```
RRRRRRRRRRR      UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNN         NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRR          RRR UUU          UUU  NNNNNN      NNN  000          000 FFF               FFF
RRRRRRRRRRR      UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN   NNN   NNN  000          000 FFFFFFFFFFF       FFFFFFFFFFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR   RRR        UUU          UUU  NNN       NNNNNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR      RRR     UUU          UUU  NNN          NNN 000          000 FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
RRR         RRR  UUUUUUUUUUUUUUU  NNN          NNN    000000000     FFF               FFF
```

```
_$2

Sym
---
NDX
NDX
NUM
NUM
OPE
OUT
PAC
PAC
PAC
PAC
PAC
PAC
PAC
PAD
PAG
PAG
PAG
PAG
PAG
PAG
PAG
PER
PUT
RCO
RIN

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
STR
```

```
  CCCCCCCC  NN      NN  VV      VV  DDDDDDDD      AAAAAA    TTTTTTTTTT
  CCCCCCCC  NN      NN  VV      VV  DDDDDDDD      AAAAAA    TTTTTTTTTT
CC          NN      NN  VV      VV  DD      DD   AA    AA      TT
CC          NN      NN  VV      VV  DD      DD   AA    AA      TT
CC          NNNN    NN  VV      VV  DD      DD   AA    AA      TT
CC          NNNN    NN  VV      VV  DD      DD   AA    AA      TT
CC          NN  NN  NN  VV      VV  DD      DD   AA    AA      TT
CC          NN  NN  NN  VV      VV  DD      DD   AA    AA      TT
CC          NN    NNNN  VV      VV  DD      DD   AAAAAAAAAA      TT
CC          NN    NNNN  VV      VV  DD      DD   AAAAAAAAAA      TT
CC          NN      NN    VV  VV    DD      DD   AA    AA      TT
CC          NN      NN    VV  VV    DD      DD   AA    AA      TT
  CCCCCCCC  NN      NN      VV      DDDDDDDD     AA    AA      TT
  CCCCCCCC  NN      NN      VV      DDDDDDDD     AA    AA      TT

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

```
;  1          0001  0 %TITLE 'CNVDAT - Converts binary date and time into ascii'
;  2          0002  0 MODULE CNVDAT ( IDENT = 'V04-000'
;  3        P 0003  0                 %BLISS32 [ , ADDRESSING_MODE ( EXTERNAL =    LONG_RELATIVE,
;  4          0004  0                                                NONEXTERNAL = LONG_RELATIVE)]
;  5          0005  0                 ) =
;  6          0006  1 BEGIN
;  7          0007  1 !
;  8          0008  1 !*****************************************************************************
;  9          0009  1 !*                                                                           *
; 10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 12          0012  1 !*  ALL RIGHTS RESERVED.                                                     *
; 13          0013  1 !*                                                                           *
; 14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
; 17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 19          0019  1 !*  TRANSFERRED.                                                             *
; 20          0020  1 !*                                                                           *
; 21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 23          0023  1 !*  CORPORATION.                                                             *
; 24          0024  1 !*                                                                           *
; 25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 27          0027  1 !*                                                                           *
; 28          0028  1 !*                                                                           *
; 29          0029  1 !*****************************************************************************
; 30          0030  1 !
; 31          0031  1 !++
; 32          0032  1 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
; 33          0033  1 !
; 34          0034  1 ! ABSTRACT: Convert binary and date and time into something
; 35          0035  1 !                 more legible (i.e., ASCII).
; 36          0036  1 !
; 37          0037  1 !
; 38          0038  1 ! ENVIRONMENT: Transportable
; 39          0039  1 !
; 40          0040  1 ! AUTHOR: R.W.Friday     CREATION DATE: February, 1979
; 41          0041  1 !
```

```
;    43          0042  1 %SBTTL 'Revision History'
;    44          0043  1 !
;    45          0044  1 ! MODIFIED BY:
;    46          0045  1 !
;    47          0046  1 !       009     REM00009        Ray Marshall    30-Mar-1984
;    48          0047  1 !               Added conditionals for foreign language support.  Provided
;    49          0048  1 !                 German translations for the names of the months.
;    50          0049  1 !
;    51          0050  1 !       008     KFA00008        Ken Alden       16-Sep-1983
;    52          0051  1 !               Did the same as 007 but did it for RUNOFF.
;    53          0052  1 !
;    54          0053  1 !       007     KFA00007        Ken Alden       8-Jul-1983
;    55          0054  1 !               Removed hack that always assumed that the date was
;    56          0055  1 !                 two characters long.
;    57          0056  1 !
;    58          0057  1 !       006     KFA00006        Ken Alden       8-Jul-1983
;    59          0058  1 !               Removed the leading '0' from the date (if any).
;    60          0059  1 !
;    61          0060  1 !       005     KFA00005        Ken Alden       13-April-1983
;    62          0061  1 !               For DSRPLUS: Format of the date will now read like
;    63          0062  1 !               the date of this rev. history.(without the hyphens)
;    64          0063  1 !
;    65          0064  1 !       004     KFA00004        Ken Alden       07-Mar-1983
;    66          0065  1 !               Global edit of all modules. Updated module names, idents,
;    67          0066  1 !               copyright dates. Changed require files to BLISS library.
;    68          0067  1 !
;    69          0068  1 !--
```

