDD         DDD  CCC              XXX         XXX
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
DDDDDDDDDDDD        CCCCCCCCCCCC  XXX         XXX
```

```
**FILE**ID**ANALYZE                                              F  9




  AAAAAA    NN      NN    AAAAAA    LL          YY      YY  ZZZZZZZZZZ  EEEEEEEEEE
  AAAAAA    NN      NN    AAAAAA    LL          YY      YY  ZZZZZZZZZZ  EEEEEEEEEE
AA      AA  NN      NN  AA      AA  LL          YY      YY          ZZ  EE
AA      AA  NN      NN  AA      AA  LL          YY      YY          ZZ  EE
AA      AA  NNNN    NN  AA      AA  LL            YY  YY          ZZ    EE
AA      AA  NNNN    NN  AA      AA  LL            YY  YY          ZZ    EE
AA      AA  NN  NN  NN  AA      AA  LL              YY          ZZ      EEEEEEEE
AA      AA  NN  NN  NN  AA      AA  LL              YY          ZZ      EEEEEEEE
AAAAAAAAAA  NN    NNNN  AAAAAAAAAA  LL              YY        ZZ        EE
AAAAAAAAAA  NN    NNNN  AAAAAAAAAA  LL              YY        ZZ        EE
AA      AA  NN      NN  AA      AA  LL              YY      ZZ          EE              ....
AA      AA  NN      NN  AA      AA  LL              YY      ZZ          EE              ....
AA      AA  NN      NN  AA      AA  LLLLLLLLLL      YY      ZZZZZZZZZZ  EEEEEEEEEE      ....
AA      AA  NN      NN  AA      AA  LLLLLLLLLL      YY      ZZZZZZZZZZ  EEEEEEEEEE      ....

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

```
0 MODULE dcx_analyze (                         . Data analysis routines
0                 LANGUAGE (BLISS32),
0                 !DENT = 'V04-000'
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 !
1 !   FACILITY:
1 !
1 !       DCX -- Data Compression / Expansion Facility
1 !
1 !   ABSTRACT:
1 !
1 !       The Data Compression / Expansion procedures provide a general
1 !       method for reducing the storage requirement for a arbitrary data.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX native, user mode.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  David Thiel
1 !
1 ! CREATION DATE: July, 1981
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-001 DWT0078        David W. Thiel          22-Feb-1983
1 !               Add support for estimated size of data to be compressed.
1 !
```

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

```
:    60           0059  1 %SBTTL  'Declarations';
:    61           0060  1
:    62           0061  1 LIBRARY
:    63           0062  1         'sys$library:starlet';  ! System macros
:    64           0063  1 REQUIRE
:    65           0064  1         'prefix';               ! DCX macros
:    66           0207  1 REQUIRE
:    67           0208  1         'dcxdef';               ! DCX public structure definitions
:    68           0302  1 REQUIRE
:    69           0303  1         'dcxprvdef';            ! DCX private structure definitions
:    70           0469  1
:    71           0470  1 ! random tree insertion macro
:    72           0471  1 !
:    73           0472  1 !       format of tree entry:
:    74           0473  1 !
:    75           0474  1 !       .-----------------------------------.
:    76           0475  1 !       ! ptr to tree with keys < this entry !
:    77           0476  1 !       ! ptr to tree with keys > this entry !
:    78           0477  1 !       ! ptr to list with keys = this entry !
:    79           0478  1 !       ! key value                          !
:    80           0479  1 !       ! ................................... !
:    81           0480  1 !
:    82           0481  1 MACRO
:    83         M 0482  1     tree_insert (header_address, item_address) =
:    84         M 0483  1         BEGIN
:    85         M 0484  1
:    86         M 0485  1         BIND
:    87         M 0486  1             _it = (item_address) : VECTOR [, LONG];
:    88         M 0487  1
:    89         M 0488  1         LOCAL
:    90         M 0489  1             _h : LONG;
:    91         M 0490  1
:    92         M 0491  1         _it [0] = 0;
:    93         M 0492  1         _it [1] = 0;
:    94         M 0493  1         _it [2] = 0;
:    95         M 0494  1         _h = (header_address);
:    96         M 0495  1         WHILE .._h NEQA 0 DO
:    97         M 0496  1             BEGIN
:    98         M 0497  1             IF ._it [3] LSSU .VECTOR [.._h, 3]
:    99         M 0498  1             THEN
:   100         M 0499  1                 _h = VECTOR [.._h, 0]
:   101         M 0500  1             ELSE IF ._it [3] GTRU .VECTOR [.._h, 3]
:   102         M 0501  1             THEN
:   103         M 0502  1                 _h = VECTOR [.._h, 1]
:   104         M 0503  1             ELSE
:   105         M 0504  1                 BEGIN
:   106         M 0505  1                 _it [2] = .VECTOR [.._h, 2];
:   107         M 0506  1                 _h = VECTOR [.._h, 2];
:   108         M 0507  1                 EXITLOOP;
:   109         M 0508  1                 END;
:   110         M 0509  1             END;
:   111         M 0510  1         ._h = _it [0];
:   112           0511  1         END%,
:   113         M 0512  1     tree_least (header_address) =
:   114         M 0513  1         BEGIN
:   115         M 0514  1
:   116         M 0515  1         LOCAL
```

```
;   117        M 0516  1                _q : REF VECTOR [, LONG],
;   118        M 0517  1                _h : LONG;
;   119        M 0518  1
;   120        M 0519  1            _h = (header_address);
;   121        M 0520  1            IF .._h EQLA 0
;   122        M 0521  1            THEN
;   123        M 0522  1                0
;   124        M 0523  1            ELSE
;   125        M 0524  1                BEGIN
;   126        M 0525  1                WHILE .VECTOR [.._h, 0] NEQA 0 DO
;   127        M 0526  1                    _h = VECTOR [.._h, 0];
;   128        M 0527  1                IF (_q = .VECTOR [.._h, 2]) NEQA 0
;   129        M 0528  1                THEN
;   130        M 0529  1                    BEGIN
;   131        M 0530  1                    VECTOR [.._h, 2] = ._q [2];
;   132        M 0531  1                    _q [0]
;   133        M 0532  1                    END
;   134        M 0533  1                ELSE
;   135        M 0534  1                    BEGIN
;   136        M 0535  1                    _q = VECTOR [.._h, 0];
;   137        M 0536  1                    ._h = ._q [1];
;   138        M 0537  1                    _q [0]
;   139        M 0538  1                    END
;   140        M 0539  1                END
;   141          0540  1            END%;
;   142          0541  1
;   143          0542  1 EXTERNAL ROUTINE
;   144          0543  1     dcx$ctx_check : lkg_ctx_check,      ! Check context block
;   145          0544  1     dcx$map_check : lkg_map_check,      ! Check map
;   146          0545  1     dcx$get_vm,                     ! Allocate memory
;   147          0546  1     dcx$free_vm,                    ! Deallocate memory
;   148          0547  1     lib$scopy_r_dx :                ! General string copy
;   149          0548  1         ADDRESSING_MODE (GENERAL);
;   150          0549  1
;   151          0550  1 EXTERNAL LITERAL
;   152          0551  1     dcx$_invarg,
;   153          0552  1     dcx$_invctx,
;   154          0553  1     dcx$_invitem,
;   155          0554  1     dcx$_normal;
;   156          0555  1
;   157          0556  1 FORWARD ROUTINE
;   158          0557  1     process_item,                   ! process item list
;   159          0558  1     make_seg,                       ! make a segment
;   160          0559  1     dcx$analyze_init,               ! initialize for data analysis
;   161          0560  1     dcx$analyze_data,               ! process data record
;   162          0561  1     huffman_size,                   ! compute size of Huffman encoded data
;   163          0562  1     remove_seg,                     ! remove one segment
;   164          0563  1     eliminate_seg,                  ! eliminate unprofitable segment
;   165          0564  1     dcx$make_map,                   ! compute comp / exp function
;   166          0565  1     dcx$analyze_done;               ! delete analysis context
```

```
1 %SBTTL  'process_item - Process Item List'
1
1 ROUTINE process_item (anl : REF BBLOCK, list : REF VECTOR [, LONG]) =
2 BEGIN
2 !++
2 !
2 ! Compute size of Huffman Encoded Data
2 !
2 ! Inputs:
2 !
2 !       anl                      Address of anl structure
2 !       list                     Address of item list
2 !
2 ! Outputs:
2 !
2 !       NONE
2 !
2 ! Return value:
2 !
2 !       status code
2 !
2 !--
2
2 IF NOT .list [0]
2 THEN
2     RETURN dcx$_invitem;                        ! even length list
2 INCR i FROM 1 TO .list [0] BY 2 DO
3     BEGIN
3     SELECTONE .list [.i] OF
3     SET
3
3     [dcx$c_bounded]:
3         anl [anl$v_bounded] = ..list [.i + 1];
3
3     [dcx$c_one_pass]:
3         anl [anl$v_one_pass] = ..list [.i + 1];
3
3     [dcx$c_est_records]:
4         BEGIN
4         anl [anl$v_est_recs] = true;
4         anl [anl$l_est_d_recs] = ..list [.i + 1];
3         END;
3
3     [dcx$c_est_bytes]:
4         BEGIN
4         anl [anl$v_est_bytes] = true;
4         anl [anl$l_est_d_bytes] = ..list [.i + 1];
3         END;
3
3     [dcx$c_list]:
3         perform (process_item (.anl, .list [.i + 1]));
3
3     TES;
2     END;
2 RETURN dcx$_normal;
2
1 END;                                            ! Of process_item
```

```
                                                                  .TITLE  DCX_ANALYZE
                                                                  .IDENT  \V04-000\

                                                                  .EXTRN  LIB$ANALYZE_SDESC_R2
                                                                  .EXTRN  DCX$CTX_CHECK, DCX$MAP_CHECK
                                                                  .EXTRN  DCX$GET_VM, DCX$FREE_VM
                                                                  .EXTRN  LIB$SCOPY_R_DX, DCX$_INVARG
                                                                  .EXTRN  DCX$_INVCTX, DCX$_INVITEM
                                                                  .EXTRN  DCX$_NORMAL

                                                                  .PSECT  $CODE$,NOWRT,2

                                                 000C 00000 PROCESS_ITEM:
                                                                  .WORD   Save R2,R3                            ; 0568
                                 53       08    AC  D0 00002      MOVL    LIST, R3                              ; 0589
                                 08             63  E8 00006      BLBS    (R3), 1$                              ;
                                 50 00000000G   8F  D0 00009      MOVL    #DCX$_INVITEM, R0                     ; 0591
                                                    04 00010      RET                                           ;
                                 52             01  CE 00011 1$:  MNEGL   #1, I                                 ; 0592
                                                7A  11 00014      BRB     7$                                    ;
                                 50           6342  D0 00016 2$:  MOVL    (R3)[I], R0                           ; 0594
                       00000101  8F             50  D1 0001A      CMPL    R0, #257                              ; 0597
                                                0D  12 00021      BNEQ    3$                                    ;
                                 50       04  A342  D0 00023      MOVL    4(R3)[I], R0                          ; 0598
      04    BC             01    00             60  F0 00028      INSV    (R0), #0, #1, @ANL                    ;
                                                60  11 0002E      BRB     7$                                    ;
                       00000102  8F             50  D1 00030 3$:  CMPL    R0, #258                              ; 0600
                                                0D  12 00037      BNEQ    4$                                    ;
                                 50       04  A342  D0 00039      MOVL    4(R3)[I], R0                          ; 0601
      04    BC             01    01             60  F0 0003E      INSV    (R0), #1, #1, @ANL                    ;
                                                4A  11 00044      BRB     7$                                    ;
                       00000201  8F             50  D1 00046 4$:  CMPL    R0, #513                              ; 0603
                                                12  12 0004D      BNEQ    5$                                    ;
                                 51       04    AC  D0 0004F      MOVL    ANL, R1                               ; 0605
                                 61             08  88 00053      BISB2   #8, (R1)                              ;
                                 50       04  A342  D0 00056      MOVL    4(R3)[I], R0                          ; 0606
                             10  A1             60  D0 0005B      MOVL    (R0), 16(R1)                          ;
                                                2F  11 0005F      BRB     7$                                    ; 0594
                       00000202  8F             50  D1 00061 5$:  CMPL    R0, #514                              ; 0609
                                                12  12 00068      BNEQ    6$                                    ;
                                 50       04    AC  D0 0006A      MOVL    ANL, R0                               ; 0611
                                 60             04  88 0006E      BISB2   #4, (R0)                              ;
                                 51       04  A342  D0 00071      MOVL    4(R3)[I], R1                          ; 0612
                             0C  A0             61  D0 00076      MOVL    (R1), 12(R0)                          ;
                                                14  11 0007A      BRB     7$                                    ; 0594
                                 01             50  D1 0007C 6$:  CMPL    R0, #1                                ; 0615
                                                0F  12 0007F      BNEQ    7$                                    ;
                                          04  A342  DD 00081      PUSHL   4(R3)[I]                              ; 0616
                                          04    AC  DD 00085      PUSHL   ANL                                   ;
                           FF73  CF             02  FB 00088      CALLS   #2, PROCESS_ITEM                      ;
                                 0D             50  E9 0008D      BLBC    STATUS, 8$                            ;
        FF80               52    02             63  F1 00090 7$:  ACBL    (R3), #2, I, 2$                       ; 0592
                                 50 00000000G   8F  D0 00096      MOVL    #DCX$_NORMAL, R0                      ; 0620
                                                    04 0009D 8$:  RET                                           ; 0622
```

```
; Routine Size:  158 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL  'make_seg - Make a tree segment'
1
1 ROUTINE make_seg (anl : REF BBLOCK, parent_seg : REF BBLOCK, char : LONG) =
2 BEGIN
2 !++
2 !
2 ! Compute size of Huffman Encoded Data
2 !
2 ! Inputs:
2 !
2 !       anl                     Address of anl structure
2 !       parent_seg              Address of parent anlseg structure or 0
2 !       char                    Transition character into segment
2 !
2 ! Outputs:
2 !
2 !       NONE
2 !
2 ! Return value:
2 !
2 !       status code
2 !
