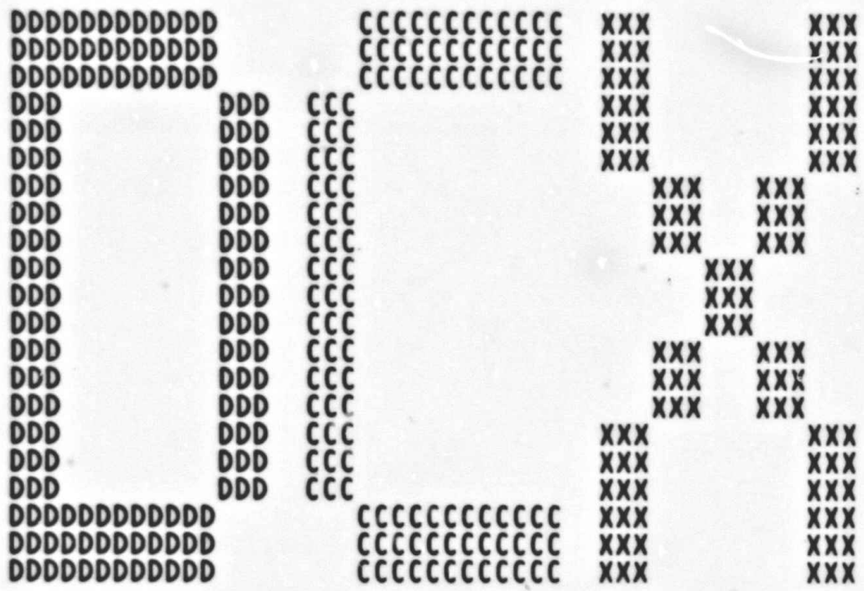
DDDDDDDDDDDD CCCCCCCCCCCC XXX XXX
DDDDDDDDDDDD CCCCCCCCCCCC XXX XXX
DDDDDDDDDDDD CCCCCCCCCCCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX
DDD DDD CCC XXX
DDD DDD CCC XXX
DDD DDD CCC XXX XXX
DDD DDD CCC XXX XXX
DDD DDD CCC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