```
;    71          0069  1 %SBTTL 'Module Level Declarations'
;    72          0070  1 !
;    73          0071  1 ! TABLE OF CONTENTS:
;    74          0072  1 !
;    75          0073  1
;    76          0074  1 !
;    77          0075  1 ! INCLUDE FILES:
;    78          0076  1 !
;    79          0077  1
;    80          0078  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    81          0079  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    82          0210  1
;    83        U 0211  1 %IF DSRPLUS %THEN
;    84        U 0212  1 LIBRARY 'REQ:DPLLIB';                 ! DSRPLUS BLISS Library
;    85          0213  1 %ELSE
;    86          0214  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    87          0215  1 %FI
;    88          0216  1
;    89          0217  1 !
;    90          0218  1 ! MACROS:
;    91          0219  1 !
;    92          0220  1
;    93          0221  1 !
;    94          0222  1 ! EQUATED SYMBOLS:
;    95          0223  1 !
;    96          0224  1
;    97          0225  1 !
;    98          0226  1 ! OWN STORAGE:
;    99          0227  1 !
;   100          0228  1
;   101          0229  1 !
;   102          0230  1 ! EXTERNAL REFERENCES:
;   103          0231  1 !
;   104          0232  1 EXTERNAL
;   105          0233  1     GCA : GCA_DEFINITION;
;   106          0234  1
;   107          0235  1 EXTERNAL ROUTINE
;   108          0236  1     CONVBB;
```

```
;  110     0237  1 %sbttl 'CNVDAT -- convert date to ASCII string'
;  111     0238  1 GLOBAL ROUTINE CNVDAT (PIECES, RESULT, RESULT_LENGTH) : NOVALUE =       !
;  112     0239  1
;  113     0240  1 !++
;  114     0241  1 ! FUNCTIONAL DESCRIPTION:
;  115     0242  1 !
;  116     0243  1 !       Converts a binary date into something legible.
;  117     0244  1 !
;  118     0245  1 ! FORMAL PARAMETERS:
;  119     0246  1 !
;  120     0247  1 !       PIECES are the separated month, day, and year.
;  121     0248  1 !       It's really the complete binary date and time, but only the
;  122     0249  1 !       date portion is used.
;  123     0250  1 !       RESULT is a CH$PTR to where the results are to go.
;  124     0251  1 !       RESULT_LENGTH is the number of characters in the result.
;  125     0252  1 !
;  126     0253  1 ! IMPLICIT INPUTS:      None
;  127     0254  1 !
;  128     0255  1 ! IMPLICIT OUTPUTS:     None
;  129     0256  1 !
;  130     0257  1 ! ROUTINE VALUE:
;  131     0258  1 ! COMPLETION CODES:     None
;  132     0259  1 !
;  133     0260  1 ! SIDE EFFECTS:
;  134     0261  1 !
;  135     0262  1 !       Advances the pointer, RESULT.
;  136     0263  1 !
;  137     0264  1 !--
;  138     0265  1
;  139     0266  2     BEGIN
;  140     0267  2
;  141     0268  2     LOCAL
;  142     0269  2         CHARS : VECTOR [20],
;  143     0270  2         COUNT;
;  144     0271  2
;  145     0272  2     MAP
;  146     0273  2         PIECES : REF VECTOR;
;  147     0274  2
;  148     0275  2     BIND
;  149     0276  2         YEAR = PIECES[0],
;  150     0277  2         MONTH = PIECES[1],
;  151     0278  2         DAY = PIECES[2];
;  152     0279  2
;  153     0280  2     !Convert the day to ASCII.
;  154     0281  2     CONVBB(.DAY,CHARS,COUNT, 10);
;  155     0282  2
;  156     0283  2 !!    !Insist that the day be two digits.
;  157     0284  2 !!    IF!
;  158     0285  2 !!       .COUNT NEQ 2
;  159     0286  2 !!    THEN
;  160     0287  2 !!       CH$WCHAR_A(%C'0', .RESULT);
;  161     0288  2
;  162     0289  2     !Now copy over what was actually converted.
;  163     0290  2     DECR I FROM .COUNT TO 1 DO
;  164     0291  2         CH$WCHAR_A(.CHARS[.I-1], .RESULT);
;  165     0292  2
;  166     0293  2     !Now, add an appropriate separator.
```