2 !--
2
2 LOCAL
2     anlseg : REF BBLOCK;
2
2 perform (dcx$get_vm (anlseg$k_length, anlseg));
2 anl [anl$w_nsegs] = .anl [anl$w_nsegs] + 1;
2 IF .parent_seg EQLA 0
2 THEN
2     anl [anl$b_depth] = 1
2 ELSE IF .parent_seg [anlseg$b_depth] EQL .anl [anl$b_depth]
2 THEN
2     anl [anl$b_depth] = .anl [anl$b_depth] + 1;
2 insque (.anlseg, .anl [anl$l_blink]);
2 IF .parent_seg NEQA 0
2 THEN
2     parent_seg [anlseg$w_sons] = .parent_seg [anlseg$w_sons] + 1;
2 anlseg [anlseg$l_size] = anlseg$k_length;
2 anlseg [anlseg$w_id] = 0;
2 anlseg [anlseg$w_char] = .char;
2 anlseg [anlseg$w_active] = 0;
2 anlseg [anlseg$w_active_r] = 0;
2 IF .parent_seg EQLA 0
2 THEN
2     anlseg [anlseg$b_depth] = 1
2 ELSE
2     anlseg [anlseg$b_depth] = 1 + .parent_seg [anlseg$b_depth];
2 anlseg [anlseg$b_max_char] = %X'00';
2 anlseg [anlseg$b_min_char] = %X'FF';
2 anlseg [anlseg$b_escape] = 0;
2 anlseg [anlseg$v_tent] = true;
2 anlseg [anlseg$v_solid] = false;
2 anlseg [anlseg$v_escape] = false;
2 anlseg [anlseg$v_base] = true;
2 anlseg [anlseg$v_unbounded] = NOT .anl [anl$v_bounded];
```

```
2 anlseg [anlseg$w_sons] = 0;
2 anlseg [anlseg$w_max_code] = 0;
2 anlseg [anlseg$w_mapseg_size] = 0;
2 anlseg [anlseg$l_prev] = .parent_seg;
2 anlseg [anlseg$l_comp_bits] = 0;
2 anlseg [anlseg$l_adj_bits] = 0;
2 anlseg [anlseg$l_chars] = 0;
2 IF .parent_seg NEQA 0
2 THEN
2     CH$COPY (
2         .parent_seg [anlseg$b_depth] - 1, parent_seg [anlseg$t_string],
2         1, char,
2         0,
2         anlseg$s_string, anlseg [anlseg$t_string]
2         );
2 IF .parent_seg EQLA 0
2 THEN
3     BEGIN
3     DECR index FROM anlseg$c_next - 1 TO 0 DO
3         VECTOR [anlseg [anlseg$l_next], .index] = .anlseg;
3     END
2 ELSE
3     BEGIN
3
3     LOCAL
3         seg : REF BBLOCK;
3
3     BIND
3         pnext = parent_seg [anlseg$l_next] : VECTOR [, LONG],
3         ptrs = anlseg [anlseg$l_next] : VECTOR [, LONG],
3         ca = anlseg [anlseg$t_string] : VECTOR [, BYTE];
3
3     pnext [.char] = .anlseg;
3     seg = .anl [anl$l_flink];
3     WHILE .seg NEQA anl [anl$q_queue] DO
4         BEGIN
4
4         BIND
4             sptrs = seg [anlseg$l_next] : VECTOR [, LONG],
4             snext = sptrs [.char] : REF BBLOCK;
4
4         IF .seg EQLA .anlseg
4         THEN
4             0
4         ELSE IF .anlseg [anlseg$b_depth] EQL 2
4         THEN
4             sptrs [.char] = .anlseg
4         ELSE IF .seg [anlseg$b_depth] LSS .anlseg [anlseg$b_depth]
4         THEN
4             0
4         ELSE IF .snext [anlseg$b_depth] GEQ .anlseg [anlseg$b_depth]
4         THEN
4             0
4         ELSE IF CH$EQL (
4             .anlseg [anlseg$b_depth] - 2,
4             anlseg [anlseg$t_string] + 1,
4             .anlseg [anlseg$b_depth] - 2,
0737  5             seg [anlseg$t_string] + 1 + (.seg [anlseg$b_depth] - .anlseg [anlseg$b_depth])
0738  4             )
0739  4         THEN
0740  4             sptrs [.char] = .anlseg;
0741  4         seg = .seg [anlseg$l_flink];
0742  3         END;
0743  3
0744  3     seg = .anl [anl$l_flink];
0745  3     INCR index FROM 1 TO .anlseg [anlseg$b_depth] - 2 DO
0746  3         seg = .VECTOR [seg [anlseg$l_next], .ca [.index]];
0747  3     CH$MOVE (4*anlseg$c_next, seg [anlseg$l_next], anlseg [anlseg$l_next]);
0748  2     END;
0749  2 RETURN dcx$_normal;
0750  2
0751  1 END;                                        ! Of make_seg
```

```
                                              OFFC 00000 MAKE_SEG:
                                                                 .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11          ; 0625
                         5E              0C  C2 00002            SUBL2   #12, SP
                                   08    AE  9F 00005            PUSHAB  ANLSEG                                        ; 0650
                         7E      0840    8F  3C 00008            MOVZWL  #2112, -(SP)
                 0000G   CF              02  FB 0000D            CALLS   #2, DCX$GET_VM                                ;
                         01              50  E8 00012            BLBS    STATUS, 1$                                    ;
                                             04 00015            RET                                                   ;
                         50        04    AC  7D 00016 1$:        MOVQ    ANL, R0                                       ; 0651
                                   16    A0  B6 0001A            INCW    22(R0)
                                         51  D5 0001D            TSTL    R1                                            ; 0652
                                         06  12 0001F            BNEQ    2$                                            ;
                   14    A0              01  90 00021            MOVB    #1, 20(R0)                                    ; 0654
                                         0A  11 00025            BRB     3$                                            ;
                   14    A0        14    A1  91 00027 2$:        CMPB    20(R1), 20(R0)                                ; 0655
                                         03  12 0002C            BNEQ    3$                                            ;
                                   14    A0  96 0002E            INCB    20(R0)                                        ; 0657
                   1C    B0        08    BE  0E 00031 3$:        INSQUE  @ANLSEG, @28(R0)                              ; 0658
                         57        08    AC  D0 00036            MOVL    PARENT_SEG, R7                                ; 0659
                                         51  D4 0003A            CLRL    R1                                            ;
                                         57  D5 0003C            TSTL    R7                                            ;
                                         05  13 0003E            BEQL    4$                                            ;
                                         51  D6 00040            INCL    R1                                            ;
                                   1C    A7  B6 00042            INCW    28(R7)                                        ; 0661
                         56        08    AE  D0 00045 4$:        MOVL    ANLSEG, R6                                    ; 0662
                   08    A6      0840    8F  3C 00049            MOVZWL  #2112, 8(R6)
                                   0C    A6  B4 0004F            CLRW    12(R6)                                        ; 0663
                         59        0C    AC  D0 00052            MOVL    CHAR, R9                                      ; 0664
                   0E    A6              59  3C 00056            MOVZWL  R9, 14(R6)
                                   12    A6  B4 0005A            CLRW    18(R6)                                        ; 0666
                                   04    AE  D4 0005D            CLRL    4(SP)                                         ; 0667
                                         57  D5 00060            TSTL    R7                                            ;
                                         09  12 00062            BNEQ    5$                                            ;
                                   04    AE  D6 00064            INCL    4(SP)                                         ;
                   14    A6              01  90 00067            MOVB    #1, 20(R6)                                    ; 0669
                                         06  11 0006B            BRB     6$                                            ;
          14    A6  14    A7             01  81 0006D 5$:        ADDB3   #1, 20(R7), 20(R6)                            ; 0671
```

```
                                                          D 10
DCX_ANALYZE                                              15-Sep-1984 23:38:18    VAX-11 Bliss-32 V4.0-742            Page  11
V04-000         make_seg - Make a tree segment           14-Sep-1984 12:15:55    DISK$VMSMASTER:[DCX.SRC]ANALYZE.B32;1    (4)

                              15  A6      FF  8F  9B 00073 6$:    MOVZBW  #255, 21(R6)                          ; 0673
                                          17  A6  94 00078        CLRB    23(R6)                                ; 0674
                                  50      18  A6  9E 0007B        MOVAB   24(R6), R0                            ; 0675
                                  60          01  88 0007F        BISB2   #1, (R0)                              ;
                                  60          0A  8A 00082        BICB2   #10, (R0)                             ; 0677
                                  60          10  88 00085        BISB2   #16, (R0)                             ; 0678
                                  58      04  AC  D0 00088        MOVL    ANL, R8                               ; 0679
       52         68              01          00  EF 0008C        EXTZV   #0, #1, (R8), R2                      ;
                                  52          52  92 00091        MCOMB   R2, R2                                ;
       60         01              05          52  F0 00094        INSV    R2, #5, #1, (R0)                      ;
                                          1A  A6  D4 00099        CLRL    26(R6)                                ; 0681
                                          1E  A6  B4 0009C        CLRW    30(R6)                                ; 0682
                              20  A6          57  D0 0009F        MOVL    R7, 32(R6)                            ; 0683
                                          24  A6  7C 000A3        CLRQ    36(R6)                                ; 0684
                                          2C  A6  D4 000A6        CLRL    44(R6)                                ; 0686
                                  23          51  E9 000A9        BLBC    R1, 7$                                ; 0687
                                  6E      14  A7  9A 000AC        MOVZBL  20(R7), (SP)                          ; 0690
                                              6E  D7 000B0        DECL    (SP)                                  ;
                                  5B          08  D0 000B2        MOVL    #8, R11                               ;
                                  5A      30  A6  9E 000B5        MOVAB   48(R6), R10                           ; 0693
       5B         00          30  A7          6E  2C 000B9        MOVC5   (SP), 48(R7), #0, R11, (R10)          ;
                                              6A     000BF                                                      ;
                                              0D  18 000C0        BGEQ    7$                                    ;
                                  5A          6E  C0 000C2        ADDL2   (SP), R10                             ;
                                  5B          6E  C2 000C5        SUBL2   (SP), R11                             ;
       5B         00          0C  AC          01  2C 000C8        MOVC5   #1, CHAR, #0, R11, (R10)              ;
                                              6A     000CE                                                      ;
                                  5A    043C  C6  9E 000CF 7$:    MOVAB   1084(R6), R10                         ; 0699
                                  0E      04  AE  E9 000D4        BLBC    4(SP), 9$                             ;
                                  50    0100  8F  3C 000D8        MOVZWL  #256, INDEX                           ; 0698
                                6A40          56  D0 000DD 8$:    MOVL    R6, (R10)[INDEX]                      ; 0699
                                  F9          50  F4 000E1        SOBGEQ  INDEX, 8$                             ;
                                              78  11 000E4        BRB     16$                                   ; 0695
                           043C C749          56  D0 000E6 9$:    MOVL    R6, 1084(R7)[R9]                      ; 0712
                                  54      18  A8  D0 000EC        MOVL    24(R8), SEG                           ; 0713
                                  50      18  A8  9E 000F0 10$:   MOVAB   24(R8), R0                            ; 0714
                                  50          54  D1 000F4        CMPL    SEG, R0                               ;
                                              3F  13 000F7        BEQL    13$                                   ;
                                  55    043C C449 DE 000F9        MOVAL   1084(SEG)[R9], R5                     ; 0719
                                  56          54  D1 000FF        CMPL    SEG, R6                               ; 0721
                                              2F  13 00102        BEQL    12$                                   ;
                                  51      14  A6  9A 00104        MOVZBL  20(R6), R1                            ; 0724
                                  02          51  91 00108        CMPB    R1, #2                                ;
                                              23  13 0010B        BEQL    11$                                   ;
                                  51      14  A4  91 0010D        CMPB    20(SEG), R1                           ; 0727
                                              20  1F 00111        BLSSU   12$                                   ;
                                  50          65  D0 00113        MOVL    (R5), R0                              ; 0730
                                  51      14  A0  91 00116        CMPB    20(R0), R1                            ;
                                              17  1E 0011A        BGEQU   12$                                   ;
                                  52      FE  A1  9E 0011C        MOVAB   -2(R1), R2                            ; 0734
                                  50      14  A4  9A 00120        MOVZBL  20(SEG), R0                           ; 0737
                                  50          51  C2 00124        SUBL2   R1, R0                                ;
                31 A044       31  A6          52  29 00127        CMPC3   R2, 49(R6), 49(R0)[SEG]               ; 0733
                                              03  12 0012E        BNEQ    12$                                   ;
                                  65          56  D0 00130 11$:   MOVL    R6, (R5)                              ; 0740
                                  54          64  D0 00133 12$:   MOVL    (SEG), SEG                            ; 0741
                                              B8  11 00136        BRB     10$                                   ; 0714
```

```
                        54        18   A8  D0 00138 13$:    MOVL     24(R8), SEG                                          ; 0744
                        52        14   A6  9A 0013C         MOVZBL   20(R6), R2                                           ; 0745
                        52             02  C2 00140         SUBL2    #2, R2                                               ;
                                       51  D4 00143         CLRL     INDEX                                                ;
                                       0B  11 00145         BRB      15$                                                  ;
                        50        30 A641  9A 00147 14$:    MOVZBL   48(R6)[INDEX], R0                                    ; 0746
                        54      043C C440  D0 0014C         MOVL     1084(SEG)[R0], SEG                                   ;
           F1           51             52  F3 00152 15$:    AOBLEQ   R2, INDEX, 14$                                       ;
           6A     043C  C4      0404   8F  28 00156         MOVC3    #1028, 1084(SEG), (R10)                              ; 0747
                        50 00000000G   8F  D0 0015E 16$:    MOVL     #DCX$_NORMAL, R0                                     ; 0749
                                       04 00165             RET                                                           ; 0751

; Routine Size:  358 bytes,    Routine Base:  $CODE$ + 009E
```

```
1 %SBTTL  'dcx$analyze_init - Initialization for data analysis'
1
1 GLOBAL ROUTINE dcx$analyze_init (context_addr, item, value) =
2 BEGIN
2 !++
2 !
2 ! Initialization for data analysis.
2 ! Allocate and initialize context area.
2 !
2 ! Inputs:
2 !
2 !       context_addr.wz.r       Address of context longword
2 !       item.rl.r               Item code (optional)
2 !       value.rl.r              Value associated with item (optional)
2 !
2 ! Outputs:
2 !
2 !       context_addr            Address of context block is stored
2 !
2 ! Return value:
2 !
2 !       status.wlc.v
2 !