FILEID**TRANSFER

```
TTTTTTTTTT  RRRRRRRR      AAAAAA    NN      NN   SSSSSSSS  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR
TTTTTTTTTT  RRRRRRRR      AAAAAA    NN      NN   SSSSSSSS  FFFFFFFFFF  EEEEEEEEEE  RRRRRRRR
    TT      RR      RR   AA    AA   NN      NN  SS         FF          EE          RR      RR
    TT      RR      RR   AA    AA   NN      NN  SS         FF          EE          RR      RR
    TT      RR      RR   AA    AA   NNNN    NN  SS         FF          EE          RR      RR
    TT      RR      RR   AA    AA   NNNN    NN  SS         FF          EE          RR      RR
    TT      RRRRRRRR     AA    AA   NN  NN  NN    SSSSSS   FFFFFFFF    EEEEEEEE    RRRRRRRR
    TT      RRRRRRRR     AA    AA   NN  NN  NN    SSSSSS   FFFFFFFF    EEEEEEEE    RRRRRRRR
    TT      RR  RR       AAAAAAAAAA   NN    NNNN        SS  FF          EE          RR  RR
    TT      RR  RR       AAAAAAAAAA   NN    NNNN        SS  FF          EE          RR  RR
    TT      RR    RR     AA    AA   NN      NN        SS  FF          EE          RR    RR      ....
    TT      RR    RR     AA    AA   NN      NN        SS  FF          EE          RR    RR      ....
    TT      RR      RR   AA    AA   NN      NN  SSSSSSSS  FF          EEEEEEEEEE  RR      RR    ....
    TT      RR      RR   AA    AA   NN      NN  SSSSSSSS  FF          EEEEEEEEEE  RR      RR    ....

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
```

```
0000     1          .TITLE  DCX_TRANSFER    transfer vectors for data compression / expansion
0000     2          .IDENT  'V04-000'
0000     3  ;
0000     4  ;*****************************************************************************
0000     5  ;*                                                                           *
0000     6  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     7  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     8  ;*  ALL RIGHTS RESERVED.                                                     *
0000     9  ;*                                                                           *
0000    10  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    11  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    12  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    13  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    14  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    15  ;*  TRANSFERRED.                                                             *
0000    16  ;*                                                                           *
0000    17  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    18  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    19  ;*  CORPORATION.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    22  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    23  ;*                                                                           *
0000    24  ;*                                                                           *
0000    25  ;*****************************************************************************
0000    26  ;
0000    27  ;++
0000    28  ;   FACILITY:
0000    29  ;
0000    30  ;       DCX -- Data Compression / Expansion Facility
0000    31  ;
0000    32  ;   ABSTRACT:
0000    33  ;
0000    34  ;       The Data Compression / Expansion procedures provide a general
0000    35  ;       method for reducing the storage requirement for a arbitrary data.
0000    36  ;
0000    37  ;   ENVIRONMENT:
0000    38  ;
0000    39  ;       Native mode, user mode
0000    40  ;
0000    41  ;   AUTHOR:
0000    42  ;
0000    43  ;       David Thiel     June 1982
0000    44  ;
0000    45  ;   MODIFIED BY:
0000    46  ;
0000    47  ;       V03-001 JWT0101         Jim Teague              04-Mar-1983
0000    48  ;               Change psect name to help transfer vector find its
0000    49  ;               way to the front of the image when linked.
0000    50  ;
0000    51  ;--
```

```
                     0000     53 ;
                     0000     54 ; Symbol definitions
                     0000     55 ;
                     0000     56
                     0000     57          $DCXDEF GLOBAL  ; Define facility symbols globally
                     0000     58
                     0000     59
                     0000     60
                 00000000     61          .PSECT  $$VECTOR_0_DCX, PIC, SHR, NOWRT, EXE, PAGE
                     0000     62
                     0000     63 ;
                     0000     64 ; Define macro to set up transfer vectors
                     0000     65 ;
                     0000     66
                     0000     67          .MACRO          TRANSFER ENTRY_POINT
                     0000     68          .SHOW           BINARY                  ;Display code produced
                     0000     69          .ALIGN          QUAD                    ;For style, speed, and space
                     0000     70          .TRANSFER       ENTRY_POINT
                     0000     71          .MASK           ENTRY_POINT             ;Copy entry point mask
                     0000     72          BRW             ENTRY_POINT+2           ;Go to routine code
                     0000     73          .NOSHOW         BINARY
                     0000     74          .ENDM           TRANSFER
                     0000     75
                     0000     76          .ALIGN          PAGE
                     0000     77 DCX_TRANSFER:
                     0000     78 ;
                     0000     79 ; Each of these macro invocated defines a universal symbol
                     0000     80 ; which is an entry point for this shareable library.
                     0000     81 ; These vectors must never** be moved in order to preserve
                     0000     82 ; compatibility with previously linked images.
                     0000     83 ;
                     0000     84          TRANSFER        DCX$ANALYZE_INIT        ; Initialize data analysis
               0000'  0000               .MASK           DCX$ANALYZE_INIT                ;Copy entry point mask
           FFFD'  31  0002               BRW             DCX$ANALYZE_INIT+2              ;Go to routine code
                     0005     85          TRANSFER        DCX$ANALYZE_DATA        ; Perform data analysis
                     0005               .ALIGN          QUAD                    ;For style, speed, and space
               0000'  0008               .MASK           DCX$ANALYZE_DATA                ;Copy entry point mask
           FFF5'  31  000A               BRW             DCX$ANALYZE_DATA+2              ;Go to routine code
                     000D     86          TRANSFER        DCX$MAKE_MAP            ; Compute compression function
                     000D               .ALIGN          QUAD                    ;For style, speed, and space
               0000'  0010               .MASK           DCX$MAKE_MAP            ;Copy entry point mask
           FFED'  31  0012               BRW             DCX$MAKE_MAP+2          ;Go to routine code
                     0015     87          TRANSFER        DCX$ANALYZE_DONE        ; Release data analysis context
                     0015               .ALIGN          QUAD                    ;For style, speed, and space
               0000'  0018               .MASK           DCX$ANALYZE_DONE                ;Copy entry point mask
           FFE5'  31  001A               BRW             DCX$ANALYZE_DONE+2              ;Go to routine code
                     001D     88          TRANSFER        DCX$COMPRESS_INIT       ; Initialize data compression
                     001D               .ALIGN          QUAD                    ;For style, speed, and space
               0000'  0020               .MASK           DCX$COMPRESS_INIT               ;Copy entry point mask
           FFDD'  31  0022               BRW             DCX$COMPRESS_INIT+2             ;Go to routine code
                     0025     89          TRANSFER        DCX$COMPRESS_DATA       ; Perform data compression
                     0025               .ALIGN          QUAD                    ;For style, speed, and space
               0000'  0028               .MASK           DCX$COMPRESS_DATA               ;Copy entry point mask
           FFD5'  31  002A               BRW             DCX$COMPRESS_DATA+2             ;Go to routine code
                     002D     90          TRANSFER        DCX$COMPRESS_DONE       ; Release data compression context
                     002D               .ALIGN          QUAD                    ;For style, speed, and space
                0000'  0030              .MASK           DCX$COMPRESS_DONE               ;Copy entry point mask
```

```
                FFCD'  31  0032                BRW         DCX$COMPRESS_DONE+2     ;Go to routine code
                       0035    91              TRANSFER    DCX$EXPAND_INIT         ; Initialize data expansion
                       0035                    .ALIGN      QUAD                    ;For style, speed, and space
                0000'  0038                    .MASK       DCX$EXPAND_INIT         ;Copy entry point mask
                FFC5'  31  003A                BRW         DCX$EXPAND_INIT+2       ;Go to routine code
                       003D    92              TRANSFER    DCX$EXPAND_DATA         ; Perform data expansion
                       003D                    .ALIGN      QUAD                    ;For style, speed, and space
                0000'  0040                    .MASK       DCX$EXPAND_DATA         ;Copy entry point mask
                FFBD'  31  0042                BRW         DCX$EXPAND_DATA+2       ;Go to routine code
                       0045    93              TRANSFER    DCX$EXPAND_DONE         ; Release data expansion context
                       0045                    .ALIGN      QUAD                    ;For style, speed, and space
                0000'  0048                    .MASK       DCX$EXPAND_DONE         ;Copy entry point mask
                FFB5'  31  004A                BRW         DCX$EXPAND_DONE+2       ;Go to routine code
                       004D    94
                       004D    95              .ALIGN      PAGE
                       0200    96
                       0200    97              .END
```

```
DCX$ANALYZE_DATA                ********   X   02
DCX$ANALYZE_DONE                ********   X   02
DCX$ANALYZE_INIT                ********   X   02
DCX$COMPRESS_DATA               ********   X   02
DCX$COMPRESS_DONE               ********   X   02
DCX$COMPRESS_INIT               ********   X   02
DCX$C_BOUNDED                 = 00000101   G
DCX$C_EST_BYTES               = 00000202   G
DCX$C_EST_RECORDS             = 00000201   G
DCX$C_LIST                    = 00000001   G
DCX$C_ONE_PASS                = 00000102   G
DCX$EXPAND_DATA                 ********   X   02
DCX$EXPAND_DONE                 ********   X   02
DCX$EXPAND_INIT                 ********   X   02
DCX$MAKE_MAP                    ********   X   02
DCX_TRANSFER                    00000000 R     02
```