```
;  167          0294  2      CH$WCHAR_A(%C' ', .RESULT);
;  168          0295  2
;  169          0296  2      !Add the name of the month.
;  170          0297  2      BEGIN
;  171          0298  2      BIND
;  172          0299  2          MONTH_NAME= UPLIT (
;  173        C 0300  2  %(                The following block of code is commented out because the
;  174        C 0301  2                    one that follows defines the month names in their fully
;  175        C 0302  2                    spelled out forms.
;  176        C 0303  2  %IF german %THEN
;  177        C 0304  2                                CH$PTR(UPLIT('Jan')),
;  178        C 0305  2                                CH$PTR(UPLIT('Feb')),
;  179        C 0306  2                                CH$PTR(UPLIT('Mar')),
;  180        C 0307  2                                CH$PTR(UPLIT('Apr')),
;  181        C 0308  2                                CH$PTR(UPLIT('Mai')),
;  182        C 0309  2                                CH$PTR(UPLIT('Jun')),
;  183        C 0310  2                                CH$PTR(UPLIT('Jul')),
;  184        C 0311  2                                CH$PTR(UPLIT('Aug')),
;  185        C 0312  2                                CH$PTR(UPLIT('Sep')),
;  186        C 0313  2                                CH$PTR(UPLIT('Okt')),
;  187        C 0314  2                                CH$PTR(UPLIT('Nov')),
;  188        C 0315  2                                CH$PTR(UPLIT('Dez'))
;  189        C 0316  2  %ELSE
;  190        C 0317  2  %IF french %THEN
;  191        C 0318  2                                CH$PTR(UPLIT('Jan')),
;  192        C 0319  2                                CH$PTR(UPLIT('Feb')),
;  193        C 0320  2                                CH$PTR(UPLIT('Mar')),
;  194        C 0321  2                                CH$PTR(UPLIT('Apr')),
;  195        C 0322  2                                CH$PTR(UPLIT('May')),
;  196        C 0323  2                                CH$PTR(UPLIT('Jun')),
;  197        C 0324  2                                CH$PTR(UPLIT('Jul')),
;  198        C 0325  2                                CH$PTR(UPLIT('Aug')),
;  199        C 0326  2                                CH$PTR(UPLIT('Sep')),
;  200        C 0327  2                                CH$PTR(UPLIT('Oct')),
;  201        C 0328  2                                CH$PTR(UPLIT('Nov')),
;  202        C 0329  2                                CH$PTR(UPLIT('Dec'))
;  203        C 0330  2  %ELSE
;  204        C 0331  2  %IF italian %THEN
;  205        C 0332  2                                CH$PTR(UPLIT('Jan')),
;  206        C 0333  2                                CH$PTR(UPLIT('Feb')),
;  207        C 0334  2                                CH$PTR(UPLIT('Mar')),
;  208        C 0335  2                                CH$PTR(UPLIT('Apr')),
;  209        C 0336  2                                CH$PTR(UPLIT('May')),
;  210        C 0337  2                                CH$PTR(UPLIT('Jun')),
;  211        C 0338  2                                CH$PTR(UPLIT('Jul')),
;  212        C 0339  2                                CH$PTR(UPLIT('Aug')),
;  213        C 0340  2                                CH$PTR(UPLIT('Sep')),
;  214        C 0341  2                                CH$PTR(UPLIT('Oct')),
;  215        C 0342  2                                CH$PTR(UPLIT('Nov')),
;  216        C 0343  2                                CH$PTR(UPLIT('Dec'))
;  217        C 0344  2  %ELSE
;  218        C 0345  2                                CH$PTR(UPLIT('Jan')),
;  219        C 0346  2                                CH$PTR(UPLIT('Feb')),
;  220        C 0347  2                                CH$PTR(UPLIT('Mar')),
;  221        C 0348  2                                CH$PTR(UPLIT('Apr')),
;  222        C 0349  2                                CH$PTR(UPLIT('May')),
;  223        C 0350  2                                CH$PTR(UPLIT('Jun')),
```