2 !               dcx$_normal     All is well
2 !               dcx$_invitem    Invalid item code or missing item value
2 !               lib$_insvirmem  Error allocating memory
2 !--
2
2 BIND
2     ctx = .context_addr : REF BBLOCK;   ! address of context block
2
2 LOCAL
2     anl : REF BBLOCK,             ! address of anl block
2     anlseg : REF BBLOCK;          ! address of anlseg block
2
2 BUILTIN
2     ACTUALPARAMETER,
2     ACTUALCOUNT,
2     NULLPARAMETER;
2
2 perform (dcx$get_vm (ctx$k_fixed_len + anl$k_length, ctx));
2 ctx [ctx$l_size] = ctx$k_fixed_len + anl$k_length;
2 ctx [ctx$b_type] = ctx$c_anlyz;
2 ctx [ctx$w_version] = ctx$c_version;
2 ctx [ctx$l_sanity] = ctx$c_sanity;
2 anl = ctx [ctx$l_specific];
2 anl [anl$v_bounded] = false;
2 anl [anl$v_one_pass] = false;
2 anl [anl$v_est_bytes] = false;
2 anl [anl$v_est_recs] = false;
2 anl [anl$l_est_d_bytes] = 0;
2 anl [anl$l_est_d_recs] = 0;
2 anl [anl$l_d_bytes] = 0;
2 anl [anl$l_d_recs] = 0;
2 anl [anl$b_depth] = 0;
2 anl [anl$w_nsegs] = 0;
2 anl [anl$l_flink] = anl [anl$l_blink] = anl [anl$q_queue];
```

```
;   413          0809  2 IF NOT ACTUALCOUNT ()
;   414          0810  2 THEN
;   415          0811  2     RETURN dcx$_invitem;                          ! even number of arguments
;   416          0812  2 INCR i FROM 2 TO ACTUALCOUNT () BY 2 DO
;   417          0813  3     BEGIN
;   418          0814  3
;   419          0815  3     LOCAL
;   420          0816  3         list : VECTOR [3, LONG];
;   421          0817  3
;   422          0818  3     list [0] = 3;
;   423          0819  3     list [1] = .ACTUALPARAMETER (.i);
;   424          0820  3     list [2] = ACTUALPARAMETER (.i + 1);
;   425          0821  3     perform (process_item (.anl, list));
;   426          0822  2     END;
;   427          0823  2 perform (make_seg (.anl, 0, -1));
;   428          0824  2 DECR char FROM anlseg$c_next - 1 TO 0 DO
;   429          0825  2     perform (make_seg (.anl, .anl [anl$l_flink], .char));
;   430          0826  2 RETURN dcx$_normal;
;   431          0827  2
;   432          0828  1 END;                                              ! of dcx$analyze_init





                                           001C 00000          .ENTRY  DCX$ANALYZE_INIT, Save R2,R3,R4                    ; 0754
                     5E                 0C   C2 00002          SUBL2   #12, SP                                            ;
                             04         AC   DD 00005          PUSHL   CONTEXT_ADDR                                       ; 0792
                                        3C   DD 00008          PUSHL   #60                                                ;
               0000G CF                 02   FB 0000A          CALLS   #2, DCX$GET_VM                                     ;
                     73                 50   E9 0000F          BLBC    STATUS, 4$                                         ;
                     52      04         BC   D0 00012          MOVL    @CONTEXT_ADDR, R2                                  ; 0793
                     82                 3C   D0 00016          MOVL    #60, (R2)+                                         ;
                                        82   94 00019          CLRB    (R2)+                                              ; 0794
                             03         A2   B4 0001B          CLRW    3(R2)                                              ; 0795
                  07 A2 4F317C65        8F   D0 0001E          MOVL    #1328643173, 7(R2)                                 ; 0796
                     52                 0F   C0 00026          ADDL2   #15, ANL                                           ; 0797
                     62                 0F   8A 00029          BICB2   #15, (ANL)                                         ; 0801
                             0C         A2   7C 0002C          CLRQ    12(ANL)                                            ; 0802
                             04         A2   7C 0002F          CLRQ    4(ANL)                                             ; 0804
                             14         A2   94 00032          CLRB    20(ANL)                                            ; 0806
                             16         A2   B4 00035          CLRW    22(ANL)                                            ; 0807
                     50      18         A2   9E 00038          MOVAB   24(ANL), R0                                        ; 0808
                  1C A2                 50   D0 0003C          MOVL    R0, 28(ANL)                                        ;
                  18 A2                 50   D0 00040          MOVL    R0, 24(ANL)                                        ;
                     08                 6C   E8 00044          BLBS    (AP), 1$                                           ; 0809
                     50 00000000G       8F   D0 00047          MOVL    #DCX$_INVITEM, R0                                  ; 0811
                                             04 0004E          RET                                                        ;
                     54                 6C   9A 0004F  1$:     MOVZBL  (AP), R4                                           ; 0812
                                        53   D4 00052          CLRL    I                                                  ;
                                        1D   11 00054          BRB     3$                                                 ;
                     6E                 03   D0 00056  2$:     MOVL    #3, LIST                                           ; 0818
                     50               6C43   D0 00059          MOVL    (AP)[I], R0                                        ; 0819
                  04 AE                 60   D0 0005D          MOVL    (R0), LIST+4                                       ;
                  08 AE           04 AC43    D0 00061          MOVL    4(AP)[I], LIST+8                                   ; 0820
                             4004       8F   BB 00067          PUSHR   #^M<R2,SP>                                         ; 0821
               FD8C CF                  02   FB 0006B          CALLS   #2, PROCESS_ITEM                                   ;
```

```
                                                                     H 10
DCX_ANALYZE                                                   15-Sep-1984 23:38:18    VAX-11 Bliss-32 V4.0-742                 Page  15
V04-000        dcx$analyze_init - Initialization for data anal 14-Sep-1984 12:15:55    DISK$VMSMASTER:[DCX.SRC]ANALYZE.B32;1      (5)

                                     33          50  E9 00070          BLBC     STATUS, 6$                                   ;
        FFDD          53             02          54  F1 00073 3$:      ACBL     R4, #2, I, 2$                                ; 0812
                                     7E          01  CE 00079          MNEGL    #1, -(SP)                                    ; 0823
                                                 7E  D4 0007C          CLRL     -(SP)                                        ;
                                                 52  DD 0007E          PUSHL    ANL                                          ;
                          FE15       CF          03  FB 00080          CALLS    #3, MAKE_SEG                                 ;
                                     1E          50  E9 00085 4$:      BLBC     STATUS, 6$                                   ;
                                     53    0100  8F  3C 00088          MOVZWL   #256, CHAR                                   ; 0824
                                                 53  DD 0008D 5$:      PUSHL    CHAR                                         ; 0825
                                             18  A2  DD 0008F          PUSHL    24(ANL)                                      ;
                                                 52  DD 00092          PUSHL    ANL                                          ;
                          FE01       CF          03  FB 00094          CALLS    #3, MAKE_SEG                                 ;
                                     0A          50  E9 00099          BLBC     STATUS, 6$                                   ;
                                     EE          53  F4 0009C          SOBGEQ   CHAR, 5$                                     ;
                                     50 00000000G 8F  D0 0009F          MOVL     #DCX$_NORMAL, R0                             ; 0826
                                                 04 000A6 6$:          RET                                                   ; 0828

; Routine Size:  167 bytes     Routine Base:  $CODE$ + 0204
```

```
1 %SBTTL  'dcx$analyze_data - Analyze data record'
1
1 GLOBAL ROUTINE dcx$analyze_data (context_addr, rec : REF BBLOCK) =
2 BEGIN
2 !++
2 !
2 ! Analyze data record
2 !
2 ! Inputs:
2 !
2 !       context_addr.mz.r       Address of context longword
2 !       rec.rt.dx               Descriptor for (text) data record
2 !
2 ! Outputs:
2 !
2 !       context_addr.mz.r       Context block accumulates data
2 !
2 ! Return value:
2 !
2 !       status.wlc.v
2 !
2 !               dcx$_normal     All is well
2 !               dcx$_invctx     Invalid context block
2 !--
2
2 BIND
2     ctx = .context_addr : REF BBLOCK;
2
2 LOCAL
2     addr : REF VECTOR [, BYTE], ! address of data
2     len,                        ! lengh of data
2     anl : REF BBLOCK,
2     anlseg : REF BBLOCK,
2     status;                     ! returr status
2
2 ! check context block
2 !
2 perform (dcx$ctx_check (.ctx, ctx$c_anlyz));
2
2 ! get address of data record
2 !
2 perform (lib$analyze_sdesc_r2 (.rec; status, len, addr); .status);
2
2 ! accumulate statistical information
2 !
2 anl = ctx [ctx$l_specific];
2 anl [anl$l_d_bytes] = .anl [anl$l_d_bytes] + .len;
2 anl [anl$l_d_recs] = .anl [anl$l_d_recs] + 1;
2 anlseg = .anl [anl$l_flink];
2 DECR i FROM .len-1 TO 0 DO
3     BEGIN
3
3     BIND
3         count = anlseg [anlseg$l_count] : VECTOR [, LONG],
3         next = anlseg [anlseg$l_next] : VECTOR [, LONG];
3
3     anlseg [anlseg$l_chars] = .anlseg [anlseg$l_chars] + 1;
```

```
;  491          0886  3      count [.addr [0]] = .count [.addr [0]] + 1;
;  492          0887  3      anlseg = .next [.addr [0]];
;  493          0888  3      addr = .addr + 1;
;  494          0889  2      END;
;  495          0890  2 IF true
;  496          0891  2 THEN
;  497          0892  3      BEGIN
;  498          0893  3
;  499          0894  3      BIND
;  500          0895  3          count = anlseg [anlseg$l_count] : VECTOR [, LONG];
;  501          0896  3
;  502          0897  3      anlseg [anlseg$l_chars] = .anlseg [anlseg$l_chars] + 1;
;  503          0898  3      count [dcx$c_eor] = .count [dcx$c_eor] + 1;
;  504          0899  2      END;
;  505          0900  2 RETURN dcx$_normal;
;  506          0901  2
;  507          0902  1 END;                                        ! Of dcx$analyze_data



                                   000C 00000          .ENTRY  DCX$ANALYZE_DATA, Save R2,R3                           ; 0831
                                 51  D4 00002          CLRL    R1                                                     ; 0866
                      50      04  BC  D0 00004          MOVL    @CONTEXT_ADDR, R0                                      ;
                            0000G  30 00008          BSBW    DCX$CTX_CHECK                                          ;
                      44          50  E9 0000B          BLBC    STATUS, 3$                                             ;
                      50      08  AC  D0 0000E          MOVL    REC, R0                                                ; 0870
                        00000000G 00  16 00012          JSB     LIB$ANALYZE_SDESC_R2                                   ;
                      37          50  E9 00018          BLBC    STATUS, 3$                                             ;
           50       04  BC        14  C1 0001B          ADDL3   #20, @CONTEXT_ADDR, ANL                                ; 0874
                    04  A0        51  C0 00020          ADDL2   LEN, 4(ANL)                                            ; 0875
                              08  A0  D6 00024          INCL    8(ANL)                                                 ; 0876
                      53      18  A0  D0 00027          MOVL    24(ANL), ANLSEG                                        ; 0877
                                  10  11 0002B          BRB     2$                                                     ; 0878
                              2C  A3  D6 0002D 1$:      INCL    44(ANLSEG)                                             ; 0885
                      50          82  9A 00030          MOVZBL  (ADDR)+, R0                                            ; 0886
                             38 A340  D6 00033          INCL    56(ANLSEG)[R0]                                         ;
                      53    043C C340  D0 00037          MOVL    1084(ANLSEG)[R0], ANLSEG                               ; 0887
                      ED          51  F4 0003D 2$:      SOBGEQ  I, 1$                                                  ; 0878
                      50      38  A3  9E 00040          MOVAB   56(ANLSEG), R0                                         ; 0895
                              2C  A3  D6 00044          INCL    44(ANLSEG)                                             ; 0897
                            0400  C0  D6 00047          INCL    1024(R0)                                               ; 0898
                      50 00000000G 8F  D0 0004B          MOVL    #DCX$_NORMAL, R0                                       ; 0900
                                  04 00052 3$:          RET                                                            ; 0902

; Routine Size:  83 bytes,    Routine Base:  $CODE$ + 02AB
```

```
1 %SBTTL  'huffman_size - Compute size of Huffman Encoded Data'
1
1 ROUTINE huffman_size (anl : REF BBLOCK, anlseg : REF BBLOCK) =
2 BEGIN
2 !++
2 !
2 ! Compute size of Huffman Encoded Data in bits
2 ! Store size in anlseg$l_comp_bits field
2 ! Also fill in anlseg$w_active and anlseg$w_active_r fields
2 ! Also fill in anlseg$b_min_char and anlseg$b_max_char fields
2 !
2 ! Inputs:
2 !
2 !       anl                             Address of anl structure
2 !       anlseg                          Address of anlseg structure
2 !
2 ! Outputs:
2 !
2 !       NONE
2 !
2 ! Return value:
2 !
2 !       Status code
2 !