Psect synopsis

```
PSECT name                      Allocation          PSECT No.  Attributes
----------                      ----------          ---------  ----------
. ABS .                         00000000  (    0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD  NOWRT NOVEC BYTE
$ABS$                           00000000  (    0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD    WRT NOVEC BYTE
$$VECTOR_0_DCX                  00000200  (  512.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC PAGE
```

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	48	00:00:00.13	00:00:00.88
Command processing	151	00:00:00.59	00:00:03.63
Pass 1	94	00:00:00.84	00:00:02.25
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	31	00:00:00.32	00:00:01.28
Symbol table output	2	00:00:00.02	00:00:00.02
Psect synopsis output	2	00:00:00.02	00:00:00.63
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	328	00:00:01.93	00:00:08.69

```
The working set limit was 1200 pages.
4448 bytes (9 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 16 non-local and 0 local symbols.
97 source lines were read in Pass 1, producing 14 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.
```

```
                              +----------------------------+
                              ! Macro library statistics !
                              +----------------------------+

Macro library name                                Macros defined
------------------                                --------------
_$255$DUA28:[DCX.OBJ]DCX.MLB;1                            1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                         3
TOTALS (all libraries)                                    4

69 GETS were required to define 4 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:TRANSFER/OBJ=OBJ$:TRANSFER MSRC$:TRANSFER/UPDATE=(ENH$:TRANSFER)+LIB$:DCX/LIB
```

PREFIX
REQ

STACLINT
LIS

STATUS
LIS

ANALYZE
LIS

DCXMSG
LIS

STASTUB
LIS

DCXDEF
MDL

EXPAND
LIS

SYSOUTPUT
LIS

DCXPRVDEF
MDL

STATEMENT
LIS

DCX

COMPRESS
LIS

TRANSFER
LIS

SYMBOL
LIS

DCXSHR
MAP

SUBS
LIS

S0DEBUG
MAP

DBGLIB
REQ

DEBUG

DEBUG
MAP

DBGSSISHR
MAP

DBGEXT
REQ

DBGMSG
MDL

DBGGEN
REQ