```
;  224        C 0351  3                                CH$PTR(UPLIT('Jul')),
;  225        C 0352  3                                CH$PTR(UPLIT('Aug')),
;  226        C 0353  3                                CH$PTR(UPLIT('Sep')),
;  227        C 0354  3                                CH$PTR(UPLIT('Oct')),
;  228        C 0355  3                                CH$PTR(UPLIT('Nov')),
;  229        C 0356  3                                CH$PTR(UPLIT('Dec'))
;  230        C 0357  3    %FI %FI %FI
;  231          0358  3    )%
;  232        U 0359  3    %IF german %THEN
;  233        U 0360  3                                CH$PTR(UPLIT('Jaenner')),
;  234        U 0361  3                                CH$PTR(UPLIT('Februar')),
;  235        U 0362  3                                CH$PTR(UPLIT('Maerz')),
;  236        U 0363  3                                CH$PTR(UPLIT('April')),
;  237        U 0364  3                                CH$PTR(UPLIT('Mai')),
;  238        U 0365  3                                CH$PTR(UPLIT('Juni')),
;  239        U 0366  3                                CH$PTR(UPLIT('Juli')),
;  240        U 0367  3                                CH$PTR(UPLIT('August')),
;  241        U 0368  3                                CH$PTR(UPLIT('September')),
;  242        U 0369  3                                CH$PTR(UPLIT('Oktober')),
;  243        U 0370  3                                CH$PTR(UPLIT('November')),
;  244        U 0371  3                                CH$PTR(UPLIT('Dezember'))
;  245          0372  3    %ELSE
;  246        U 0373  3    %IF french %THEN
;  247        U 0374  3                                CH$PTR(UPLIT('January')),
;  248        U 0375  3                                CH$PTR(UPLIT('February')),
;  249        U 0376  3                                CH$PTR(UPLIT('March')),
;  250        U 0377  3                                CH$PTR(UPLIT('April')),
;  251        U 0378  3                                CH$PTR(UPLIT('May')),
;  252        U 0379  3                                CH$PTR(UPLIT('June')),
;  253        U 0380  3                                CH$PTR(UPLIT('July')),
;  254        U 0381  3                                CH$PTR(UPLIT('August')),
;  255        U 0382  3                                CH$PTR(UPLIT('September')),
;  256        U 0383  3                                CH$PTR(UPLIT('October')),
;  257        U 0384  3                                CH$PTR(UPLIT('November')),
;  258        U 0385  3                                CH$PTR(UPLIT('December'),
;  259          0386  3    %ELSE
;  260        U 0387  3    %IF italian %THEN
;  261        U 0388  3                                CH$PTR(UPLIT('January')),
;  262        U 0389  3                                CH$PTR(UPLIT('February')),
;  263        U 0390  3                                CH$PTR(UPLIT('March')),
;  264        U 0391  3                                CH$PTR(UPLIT('April')),
;  265        U 0392  3                                CH$PTR(UPLIT('May')),
;  266        U 0393  3                                CH$PTR(UPLIT('June')),
;  267        U 0394  3                                CH$PTR(UPLIT('July')),
;  268        U 0395  3                                CH$PTR(UPLIT('August')),
;  269        U 0396  3                                CH$PTR(UPLIT('September')),
;  270        U 0397  3                                CH$PTR(UPLIT('October')),
;  271        U 0398  3                                CH$PTR(UPLIT('November')),
;  272        U 0399  3                                CH$PTR(UPLIT('December'))
;  273          0400  3    %ELSE
;  274          0401  3                                CH$PTR(UPLIT('January')),
;  275          0402  3                                CH$PTR(UPLIT('February')),
;  276          0403  3                                CH$PTR(UPLIT('March')),
;  277          0404  3                                CH$PTR(UPLIT('April')),
;  278          0405  3                                CH$PTR(UPLIT('May')),
;  279          0406  3                                CH$PTR(UPLIT('June')),
;  280          0407  3                                CH$PTR(UPLIT('July')),
```

```
;  281          0408  %                                   CH$PTR(UPLIT('August')),
;  282          0409  %                                   CH$PTR(UPLIT('September')),
;  283          0410  %                                   CH$PTR(UPLIT('October')),
;  284          0411  %                                   CH$PTR(UPLIT('November')),
;  285          0412  %                                   CH$PTR(UPLIT('December'))
;  286          0413  %FI %FI %FI
;  287          0414  %                                   ) : VECTOR;
;  288          0415                  OWN
;  289          0416                      MONTHL : VECTOR [12]
;  290        U 0417  %IF german %THEN
;  291        U 0418  %                         INITIAL (7, 7, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8);
;  292          0419  %ELSE
;  293        U 0420  %IF french %THEN
;  294        U 0421  %                         INITIAL (7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8);
;  295          0422  %ELSE
;  296        U 0423  %IF italian %THEN
;  297        U 0424  %                         INITIAL (7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8);
;  298          0425  %ELSE
;  299          0426  %                         INITIAL (7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8);
;  300          0427  %FI %FI %FI
;  301          0428
;  302          0429
;  303          0430                  LOCAL
;  304          0431                      len,
;  305          0432                      ptr;
;  306          0433
;  307          0434                  len = .monthl[.month-1];
;  308          0435                  PTR = .MONTH_NAME[.MONTH-1];
;  309          0436                  INCR I FROM 1 TO .len DO CH$WCHAR_A(CH$RCHAR_A(PTR), .RESULT);
;  310          0437
;  311          0438  !!                INCR I FROM 1 TO 3 DO CH$WCHAR_A(CH$RCHAR_A(PTR), .RESULT);
;  312          0439
;  313          0440                  !Add a separator.
;  314          0441                  CH$WCHAR_A(%C' ', .RESULT);
;  315          0442
;  316          0443                  !Result length prior to year string:
;  317          0444                  .RESULT_LENGTH = .count + 1 + .len + 1;
;  318          0445              2   END;
;  319          0446              2
;  320          0447              2   !Put in the year.
;  321          0448              2   CONVBB(.YEAR,CHARS,COUNT, 10);
;  322          0449              2
;  323          0450              2
;  324          0451              2   IF .count EQL 1 THEN
;  325          0452              3       BEGIN
;  326          0453              3       CH$WCHAR_A(%C'0', .RESULT);
;  327          0454              3       CH$WCHAR_A(.CHARS[0], .RESULT);
;  328          0455              3       .RESULT_LENGTH = ..RESULT_LENGTH + 2
;  329          0456              2       END;
;  330          0457              2
;  331          0458              2   IF .count GTR 1 THEN
;  332          0459              2
;  333          0460              2 !%IF DSRPLUS %THEN
;  334          0461              3               BEGIN
;  335          0462              3               DECR i FROM .count TO 1 DO
;  336          0463              3                   CH$WCHAR_A(.CHARS[.I-1], .RESULT);
;  337          0464              3               .RESULT_LENGTH = ..RESULT_LENGTH + .count;
```