2 !--
2
2 BIND
2     count = anlseg [anlseg$l_count] : VECTOR [, LONG];
2
2 IF NOT .anl [anl$v_bounded]
2 THEN
3     BEGIN
3     anlseg [anlseg$w_active] = dcx$c_chars;
3     anlseg [anlseg$w_active_r] = dcx$c_chars;
3     anlseg [anlseg$b_min_char] = %X'00';
3     anlseg [anlseg$b_max_char] = %X'FF';
3     END
2 ELSE IF .anlseg [anlseg$l_chars] EQL 0
2 THEN
3     BEGIN
3     anlseg [anlseg$b_min_char] = %X'00';
3     anlseg [anlseg$b_max_char] = %X'00';
3     anlseg [anlseg$w_active] = 0;
3     anlseg [anlseg$w_active_r] = 0;
3     END
2 ELSE
3     BEGIN
3     anlseg [anlseg$b_min_char] = %X'00';
3     anlseg [anlseg$b_max_char] = %X'00';
3     anlseg [anlseg$w_active] = 0;
3     DECR i FROM anlseg$c_count-2 TO 0 DO
3         IF .count [.i] NEQ 0
3         THEN
4             BEGIN
4             anlseg [anlseg$w_active] = .anlseg [anlseg$w_active] + 1;
4             anlseg [anlseg$b_min_char] = .i;
3             END;
```

```
3     IF .count [dcx$c_eor] NEQ 0
3     THEN
3         anlseg [anlseg$w_active] = .anlseg [anlseg$w_active] + 1;
3     IF .anlseg [anlseg$w_active] NEQ 1
3     THEN
3         anlseg [anlseg$w_active_r] = .anlseg [anlseg$w_active]
3     ELSE
3         anlseg [anlseg$w_active_r] = 2;
3     DECR i FROM anlseg$c_count-2 TO 0 DO
3         IF .count [.i] NEQ 0
3         THEN
4             BEGIN
4             anlseg [anlseg$b_max_char] = .i;
4             EXITLOOP;
3             END;
2     END;
2 IF .anlseg [anlseg$l_chars] EQL 0
2 THEN
2     anlseg [anlseg$l_comp_bits] = 0
2 ELSE IF .anlseg [anlseg$w_active] EQL 0
2 THEN
2     anlseg [anlseg$l_comp_bits] = 0
2 ELSE IF .anlseg [anlseg$w_active] EQL 1
2 THEN
2     anlseg [anlseg$l_comp_bits] = .anlseg [anlseg$l_chars]
2 ELSE
3     BEGIN
3
3     LOCAL
3         p1 : REF VECTOR [4, LONG],
3         p2 : REF VECTOR [4, LONG],
3         ptr : LONG,                                         ! pointer into list
3         list : VECTOR [4 * anlseg$c_count, LONG],           ! storage list
3         zero : LONG;
3
3     anlseg [anlseg$l_comp_bits] = 0;
3     zero = false;
3     ptr = 0;
3     p1 = list [0];
3     DECR i FROM anlseg$c_count-1 TO 0 DO
4         BEGIN
4         p1 [3] = .count [.i];
4         IF .p1 [3] NEQ 0
4         THEN
5             BEGIN
5             tree_insert (ptr, p1 [0]);
5             p1 = p1 [4];          ! bump to next cell
5             END
5         ELSE IF (NOT .anl [anl$v_bounded]) AND (NOT .zero)
4         THEN
5             BEGIN
5             ! all zeroes logically consolidate to one zero
5             zero = true;
5             tree_insert (ptr, p1 [0]);
5             p1 = p1 [4];          ! bump to next cell
4             END;
3         END;
```

```
:  623          1017  3        WHILE (p1 = tree_least (ptr); (p2 = tree_least (ptr)) NEQA 0) DO
:  624          1018  4            BEGIN
:  625          1019  4            p1 [3] = .p1 [3] + .p2 [3];
:  626          1020  4            anlseg [anlseg$l_comp_bits] = .anlseg [anlseg$l_comp_bits] + .p1 [3];
:  627          1021  4            tree_insert (ptr, p1 [0]);
:  628          1022  3            END;
:  629          1023  2        END;
:  630          1024  2  anlseg [anlseg$l_adj_bits] = .anlseg [anlseg$l_comp_bits];
:  631          1025  2  RETURN dcx$_normal;
:  632          1026  2
:  633          1027  1  END;                                         ! Of huffman_size
```

```
                                            00FC 00000 HUFFMAN_SIZE:
                                                               .WORD    Save R2,R3,R4,R5,R6,R7                          ; 0905
                 5E      EFEC     CE   9E 00002                MOVAB    -4116(SP), SP                                   ;
                 53        08     AC   D0 00007                MOVL     ANLSEG, R3                                      ; 0930
                 55        38     A3   9E 0000B                MOVAB    56(R3), R5                                      ;
                 51        10     A3   9E 0000F                MOVAB    16(R3), R1                                      ; 0935
                 13        04     BC   E8 00013                BLBS     @ANL, 1$                                        ; 0932
                 61      0101     8F   B0 00017                MOVW     #257, (R1)                                      ; 0935
           12    A3      0101     8F   B0 0001C                MOVW     #257, 18(R3)                                    ; 0936
           15    A3      FF00     8F   B0 00022                MOVW     #65280, 21(R3)                                  ; 0937
                                  4F   11 00028                BRB      10$                                             ; 0932
                           2C     A3   D5 0002A 1$:            TSTL     44(R3)                                          ; 0940
                                  0A   12 0002D                BNEQ     2$                                              ;
                           15     A3   B4 0002F                CLRW     21(R3)                                          ; 0943
                                  61   B4 00032                CLRW     (R1)                                            ; 0945
                           12     A3   B4 00034                CLRW     18(R3)                                          ; 0946
                                  40   11 00037                BRB      10$                                             ; 0940
                           15     A3   B4 00039 2$:            CLRW     21(R3)                                          ; 0950
                                  61   B4 0003C                CLRW     (R1)                                            ; 0952
                 50        FF     8F   9A 0003E                MOVZBL   #255, I                                         ; 0953
                               6540    D5 00042 3$:            TSTL     (R5)[I]                                         ; 0954
                                  06   13 00045                BEQL     4$                                              ;
                                  61   B6 00047                INCW     (R1)                                            ; 0957
           15    A3               50   90 00049                MOVB     I, 21(R3)                                       ; 0958
                 F2               50   F4 0004D 4$:            SOBGEQ   I, 3$                                           ; 0954
                         0400     C5   D5 00050                TSTL     1024(R5)                                        ; 0960
                                  02   13 00054                BEQL     5$                                              ;
                                  61   B6 00056                INCW     (R1)                                            ; 0962
                 01               61   B1 00058 5$:            CMPW     (R1), #1                                        ; 0963
                                  06   13 0005B                BEQL     6$                                              ;
           12    A3               61   B0 0005D                MOVW     (R1), 18(R3)                                    ; 0965
                                  04   11 00061                BRB      7$                                              ;
           12    A3               02   B0 00063 6$:            MOVW     #2, 18(R3)                                      ; 0967
                 50        FF     8F   9A 00067 7$:            MOVZBL   #255, I                                         ; 0968
                               6540    D5 0006B 8$:            TSTL     (R5)[I]                                         ; 0969
                                  06   13 0006E                BEQL     9$                                              ;
           16    A3               50   90 00070                MOVB     I, 22(R3)                                       ; 0972
                                  03   11 00074                BRB      10$                                             ; 0971
                 F2               50   F4 00076 9$:            SOBGEQ   I, 8$                                           ; 0969
                 56        24     A3   9E 00079 10$:           MOVAB    36(R3), R6                                      ; 0978
                           2C     A3   D5 0007D                TSTL     44(R3)                                          ; 0976
```

```
                              04  13 00080          BEQL    11$                          ;
                              61  B5 00082          TSTW    (R1)                         ; 0979
                              04  12 00084          BNEQ    12$                          ;
                              66  D4 00086 11$:     CLRL    (R6)                         ; 0981
                              09  11 00088          BRB     13$                          ;
             01               61  B1 0008A 12$:     CMPW    (R1), #1                     ; 0982
                              07  12 0008D          BNEQ    14$                          ;
             66         2C    A3  D0 0008F          MOVL    44(R3), (R6)                 ; 0984
                            010D  31 00093 13$:     BRW     39$                          ;
                              66  D4 00096 14$:     CLRL    (R6)                         ; 0995
                              57  D4 00098          CLRL    ZERO                         ; 0996
                              6E  D4 0009A          CLRL    PTR                          ; 0997
             50         04    AE  9E 0009C          MOVAB   LIST, P1                     ; 0998
             52       0100    8F  3C 000A0          MOVZWL  #256, I                      ; 0999
      0C     A0             6542  D0 000A5 15$:     MOVL    (R5)[I], 12(P1)              ; 1001
                              21  13 000AA          BEQL    18$                          ; 1002
                              60  7C 000AC          CLRQ    (P1)                         ; 1005
                        08    A0  D4 000AE          CLRL    8(P1)                        ;
             54               6E  9E 000B1          MOVAB   PTR, _H                      ;
             51               64  D0 000B4 16$:     MOVL    (_H), R1                     ;
                              48  13 000B7          BEQL    22$                          ;
      0C     A1         0C    A0  D1 000B9          CMPL    12(P1), 12(R1)               ;
                              05  1E 000BE          BGEQU   17$                          ;
             54               51  D0 000C0          MOVL    R1, _H                       ;
                              EF  11 000C3          BRB     16$                          ;
                              31  1B 000C5 17$:     BLEQU   21$                          ;
             54         04    A1  9E 000C7          MOVAB   4(R1), _H                    ;
                              E7  11 000CB          BRB     16$                          ;
             36         04    BC  E8 000CD 18$:     BLBS    @ANL, 23$                    ; 1008
             33               57  E8 000D1          BLBS    ZERO, 23$                    ;
             57               01  D0 000D4          MOVL    #1, ZERO                     ; 1012
                              60  7C 000D7          CLRQ    (P1)                         ; 1013
                        08    A0  D4 000D9          CLRL    8(P1)                        ;
             54               6E  9E 000DC          MOVAB   PTR, _H                      ;
             51               64  D0 000DF 19$:     MOVL    (_H), R1                     ;
                              1D  13 000E2          BEQL    22$                          ;
      0C     A1         0C    A0  D1 000E4          CMPL    12(P1), 12(R1)               ;
                              05  1E 000E9          BGEQU   20$                          ;
             54               51  D0 000EB          MOVL    R1, _H                       ;
                              EF  11 000EE          BRB     19$                          ;
                              06  1B 000F0 20$:     BLEQU   21$                          ;
             54         04    A1  9E 000F2          MOVAB   4(R1), _H                    ;
                              E7  11 000F6          BRB     19$                          ;
      08     A0         08    A1  D0 000F8 21$:     MOVL    8(R1), 8(P1)                 ;
             54         08    A1  9E 000FD          MOVAB   8(R1), _H                    ;
             64               80  7E 00101 22$:     MOVAQ   (P1)+, (_H)                  ;
             50               08  C0 00104          ADDL2   #8, P1                       ; 1014
             9B               52  F4 00107 23$:     SOBGEQ  I, 15$                       ; 0999
             55               6F  9E 0010A 24$:     MOVAB   PTR, _H                      ; 1017
                              65  D5 0010D          TSTL    (_H)                         ;
                              04  12 0010F          BNEQ    25$                          ;
                              50  D4 00111          CLRL    P1                           ;
                              24  11 00113          BRB     29$                          ;
                        00    B5  D5 00115 25$:     TSTL    @0(_H)                       ;
                              05  13 00118          BEQL    26$                          ;
             55               65  D0 0011A          MOVL    (_H), _H                     ;
                              F6  11 0011D          BRB     25$                          ;
```

```
              54            65  D0 0011F 26$:    MOVL    (_H), R4                           ;
              51       08   A4  D0 00122         MOVL    8(R4), _Q                          ;
                            07  13 00126         BEQL    27$                                ;
         08   A4       08   A1  D0 00128         MOVL    8(_Q), 8(R4)                       ;
                            07  11 0012D         BRB     28$                                ;
              51            54  D0 0012F 27$:    MOVL    R4, _Q                             ;
              65       04   A1  D0 00132         MOVL    4(_Q), (_H)                        ;
              50            51  D0 00136 28$:    MOVL    R1, P1                             ;
              55            6E  9E 00139 29$:    MOVAB   PTR, _H                            ;
                            65  D5 0013C         TSTL    (_H)                               ;
                            04  12 0013E         BNEQ    30$                                ;
                            52  D4 00140         CLRL    P2                                 ;
                            24  11 00142         BRB     34$                                ;
                       00   B5  D5 00144 30$:    TSTL    @0(_H)                             ;
                            05  13 00147         BEQL    31$                                ;
              55            65  D0 00149         MOVL    (_H), _H                           ;
                            F6  11 0014C         BRB     30$                                ;
              54            65  D0 0014E 31$:    MOVL    (_H), R4                           ;
              51       08   A4  D0 00151         MOVL    8(R4), _Q                          ;
                            07  13 00155         BEQL    32$                                ;
         08   A4       08   A1  D0 00157         MOVL    8(_Q), 8(R4)                       ;
                            07  11 0015C         BRB     33$                                ;
              51            54  D0 0015E 32$:    MOVL    R4, _Q                             ;
              65       04   A1  D0 00161         MOVL    4(_Q), (_H)                        ;
              52            51  D0 00165 33$:    MOVL    R1, P2                             ;
                            39  13 00168 34$:    BEQL    39$                                ;
         0C   A0       0C   A2  C0 0016A         ADDL2   12(P2), 12(P1)                     ; 1019
              66       0C   A0  C0 0016F         ADDL2   12(P1), (R6)                       ; 1020
                            60  7C 00173         CLRQ    (P1)                               ; 1021
                       08   A0  D4 00175         CLRL    8(P1)                              ;
              54            6E  9E 00178         MOVAB   PTR, _H                            ;
              51            64  D0 0017B 35$:    MOVL    (_H), R1                           ;
                            1D  13 0017E         BEQL    38$                                ;
         0C   A1       0C   A0  D1 00180         CMPL    12(P1), 12(R1)                     ;
                            05  1E 00185         BGEQU   36$                                ;
              54            51  D0 00187         MOVL    R1, _H                             ;
                            EF  11 0018A         BRB     35$                                ;
                            06  1B 0018C 36$:    BLEQU   37$                                ;
              54       04   A1  9E 0018E         MOVAB   4(R1), _H                          ;
                            E7  11 00192         BRB     35$                                ;
         08   A0       08   A1  D0 00194 37$:    MOVL    8(R1), 8(P1)                       ;
              54       08   A1  9E 00199         MOVAB   8(R1), _H                          ;
              64            50  D0 0019D 38$:    MOVL    P1, (_H)                           ;
                          FF67  31 001A0         BRW     24$                                ; 1017
         28   A3            66  D0 001A3 39$:    MOVL    (R6), 40(R3)                       ; 1024
              50 00000000G  8F  D0 001A7         MOVL    #DCX$_NORMAL, R0                   ; 1025
                                04 001AE         RET                                        ; 1027

; Routine Size:  431 bytes,    Routine Base:  $CODE$ + 02FE
```

```
1 %SBTTL  'remove_seg - Remove one segment'
1
1 ROUTINE remove_seg (anl : REF BBLOCK, anlseg : REF BBLOCK) =
2 BEGIN
2 !++
2 !
2 ! Remove one segment fixing pointers and freeing storage
2 !
2 ! Inputs:
2 !
2 !       anl                         Address of anl structure
2 !       anlseg                      Address of anlseg structure to be removed
2 !
2 ! Outputs:
2 !
2 !       NONE
2 !
2 ! Return value:
2 !
2 !       dcx$_normal
2 !       memory deallocation error
2 !
2 !--
2
2 ! this segment is unprofitable, eliminate it
2 !