```
;  338          0465  2                END;
;  339          0466  2            %ELSE
;  340          0467  2            BEGIN
;  341          0468  2            CH$WCHAR_A(.CHARS[1], .RESULT);
;  342          0469  2            CH$WCHAR_A(.CHARS[0], .RESULT);
;  343          0470  2            .RESULT_LENGTH = ..RESULT_LENGTH + 2
;  344          0471  2            END
;  345          0472  2        %FI;
;  346          0473  2
;  347          0474  2        RETURN;
;  348          0475  2
   349          0476  2
;  350          0477  2
;  351          0478  1        END;                        !End of CNVDAT


                                                                .TITLE  CNVDAT CNVDAT - Converts binary date and time i
;                                                                       nto asc
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                        00  79  72  61  75  6E  61  4A  00000 P.AAB:    .ASCII  \January\<0>                                    ;
                        79  72  61  75  72  62  65  46  00008 P.AAC:    .ASCII  \February\                                      ;
                        00  00  00  68  63  72  61  4D  00010 P.AAD:    .ASCII  \March\<0><0><0>                                ;
                        00  00  00  6C  69  72  70  41  00018 P.AAE:    .ASCII  \April\<0><0><0>                                ;
                                        00  79  61  4D  00020 P.AAF:    .ASCII  \May\<0>                                        ;
                                        65  6E  75  4A  00024 P.AAG:    .ASCII  \June\                                          ;
                                        79  6C  75  4A  00028 P.AAH:    .ASCII  \July\                                          ;
                                00  00  74  73  75  67  75  41  0002C P.AAI:    .ASCII  \August\<0><0>                          ;
                    00  00  00  72  65  62  6D  65  74  70  65  53  00034 P.AAJ:    .ASCII  \September\<0><0><0>            ;
                                00  72  65  62  6F  74  63  4F  00040 P.AAK:    .ASCII  \October\<0>                            ;
                                72  65  62  6D  65  76  6F  4E  00048 P.AAL:    .ASCII  \November\                              ;
                                72  65  62  6D  65  63  65  44  00050 P.AAM:    .ASCII  \December\                              ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00058 P.AAA:    .ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, -           ;
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00070                   P.AAG, P.AAH, P.AAI, P.AAJ, P.AAK, P.AAL, -     ;
                                                                                    P.AAM                                       ;

                                                                .PSECT  $OWN$,NOEXE,2

00000004  00000003  00000005  00000005  00000008  00000007  00000 MONTHL:  .LONG   7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8          ;
00000008  00000008  00000007  00000009  00000006  00000004  00018

                                                        MONTH_NAME=         P.AAA
                                                                .EXTRN  GCA, CONVBB

                                                                .PSECT  $CODE$,NOWRT,2

                                                    001C 00000          .ENTRY  CNVDAT, Save R2,R3,R4                           ; 0238
                                        54 00000000G    EF  9E 00002    MOVAB   CONVBB, R4                                      ;
                                        5E      AC      AE  9E 00009    MOVAB   -84(SP), SP                                     ;
                                52      04      AC      04  C1 0000D    ADDL3   #4, PIECES, R2                                  ; 0277
                                50      04      AC      08  C1 00012    ADDL3   #8, PIECES, R0                                  ; 0278
                                                        0A  DD 00017    PUSHL   #10                                             ; 0281
                                                    04  AE  9F 00019    PUSHAB  COUNT                                           ;
                                                    0C  AE  9F 0001C    PUSHAB  CHARS                                           ;
                                                        60  DD 0001F    PUSHL   (R0)                                            ;
```

```
                                64              04  FB 00021          CALLS   #4, CONVBB
                            50  6E              01  C1 00024          ADDL3   #1, COUNT, I                                    ;  0291
                                                0B  11 00028          BRB     2$
                                51          08  AC  D0 0002A 1$:      MOVL    RESULT, R1
                            00  B1              6E40 F6 0002E          CVTLB   CHARS-4[I], @0(R1)
                                                61  D6 00033          INCL    (R1)
                                F2              50  F5 00035 2$:      SOBGTR  I, 1$
                                50          08  AC  D0 00038          MOVL    RESULT, R0                                     ;  0294
                            00  B0              20  90 0003C          MOVB    #32, @0(R0)
                                                60  D6 00040          INCL    (R0)
                                50              62  D0 00042          MOVL    (R2), R0                                       ;  0434
                                53 00000000'EF40  D0 00045          MOVL    MONTHL-4[R0], LEN                              ;  0435
                                52 00000000'EF40  D0 0004D          MOVL    MONTH_NAME-4[R0], PTR                          ;  0436
                                                51  D4 00055          CLRL    I
                                                0A  11 00057          BRB     4$
                                50          08  AC  D0 00059 3$:      MOVL    RESULT, R0
                            00  B0              82  90 0005D          MOVB    (PTR)+, @0(R0)
                                                60  D6 00061          INCL    (R0)
                            F2                  51  F3 00063 4$:      AOBLEQ  LEN, I, 3$
                                51          08  AC  D0 00067          MOVL    RESULT, R1                                     ;  0441
                            00  B1              20  90 0006B          MOVB    #32, @0(R1)
                                                61  D6 0006F          INCL    (R1)
                            50                  6E  C1 00071          ADDL3   LEN, COUNT, R0                                 ;  0444
                            0C  BC          02  A0  9E 00075          MOVAB   2(R0), @RESULT_LENGTH
                                                0A  DD 0007A          PUSHL   #10                                            ;  0448
                                            04  AE  9F 0007C          PUSHAB  COUNT
                                            0C  AE  9F 0007F          PUSHAB  CHARS
                                            04  BC  DD 00082          PUSHL   @PIECES
                                    64          04  FB 00085          CALLS   #4, CONVBB
                                    01          6E  D1 00088          CMPL    COUNT, #1                                      ;  0451
                                                19  12 0008B          BNEQ    5$
                                50          08  AC  D0 0008D          MOVL    RESULT, R0                                     ;  0453
                            00  B0              30  90 00091          MOVB    #48, @0(R0)
                                                60  D6 00095          INCL    (R0)
                                50          08  AC  D0 00097          MOVL    RESULT, R0                                     ;  0454
                            00  B0          04  AE  90 0009B          MOVB    CHARS, @0(R0)
                                                60  D6 000A0          INCL    (R0)
                            0C  BC              02  C0 000A2          ADDL2   #2, @RESULT_LENGTH                             ;  0455
                                01              6E  D1 000A6 5$:      CMPL    COUNT, #1                                      ;  0458
                                                18  15 000A9          BLEQ    8$
                        50                      6E  C1 000AB          ADDL3   #1, COUNT, I                                   ;  0463
                                                0B  11 000AF          BRB     7$
                                51          08  AC  D0 000B1 6$:      MOVL    RESULT, R1
                            00  B1              6E40 F6 000B5          CVTLB   CHARS-4[I], @0(R1)
                                                61  D6 000BA          INCL    (R1)
                                F2              50  F5 000BC 7$:      SOBGTR  I, 6$
                            0C  BC              6E  C0 000BF          ADDL2   COUNT, @RESULT_LENGTH                          ;  0464
                                                04  000C3 8$:        RET                                                    ;  0478