2 LOCAL
2     seg : REF BBLOCK,    ! replacement
2     anl_p : REF BBLOCK;
2
2 remque (anlseg [anlseg$q_queue], anl_p);
2 IF .anlseg [anlseg$l_prev] NEQA 0
2 THEN
3     BEGIN
3
3     BIND
3         p_seg = anlseg [anlseg$l_prev] : REF BBLOCK;
3
3     p_seg [anlseg$w_sons] = .p_seg [anlseg$w_sons] - 1;
3     seg = .anl [anl$l_flink];
3     INCR index FROM 1 TO .anlseg [anlseg$b_depth] - 2 DO
4         BEGIN
4
4         BIND
4             seg_next = seg [anlseg$l_next] : VECTOR [, LONG],
4             anlseg_string = anlseg [anlseg$t_string] : VECTOR [, BYTE];
4
4         seg = .seg_next [.anlseg_string [.index]];
3         END;
3     anl_p = .anl [anl$l_flink];
3     WHILE .anl_p NEQA anl [anl$q_queue] DO
4         BEGIN
4
4         BIND
4             next_p = anl_p [anlseg$l_next] : VECTOR [, LONG];
4
4         IF .next_p [.anlseg [anlseg$w_char]] EQLA .anlseg
```

```
:  692          1085  4          THEN
:  693          1086  4              next_p [.anlseg [anlseg$w_char]] = .seg;
:  694          1087  4          anl_p = .anl_p [anlseg$l_flink];
:  695          1088  3          END;
:  696          1089  2      END;
:  697          1090  2  perform (dcx$free_vm (.anlseg [anlseg$l_size], .anlseg));
:  698          1091  2  anl [anl$w_nsegs] = .anl [anl$w_nsegs] - 1;
:  699          1092  2  RETURN dcx$_normal;
:  700          1093  2
:  701          1094  1  END;                                       ! 0' remove_seg



                                          00FC 00000 REMOVE_SEG:
                                                           .WORD    Save R2,R3,R4,R5,R6,R7                       ; 1030
                           53      08    BC  0F 00002      REMQUE   @ANLSEG, ANL_P                               ; 1058
                           54      08    AC  D0 00006      MOVL     ANLSEG, R4                                   ; 1059
                                   20    A4  D5 0000A      TSTL     32(R4)                                       ;
                                         53  13 0000D      BEQL     5$                                           ;
                           50      20    A4  D0 0000F      MOVL     32(R4), R0                                   ; 1066
                                   1C    A0  B7 00013      DECW     28(R0)                                       ;
                           52      04    AC  D0 00016      MOVL     ANL, R2                                      ; 1067
                           56      18    A2  D0 0001A      MOVL     24(R2), SEG                                  ;
                           57      14    A4  9A 0001E      MOVZBL   20(R4), R7                                   ; 1068
                           57            02  C2 00022      SUBL2    #2, R7                                       ;
                           51      30    A4  9E 00025      MOVAB    48(R4), R1                                   ; 1073
                                         55  D4 00029      CLRL     INDEX                                        ;
                                         0A  11 0002B      BRB      2$                                           ;
                           50          6541  9A 0002D 1$:  MOVZBL   (INDEX)[R1], R0                              ; 1075
                           56     043C C640  D0 00031      MOVL     1084(SEG)[R0], SEG                           ;
                  F2       55            57  F3 00037 2$:  AOBLEQ   R7, INDEX, 1$                                ; 1068
                           53      18    A2  D0 0003B      MOVL     24(R2), ANL_P                                ; 1077
                           50      0E    A4  9E 0003F      MOVAB    14(R4), R0                                   ; 1084
                           51      18    A2  9E 00043 3$:  MOVAB    24(R2), R1                                   ; 1078
                           51            53  D1 00047      CMPL     ANL_P, R1                                    ;
                                         16  13 0004A      BEQL     5$                                           ;
                           51            60  3C 0004C      MOVZWL   (R0), R1                                     ; 1084
                           54     043C C341  D1 0004F      CMPL     1084(ANL_P)[R1], R4                          ;
                                         06  12 00055      BNEQ     4$                                           ;
                    043C C341            56  D0 00057      MOVL     SEG, 1084(ANL_P)[R1]                         ; 1086
                           53            63  D0 0005D 4$:  MOVL     (ANL_P), ANL_P                               ; 1087
                                         E1  11 00060      BRB      3$                                           ; 1078
                                         54  DD 00062 5$:  PUSHL    R4                                           ; 1090
                                   08    A4  DD 00064      PUSHL    8(R4)                                        ;
                    0000G  CF            02  FB 00067      CALLS    #2, DCX$FREE_VM                              ;
                           0E            50  E9 0006C      BLBC     STATUS, 6$                                   ;
                           50      04    AC  D0 0006F      MOVL     ANL, R0                                      ; 1091
                                   16    A0  B7 00073      DECW     22(R0)                                       ;
                           50 00000000G  8F  D0 00076      MOVL     #DCX$_NORMAL, R0                             ; 1092
                                         04 0007D 6$:      RET                                                   ; 1094

; Routine Size:  126 bytes,    Routine Base:  $CODE$ + 04AD
;      1 %SBTTL  'Eliminate_seg - Remove unprofitable segments'
;      1
;      1 ROUTINE eliminate_seg (anl : REF BBLOCK, this : REF BBLOCK) =
;      2 BEGIN
;      2 !++
;      2 !
;      2 ! Determine if a segment is unprofitable and ifso, remove it
;      2 !
;      2 ! Inputs:
;      2 !
;      2 !       anl                         Address of anl structure
;      2 !       this                        Address of anlseg structure in question
;      2 !
;      2 ! Outputs:
;      2 !
;      2 !       NONE
;      2 !
;      2 ! Return value:
;      2 !
;      2 !       dcx$_normal
;      2 !       memory deallocation error
;      2 !
;      2 !--
;      2
;      2 ! Macro to scale a size based on the ratio of observed to estimated
;      2 ! data.
;      2 !
;    M 2 MACRO scale_map (size) =
;    M 2     BEGIN
;    M 2
;    M 2     LOCAL
;    M 2         scaled_size : LONG,             ! result
;    M 2         remainder : LONG,               ! throw away
;    M 2         ext_prod : VECTOR [2, LONG];    ! extended intermediate value
;    M 2
;    M 2     BUILTIN
;    M 2         emul,
;    M 2         ediv;
;    M 2
;    M 2     emul (%REF (size), anl [anl$l_ratio_num], %REF (0), ext_prod);
;    M 2     ediv (anl [anl$l_ratio_denom], ext_prod, scaled_size, remainder);
;    M 2     .scaled_size
;    M 2
;      2     END%;
;      2
;      2 BIND
;      2     aff = this [anlseg$l_prev] : REF BBLOCK,
;      2     this_count = this [anlseg$l_count] : VECTOR [, LONG],
;      2     next = this [anlseg$l_next] : VECTOR [, LONG];
;      2
;      2 ! If we have no true successors then we can
;      2 ! consider eliminating the current anlseg block
;      2 !
;      2 IF .this [anlseg$w_sons] NEQ 0
;      2 THEN
;      2     0
;      2 ELSE IF .this [anlseg$l_chars] EQL 0
```

```
       2 THEN
       2     remove_seg (.anl, .this)              ! Basic map is bigger than data
       2 ELSE IF .this [anlseg$w_active] EQL 0
       2 THEN
       2     remove_seg (.anl, .this)
       2 ELSE IF .this [anlseg$l_prev] EQLA 0
       2 THEN
       3     BEGIN
       3
       3     ! check profitability of eliminating the last block
       3     !
       4     IF (
       5             (
       6                 (.this [anlseg$l_comp_bits])                                  ! this data
       6             -   ((.this [anlseg$l_chars] - .this_count [dcx$c_eor]) * 8)      ! raw data
       5             )
P      4         +   scale_map (
P      4                 (dcxsbm$k_length*8                                            ! this map
P      4                 +   (.this [anlseg$w_active_r] - 1) * (2 + 16))
P      4             -   (0)                                                           ! null map
       5             )
       3         ) GEQ 0                  ! true if current total size is greater
       3     THEN
       3         remove_seg (.anl, .this);
       3     END
       2 ELSE
       3     BEGIN
       3
       3     ! check profitability of eliminating the block
       3     !
       3     LOCAL
       3         aff2 : BBLOCK [anlseg$k_length];
       3
       3     BIND
       3         aff2_count = aff2 [anlseg$l_count] : VECTOR [, LONG];
       3
       3     CH$MOVE (.aff [anlseg$l_size], .aff, aff2);
       3     aff2 [anlseg$l_chars] = .aff2 [anlseg$l_chars] + .this [anlseg$l_chars];
       3     DECR i FROM anlseg$c_count - 1 TO 0 DO
       3         aff2_count [.i] = .aff2_count [.i] + .this_count [.i];
       3     perform (huffman_size (.anl, aff2));
       4     IF (
       5             (
       6                 (.this [anlseg$l_adj_bits])                                          ! this data
       6             +   (.aff [anlseg$l_adj_bits])                                           ! aff data
       6             -   (.aff2 [anlseg$l_adj_bits])                                          ! aff2 data
       5             )
P      4         + scale_map (
P      4                 (dcxsbm$k_length*8  +  (.this [anlseg$w_active_r] - 1) * (2 + 16)  +    ! this map
P      4                     (.this [anlseg$b_max_char] - .this [anlseg$b_min_char] + 1) * 16)
P      4             +   (dcxsbm$k_length*8  +  (.aff [anlseg$w_active_r] - 1) * (2 + 16)  +     ! aff map
P      4                     (.aff [anlseg$b_max_char] - .aff [anlseg$b_min_char] + 1) * 16)
P      4             -   (dcxsbm$k_length*8  +  (.aff2 [anlseg$w_active_r] - 1) * (2 + 16)  +    ! aff2 map
P      4                     (IF .anl [anl$w_nsegs] GTR 2
P      4                      THEN ((.aff2 [anlseg$b_max_char] - .aff2 [anlseg$b_min_char] + 1) * 16)
P      4                      ELSE 0))
       5             )
;  817          1209  3          ) GEQ 0                     ! true if current total size is greater
;  818          1210  3      THEN
;  819          1211  4          BEGIN
;  820          1212  4          CH$MOVE (.aff [anl$seg$l_size], aff2, .aff);
;  821          1213  4          remove_seg (.anl, .this);
;  822          1214  4          END
;  823          1215  2      END;
;  824          1216  2  RETURN dcx$_normal;
;  825          1217  2
;  826          1218  1  END;                                    ! Of eliminate_seg



                                                     07FC 00000 ELIMINATE_SEG:
                                                                     .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10                   ; 1097
                                  5E      F7B8   CE   9E 00002       MOVAB    -2120(SP), SP                                      ;
                                  56        08   AC   D0 00007       MOVL     THIS, R6                                           ; 1141
                                  5A        38   A6   9E 0000B       MOVAB    56(R6), R10                                        ; 1142
                                            1C   A6   B5 0000F       TSTW     28(R6)                                             ; 1148
                                                 03   13 00012       BEQL     2$                                                 ;
                                               00F3   31 00014 1$:   BRW      10$                                                ;
                                  58        04   AC   D0 00017 2$:   MOVL     ANL, R8                                            ; 1153
                                  59        2C   A6   D0 0001B       MOVL     44(R6), R9                                         ; 1151
                                                 33   13 0001F       BEQL     3$                                                 ;
                                            10   A6   B5 00021       TSTW     16(R6)                                             ; 1154
                                                 2E   13 00024       BEQL     3$                                                 ;
                                            20   A6   D5 00026       TSTL     32(R6)                                             ; 1157
                                                 2C   12 00029       BNEQ     4$                                                 ;
                        50      0400   CA             59   C3 0002B  SUBL3    R9, 1024(R10), R0                                  ; 1166
                                       51       24 B640   7E 00031   MOVAQ    @36(R6)[R0], R1                                    ; 1164
                                       50       12   A6   3C 00036   MOVZWL   18(R6), R0                                         ; 1172
                                       50            12   C4 0003A   MULL2    #18, R0                                            ;
                                       50       4E   A0   9E 0003D   MOVAB    78(R0), R0                                         ;
      F8    AD          00        20   A8            50   7A 00041   EMUL     R0, 32(R8), #0, EXT_PROD                           ;
            52          50        F8   AD       24   A8   7B 00048   EDIV     36(R8), EXT_PROD, SCALED_SIZE, REMAINDER           ;
                                       51            50   C0 0004F   ADDL2    SCALED_SIZE, R1                                    ;
                                                     C0   19 00052   BLSS     1$                                                 ; 1173
                                                   00AA   31 00054 3$: BRW    9$                                                 ; 1175
                                       57       20   A6   D0 00057 4$: MOVL   32(R6), R7                                         ; 1188
                  08    AE             67       08   A7   28 0005B   MOVC3    8(R7), (R7), AFF2                                  ;
                                  34   AE            59   C0 00061   ADDL2    R9, AFF2+44                                        ; 1189
                                       50     0100   8F   3C 00065   MOVZWL   #256, I                                            ; 1190
                               40 AE40             6A40   C0 0006A 5$: ADDL2  (R10)[I], AFF2_COUNT[I]                            ; 1191
                                       F7            50   F4 00070   SOBGEQ   I, 5$                                              ;
                                            08   AE   9F 00073       PUSHAB   AFF2                                               ; 1192
                                                 58   DD 00076       PUSHL    R8                                                 ;
                              FD56   CF          02   FB 00078       CALLS    #2, HUFFMAN_SIZE                                   ;
                                       01        50   E8 0007D       BLBS     STATUS, 6$                                         ;
                                                      04 00080       RET                                                         ;
                        53      28   A6         28   A7   C1 00081 6$: ADDL3  40(R7), 40(R6), R3                                 ; 1196
                                     53         30   AE   C2 00087   SUBL2    AFF2+40, R3                                        ; 1197
                                     50         12   A6   3C 0008B   MOVZWL   18(R6), R0                                         ; 1208
                                     50              12   C4 0008F   MULL2    #18, R0                                            ;
                                     51         16   A6   9A 00092   MOVZBL   22(R6), R1                                         ;
                                     52         15   A6   9A 00096   MOVZBL   21(R6), R2                                         ;
                               51                     [illegible]9A          SUBL2    R2, R1                                           ;
                               51                                            [illegible]LL2    #16, R1                                          ;
                               51              50  [illegible]               [illegible]DL2    R0, R1                                           ;
                               50        12    A7  [illegible]0A3            MOVZWL   18(R7), R0                                       ;
                               50              12  C4 [illegible]00A7        MULL2    #18, R0                                          ;
                               52        16    A7  9A 000AA                  MOVZBL   22(R7), R2                                       ;
                               54        15    A7  9A 000AE                  MOVZBL   21(R7), R4                                       ;
                               52              54  C2 000B2                  SUBL2    R4, R2                                           ;
                               52              10  C4 000B5                  MULL2    #16, R2                                          ;
                               50              52  C0 000B8                  ADDL2    R2, R0                                           ;
                               50              51  C0 000BB                  ADDL2    R1, R0                                           ;
                               52        1A    AE  3C 000BE                  MOVZWL   AFF2+18, R2                                      ;
                               52              12  C4 000C2                  MULL2    #18, R2                                          ;
                               02        16    A8  B1 000C5                  CMPW     22(R8), #2                                       ;
                                               13  1B 000C9                  BLEQU    7$                                               ;
                               51        1E    AE  9A 000CB                  MOVZBL   AFF2+22, R1                                      ;
                               54        1D    AE  9A 000CF                  MOVZBL   AFF2+21, R4                                      ;
                               51              54  C2 000D3                  SUBL2    R4, R1                                           ;
                               51              10  C4 000D6                  MULL2    #16, R1                                          ;
                               51              10  C0 000D9                  ADDL2    #16, R1                                          ;
                                               02  11 000DC                  BRB      8$                                               ;
                                               51  D4 000DE 7$:              CLRL     R1                                               ;
                               51              52  C0 000E0 8$:              ADDL2    R2, R1                                           ;
                               50              51  C2 000E3                  SUBL2    R1, R0                                           ;
                               50        6E    A0  9E 000E6                  MOVAB    110(R0), R0                                      ;
        6E          00       20    A8              50  7A 000EA                  EMUL     R0, 32(R8), #0, EXT_PROD                         ;
        51          50             6E        24    A8  7B 000F0                  EDIV     36(R8), EXT_PROD, SCALED_SIZE, REMAINDER         ;
                               53              50  C0 000F6                  ADDL2    SCALED_SIZE, R3                                  ;
                                               0F  19 000F9                  BLSS     10$                                              ; 1209
                    67       08    AE        08    A7  28 000FB                  MOVC3    8(R7), AFF2, (R7)                                ; 1212
                                               56  DD 00101 9$:              PUSHL    R6                                               ; 1213
                                               58  DD 00103                  PUSHL    R8                                               ;
                          FE78     CF              02  FB 00105                  CALLS    #2, REMOVE_SEG                                   ;
                               50 00000000G    8F  D0 0010A 10$:             MOVL     #DCX$_NORMAL, R0                                 ; 1216
                                               04 00111                      RET                                                       ; 1218

; Routine Size:  274 bytes,    Routine Base:  $CODE$ + 052B
```

```
1 %SBTTL  'build_map_seg - Build a map segment'
1
1 ROUTINE build_map_seg (anl : REF BBLOCK, anlseg : REF BBLOCK, dcxsbm : REF BBLOCK) =
2 BEGIN
2 !++
2 !