; Routine Size:  196 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  353     0479  1 %sbttl 'CNVTIM -- Convert time to ASCII string'
;  354     0480  1 GLOBAL ROUTINE CNVTIM (PIECES, RESULT, RESULT_LENGTH) : NOVALUE =       !
;  355     0481  1
;  356     0482  1 !++
;  357     0483  1 ! FUNCTIONAL DESCRIPTION:
;  358     0484  1 !
;  359     0485  1 !       Converts a binary time into something legible.
;  360     0486  1 !
;  361     0487  1 ! FORMAL PARAMETERS:
;  362     0488  1 !
;  363     0489  1 !       PIECES is the complete binary date and time.  This routine
;  364     0490  1 !       uses only the time information.
;  365     0491  1 !       RESULT is a CH$PTR to where the results are to go.
;  366     0492  1 !       RESULT_LENGTH is the number of characters in the result.
;  367     0493  1 !
;  368     0494  1 ! IMPLICIT INPUTS:      None
;  369     0495  1 !
;  370     0496  1 ! IMPLICIT OUTPUTS:     None
;  371     0497  1 !
;  372     0498  1 ! ROUTINE VALUE:
;  373     0499  1 ! COMPLETION CODES:     None
;  374     0500  1 !
;  375     0501  1 ! SIDE EFFECTS:
;  376     0502  1 !
;  377     0503  1 !       Advances the pointer, RESULT.
;  378     0504  1 !
;  379     0505  1 !--
;  380     0506  1
;  381     0507  2     BEGIN
;  382     0508  2
;  383     0509  2     LOCAL
;  384     0510  2         CHARS : VECTOR[20],
;  385     0511  2         COUNT;
;  386     0512  2
;  387     0513  2     MAP
;  388     0514  2         PIECES : REF VECTOR;
;  389     0515  2
;  390     0516  2     BIND
;  391     0517  2         HOURS = PIECES[3],
;  392     0518  2         MINUTES = PIECES [4],
;  393     0519  2         SECONDS = PIECES [5];
;  394     0520  2
;  395     0521  2     !Convert the hours.
;  396     0522  2     CONVBB(.HOURS,CHARS,COUNT, 10);
;  397     0523  2
;  398     0524  2     !Put the characters into the string.  Force it to be exactly
;  399     0525  2     !two digits.
;  400     0526  2     INCR I FROM 1 TO (2 - .COUNT) DO
;  401     0527  2         CH$WCHAR_A(%C'0', .RESULT);
;  402     0528  2     DECR I FROM .COUNT TO 1 DO
;  403     0529  2         CH$WCHAR_A(.CHARS[.I-1], .RESULT);
;  404     0530  2
;  405     0531  2     !Insert a separator
;  406     0532  2     CH$WCHAR_A(%C':', .RESULT);
;  407     0533  2
;  408     0534  2     !Convert the minutes
;  409     0535  2     CONVBB(.MINUTES,CHARS,COUNT, 10);
```

```
;  410          0536  2
;  411          0537  2      !Put the characters into the string.  Force it to be exactly
;  412          0538  2      !two digits.
;  413          0539  2      INCR I FROM 1 TO (2 - .COUNT) DO
;  414          0540  2          CH$WCHAR_A(%C'0', .RESULT);
;  415          0541  2      DECR I FROM .COUNT TO 1 DO
;  416          0542  2          CH$WCHAR_A(.CHARS[.I-1], .RESULT);
;  417          0543  2
;  418          0544  2      !Insert a separator
;  419          0545  2      CH$WCHAR_A(%C':', .RESULT);
;  420          0546  2
;  421          0547  2      !Convert the seconds
;  422          0548  2      CONVBB(.SECONDS,CHARS,COUNT, 10);
;  423          0549  2
;  424          0550  2      !Put the characters into the string.  Force it to be exactly
;  425          0551  2      !two digits.
;  426          0552  2      INCR I FROM 1 TO (2 - .COUNT) DO
;  427          0553  2          CH$WCHAR_A(%C'0', .RESULT);
;  428          0554  2      DECR I FROM .COUNT TO 1 DO
;  429          0555  2          CH$WCHAR_A(.CHARS[.I-1], .RESULT);
;  430          0556  2
;  431          0557  2      !Return the length.
;  432          0558  2      .RESULT_LENGTH = 2 + 1 + 2 + 1 + 2;
;  433          0559  2
;  434          0560  2      RETURN;
;  435          0561  2
;  436          0562  1      END;                            !End of CNVTIM


                                    003C 00000          .ENTRY  CNVTIM, Save R2,R3,R4,R5                        ; 0480
                 55 00000000G   EF   9E 00002          MOVAB   CONVBB, R5                                      ;
                 5E        AC   AE   9E 00009          MOVAB   -84(SP), SP                                     ;
50          04   AC             0C   C1 0000D          ADDL3   #12, PIECES, R0                                 ; 0517
53          04   AC             10   C1 00012          ADDL3   #16, PIECES, R3                                 ; 0518
54          04   AC             14   C1 00017          ADDL3   #20, PIECES, R4                                 ; 0519
                                 0A   DD 0001C          PUSHL   #10                                             ; 0522
                           04    AE   9F 0001E          PUSHAB  COUNT                                           ;
                           0C    AE   9F 00021          PUSHAB  CHARS                                           ;
                                 60   DD 00024          PUSHL   (R0)                                            ;
                 65              04   FB 00026          CALLS   #4, CONVBB                                      ;
51               02              6E   C3 00029          SUBL3   COUNT, #2, R1                                   ; 0526
                                 52   D4 0002D          CLRL    I                                               ; 0527
                                 0A   11 0002F          BRB     2$                                              ;
                 50        08    AC   D0 00031 1$:      MOVL    RESULT, R0                                      ;
             00  B0              30   90 00035          MOVB    #48, @0(R0)                                     ;
                                 60   D6 00039          INCL    (R0)                                            ;
F2               52              51   F3 0003B 2$:      AOBLEQ  R1, I, 1$                                       ;
50               6E              01   C1 0003F          ADDL3   #1, COUNT, I                                    ; 0529
                                 0B   11 00043          BRB     4$                                              ;
                 51        08    AC   D0 00045 3$:      MOVL    RESULT, R1                                      ;
             00  B1           6E40   F6 00049          CVTLB   CHARS-4[I], @0(R1)                              ;
                                 61   D6 0004E          INCL    (R1)                                            ;
                 F2              50   F5 00050 4$:      SOBGTR  I, 3$                                           ;
                 50        08    AC   D0 00053          MOVL    RESULT, R0                                      ; 0532
```

```
                 00    B0              3A  90 00057          MOVB    #58, @0(R0)                          ;
                                       60  D6 0005B          INCL    (R0)                                 ;
                                       0A  DD 0005D          PUSHL   #10                                  ; 0535
                             04        AE  9F 0005F          PUSHAB  COUNT                                ;
                             0C        AE  9F 00062          PUSHAB  CHARS                                ;
                                       63  DD 00065          PUSHL   (R3)                                 ;
                       65              04  FB 00067          CALLS   #4, CONVBB                           ;
              51               02      6E  C3 0006A          SUBL3   COUNT, #2, R1                        ; 0539
                                       53  D4 0006E          CLRL    I                                    ; 0540
                                       0A  11 00070          BRB     6$                                   ;
                       50      08      AC  D0 00072 5$:      MOVL    RESULT, R0                           ;
                 00    B0              30  90 00076          MOVB    #48, @0(R0)                          ;
                                       60  D6 0007A          INCL    (R0)                                 ;
              F2               53      51  F3 0007C 6$:      AOBLEQ  R1, I, 5$                            ;
              50               6E      01  C1 00080          ADDL3   #1, COUNT, I                         ; 0542
                                       0B  11 00084          BRB     8$                                   ;
                       51      08      AC  D0 00086 7$:      MOVL    RESULT, R1                           ;
                 00    B1          6E40  F6 0008A          CVTLB   CHARS-4[I], @0(R1)                   ;
                                       61  D6 0008F          INCL    (R1)                                 ;
                       F2              50  F5 00091 8$:      SOBGTR  I, 7$                                ;
                       50      08      AC  D0 00094          MOVL    RESULT, R0                           ; 0545
                 00    B0              3A  90 00098          MOVB    #58, @0(R0)                          ;
                                       60  D6 0009C          INCL    (R0)                                 ;
                                       0A  DD 0009E          PUSHL   #10                                  ; 0548
                             04        AE  9F 000A0          PUSHAB  COUNT                                ;
                             0C        AE  9F 000A3          PUSHAB  CHARS                                ;
                                       64  DD 000A6          PUSHL   (R4)                                 ;
                       65              04  FB 000A8          CALLS   #4, CONVBB                           ;
              51               02      6E  C3 000AB          SUBL3   COUNT, #2, R1                        ; 0552
                                       52  D4 000AF          CLRL    I                                    ; 0553
                                       0A  11 000B1          BRB     10$                                  ;
                       50      08      AC  D0 000B3 9$:      MOVL    RESULT, R0                           ;
                 00    B0              30  90 000B7          MOVB    #48, @0(R0)                          ;
                                       60  D6 000BB          INCL    (R0)                                 ;
              F2               52      51  F3 000BD 10$:     AOBLEQ  R1, I, 9$                            ;
              50               6E      01  C1 000C1          ADDL3   #1, COUNT, I                         ; 0555
                                       0B  11 000C5          BRB     12$                                  ;
                       51      08      AC  D0 000C7 11$:     MOVL    RESULT, R1                           ;
                 00    B1          6E40  F6 000CB          CVTLB   CHARS-4[I], @0(R1)                   ;
                                       61  D6 000D0          INCL    (R1)                                 ;
                       F2              50  F5 000D2 12$:     SOBGTR  I, 11$                               ;
                 0C    BC              08  D0 000D5          MOVL    #8, @RESULT_LENGTH                   ; 0558
                                       04  000D9          RET                                          ; 0562

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 00C4

;  437          0563  1 END                                !End of module
;  438          0564  0 ELUDOM



;                                 PSECT SUMMARY
;
```

```
;       Name                    Bytes                   Attributes
;
;  $PLIT$                        136  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          48  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        414  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]XPORT.L32;1     590         0         0         252       00:00.1
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1   1248         2         0          86       00:00.3
```

```
;                                   COMMAND QUALIFIERS

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

; Size:         414 code + 184 data bytes
; Run Time:       00:09.6
; Elapsed Time:   00:27.0
; Lines/CPU Min:   3536
; Lexemes/CPU-Min: 15529
; Memory Used:  80 pages
; Compilation Complete
```

CNVDAT
LIS

CONVBB
LIS

CONVBR
LIS

CNTVMSMSG
LIS

CONTENTS
LIS

CNTVMS
LIS

CNTVRS
LIS

CONTRL
LIS

CONVBI
LIS