2 ! Build a map segment
2 !
2 ! Inputs:
2 !
2 !       anl                     Address of anl block
2 !       anlseg                  Analysis segment structure address
2 !       dcxsbm                  Analysis of allocated map segment
2 !
2 ! Outputs:
2 !
2 !       dcxsbm                  Filled in
2 !
2 ! Return value:
2 !
2 !       dcx$_rormal             All is well
2 !--
2
2 LOCAL
2     flags : REF BITVECTOR,
2     nodes : REF VECTOR [, BYTE],
2     next : REF VECTOR [, WORD],
2     p1 : REF VECTOR [6, LONG],
2     p2 : REF VECTOR [6, LONG],
2     ptr : LONG,                       ! pointer into list
2     list : VECTOR [6 * anlseg$c_count, LONG];   ! storage list
2
2 BIND
2     count = anlseg [anlseg$l_count] : VECTOR [, LONG];
2
2 flags = .dcxsbm + .dcxsbm [dcxsbm$w_flags];
2 nodes = .dcxsbm + .dcxsbm [dcxsbm$w_nodes];
2 IF .dcxsbm [dcxsbm$w_next] NEQ 0
2 THEN
2     next = .dcxsbm + .dcxsbm [dcxsbm$w_next]
2 ELSE
2     next = 0;
2 ptr = 0;
2 p1 = list [0];
2 DECR i FROM anlseg$c_count-1 TO 0 DO
3     BEGIN
3     p1 [3] = .count [.i];
3     IF .p1 [3] NEQ 0 OR NOT .anl [anl$v_bounded]
3     THEN
4         BEGIN
4         p1 [4] = 0;               ! node offset
4         p1 [5] = .i;              ! character code
4         tree_insert (ptr, p1 [0]);
4         p1 = p1 [6];
3         END;
2     END;
2 IF .anlseg [anlseg$w_active] EQL 1
```

```
THEN
    BEGIN
    p2 = .ptr;
    p1 [3] = .p2 [3];
    p1 [4] = .p2 [4];
    p1 [5] = .p2 [5];
    tree_insert (ptr, p1 [0]);
    END;
DECR high FROM .anlseg [anlseg$w_active_r] - 2 TO 0 DO
    BEGIN

    LOCAL
        p : VECTOR [2, LONG];

    BIND
        p2 = p [0] : REF VECTOR [6, LONG],
        p1 = p [1] : REF VECTOR [6, LONG];

    DECR i FROM 1 TO 0 DO
        BEGIN

        LOCAL
            q : REF VECTOR [6, LONG];

        p [.i] = q = tree_least (ptr);
        IF .q [5] EQL dcx$c_eor
        THEN
            BEGIN
            flags [2*.high + .i] = false;
            nodes [2*.high + .i] = 0;
            END
        ELSE IF .q [5] GEQ 0
        THEN
            BEGIN
            flags [2*.high + .i] = true;
            nodes [2*.high + .i] = .q [5];
            IF .next NEQA 0
            THEN
                BEGIN

                BIND
                    next_seg_a = anlseg [anlseg$l_next] : VECTOR [, LONG],
                    next_seg = next_seg_a [.q [5]] : REF BBLOCK;

                next [.q [5] - .anlseg [anlseg$b_min_char]] = .next_seg [anlseg$w_id];
                END;
            END
        ELSE
            BEGIN
            flags [2*.high + .i] = false;
            nodes [2*.high + .i] = .q [4];
            END;
        END;
    p1 [3] = .p1 [3] + .p2 [3];
    p1 [4] = .high;
    p1 [5] = -1;
    tree_insert (ptr, p1 [0]);
```

```
;  942           1333  2      END;
;  943           1334  2
;  944           1335  2 RETURN dcx$_normal;
;  945           1336  2
;  946           1337  1 END;                           ! of build_map_seg



                                         03FC 00000 BUILD_MAP_SEG:
                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                  ; 1221
                      5E      E7DC   CE  9E 00002         MOVAB   -6180(SP), SP                                 ;
                      55        08   AC  D0 00007         MOVL    ANLSEG, R5                                    ; 1252
                      50        0C   AC  D0 0000B         MOVL    DCXSBM, R0                                    ; 1254
                      59        06   A0  3C 0000F         MOVZWL  6(R0), FLAGS                                  ;
                      59             50  C0 00013         ADDL2   R0, FLAGS                                     ;
                      56        08   A0  3C 00016         MOVZWL  8(R0), NODES                                  ; 1255
                      56             50  C0 0001A         ADDL2   R0, NODES                                     ;
                                0A   A0  B5 0001D         TSTW    10(R0)                                        ; 1256
                                     09  13 00020         BEQL    1$                                            ;
                      58        0A   A0  3C 00022         MOVZWL  10(R0), NEXT                                  ; 1258
                      58             50  C0 00026         ADDL2   R0, NEXT                                      ;
                                     02  11 00029         BRB     2$                                            ;
                                     58  D4 0002B 1$:     CLRL    NEXT                                          ; 1260
                                     6E  D4 0002D 2$:     CLRL    PTR                                           ; 1261
                      50        0C   AE  9E 0002F         MOVAB   LIST, P1                                      ; 1262
                      52      0100   8F  3C 00033         MOVZWL  #256, I                                       ; 1263
               0C     A0        38 A542  D0 00038 3$:     MOVL    56(R5)[I], 12(P1)                             ; 1265
                                     04  12 0003E         BNEQ    4$                                            ; 1266
                      37        04   BC  E8 00040         BLBS    @ANL, 9$                                      ;
                                10   A0  D4 00044 4$:     CLRL    16(P1)                                        ; 1269
               14     A0             52  D0 00047         MOVL    I, 20(P1)                                     ; 1270
                                     60  7C 0004B         CLRQ    (P1)                                          ; 1271
                                08   A0  D4 0004D         CLRL    8(P1)                                         ;
                      53             6E  9E 00050         MOVAB   PTR, _H                                       ;
                      51             63  D0 00053 5$:     MOVL    (_H), R1                                      ;
                                     1D  13 00056         BEQL    8$                                            ;
               0C     A1        0C   A0  D1 00058         CMPL    12(P1), 12(R1)                                ;
                                     05  1E 0005D         BGEQU   6$                                            ;
                      53             51  D0 0005F         MOVL    R1, _H                                        ;
                                     EF  11 00062         BRB     5$                                            ;
                                     06  1B 00064 6$:     BLEQU   7$                                            ;
                      53        04   A1  9E 00066         MOVAB   4(R1), _H                                     ;
                                     E7  11 0006A         BRB     5$                                            ;
               08     A0        08   A1  DC 0006C 7$:     MOVL    8(R1), 8(P1)                                  ;
                      53        08   A1  9E 00071         MOVAB   8(R1), _H                                     ;
                      63             80  7E 00075 8$:     MOVAQ   (P1)+, (_H)                                   ;
                      50             10  C0 00078         ADDL2   #16, P1                                       ; 1272
                      BA             52  F4 0007B 9$:     SOBGEQ  I, 3$                                         ; 1263
                      01        10   A5  B1 0007E         CMPW    16(R5), #1                                    ; 1275
                                     3A  12 00082         BNEQ    14$                                           ;
                      51             6E  D0 00084         MOVL    PTR, P2                                       ; 1278
               0C     A0        0C   A1  7D 00087         MOVQ    12(P2), 12(P1)                                ; 1279
               14     A0        14   A1  D0 0008C         MOVL    20(P2), 20(P1)                                ; 1281
                                     60  7C 00091         CLRQ    (P1)                                          ; 1282
                                08   A0  D4 00093         CLRL    8(P1)                                         ;
```

```
                     52           6E  9E 00096          MOVAB   PTR, _H                                 ;
                     51           62  D0 00099 10$:     MOVL    (_H), R1                                ;
                                  1D  13 0009C          BEQL    13$                                     ;
                0C   A1     0C    A0  D1 0009E          CMPL    12(P1), 12(R1)                          ;
                                  05  1E 000A3          BGEQU   11$                                     ;
                     52           51  D0 000A5          MOVL    R1, _H                                  ;
                                  EF  11 000A8          BRB     10$                                     ;
                                  06  1B 000AA 11$:     BLEQU   12$                                     ;
                     52     04    A1  9E 000AC          MOVAB   4(R1), _H                               ;
                                  E7  11 000B0          BRB     10$                                     ;
                08   A0     08    A1  D0 000B2 12$:     MOVL    8(R1), 8(P1)                            ;
                     52     08    A1  9E 000B7          MOVAB   8(R1), _H                               ;
                     62           50  D0 000BB 13$:     MOVL    P1, (_H)                                ;
                     51     12    A5  3C 000BE 14$:     MOVZWL  18(R5), HIGH                            ; 1305
                                  51  D7 000C2          DECL    HIGH                                    ;
                                00C5  31 000C4          BRW     31$                                     ;
          57         51           01  78 000C7 15$:     ASHL    #1, HIGH, R7                            ; 1304
                     54           01  D0 000CB          MOVL    #1, I                                   ;
                     53           6E  9E 000CE 16$:     MOVAB   PTR, _H                                 ; 1300
                                  63  D5 000D1          TSTL    (_H)                                    ;
                                  04  12 000D3          BNEQ    17$                                     ;
                                  50  D4 000D5          CLRL    Q                                       ;
                                  21  11 000D7          BRB     20$                                     ;
                            00    B3  D5 000D9 17$:     TSTL    @0(_H)                                  ;
                                  05  13 000DC          BEQL    18$                                     ;
                     53           63  D0 000DE          MOVL    (_H), _H                                ;
                                  F6  11 000E1          BRB     17$                                     ;
                     52           63  D0 000E3 18$:     MOVL    (_H), R2                                ;
                     50     08    A2  D0 000E6          MOVL    8(R2), _Q                               ;
                                  07  13 000EA          BEQL    19$                                     ;
                08   A2     08    A0  D0 000EC          MOVL    8(_Q), 8(R2)                            ;
                                  07  11 000F1          BRB     20$                                     ;
                     50           52  D0 000F3 19$:     MOVL    R2, _Q                                  ;
                     63     04    A0  D0 000F6          MOVL    4(_Q), (_H)                             ;
                04 AE44           50  D0 000FA 20$:     MOVL    Q, P[I]                                 ;
          53         57           54  C1 000FF          ADDL3   I, R7, R3                               ; 1304
                     52     14    A0  D0 00103          MOVL    20(Q), R2                               ; 1301
            00000100 8F           52  D1 00107          CMPL    R2, #256                                ;
                                  09  12 0010E          BNEQ    22$                                     ;
          00         69           53  E5 00110          BBCC    R3, (FLAGS), 21$                        ; 1304
                                6346  94 00114 21$:     CLRB    (R3)[NODES]                             ; 1305
                                  2E  11 00117          BRB     26$                                     ; 1301
                                  52  D5 00119 22$:     TSTL    R2                                      ; 1307
                                  21  19 0011B          BLSS    24$                                     ;
          00         69           53  E2 0011D          BBSS    R3, (FLAGS), 23$                        ; 1310
                   6346           52  90 00121 23$:     MOVB    R2, (R3)[NODES]                         ; 1311
                                  58  D5 00125          TSTL    NEXT                                    ; 1312
                                  1E  13 00127          BEQL    26$                                     ;
                     50     15    A5  9A 00129          MOVZBL  21(R5), R0                              ; 1320
          50         52           50  C3 0012D          SUBL3   R0, R2, R0                              ;
                     52   043C  C542  D0 00131          MOVL    1084(R5)[R2], R2                        ;
                   6840     0C    A2  B0 00137          MOVW    12(R2), (NEXT)[R0]                      ;
                                  09  11 0013C          BRB     26$                                     ; 1307
          00         69           53  E5 0013E 24$:     BBCC    R3, (FLAGS), 25$                        ; 1325
                   6346     10    A0  90 00142 25$:     MOVB    16(Q), (R3)[NODES]                      ; 1326
                     84           54  F4 00147 26$:     SOBGEQ  I, 16$                                  ; 1294
                     50     08    AE  D0 0014A          MOVL    P1, R0                                  ; 1329
```

```
                    52          04  AE  D0 0014E            MOVL    P2, R2                          ;
               0C   A0          0C  A2  C0 00152            ADDL2   12(R2), 12(R0)                  ;
               10   A0              51  D0 00157            MOVL    HIGH, 16(R0)                    ; 1330
               14   A0              01  CE 0015B            MNEGL   #1, 20(R0)                      ; 1331
                                    60  7C 0015F            CLRQ    (R0)                            ; 1332
                                08  A0  D4 00161            CLRL    8(R0)                           ;
                    53              6E  9E 00164            MOVAB   PTR, _H                         ;
                    52              63  D0 00167 27$:       MOVL    (_H), R2                        ;
                                    1D  13 0016A            BEQL    30$                             ;
               0C   A2          0C  A0  D1 0016C            CMPL    12(R0), 12(R2)                  ;
                                    05  1E 00171            BGEQU   28$                             ;
                    53              52  D0 00173            MOVL    R2, _H                          ;
                                    EF  11 00176            BRB     27$                             ;
                                    06  1B 00178 28$:       BLEQU   29$                             ;
                    53          04  A2  9E 0017A            MOVAB   4(R2), _H                       ;
                                    E7  11 0017E            BRB     27$                             ;
               08   A0          08  A2  D0 00180 29$:       MOVL    8(R2), 8(R0)                    ;
                    53          08  A2  9E 00185            MOVAB   8(R2), _H                       ;
                    63              50  D0 00189 30$:       MOVL    R0, (_H)                        ;
                    02              51  F4 0018C 31$:       SOBGEQ  HIGH, 32$                       ; 1284
                                    03  11 0018F            BRB     33$                             ;
                                  FF33  31 00191 32$:       BRW     15$                             ;
                    50 00000000G    8F  D0 00194 33$:       MOVL    #DCX$_NORMAL, R0                ; 1335
                                        04 0019B            RET                                     ; 1337

; Routine Size:  412 bytes,    Routine Base:  $CODE$ + 063D
```

```
1 %SBTTL  'dcx$make_map - Compute mapping function'
1
1 GLOBAL ROUTINE dcx$make_map (context_addr, map_addr, map_size) =
2 BEGIN
2 !++
2 !
2 ! Compute mapping function
2 !
2 ! Inputs:
2 !
2 !       context_addr.mz.r       Address of context longword
2 !
2 ! Outputs:
2 !
2 !       context_addr.mz.r       Context block accumulates data
2 !       map_addr.wa.r           Address of longword to receive map address
2 !       map_size.wl.r           Address of longword to receive map length
2 !
2 ! Return value:
2 !
2 !       status.wlc.v
2 !
2 !               dcx$_normal     All is well
2 !               dcx$_invctx     Invalid context block
2 !               lib$_insvirmem
2 !--
2
2 BUILTIN
2     NULLPARAMETER;
2
2 BIND
2     ctx = .context_addr : REF BBLOCK,
2     anl = ctx [ctx$l_specific] : BBLOCK,
2     dcxmap = ctx [ctx$l_map] : REF BBLOCK;
2
2 LOCAL
2     id,
2     seg : REF BBLOCK,
2     size;
2
2 .map_addr = 0;
2 IF NOT NULLPARAMETER (3)
2 THEN
2     .map_size = 0;
2
2 ! validate context block
2 !
2 perform (dcx$ctx_check (.ctx, ctx$c_anlyz));
2
2 ! Compute ratio of estimated to observed data size.
2 ! Give priority first to an estimate of data size,
2 ! then to an estimate of number of records, and if
2 ! neither of these is provided, assume that estimated
2 ! is equal to observed.
2 !
2 IF .anl [anl$v_est_bytes] AND .anl [anl$l_est_d_bytes] GTR 0
2 THEN
```

```
3           BEGIN
3           anl [anl$l_ratio_num] = .anl [anl$l_d_bytes];
3           anl [anl$l_ratio_denom] = .anl [anl$l_est_d_bytes];
3           END
2       ELSE IF .anl [anl$v_est_recs] AND .anl [anl$l_est_d_recs] GTR 0
2       THEN
3           BEGIN
3           anl [anl$l_ratio_num] = .anl [anl$l_d_recs];
3           anl [anl$l_ratio_denom] = .anl [anl$l_est_d_recs];
3           END
2       ELSE
3           BEGIN
3           anl [anl$l_ratio_num] = 1;
3           anl [anl$l_ratio_denom] = 1;
2           END;
2
2       ! pre-process data
2       !
2       DECR depth FROM .anl [anl$b_depth] TO 1 DO
3           BEGIN
3
3           LOCAL
3               seg : REF BBLOCK;
3
3           seg = .anl [anl$l_flink];
3           WHILE .seg NEQA anl [anl$q_queue] DO
4               BEGIN
4
4               LOCAL
4                   next_seg : REF BBLOCK;
4
4               next_seg = .seg [anlseg$l_flink];
4               IF
4                       .seg [anlseg$b_depth] EQL .depth
4                   AND
4                       .seg [anlseg$l_chars] EQL 0
4                   AND
4                       .seg [anlseg$w_sons] EQL 0
4               THEN
4                   remove_seg (anl, .seg)
4               ELSE
4                   perform (huffman_size (anl, .seg));         ! compute compressed size
4               seg = .next_seg;
3               END;
2           END;
2       DECR depth FROM .anl [anl$b_depth] TO 1 DO
3           BEGIN
3
3           LOCAL
3               ptr : LONG,                 ! random tree listhead
3               p1 : REF VECTOR [5, LONG],
3               list : VECTOR [5 * dcx$c_max_segs, LONG],
3               seg : REF BBLOCK;
3
3           ptr = 0;
3           p1 = list [0];
3           seg = .anl [anl$l_flink];
```

```
3     WHILE .seg NEQA anl [anl$q_queue] DO
4         BEGIN
4         IF
4                 .seg [anlseg$b_depth] EQL .depth
4             AND
4                 .seg [anlseg$w_sors] EQL 0
4         THEN
5             BEGIN
5             p1 [3] = .seg [anlseg$l_chars];
5             p1 [4] = .seg;
5             tree_insert (ptr, p1 [0]);
5             p1 = p1 [5];
4             END;
4         seg = .seg [anlseg$l_flink];
3         END;
3     WHILE (p1 = tree_least (ptr)) NEQA 0 DO
3         perform (eliminate_seg (anl, .p1 [4]));
2     END;
2
2 ! Compute size of map
2 !
2 id = 0;
2 size = dcxmap$k_length;
2 seg = .anl [anl$l_flink];
2 WHILE .seg NEQA anl [anl$q_queue] DO
3     BEGIN
3     seg [anlseg$w_id] = .id;
3     id = .id + 1;
3     size = .size + dcxsbm$k_length;
3     size = .size + (2*(.seg [anlseg$w_active_r]-1) + 7) / 8;
3     size = .size + 2*(.seg [anlseg$w_active_r]-1);
3     IF .seg [anlseg$l_flink] NEQA .seg [anlseg$l_blink]
3     THEN
3         size = .size + 2 * (.seg [anlseg$b_max_char] - .seg [anlseg$b_min_char] + 1);
3     seg = .seg [anlseg$l_flink];
2     END;
2
2 ! Allocate map and populate
2 !
2 perform (dcx$get_vm (.size, dcxmap));
2 dcxmap [dcxmap$l_size] = .size;
2 dcxmap [dcxmap$w_version] = dcxmap$c_version;
2 dcxmap [dcxmap$l_sanity] = dcxmap$c_sanity;
2 dcxmap [dcxmap$l_flags] = 0;
2 dcxmap [dcxmap$w_nsubs] = .id;
2 IF .id NEQ 0
2 THEN
3     BEGIN
3
3     LOCAL
3         seg : REF BBLOCK,
3         dcxsbm : REF BBLOCK;
3
3     dcxmap [dcxmap$w_sub0] = dcxmap$k_length;
3     seg = .anl [anl$l_flink];
3     dcxsbm = .dcxmap + .dcxmap [dcxmap$w_sub0];
3     WHILE .seg NEQA anl [anl$q_queue] DO
```

```
; 1119          1509 4          BEGIN
; 1120          1510 4          dcxsbm [dcxsbm$w_size] = dcxsbm$k_length;
; 1121          1511 4          dcxsbm [dcxsbm$b_min_char] = .seg [anlseg$b_min_char];
; 1122          1512 4          dcxsbm [dcxsbm$b_max_char] = .seg [anlseg$b_max_char];
; 1123          1513 4          dcxsbm [dcxsbm$b_escape] = 0;
; 1124          1514 4          dcxsbm [dcxsbm$v_escape] = false;
; 1125          1515 4          dcxsbm [dcxsbm$v_unbounded] = false;
; 1126          1516 4          dcxsbm [dcxsbm$w_flags] = .dcxsbm [dcxsbm$w_size];
; 1127          1517 4          dcxsbm [dcxsbm$w_size] = .dcxsbm [dcxsbm$w_size] + (2*(.seg [anlseg$w_active_r]-1) + 7) / 8;
; 1128          1518 4          dcxsbm [dcxsbm$w_nodes] = .dcxsbm [dcxsbm$w_size];
; 1129          1519 4          dcxsbm [dcxsbm$w_size] = .dcxsbm [dcxsbm$w_size] + 2*(.seg [anlseg$w_active_r]-1);
; 1130          1520 4          IF .seg [anlseg$l_flink] NEQA .seg [anlseg$l_blink]
; 1131          1521 4          THEN
; 1132          1522 5              BEGIN
; 1133          1523 5              dcxsbm [dcxsbm$w_next] = .dcxsbm [dcxsbm$w_size];
; 1134          1524 5              dcxsbm [dcxsbm$w_size] = .dcxsbm [dcxsbm$w_size] + 2 * (.seg [anlseg$b_max_char] - .seg [anlseg$
; 1135          1525 5              END
; 1136          1526 4          ELSE
; 1137          1527 4              dcxsbm [dcxsbm$w_next] = 0;
; 1138          1528 4          build_map_seg (anl, .seg, .dcxsbm);
; 1139          1529 4          dcxsbm = .dcxsbm + .dcxsbm [dcxsbm$w_size];
; 1140          1530 4          seg = .seg [anlseg$l_flink];
; 1141          1531 3          END;
; 1142          1532 4      IF .dcxsbm NEQA (.dcxmap + .dcxmap [dcxmap$l_size])
; 1143          1533 3      THEN
; 1144          1534 3          RETURN ss$_accvio;
; 1145          1535 2      END;
; 1146          1536 2 .map_addr = .dcxmap;
; 1147          1537 2 IF NOT NULLPARAMETER (3)
; 1148          1538 2 THEN
; 1149          1539 2     .map_size = .size;
; 1150          1540 2 RETURN dcx$_normal;
; 1151          1541 2
; 1152          1542 1 END;                                          ! Of dcx$make_map



                                         00FC 00000          .ENTRY  DCX$MAKE_MAP, Save R2,R3,R4,R5,R6,R7                  ; 1340
                  5E        AFFC     CE   9E 00002          MOVAB   -20484(SP), SP                                              ;
        54     04 BC                 14   C1 00007          ADDL3   #20, @CONTEXT_ADDR, R4                                      ; 1370
        55     04 BC                 10   C1 0000C          ADDL3   #16, @CONTEXT_ADDR, R5                                      ; 1371
                          08         BC   D4 00011          CLRL    @MAP_ADDR                                                   ; 1378
                  03                 6C   91 00014          CMPB    (AP), #3                                                    ; 1379
                                     08   1F 00017          BLSSU   1$                                                          ;
                          0C         AC   D5 00019          TSTL    12(AP)                                                      ;
                                     03   13 0001C          BEQL    1$                                                          ;
                          0C         BC   D4 0001E          CLRL    @MAP_SIZE                                                   ; 1381
                                     51   D4 00021  1$:     CLRL    R1                                                          ; 1385
                  50      04         BC   D0 00023          MOVL    @CONTEXT_ADDR, R0                                           ;
                                  0000G   30 00027          BSBW    DCX$CTX_CHECK                                               ;
                  70                 50   E9 0002A          BLBC    STATUS, 8$                                                  ;
        11        64                 02   E1 0002D          BBC     #2, (R4), 2$                                                ; 1393
                          0C         A4   D5 00031          TSTL    12(R4)                                                      ;
                                     0C   15 00034          BLEQ    2$                                                          ;
             20   A4      04         A4   D0 00036          MOVL    4(R4), 32(R4)                                               ; 1396
```

```
                        24   A4      0C   A4  D0 0003B          MOVL    12(R4), 36(R4)                                          ; 1397
                                          1D  11 00040          BRB     4$                                                      ; 1393
                  11         64           03  E1 00042 2$:      BBC     #3, (R4), 3$                                            ; 1399
                                     10   A4  D5 00046          TSTL    16(R4)                                                  ;
                                          0C  15 00049          BLEQ    3$                                                      ;
                        20   A4      08   A4  D0 0004B          MOVL    8(R4), 32(R4)                                           ; 1402
                        24   A4      10   A4  D0 00050          MOVL    16(R4), 36(R4)                                          ; 1403
                                          08  11 00055          BRB     4$                                                      ; 1399
                        20   A4           01  D0 00057 3$:      MOVL    #1, 32(R4)                                              ; 1407
                        24   A4           01  D0 0005B          MOVL    #1, 36(R4)                                              ; 1408
                             53      14   A4  9A 0005F 4$:      MOVZBL  20(R4), DEPTH                                           ; 1413
                                          53  D6 00063          INCL    DEPTH                                                   ;
                                          3F  11 00065          BRB     10$                                                     ;
                             52      18   A4  D0 00067 5$:      MOVL    24(R4), SEG                                             ; 1419
                             50      18   A4  9E 0006B 6$:      MOVAB   24(R4), R0                                              ; 1420
                             50           52  D1 0006F          CMPL    SEG, R0                                                 ;
                                          32  13 00072          BEQL    10$                                                     ;
                             56           62  D0 00074          MOVL    (SEG), NEXT_SEG                                         ; 1426
     53           14    A2   08           00  ED 00077          CMPZV   #0, #8, 20(SEG), DEPTH                                  ; 1428
                                          15  12 0007D          BNEQ    7$                                                      ;
                                     2C   A2  D5 0007F          TSTL    44(SEG)                                                 ; 1430
                                          10  12 00082          BNEQ    7$                                                      ;
                                     1C   A2  B5 00084          TSTW    28(SEG)                                                 ; 1432
                                          0B  12 00087          BNEQ    7$                                                      ;
                                          52  DD 00089          PUSHL   SEG                                                     ; 1434
                                          54  DD 0008B          PUSHL   R4                                                      ;
                      FC42   CF           02  FB 0008D          CALLS   #2, REMOVE_SEG                                          ;
                                          0D  11 00092          BRB     9$                                                      ;
                                          52  DD 00094 7$:      PUSHL   SEG                                                     ; 1436
                                          54  DD 00096          PUSHL   R4                                                      ;
                      FA88   CF           02  FB 00098          CALLS   #2, HUFFMAN_SIZE                                        ;
                             01           50  E8 0009D 8$:      BLBS    STATUS, 9$                                              ;
                                          04     000A0          RET                                                             ;
                             52           56  D0 000A1 9$:      MOVL    NEXT_SEG, SEG                                           ; 1437
                                          C5  11 000A4          BRB     6$                                                      ; 1420
                             BE           53  F5 000A6 10$:     SOBGTR  DEPTH, 5$                                               ; 1413
                             56      14   A4  9A 000A9          MOVZBL  20(R4), DEPTH                                           ; 1440
                                          56  D6 000AD          INCL    DEPTH                                                   ;
                                        009D  31 000AF          BRW     24$                                                     ;
                                          6E  D4 000B2 11$:     CLRL    PTR                                                     ; 1449
                             52      04   AE  9E 000B4          MOVAB   LIST, P1                                                ; 1450
                             51      18   A4  D0 000B8          MOVL    24(R4), SEG                                             ; 1451
                             50      18   A4  9E 000BC 12$:     MOVAB   24(R4), R0                                              ; 1452
                             50           51  D1 000C0          CMPL    SEG, R0                                                 ;
                                          4B  13 000C3          BEQL    18$                                                     ;
     56           14    A1   08           00  ED 000C5          CMPZV   #0, #8, 20(SEG), DEPTH                                  ; 1455
                                          3E  12 000CB          BNEQ    17$                                                     ;
                                     1C   A1  B5 000CD          TSTW    28(SEG)                                                 ; 1457
                                          39  12 000D0          BNEQ    17$                                                     ;
                        0C   A2      2C   A1  D0 000D2          MOVL    44(SEG), 12(P1)                                         ; 1460
                        10   A2           51  D0 000D7          MOVL    SEG, 16(P1)                                             ; 1461
                                          62  7C 000DB          CLRQ    (P1)                                                    ; 1462
                                     08   A2  D4 000DD          CLRL    8(P1)                                                   ;
                             53           6E  9E 000E0          MOVAB   PTR, _H                                                 ;
                             50           63  D0 000E3 13$:     MOVL    (_H), R0                                                ;
                                          1D  13 000E6          BEQL    16$                                                     ;
                        0C   A0      0C   A2  D1 000E8          CMPL    12(P1), 12(R0)                                          ;
```

```
                             05  1E 000ED        BGEQU   14$                              ;
               53            50  D0 000EF        MOVL    R0, _H                           ;
                             EF  11 000F2        BRB     13$                              ;
                             06  1B 000F4  14$:  BLEQU   15$                              ;
               53      04    A0  9E 000F6        MOVAB   4(R0), _H                        ;
                             E7  11 000FA        BRB     13$                              ;
          08   A2      08    A0  D0 000FC  15$:  MOVL    8(R0), 8(P1)                     ;
               53      08    A0  9E 00101        MOVAB   8(R0), _H                        ;
               63            82  7E 00105  16$:  MOVAQ   (P1)+, (_H)                      ;
               52            0C  C0 00108        ADDL2   #12, P1                          ; 1463
               51            61  D0 0010B  17$:  MOVL    (SEG), SEG                       ; 1465
                             AC  11 0010E        BRB     12$                              ; 1452
               53            6E  9E 00110  18$:  MOVAB   PTR, _H                          ; 1467
                             63  D5 00113        TSTL    (_H)                             ;
                             04  12 00115        BNEQ    19$                              ;
                             52  D4 00117        CLRL    P1                               ;
                             24  11 00119        BRB     23$                              ;
                       00    B3  D5 0011B  19$:  TSTL    @0(_H)                           ;
                             05  13 0011E        BEQL    20$                              ;
               53            63  D0 00120        MOVL    (_H), _H                         ;
                             F6  11 00123        BRB     19$                              ;
               51            63  D0 00125  20$:  MOVL    (_H), R1                         ;
               50      08    A1  D0 00128        MOVL    8(R1), _Q                        ;
                             07  13 0012C        BEQL    21$                              ;
          08   A1      08    A0  D0 0012E        MOVL    8(_Q), 8(R1)                     ;
                             07  11 00133        BRB     22$                              ;
               50            51  D0 00135  21$:  MOVL    R1, _Q                           ;
               63      04    A0  D0 00138        MOVL    4(_Q), (_H)                      ;
               52            50  D0 0013C  22$:  MOVL    R0, P1                           ;
                             0E  13 0013F  23$:  BEQL    24$                              ;
                       10    A2  DD 00141        PUSHL   16(P1)                           ; 1468
                             54  DD 00144        PUSHL   R4                               ;
        FC07   CF            02  FB 00146        CALLS   #2, ELIMINATE_SEG                ;
               C2            50  E8 0014B        BLBS    STATUS, 18$                      ;
                                 04 0014E        RET                                      ;
               02            56  F5 0014F  24$:  SOBGTR  DEPTH, 25$                       ; 1440
                             03  11 00152        BRB     26$                              ;
                           FF5B  31 00154  25$:  BRW     11$                              ;
                             53  D4 00157  26$:  CLRL    ID                               ; 1473
               56            14  D0 00159        MOVL    #20, SIZE                        ; 1474
               50      18    A4  D0 0015C        MOVL    24(R4), SEG                      ; 1475
               51      18    A4  9E 00160  27$:  MOVAB   24(R4), R1                       ; 1476
               51            50  D1 00164        CMPL    SEG, R1                          ;
                             3A  13 00167        BEQL    29$                              ;
          0C   A0            53  B0 00169        MOVW    ID, 12(SEG)                      ; 1478
                             53  D6 0016D        INCL    ID                               ; 1479
               56            0C  C0 0016F        ADDL2   #12, SIZE                        ; 1480
               51      12    A0  3C 00172        MOVZWL  18(SEG), R1                      ; 1481
     52        51            01  78 00176        ASHL    #1, R1, R2                       ;
               52            05  C0 0017A        ADDL2   #5, R2                           ;
               52            08  C6 0017D        DIVL2   #8, R2                           ;
               56            52  C0 00180        ADDL2   R2, SIZE                         ;
               56      FE  A641  3E 00183        MOVAW   -2(SIZE)[R1], SIZE               ; 1482
          04   A0            60  D1 00188        CMPL    (SEG), 4(SEG)                    ; 1483
                             10  13 0018C        BEQL    28$                              ;
               51      16    A0  9A 0018E        MOVZBL  22(SEG), R1                      ; 1485
               52      15    A0  9A 00192        MOVZBL  21(SEG), R2                      ;
```

```
                         51           52  C2 00196        SUBL2   R2, R1                          ;
                         56      02 A641  3E 00199        MOVAW   2(SIZE)[R1], SIZE               ;
                         50           60  D0 0019E 28$:   MOVL    (SEG), SEG                      ; 1486
                                      BD  11 001A1        BRB     27$                             ; 1476
                                      55  DD 001A3 29$:   PUSHL   R5                              ; 1491
                                      56  DD 001A5        PUSHL   SIZE                            ;
                 0000G   CF           02  FB 001A7        CALLS   #2, DCX$GET_VM                  ;
                         01           50  E8 001AC        BLBS    STATUS, 30$                     ;
                                          04 001AF        RET                                     ;
                         55           65  D0 001B0 30$:   MOVL    (R5), R5                        ; 1492
                         65           56  D0 001B3        MOVL    SIZE, (R5)                      ;
                                 04   A5  B4 001B6        CLRW    4(R5)                           ; 1493
                   08    A5 5BF5A3A7  8F  D0 001B9        MOVL    #1542324871, 8(R5)              ; 1494
                                 0C   A5  D4 001C1        CLRL    12(R5)                          ; 1495
                   10    A5           53  B0 001C4        MOVW    ID, 16(R5)                      ; 1496
                                      53  D5 001C8        TSTL    ID                              ; 1497
                                      03  12 001CA        BNEQ    31$                             ;
                                    0090  31 001CC        BRW     36$                             ;
                   12    A5           14  B0 001CF 31$:   MOVW    #20, 18(R5)                     ; 1505
                         52      18   A4  D0 001D3        MOVL    24(R4), SEG                     ; 1506
                         53      12   A5  3C 001D7        MOVZWL  18(R5), DCXSBM                  ; 1507
                         53           55  C0 001DB        ADDL2   R5, DCXSBM                      ;
                         50      18   A4  9E 001DE 32$:   MOVAB   24(R4), R0                      ; 1508
                         50           52  D1 001E2        CMPL    SEG, R0                         ;
                                      6B  13 001E5        BEQL    35$                             ;
                         63           0C  B0 001E7        MOVW    #12, (DCXSBM)                   ; 1510
                   02    A3      15   A2  B0 001EA        MOVW    21(SEG), 2(DCXSBM)              ; 1511
                   04    A3    03FF   8F  AA 001EF        BICW2   #1023, 4(DCXSBM)                ; 1515
                   06    A3           63  B0 001F5        MOVW    (DCXSBM), 6(DCXSBM)             ; 1516
                         50      12   A2  3C 001F9        MOVZWL  18(SEG), R0                     ; 1517
             51          50           01  78 001FD        ASHL    #1, R0, R1                      ;
                         51           05  C0 00201        ADDL2   #5, R1                          ;
                         51           08  C6 00204        DIVL2   #8, R1                          ;
                         63           51  A0 00207        ADDW2   R1, (DCXSBM)                    ;
                   08    A3           63  B0 0020A        MOVW    (DCXSBM), 8(DCXSBM)             ; 1518
                         51           63  3C 0020E        MOVZWL  (DCXSBM), R1                    ; 1519
                         57      FE A140  3E 00211        MOVAW   -2(R1)[R0], R7                  ;
                         63           57  B0 00216        MOVW    R7, (DCXSBM)                    ;
                   04    A2           62  D1 00219        CMPL    (SEG), 4(SEG)                   ; 1520
                                      1C  13 0021D        BEQL    33$                             ;
                   0A    A3           63  B0 0021F        MOVW    (DCXSBM), 10(DCXSBM)            ; 1523
                         51           63  3C 00223        MOVZWL  (DCXSBM), R1                    ; 1524
                         50      16   A2  9A 00226        MOVZBL  22(SEG), R0                     ;
                         57      15   A2  9A 0022A        MOVZBL  21(SEG), R7                     ;
                         50           57  C2 0022E        SUBL2   R7, R0                          ;
                         57      02 A140  3E 00231        MOVAW   2(R1)[R0], R7                   ;
                         63           57  B0 00236        MOVW    R7, (DCXSBM)                    ;
                                      03  11 00239        BRB     34$                             ; 1520
                                 0A   A3  B4 0023B 33$:   CLRW    10(DCXSBM)                      ; 1527
                                      0C  BB 0023E 34$:   PUSHR   #^M<R2,R3>                      ; 1528
                                      54  DD 00240        PUSHL   R4                              ;
                  FC1D   CF           03  FB 00242        CALLS   #3, BUILD_MAP_SEG               ;
                         50           63  3C 00247        MOVZWL  (DCXSBM), R0                    ; 1529
                         53           50  C0 0024A        ADDL2   R0, DCXSBM                      ;
                         52           62  D0 0024D        MOVL    (SEG), SEG                      ; 1530
                                      8C  11 00250        BRB     32$                             ; 1508
             50          55           65  C1 00252 35$:   ADDL3   (R5), R5, R0                    ; 1532
```

```
                              50               53  D1 00256        CMPL    DCXSBM, R0                                  ;
                              04               13 00259            BEQL    36$                                         ;
                              50               0C  D0 0025B        MOVL    #12, R0                                     ; 1534
                                               04 0025E            RET                                                 ;
                        08    BC               55  D0 0025F 36$:   MOVL    R5, @MAP_ADDR                               ; 1536
                              03               6C  91 00263        CMPB    (AP), #3                                    ; 1537
                                               09  1F 00266        BLSSU   37$                                         ;
                                      0C       AC  D5 00268        TSTL    12(AP)                                      ;
                                               04  13 0026B        BEQL    37$                                         ;
                        0C    BC               56  D0 0026D        MOVL    SIZE, @MAP_SIZE                             ; 1539
                              50 00000000G     8F  D0 00271 37$:   MOVL    #DCX$_NORMAL, R0                            ; 1540
                                               04 00278            RET                                                 ; 1542

; Routine Size:  633 bytes,    Routine Base:  $CODE$ + 07D9
; 1154         1543  1 %SBTTL  'dcx$analyze_done -- release context and map'
; 1155         1544  1
; 1156         1545  1 GLOBAL ROUTINE dcx$analyze_done (context_addr) =
; 1157         1546  2 BEGIN
; 1158         1547  2 !++
; 1159         1548  2 !
; 1160         1549  2 ! Release context and map
; 1161         1550  2 !
; 1162         1551  2 ! Inputs:
; 1163         1552  2 !
; 1164         1553  2 !       context_addr.mz.r       Address of context longword
; 1165         1554  2 !
; 1166         1555  2 ! Outputs:
; 1167         1556  2 !
; 1168         1557  2 !       context_addr.mz.r       Context block accumulates data
; 1169         1558  2 !
; 1170         1559  2 ! Return value:
; 1171         1560  2 !
; 1172         1561  2 !       status.wlc.v
; 1173         1562  2 !
; 1174         1563  2 !               dcx$_normal     All is well
; 1175         1564  2 !--
; 1176         1565  2
; 1177         1566  2 BIND
; 1178         1567  2     ctx = .context_addr : REF BBLOCK,   ! address of context block
; 1179         1568  2     anl = ctx [ctx$l_specific] : BBLOCK;
; 1180         1569  2
; 1181         1570  2 LOCAL
; 1182         1571  2     anlseg : REF BBLOCK;
; 1183         1572  2
; 1184         1573  2 perform (dcx$ctx_check (.ctx, ctx$c_anlyz));
; 1185         1574  2 WHILE NOT remque (.anl [anl$l_flink], anlseg) DO
; 1186         1575  2     perform (dcx$free_vm (.anlseg [anlseg$l_size], .anlseg));
; 1187         1576  2 perform (dcx$free_vm (.ctx [ctx$l_size], .ctx));
; 1188         1577  2 ctx = 0;                            ! mark context as gone
; 1189         1578  2 RETURN dcx$_normal;
; 1190         1579  2
; 1191         1580  1 END;                                        ! Of dcx$analyze_done



                                              001C 00000          .ENTRY  DCX$ANALYZE_DONE, Save R2,R3,R4              ; 1545
                           54       04    AC   D0 00002          MOVL    CONTEXT_ADDR, R4                             ; 1567
              53           64             14   C1 00006          ADDL3   #20, (R4), R3                                ; 1568
                                          51   D4 0000A          CLRL    R1                                           ; 1573
                           50             64   D0 0000C          MOVL    (R4), R0                                     ;
                                       0000G   30 0000F          BSBW    DCX$CTX_CHECK                                ;
                                          10   11 00012          BRB     2$                                           ;
                           52       18    B3   0F 00014 1$:      REMQUE  @24(R3), ANLSEG                              ; 1574
                                          0E   1D 00018          BVS     3$                                           ;
                                          52   DD 0001A          PUSHL   ANLSEG                                       ; 1575
                                    08    A2   DD 0001C          PUSHL   8(ANLSEG)                                    ;
                    0000G  CF             02   FB 0001F          CALLS   #2, DCX$FREE_VM                              ;
                           ED             50   E8 00024 2$:      BLBS    STATUS, 1$                                   ;
                                               04 00027          RET                                                  ;
```

```
                                   64  DD 00028 3$:     PUSHL   (R4)                                   ; 1576
                          00       B4  DD 0002A         PUSHL   @0(R4)                                 ;
                  0000G  CF        02  FB 0002D         CALLS   #2, DCX$FREE_VM                        ;
                         09        50  E9 00032         BLBC    STATUS, 4$                             ;
                                   64  D4 00035         CLRL    (R4)                                   ; 1577
                         50 00000000G  8F  D0 00037     MOVL    #DCX$_NORMAL, R0                       ; 1578
                                       04 0003E 4$:     RET                                            ; 1580

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 0A52
; 1193           1581  1 END                                  . Of module analyze
; 1194           1582  0 ELUDOM



;                                  PSECT SUMMARY
;
;      Name                      Bytes                      Attributes
;
;  $CODE$                           2705  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                             Library Statistics
;
;                                             -------- Symbols --------        Pages        Processing
;      File                                   Total     Loaded   Percent       Mapped       Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1           9776          7         0         581          00:01.0




;                                  COMMAND QUALIFIERS

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

; Size:          2705 code + 0 data bytes
; Run Time:         00:58.4
; Elapsed Time:     02:55.8
; Lines/CPU Min:     1624
; Lexemes/CPU-Min: 27047
; Memory Used:  272 pages
; Compilation Complete
```

0074 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY