```
RRRRRRRRRRRR    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
RRRRRRRRRRRR    UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR    UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
RRRRRRRRRRRR    UUU         UUU  NNN   NNN   NNN  000         000  FFFFFFFFFFFF     FFFFFFFFFFFF
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
```

_$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
STK
STR
STR
STR
STR
STR
STR
STR
STR
STR

FILEID**DOOPTS

```
DDDDDDDD      OOOOOO      OOOOOO    PPPPPPPP    TTTTTTTTTT    SSSSSSSS
DDDDDDDD      OOOOOO      OOOOOO    PPPPPPPP    TTTTTTTTTT    SSSSSSSS
DD      DD  OO      OO  OO      OO  PP      PP      TT      SS
DD      DD  OO      OO  OO      OO  PP      PP      TT      SS
DD      DD  OO      OO  OO      OO  PP      PP      TT      SS
DD      DD  OO      OO  OO      OO  PP      PP      TT      SS
DD      DD  OO      OO  OO      OO  PPPPPPPP        TT        SSSSSS
DD      DD  OO      OO  OO      OO  PPPPPPPP        TT        SSSSSS
DD      DD  OO      OO  OO      OO  PP              TT              SS
DD      DD  OO      OO  OO      OO  PP              TT              SS
DD      DD  OO      OO  OO      OO  PP              TT              SS      ....
DD      DD  OO      OO  OO      OO  PP              TT              SS      ....
DDDDDDDD      OOOOOO      OOOOOO    PP              TT      SSSSSSSS        ....
DDDDDDDD      OOOOOO      OOOOOO    PP              TT      SSSSSSSS        ....

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 'Digests and distributes command line information.'
;  2          0002  0 MODULE DOOPTS ( 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: Digests and distributes options specified to RUNOFF.
; 35          0035  1 !
; 36          0036  1 !
; 37          0037  1 ! ENVIRONMENT: Transportable
; 38          0038  1 !
; 39          0039  1 ! AUTHOR: R.W.Friday     CREATION DATE: September, 1978
```

```
;    41     0040  1 %SBTTL 'Revision History'
;    42     0041  1 !
;    43     0042  1 ! MODIFIED BY:
;    44     0043  1 !
;    45     0044  1 !       064     REM00064        Ray Marshall    17-May-1984
;    46     0045  1 !               Added support to pickup DIAG2_15 and move it into GCA_CMD_OSQ.
;    47     0046  1 !                 This diag flag was set by /DEC_INTERNAL=OUTPUT_LINE_NUMBER.
;    48     0047  1 !
;    49     0048  1 !       063     REM00063        Ray Marshall    10-April-1984
;    50     0049  1 !               Changed the referenct to the string terminator module for the
;    51     0050  1 !                 LN01 font load sequence from ST to ANSI$ST.
;    52     0051  1 !
;    53     0052  1 !       062     REM00062        Ray Marshall    3-4-April-1984
;    54     0053  1 !               Changed the NOHEADER qualifier back to the way it worked before
;    55     0054  1 !                 -- it now will only suppress the OSC escape sequences, but
;    56     0055  1 !                 not the other escape sequences.
;    57     0056  1 !               Added logic to force the output of a <FF> record after all of
;    58     0057  1 !                 the normal LN01 escape sequences.  This is done by magic:
;    59     0058  1 !                 we increment PHAN_LINES_TP, and later on the new page gets
;    60     0059  1 !                 thrown.  This method also makes sure that the proper number
;    61     0060  1 !                 of lines are output on the first page.  If we write the
;    62     0061  1 !                 formfeed ourselves (herein), we end up with one more line
;    63     0062  1 !                 on the first page then was requested (or expected).
;    64     0063  1 !               We also reduced the automatic /DOWN=3 to /DOWN=2 for LN01
;    65     0064  1 !                 output files.  As before, this automatic value will be
;    66     0065  1 !                 overriden by any user specified value.
;    67     0066  1 !
;    68     0067  1 !       061     REM00061        Ray Marshall    6-December-1983
;    69     0068  1 !               Made still more changes to the implementation of [NO]HEADER.
;    70     0069  1 !                 It seems that we do want to redefine the fonts, but just
;    71     0070  1 !                 not load them.  Also, the logic was modified to suppress
;    72     0071  1 !                 the building of the font module name strings if NOHEADER
;    73     0072  1 !                 was specified.
;    74     0073  1 !               Reformatted some the code around the LN01 processing to make
;    75     0074  1 !                 it read easier and take up fewer lines.
;    76     0075  1 !
;    77     0076  1 !       060     REM00060        Ray Marshall    5-December-1983
;    78     0077  1 !               Made the NOHEADER parameter suppress everything between and
;    79     0078  1 !                 including the two OSC escape sequences.  This is an LN01
;    80     0079  1 !                 feature.
;    81     0080  1 !               Also added RNO$V_LN01_LOAD to control the writing of the OSC
;    82     0081  1 !                 escape sequence that actually causes the loading of fonts.
;    83     0082  1 !
;    84     0083  1 !       059     REM00059        Ray Marshall    22-November-1983
;    85     0084  1 !               Made final (for now) changes to the LN01 output.  It now
;    86     0085  1 !                 conforms to what both VMS and LN01 development groups
;    87     0086  1 !                 say it should be.
;    88     0087  1 !
;    89     0088  1 !       058     REM00058        Ray Marshall    18-November-1983
;    90     0089  1 !               Made more changes to the LN01 output.  Amoung other things,
;    91     0090  1 !                 all DSR_ strings were changed to DSR$.
;    92     0091  1 !               Also fixed a slight bug in the output for the EXTRACT.
;    93     0092  1 !
;    94     0093  1 !       057     KFA00057        Ken Alden       11-Oct-1983
;    95     0094  1 !               Made LN01 margin fix.
;    96     0095  1 !
;    97     0096  1 !       056     KFA00056        Ken Alden       10-Oct-1983
```

```
;   98     0097  1 !                 Fixed 055.
;   99     0098  1 !
;  100     0099  1 !       055       KFA00055        Ken Alden       07-Oct-1983
;  101     0100  1 !                 Added gca_cmd_pages to flag the /PAGES qualifier.
;  102     0101  1 !
;  103     0102  1 !
;  104     0103  1 !       054       REM00054        Ray Marshall    4-Oct-1983
;  105     0104  1 !                 Correct LN01 font loading sequence.  The OSC sequence has
;  106     0105  1 !                   been redefined by VMSland.  Also, the names of the modules
;  107     0106  1 !                   to be extracted from the symbiont's text library have been
;  108     0107  1 !                   changed from COU72... to DSR_FONT_... and the font loading
;  109     0108  1 !                   module name has been changed from FL to DSR_FONT_LOAD.
;  110     0109  1 !
;  111     0110  1 !       053       KFA00053        Ken Alden       12-Sep-1983
;  112     0111  1 !                 Added functionality to LN01 output since the user
;  113     0112  1 !                   may now elect to suppress the ln01 header output(symbiont
;  114     0113  1 !                   information).
;  115     0114  1 !
;  116     0115  1 !       052       KFA00052        Ken Alden       18-Aug-1983
;  117     0116  1 !                 Fixed a logical name translation bug with brn file names.
;  118     0117  1 !                   Problem was in using an outdated XPORT macro.
;  119     0118  1 !
;  120     0119  1 !       051       KFA00051        Ken Alden       28-Jun-1983
;  121     0120  1 !                 /QUICK processing now just calls SETQUICK (false).
;  122     0121  1 !                 The routine SETQUICK is now located in this module.
;  123     0122  1 !
;  124     0123  1 !       050       KFA00050        Ken Alden       24-Jun-1983
;  125     0124  1 !                 Deleted some of Ray's logic in setting setquick to
;  126     0125  1 !                   prevent making a .mem file on the first run of a /auto
;  127     0126  1 !                   file, regardless of whether it had an old brn file or not.
;  128     0127  1 !
;  129     0128  1 !       049       REM00049        Ray Marshall    22-June-1983
;  130     0129  1 !                 Modified logic around call to REABRN.  That routine now
;  131     0130  1 !                   validates the BRN and if invalid, will set GCA_OLD_BRN_EXISTS
;  132     0131  1 !                   to be false.  Therefore, we must test it again after returning
;  133     0132  1 !                   from REABRN to determine /QUICK & /AUTOMATIC processing.
;  134     0133  1 !
;  135     0134  1 !       048       KFA00048        Ken Alden       14-Jun--1983
;  136     0135  1 !                 /AUTOMATIC now implies /CROSS.
;  137     0136  1 !
;  138     0137  1 !       047       KAD00047        Keith Dawson    2-Jun-1983
;  139     0138  1 !                 For LN01, make the user's /DOWN value override if specified.
;  140     0139  1 !
;  141     0140  1 !       046       KAD00046        Keith Dawson    25-May-1983
;  142     0141  1 !                 Minor changes to LN01 Font Assignment escape sequence
;  143     0142  1 !                   due to updated microcode.
;  144     0143  1 !
;  145     0144  1 !       045       KAD00045        Keith Dawson    16-May-1983
;  146     0145  1 !                 /DEVICE=FLIP again asserts gca_bix and gca_btc. Lost call
;  147     0146  1 !                   to PUTRTY (for .BFL-initialization information) reinserted.
;  148     0147  1 !
;  149     0148  1 !       044       KAD00044        Keith Dawson    11-May-1983
;  150     0149  1 !                 /DEVICE=FLIP does not assert gca_bix and gca_btc.
;  151     0150  1 !
;  152     0151  1 !       043       KAD00043        Keith Dawson    4-May-1983
;  153     0152  1 !                 For LN01 output, initially issue Default Font Invocation
;  154     0153  1 !                   escape sequence.
```

```
;  155          0154  1 !
;  156          0155  1 !       042     KAD00042        Keith Dawson    18-April-1983
;  157          0156  1 !               Minor tweaks to LN01 output: issue escape sequence to
;  158          0157  1 !                 return to top of page. Do not quote font names. Set the
;  159          0158  1 !                 values of /RIGHT and /DOWN (unless the user gave values
;  160          0159  1 !                 for them) to move LN01 output off the upper-left of the
;  161          0160  1 !                 paper.
;  162          0161  1 !               Remove all LN01 conditionals.
;  163          0162  1 !
;  164          0163  1 !       041     KAD00041        Keith Dawson    14-April-1983
;  165          0164  1 !               /CROSS now ==> /INTERMEDIATE. Bit gca_expand_cref is not
;  166          0165  1 !                 turned off if no pre-existing .BRN.
;  167          0166  1 !
;  168          0167  1 !       040     KAD00040        Keith Dawson    11-April-1983
;  169          0168  1 !               Added support for new termination error messages for
;  170          0169  1 !                 information written to .BRN file. This involved adding
;  171          0170  1 !                 another formal to the RGH routine; this third formal is
;  172          0171  1 !                 TRUE or FALSE, as the caller determines whether or not to
;  173          0172  1 !                 increment the count of information written to the .BRN file.
;  174          0173  1 !
;  175          0174  1 !       039     KAD00039        Keith Dawson    8-Apr-1983
;  176          0175  1 !               Make /NOCROSS the default.
;  177          0176  1 !
;  178          0177  1 !       038     KAD00038        Keith Dawson    7-Apr-1983
;  179          0178  1 !               Consolidate support for SETQUICK routine (which is in
;  180          0179  1 !                 DOAUTO). Do not try to open old .BRN file if input
;  181          0180  1 !                 file is a terminal.
;  182          0181  1 !
;  183          0182  1 !       037     KAD00037        Keith Dawson    5-Apr-1983
;  184          0183  1 !               Full support for /CROSS and /AUTO.
;  185          0184  1 !
;  186          0185  1 !       036     REM00036        Ray Marshall    4-April-1983
;  187          0186  1 !               Added support for /DEBUG=(CROSS,SAVE).
;  188          0187  1 !
;  189          0188  1 !       035     KAD00035        Keith Dawson    21-Mar-1983
;  190          0189  1 !               Added LN01 support. Got rid of /OVERPRINT.
;  191          0190  1 !
;  192          0191  1 !       034     KAD00034        Keith Dawson    20-Mar-1983
;  193          0192  1 !               Removed all references to .BIX and .BTC files.
;  194          0193  1 !
;  195          0194  1 !       033     KFA00033        Ken Alden       07-Mar-1983
;  196          0195  1 !               Global edit of all modules. Updated module names, idents,
;  197          0196  1 !               copyright dates. Changed require files to BLISS library.
;  198          0197  1 !
;  199          0198  1 !--
```

```
;  201          0199  1 %SBTTL 'Module Level Declarations'
;  202          0200  1 !
;  203          0201  1 ! TABLE OF CONTENTS:
;  204          0202  1 !
;  205          0203  1 REQUIRE 'REQ:RNODEF';                       ! RUNOFF variant definitions
;  206          0334  1
;  207          0335  1 FORWARD ROUTINE
;  208          0336  1     DOOPTS,
;  209          0337  1     setquick : NOVALUE,
;  210          0338  1     write_ln01_info : NOVALUE;
;  211          0339  1 !
;  212          0340  1 ! INCLUDE FILES:
;  213          0341  1 !
;  214          0342  1
;  215          0343  1 LIBRARY 'NXPORT:XPORT';                     ! XPORT Library
;  216          0344  1
;  217        U 0345  1 %IF DSRPLUS %THEN
;  218        U 0346  1 LIBRARY 'REQ:DPLLIB';                     ! DSRPLUS BLISS Library
;  219          0347  1 %ELSE
;  220          0348  1 LIBRARY 'REQ:DSRLIB';                     ! DSR BLISS Library
;  221          0349  1 %FI
;  222          0350  1
;  223          0351  1 !
;  224          0352  1 ! MACROS:
;  225          0353  1 !
;  226          0354  1 MACRO
;  227        M 0355  1     erm_t (rnfcode, str_descr) =
;  228        M 0356  1         ! This macro, ERM_T, is used to output as part of an error message a
;  229        M 0357  1         !   string described by an XPORT string descriptor.
;  230        M 0358  1         BEGIN
;  231        M 0359  1         BIND
;  232        M 0360  1             temp = str_descr : $STR_DESCRIPTOR ();
;  233        M 0361  1         erme (rnfcode, .temp [STR$A_POINTER], .temp [STR$H_LENGTH], .semcod)
;  234        M 0362  1         END
;  235          0363  1     %;
;  236          0364  1
;  237          0365  1 !
;  238          0366  1 ! EQUATED SYMBOLS:
;  239          0367  1 !
;  240          0368  1
;  241          0369  1 LITERAL
;  242          0370  1     escape = 27,                          ! Used for LN01 escape sequences.
;  243          0371  1     ppi = 300;                            ! Pixels-per-inch resolution of LN01.
;  244          0372  1
;  245          0373  1 EXTERNAL LITERAL
;  246          0374  1     RINTES : UNSIGNED (8);
;  247          0375  1 !
;  248          0376  1 ! OWN STORAGE:
;  249          0377  1 !
;  250          0378  1 !
;  251          0379  1 ! EXTERNAL REFERENCES:
;  252          0380  1 !
;  253          0381  1 EXTERNAL LITERAL                                  !Error messages
;  254          0382  1     RNFCEM,               ! Comma expected, missing: "<%S>"
;  255          0383  1     RNFCOB,               ! Can't open binary file
;  256          0384  1     RNFINM,               ! Illegal number value: <qualifier>
;  257          0385  1     RNFIVS,               ! Illegal /VARIANT qualifier
```

```
;  258          0386  1     RNFTMP,                 ! Too many page ranges on /PAGES qualifier
;  259          0387  1     RNFTMV;                 ! Too many /VARIANTs
;  260          0388  1
;  261          0389  1 EXTERNAL
;  262          0390  1     fs01 : fixed_string,
;  263          0391  1     fra : fixed_string,
;  264          0392  1     gca : gca_definition,
;  265          0393  1     hct : hct_definition,
;  266          0394  1     khar,
;  267          0395  1     spager : BLOCKVECTOR [1,page_sct_size],
;  268          0396  1     tpager : BLOCKVECTOR [1,page_sct_size],
;  269          0397  1     rnoiob : REF $XPO_IOB (),
;  270          0398  1     rniiob : REF $XPO_IOB (),
;  271          0399  1     brniob : $XPO_IOB (),
;  272          0400  1     brnoob : $XPO_IOB (),
;  273          0401  1     ffname : $STR_DESCRIPTOR (CLASS = DYNAMIC), !Failure filename destination
;  274          0402  1     semcod,
;  275          0403  1     outopt : outopt_define,
;  276          0404  1     phan : phan_definition,
;  277          0405  1     sca : sca_definition,
;  278          0406  1     tsiiob : $XPO_IOB (),
;  279          0407  1     vrcnt;
;  280          0408  1
;  281          0409  1 EXTERNAL ROUTINE
;  282          0410  1     bars,       bwait,          clh,
;  283          0411  1     erm,        erme,           grab_resultant,
;  284          0412  1     gname,      parsep,         putrty,         rgh,
;  285        U 0413  1 %IF DSRPLUS %THEN
;  286        U 0414  1     reabrn,
;  287          0415  1 %FI
;  288          0416  1     vrentr,     vrfind;
```

```
;  290          0417  1 %SBTTL 'DOOPTS -- ROUTINE header'
;  291          0418  1 GLOBAL ROUTINE DOOPTS (RNO_CMD) =
;  292          0419  1
;  293          0420  1 !++
;  294          0421  1 ! FUNCTIONAL DESCRIPTION:
;  295          0422  1 !
;  296          0423  1 !       DOOPTS interprets the qualifiers that the user specified on
;  297          0424  1 !       the command line. Information is disseminated to various
;  298          0425  1 !       RUNOFF control structures.
;  299          0426  1 !
;  300          0427  1 ! FORMAL PARAMETERS:
;  301          0428  1 !
;  302          0429  1 !       RNO_CMD is the preprocessed set of qualifiers.
;  303          0430  1 !
;  304          0431  1 ! IMPLICIT INPUTS:      None
;  305          0432  1 !
;  306          0433  1 ! IMPLICIT OUTPUTS:     None
;  307          0434  1 !
;  308          0435  1 ! ROUTINE VALUE:
;  309          0436  1 ! COMPLETION CODES:     None
;  310          0437  1 !
;  311          0438  1 ! SIDE EFFECTS:         None
;  312          0439  1 !
;  313          0440  1 !--
;  314          0441  1
;  315          0442  2     BEGIN
;  316          0443  2
;  317          0444  2     OWN
;  318          0445  2         u_ext : VECTOR [CH$ALLOCATION (4)],
;  319          0446  2         u_ext_ptr,
;  320          0447  2         type_length,
;  321          0448  2         type_ptr,
;  322          0449  2         prse_spec_block : $XPO_SPEC_BLOCK,
;  323          0450  2         range_error_flag;
;  324          0451  2
;  325          0452  2     MAP
;  326          0453  2         rno_cmd : REF $rno_cmd;
;  327          0454  2
;  328          0455  2     range_error_flag = 0;               ! Turn it off to start with.
```

```
;  330     0456  2 %SBTTL 'DOOPTS -- Process .RNH file type'
;  331     0457  2     !See if the user specified a .RNH input file, and if so apply special formatting rules.
;  332     0458  3     BEGIN
;  333     0459  3
;  334     0460  3     !Get the input file type.
;  335   P 0461  3     $XPO_PARSE_SPEC (     SPEC_BLOCK = prse_spec_block
;  336     0462  3                         ,FILE_SPEC = rniiob [IOB$T_RESULTANT] );
;  337     0463  3
;  338     0464  3     !Get the length and location of the file type
;  339     0465  4     BEGIN
;  340     0466  4     BIND
;  341     0467  4         temp = prse_spec_block [XPO$T_FILE_TYPE] : $STR_DESCRIPTOR ();
;  342     0468  4     type_length = .temp [STR$H_LENGTH];
;  343     0469  4     type_ptr = .temp [STR$A_POINTER];
;  344     0470  3     END;
;  345     0471  3
;  346     0472  3     IF  .type_length EQL 4
;  347     0473  3     THEN                                    !
;  348     0474  3     !Check further to see if the type is '.RNH'.
;  349     0475  4         BEGIN
;  350     0476  4         !First convert the file type to upper case.
;  351     0477  4         u_ext_ptr = CH$PTR (u_ext);
;  352     0478  4
;  353     0479  4         INCR i FROM 1 TO 4 DO
;  354     0480  5             BEGIN
;  355     0481  5
;  356     0482  5             LOCAL
;  357     0483  5                 kahr;
;  358     0484  5
;  359     0485  5             kahr = CH$RCHAR_A (type_ptr);
;  360     0486  5
;  361     0487  6             IF lower_letter (.kahr)
;  362     0488  5             THEN
;  363     0489  5                 CH$WCHAR_A (upper_letter (.kahr), u_ext_ptr)
;  364     0490  5             ELSE
;  365     0491  5                 CH$WCHAR_A (.kahr, u_ext_ptr)
;  366     0492  4             END;
;  367     0493  4
;  368     0494  4         u_ext_ptr = CH$PTR (u_ext);
;  369     0495  4
;  370     0496  4         IF CH$EQL (4, .u_ext_ptr, 4, CH$PTR (UPLIT ('.RNH')))
;  371     0497  4         THEN                                !
;  372     0498  4         !It is a .RNH input file.
;  373     0499  5             BEGIN
;  374     0500  5             hct_headers = false;            !No page headers wanted.
;  375     0501  5             phan_paging = false;            !Don't divide document into pages.
;  376     0502  5             phan_cmd_paging = false;        !...
;  377     0503  5             sca_rm = 72;                    !Set right margin to 72.
;  378     0504  5             gca_lwidth = 72;                !...
;  379     0505  4             END;
;  380     0506  3         END;
;  381     0507  2     END;
```

```
;  383     0508  2 %SBTTL 'DOOPTS -- Process /PAGES switch'
;  384     0509  2
;  385     0510  2     IF .rno_cmd [rno$h_pages] GTR 0
;  386     0511  2     THEN
;  387     0512  2     !User did specify some pages.
;  388     0513  3         BEGIN
;  389     0514  3
;  390     0515  3         LOCAL
;  391     0516  3             ira : VECTOR [4];
;  392     0517  3
;  393     0518  3         !Set up dummy fixed string, so other routines can do parsing.
;  394     0519  3
;  395     0520  3         MAP
;  396     0521  3             ira : fixed_string;
;  397     0522  3
;  398     0523  3         fs_start (ira) = .rno_cmd [rno$a_pages];
;  399     0524  3         fs_next (ira) = .fs_start (ira);
;  400     0525  3         gca_com_start = .fs_start (ira);        ! Hack needed to tell the error
;  401     0526  3                                                 ! message handler where the
;  402     0527  3                                                 ! start of the string is.
;  403     0528  3         fs_maxsize (ira) = .rno_cmd [rno$h_pages];
;  404     0529  3         fs_length (ira) = .fs_maxsize (ira);
;  405     0530  3
;  406     0531  3         !Initialize working string.
;  407     0532  3
;  408     0533  3         fs_init (fs01);
;  409     0534  3         kcns ();
;  410     0535  3
;  411     0536  3 !Collect list of pages.
;  412     0537  3 INCR i FROM 0 TO (max_page_ranges - 1) DO
;  413     0538  3
;  414     0539  3     !Attempt to get a page number.
;  415     0540  3     IF  NOT parsep (ira, spager [.i, sct_typ])
;  416     0541  3     THEN
;  417     0542  3         RETURN FALSE                            !Invalid page number.
;  418     0543  3     ELSE
;  419     0544  3     !Valid page number.  Attempt to pick up a terminating page.
;  420     0545  4         BEGIN
;  421     0546  4
;  422     0547  4         BIND                                    !Clear terminating page.
;  423     0548  4             x = tpager [.i, sct_typ] : VECTOR;  !...
;  424     0549  4
;  425     0550  4         INCR i FROM 0 TO (page_sct_size - 1) DO !...
;  426     0551  4             x [.i] = 0;                         !...
;  427     0552  4
;  428     0553  4         gca_orange_cnt = .i + 1;                !Remember page-range count.
;  429     0554  4
;  430     0555  4         IF .khar NEQ rintes
;  431     0556  4         THEN
;  432     0557  4         !The parse of the initial page did not exhaust the entire
;  433     0558  4         !list of pages.  If a ':' follows, a terminating page must
;  434     0559  4         !have been given.  Otherwise, there must be a ',' to
;  435     0560  4         !introduce a new page.
;  436     0561  5             BEGIN
;  437     0562  5
;  438     0563  5             !For PDP-11: recognize ¦ or : as range-indicator.
;  439     0564  5             IF .khar EQL %C':' OR .khar EQL %C'¦'
```

```
;  440          0565  5              THEN
;  441          0566  5              !A terminating page number has to follow.
;  442          0567  5              !Attempt to pick it up.
;  443          0568  6                  BEGIN
;  444          0569  6                  kcns ();                            !Skip the ':'
;  445          0570  6
;  446          0571  6                  IF NOT parsep (ira, tpager [.i, sct_typ])
;  447          0572  6                  THEN
;  448          0573  6                      RETURN false                    !Bad or missing page number.
;  449          0574  5                  END;
;  450          0575  5
;  451          0576  5              !Got the terminating page number successfully.
;  452          0577  5              !See if another page range might follow.
;  453          0578  5              IF  .KHAR EQL RINTES
;  454          0579  5              THEN
;  455          0580  5                  EXITLOOP;                           !Nothing left.
;  456          0581  5
;  457          0582  5              !Something still there after the last page number.
;  458          0583  5              IF  .khar EQL %C',' OR .khar EQL %C';'
;  459          0584  5              THEN
;  460          0585  5              !Yes, there should be another. For now just skip the ','.
;  461          0586  5              !The next pass through the loop will get the next one.
;  462          0587  6                  kcns ()
;  463          0588  5              ELSE
;  464          0589  5              !Tell user a comma is (probably) missing
;  465          0590  6                  BEGIN
;  466          0591  6                  erm (rnfcem, .fs_start(ira), .fs_maxsize(ira));
;  467          0592  6                  RETURN false;
;  468          0593  6                  END
;  469          0594  6
;  470          0595  5                  END
;  471          0596  5
;  472          0597  4              ELSE
;  473          0598  4              !The list of pages has been completely scanned, and everything went ok.
;  474          0599  4                  EXITLOOP
;  475          0600  4
;  476          0601  3              END;                                    !End of loop.
;  477          0602  3
;  478          0603  3              !Be sure the user did not specify too many page ranges.
;  479          0604  3              IF .khar NEQ rintes
;  480          0605  3              THEN
;  481          0606  3              !User specified too many page ranges.  Give up.
;  482          0607  4                  BEGIN
;  483          0608  4                  erm (rnftmp, 0, 0);
;  484          0609  4                  RETURN false;
;  485          0610  3                  END;
;  486          0611  3
;  487          0612  3              IF .rno_cmd [rno$h_pages] GTR 0
;  488          0613  3              THEN
;  489          0614  3                  gca_cmd_pages = true;               !Set a flag saying that user said /PAGES
;  490          0615  3              gca_skip_out = true;            !Start with output suppressed.
;  491          0616  2              END;
```

```
;  493          0617  2 %SBTTL 'DOOPTS -- Process /AUTOMATIC switch'
;  494          0618  2
;  495        U 0619  2 %IF DSRPLUS %THEN
;  496        U 0620  2
;  497        U 0621  2     ! /AUTOMATIC switch
;  498        U 0622  2     gca_black_box = false;                          ! Assume /NOAUTOMATIC
;  499        U 0623  2
;  500        U 0624  2 %IF %BLISS (BLISS32) %THEN
;  501        U 0625  2     IF .rno_cmd [rno$v_automatic]
;  502        U 0626  2     THEN
;  503        U 0627  2         BEGIN
;  504        U 0628  2         gca_bix = true;
;  505        U 0629  2         gca_btc = true;
;  506        U 0630  2         gca_cmd_btc = true;
;  507        U 0631  2         gca_black_box = true;
;  508        U 0632  2         rno_cmd [rno$v_intermediate] = true;     ! Force generation of a BRN file
;  509        U 0633  2         rno_cmd [rno$v_cross_reference] = true; ! Turn on cross referencing.
;  510        U 0634  2         END;
;  511        U 0635  2 %FI
```

```
;   513         U 0636  2 %SBTTL 'DOOPTS -- read crossreference information from old .BRN file'
;   514         U 0637  2
;   515         U 0638  2     gca_old_brn_exists = false;                   ! Assume it doesn't.
;   516         U 0639  2
;   517         U 0640  2     IF  ( .gca_black_box OR .rno_cmd [rno$v_cross_reference] )
;   518         U 0641  2        AND NOT
;   519         U 0642  2         ( .rniiob [IOB$V_TERMINAL] )
;   520         U 0643  2     THEN                                          ! /AUTOMATIC or /CROSS_REFERENCE
;   521         U 0644  2         BEGIN
;   522         U 0645  2         LOCAL
;   523         U 0646  2             status;
;   524         U 0647  2         $XPO_IOB_INIT (IOB = brniob);
;   525         U 0648  2
;   526         U 0649  2         ! Check to see if an old version of the BRN file exists.
;   527         U 0650  2         !
;   528         U 0651  2         status = $XPO_OPEN ( IOB     =  brniob
;   529         U 0652  2                            , FILE_SPEC  =  '.BRN'
;   530         U 0653  2                            , RELATED    =  rniiob [IOB$T_RESULTANT]
;   531         U 0654  2                            , OPTIONS    =  INPUT
;   532         U 0655  2                            , ATTRIBUTES =  BINARY
;   533         U 0656  2                            , FAILURE    =  0);
;   534         U 0657  2         IF .status
;   535         U 0658  2         THEN
;   536         U 0659  2             gca_old_brn_exists = true;
;   537         U 0660  2         END;
```

```
;  539        U 0661  2 %sbttl 'DOOPTS -- Process /CROSSREFERENCE switch'
;  540        U 0662  2
;  541        U 0663  2     gca_expand_cref = false;              ! Assume we won't expand.
;  542        U 0664  2 !+
;  543        U 0665  2 ! /NOCROSS is overridden by /AUTO. So we don't allow users to say /NOCROSS
;  544        U 0666  2 ! and /AUTO to produce a complete document, but without cross-references
;  545        U 0667  2 ! expanded.
;  546        U 0668  2 !
;  547        U 0669  2 ! /CROSS turns on /INTERMEDIATE as well, unless the user said /NOINTER.
;  548        U 0670  2 !-
;  549        U 0671  2     gca_cross_reference = ( .rno_cmd [rno$v_cross_reference] OR .rno_cmd [rno$v_automatic] );
;  550        U 0672  2
;  551        U 0673  2     IF .gca_cross_reference
;  552        U 0674  2     THEN
;  553        U 0675  2         BEGIN                                ! User said /CROSS_REFERENCE or /AUTO.
;  554        U 0676  2
;  555        U 0677  2         gca_expand_cref = true;              ! So expand cross references.
;  556        U 0678  2
;  557        U 0679  2 !**new
;  558        U 0680  2         ! The following logic will force generation of a BRN file unless
;  559        U 0681  2         ! /NOINTERMEDIATE was explicitly specified.
;  560        U 0682  2         !
;  561        U 0683  2         rno_cmd [rno$v_intermediate] =
;  562        U 0684  2             (.rno_cmd [rno$v_intermediate]
;  563        U 0685  2            OR
;  564        U 0686  2              NOT .rno_cmd [rno$v_s_intermediate]);
;  565        U 0687  2
;  566        U 0688  2 %IF %BLISS (BLISS32) %THEN                   ! Multi-pass logic only for BLISS32.
;  567        U 0689  2         IF .gca_pass_count EQL 1
;  568        U 0690  2         THEN
;  569        U 0691  2             BEGIN                            ! First pass.
;  570        U 0692  2 %FI
;  571        U 0693  2
;  572        U 0694  2             ! If there is an "old" BRN, validate it.  If valid, read it.
;  573        U 0695  2
;  574        U 0696  2             IF .gca_old_brn_exists THEN
;  575        U 0697  2                 reabrn ();                   ! Read in existing BRN file.
;  576        U 0698  2
;  577        U 0699  2 %IF %BLISS (BLISS32) %THEN
;  578        U 0700  2
;  579        U 0701  2             ! If there wasn't an "old" BRN (or REABRN found it to be invalid),
;  580        U 0702  2             ! and /AUTOMATIC has been asserted,  set /QUICK and initialize
;  581        U 0703  2             ! automatic operation.
;  582        U 0704  2
;  583        U 0705  2 !!          IF NOT .gca_old_brn_exists AND .gca_black_box
;  584        U 0706  2             IF .gca_black_box
;  585        U 0707  2             THEN                             ! User said /AUTOMATIC.
;  586        U 0708  2                 BEGIN
;  587        U 0709  2                 setquick (true);             ! Pretend user said /QUICK.
;  588        U 0710  2
;  589        U 0711  2                 IF .gca_rerun_count EQL 0
;  590        U 0712  2                 THEN
;  591        U 0713  2                     gca_rerun_count = 1 ! Need a second pass at input file.
;  592        U 0714  2                 END
;  593        U 0715  2             END;
;  594        U 0716  2 %FI
;  595        U 0717  2         END;
```

```
;  596        U 0718  2
;  597        U 0719  2      IF .gca_old_brn_exists
;  598        U 0720  2      THEN
;  599        U 0721  2          $XPO_CLOSE (IOB = brniob, OPTIONS = REMEMBER);  ! Old BRN file exists and is open.
;  600          0722  2 %FI                                                      ! Close it and remember its name.
```

```
;  602          0723  2 %SBTTL 'DOOPTS -- Process /INTERMEDIATE switch'
;  603          0724  2     IF .rno_cmd [rno$v_intermediate]
;  604          0725  2             AND NOT
;  605          0726  3         (.rno_cmd [rno$v_4_out_format] EQL op_dev_flip)
;  606          0727  3
;  607          0728  2     THEN
;  608          0729  2     !User said /INTERMEDIATE (and is not generating FLIP output).
;  609          0730  2     !So we will open a .BRN file and write both indexing and contents
;  610          0731  2     !information into it.
;  611          0732  2     !
;  612          0733  3         BEGIN
;  613          0734  3         LOCAL
;  614          0735  3             status;
;  615          0736  3
;  616          0737  3         ! Open the output BRN file.
;  617        P 0738  3         status = $XPO_OPEN ( IOB = brnoob
;  618        P 0739  3                             ,FILE_SPEC = rno_cmd [rno$t_intermediate]
;  619        P 0740  3                             ,DEFAULT = ('.BRN')
;  620        P 0741  3                             ,RELATED = rniiob [IOB$T_RESULTANT]
;  621        P 0742  3                             ,FAILURE = grab_resultant
;  622        P 0743  3                             ,OPTIONS = OUTPUT
;  623          0744  3                             ,ATTRIBUTES = BINARY );
;  624          0745  3
;  625          0746  3         IF NOT .status THEN
;  626          0747  4             BEGIN
;  627          0748  4             erm_t (rnfcob, ffname);
;  628          0749  4             RETURN false;
;  629          0750  4             END
;  630          0751  3         ELSE
;  631          0752  4             BEGIN
;  632          0753  4             !Now that the file has been opened successfully, initialize it
;  633          0754  4             !with a BRN header record.
;  634          0755  4             LOCAL
;  635          0756  4                 temp : VECTOR [2];
;  636          0757  4         !+
;  637          0758  4         !Write the .BRN File Identification Record Group. It identifies the
;  638          0759  4         !format of the .BRN file, and of the indexing, contents, and cross-
;  639          0760  4         !reference information in it. This lets INDEX, CONTENTS, and DSRPLUS
;  640          0761  4         !decide if they are prepared to read this format.
;  641          0762  4         !-
;  642          0763  4             temp [0] = brn_file;
;  643          0764  4             temp [1] = brn_ident;
;  644          0765  4             !
;  645          0766  4             !Write these records as the first information in the file.
;  646          0767  4             $XPO_PUT ( IOB = brnoob, BINARY_DATA = (2, temp) );
;  647          0768  4
;  648          0769  4             !Write a Record Group Header for Indexing information.
;  649          0770  4             ! Do count this record in the .BRN count.
;  650          0771  4             rgh (brn_index, 1, false);
;  651          0772  4             !
;  652          0773  4             !Add the New .BIX File Record Group.
;  653          0774  4             temp [0] = new_sequence + (index_format^(%BPVAL/2));
;  654          0775  4             !
;  655          0776  4             $XPO_PUT ( IOB = brnoob, BINARY_DATA = (1, temp) );
;  656          0777  4
;  657          0778  4             !Write a Record Group Header for Contents information.
;  658          0779  4             ! Do count this record in the .BRN count.
```

```
;  659     0780  4            rgh (brn_contents, 2, false);
;  660     0781  4            !
;  661     0782  4            !Add the New .BTC File Record Group.
;  662     0783  4            temp [0] = maj_new_toc;
;  663     0784  4            temp [1] = toc_format;
;  664     0785  4            !
;  665     0786  4            $XPO_PUT ( IOB = brnoob, BINARY_DATA = (2, temp) );
;  666     0787  4
;  667     0788  4            !Write a Record Group Header for Cross-Reference information.
;  668     0789  4            ! Do count this record in the .BRN count.
;  669     0790  4            rgh (brn_crossref, 2, false);
;  670     0791  4            !
;  671     0792  4            !Add the New Cross-Reference Record Group.
;  672     0793  4            temp [0] = new_crossref;
;  673     0794  4            temp [1] = crossref_format;
;  674     0795  4            !
;  675     0796  4            $XPO_PUT ( IOB = brnoob, BINARY_DATA = (2, temp) );
;  676     0797  4        !+
;  677     0798  4        ! End of .BRN File Identification Record Group.
;  678     0799  4        !-
;  679     0800  4            gca_bix = true;                 !Set binary-index flag.
;  680     0801  4            gca_btc = true;                 !Set binary-toc flag.
;  681     0802  4            gca_cmd_btc = true;             !...
;  682     0803  3            END;
;  683     0804  3        END
;  684     0805  2    ELSE
;  685     0806  3        BEGIN
;  686     0807  3        gca_bix = false;
;  687     0808  3        gca_btc = false;
;  688     0809  3        gca_cmd_btc = false;
;  689     0810  2        END;
```

```
;  691          0811  2 %SBTTL 'DOOPTS -- Process /VARIANT switch'
;  692        U 0812  2 %IF DSRPLUS %THEN
;  693        U 0813  2     !
;  694        U 0814  2     ! Hard-wire the variant "DSRPLUS", and also FLIP if the user said
;  695        U 0815  2     ! /DEC=FLIP.
;  696        U 0816  2     !
;  697        U 0817  2     BEGIN                           !Local definition block
;  698        U 0818  2     LOCAL
;  699        U 0819  2         temp_ptr;
;  700        U 0820  2     temp_ptr = CH$PTR (UPLIT ('DSRPLUS'));
;  701        U 0821  2     vrentr (.temp_ptr, 7, %c' ', %c' ', 0, true);
;  702        U 0822  2
;  703        U 0823  2 %IF FLIP %THEN
;  704        U 0824  2     IF (.gca_op_dev EQL op_dev_flip)
;  705        U 0825  2     THEN
;  706        U 0826  2         BEGIN
;  707        U 0827  2         temp_ptr = CH$PTR (UPLIT ('FLIP'));
;  708        U 0828  2         vrentr (.temp_ptr, 4, %c' ', %c' ', 0, true);
;  709        U 0829  2         END;
;  710        U 0830  2 %FI
;  711        U 0831  2     END;                            !End of local definition block
;  712          0832  2 %FI
;  713          0833  2
;  714          0834  2     IF .rno_cmd [rno$h_variant] GTR 0
;  715          0835  2     THEN                    !User did specify at least one /VARIANT.
;  716          0836  3         BEGIN
;  717          0837  3         LOCAL
;  718          0838  3             ira : VECTOR [4],
;  719          0839  3             gncc;
;  720          0840  3
;  721          0841  3         !Fix up IRA to look like a fixed string so other routines
;  722          0842  3         !can do some parsing.
;  723          0843  3         MAP
;  724          0844  3             ira : fixed_string;
;  725          0845  3
;  726          0846  3         fs_start (ira) = .rno_cmd [rno$a_variant];
;  727          0847  3         fs_next (ira) = .fs_start (ira);
;  728          0848  3         fs_maxsize (ira) = .rno_cmd [rno$h_variant];
;  729          0849  3         fs_length (ira) = .fs_maxsize (ira);
;  730          0850  3         kcns ();                                    !Start the scan.
;  731          0851  3
;  732          0852  3         WHILE .khar NEQ rintes DO
;  733          0853  4             BEGIN
;  734          0854  4             fs_init (fs01);
;  735          0855  4
;  736          0856  4             IF gname (ira, fs01) NEQ gname_normal
;  737          0857  4             THEN                    !Bad variable name. Abort entire command line.
;  738          0858  5                 BEGIN
;  739          0859  5                 erm (rnfivs, 0, 0);
;  740          0860  5                 RETURN false;
;  741          0861  4                 END;
;  742          0862  4
;  743          0863  4             IF .vrcnt GEQ max_vr_names
;  744          0864  4             THEN                    !Too many variants. Abort entire command line.
;  745          0865  5                 BEGIN
;  746          0866  5                 erm (rnftmv, 0, 0);
;  747          0867  5                 RETURN false;
```

```
;  748     0868  4                END;
;  749     0869  4
;  750     0870  4            IF vrfind (.fs_start (fs01), .fs_length (fs01)) EQL -1
;  751     0871  4            THEN                !This is a brand new name, so save it marked as TRUE.
;  752     0872  4                vrentr (.fs_start (fs01), .fs_length (fs01), %C' ', %C' ', 0, true)
;  753     0873  4            ELSE                !Ignore duplicates
;  754     0874  4                (0);
;  755     0875  4
;  756     0876  4            !For PDP-11: recognize ; or , as variant-separator.
;  757     0877  4            IF .khar EQL %C',' OR .khar EQL %C';'
;  758     0878  4            THEN                !Skip separator. This is a list of names.
;  759     0879  5                kcns ()
;  760     0880  4            ELSE
;  761     0881  4                IF .khar NEQ rintes
;  762     0882  4                THEN            !Garbage after the variable name.
;  763     0883  4                                !Abort the entire command line.
;  764     0884  5                    BEGIN
;  765     0885  5                    erm (rnfivs, 0, 0);
;  766     0886  5                    RETURN false;
;  767     0887  4                    END;
;  768     0888  3            END;
;  769     0889  2        END;
```

```
;   771     0890  2 %SBTTL 'DOOPTS -- Check range of numeric parameters'
;   772     0891  2
;   773     0892  3     IF (.rno_cmd[rno$h_bold] LSS 0  OR
;   774     0893  3         .rno_cmd[rno$h_bold] GTR 10 )
;   775     0894  2     THEN
;   776     0895  3         BEGIN
;   777     0896  3         ERM (RNFINM, CH$PTR(UPLIT ('/BOLD')), 5);
;   778     0897  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   779     0898  2         END;
;   780     0899  2
;   781     0900  3     IF (.rno_cmd[rno$h_down] LSS 0  OR
;   782     0901  3         .rno_cmd[rno$h_down] GTR 200 )
;   783     0902  2     THEN
;   784     0903  3         BEGIN
;   785     0904  3         ERM (RNFINM, CH$PTR(UPLIT ('/DOWN')), 5);
;   786     0905  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   787     0906  2         END;
;   788     0907  2
;   789     0908  3     IF (.rno_cmd[rno$h_form_size] LSS 0 OR
;   790     0909  3         .rno_cmd[rno$h_form_size] GTR 200 )
;   791     0910  2     THEN
;   792     0911  3         BEGIN
;   793     0912  3         ERM (RNFINM, CH$PTR(UPLIT ('/FORM_SIZE')), 10);
;   794     0913  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   795     0914  2         END;
;   796     0915  2
;   797     0916  3     IF (.rno_cmd[rno$h_right] LSS 0  OR
;   798     0917  3         .rno_cmd[rno$h_right] GTR 150  )
;   799     0918  2     THEN
;   800     0919  3         BEGIN
;   801     0920  3         ERM (RNFINM, CH$PTR(UPLIT ('/RIGHT')), 6);
;   802     0921  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   803     0922  2         END;
;   804     0923  2
;   805     0924  3     IF  (.rno_cmd[rno$v_s_underline] AND .rno_cmd[rno$c_underline] NEQ 0 AND
;   806     0925  4             (.rno_cmd[rno$c_underline] LSS 32 OR    ! These values describe
;   807     0926  3              .rno_cmd[rno$c_underline] GTR 125 ))   ! "printable" characters
;   808     0927  2     THEN
;   809     0928  3         BEGIN
;   810     0929  3         ERM (RNFINM, CH$PTR(UPLIT ('/UNDERLINE')), 10);
;   811     0930  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   812     0931  2         END;
;   813     0932  2
;   814     0933  3     IF  (.rno_cmd[rno$v_s_und_separ] AND .rno_cmd[rno$c_underline] NEQ 0 AND
;   815     0934  4             (.rno_cmd[rno$c_underline] LSS 32 OR    ! These values describe
;   816     0935  3              .rno_cmd[rno$c_underline] GTR 125 ))   ! "printable" characters
;   817     0936  2     THEN
;   818     0937  3         BEGIN
;   819     0938  3         ERM (RNFINM, CH$PTR(UPLIT ('/SEPARATE_UNDERLINE')), 19);
;   820     0939  3         range_error_flag = 1;           ! Indicate error to force exit later.
;   821     0940  2         END;
;   822     0941  2
;   823     0942  3     IF  (.rno_cmd[rno$v_s_und_nonsp] AND .rno_cmd[rno$c_underline] NEQ 0 AND
;   824     0943  4             (.rno_cmd[rno$c_underline] LSS 0  OR    ! These values describe
;   825     0944  3              .rno_cmd[rno$c_underline] GTR 32 ))    ! "NONprintable" characters
;   826     0945  2     THEN
;   827     0946  3         BEGIN
```

```
;  828          0947  3          ERM (RNFINM, CH$PTR(UPLIT ('/NONSPACING_UNDERLINE')), 21);
;  829          0948  3          range_error_flag = 1;           ! Indicate error to force exit later.
;  830          0949  2          END;
;  831          0950  2
;  832          0951  2      IF .range_error_flag
;  833          0952  2      THEN                                ! If one of the above was out
;  834          0953  2          RETURN FALSE;                   ! of range, we can't go on.
```

```
;  836     0954  2 %SBTTL 'DOOPTS -- Process /BACKSPACE, /BOLD, & /UNDERLINE'
;  837     0955  2
;  838     0956  2     !Process /BACKSPACE switch'
;  839     0957  2     outopt_back = .rno_cmd [rno$v_backspace];
;  840     0958  2     outopt_over = true;                             !Can always use line overprinting
;  841     0959  2
;  842     0960  2     !Process /BOLD switch information
;  843     0961  2     gca_cmd_bld = .rno_cmd [rno$h_bold] GTR 0;  !Turn off bolding if user said /BOLD:0
;  844     0962  2     sca_do_bld = .gca_cmd_bld;
;  845     0963  2     outopt_bldn = .rno_cmd [rno$h_bold];        !Copy bolding depth indicator
;  846     0964  2
;  847     0965  2     !Process /UNDERLINE switch information.
;  848     0966  2     gca_cmd_und = .rno_cmd [rno$v_underline];
;  849     0967  2     sca_do_und = .rno_cmd [rno$v_underline];
;  850     0968  2
;  851     0969  3     IF  (.rno_cmd [rno$c_underline] EQL 0)
;  852     0970  2         AND .rno_cmd [rno$v_und_char]
;  853     0971  2     THEN                            !User said /UNDERLINE:0
;  854     0972  3         BEGIN
;  855     0973  3         gca_cmd_und = false;    !Turn off underlining for entire document.
;  856     0974  3         sca_do_und = false;     !...
;  857     0975  2         END;
;  858     0976  2
;  859     0977  2     IF .rno_cmd [rno$v_underline]
;  860     0978  2     THEN
;  861     0979  3         BEGIN
;  862     0980  3         !Pick up information about how to do underlining.
;  863     0981  3         outopt_und_sep = .rno_cmd [rno$v_und_separ];    !Put dashes on next line.
;  864     0982  3         outopt_und_nosp = .rno_cmd [rno$v_und_nonsp];   !Underline character is non-spacing.
;  865     0983  3
;  866     0984  3         outopt_und_char =
;  867     0985  4             (
;  868     0986  4             IF .rno_cmd [rno$v_und_char]
;  869     0987  4             THEN
;  870     0988  4                 !User said what character to use.
;  871     0989  5                 (.rno_cmd [rno$c_underline])
;  872     0990  4             ELSE
;  873     0991  4                 !User did not specify the underline character, so figure it
;  874     0992  4                 !out, based on how underlining is to be done.
;  875     0993  4
;  876     0994  5                 (SELECTONE true OF
;  877     0995  5                 SET
;  878     0996  5                 [.outopt_und_sep]               : %C'-';        ! Hyphen.
;  879     0997  5                 [.outopt_und_nosp]              : 7;            ! Bell.
;  880     0998  5                 [.outopt_back, .outopt_over]    : %C'_';        ! Underscore.
;  881     0999  5                 TES)
;  882     1000  3             );
;  883     1001  2         END;
;  884     1002  2
;  885     1003  2     !Turn off the following command-line emphasis options ...
;  886     1004  2     !
;  887     1005  2     !   /BACKSPACE                          /UNDERLINE="k"
;  888     1006  2     !   /NONSPACING_UNDERLINE=["k"]         /SEPARATE_UNDERLINE=["k"]
;  889     1007  2     !
;  890     1008  2     ! ... if VT100 or LN01[e] output.
;  891     1009  2     !
;  892     1010  3     IF (.gca_op_dev EQL op_dev_ln01
```

```
;  893          1011  3          OR .gca_op_dev EQL op_dev_ln01e
;  894          1012  3          OR .gca_op_dev EQL op_dev_vt100)
;  895          1013  2      THEN
;  896          1014  3          BEGIN
;  897          1015  3          outopt_und_char = 0;              ! No underline character.
;  898          1016  3          outopt_und_nosp = false;          ! No nonspacing underlining.
;  899          1017  3          outopt_und_sep = false;           ! No separate-line underlining.
;  900          1018  3          outopt_back = false;              ! Don't use backspace characters.
;  901          1019  3          outopt_bldn = 0;                  ! Don't do bolding by overprinting.
;  902          1020  2          END;
```

```
;  904          1021  2 %SBTTL 'DOOPTS -- Process /QUICK switch'
;  905          1022  2
;  906          1023  2     IF .rno_cmd [rno$v_quick]
;  907          1024  2     THEN
;  908          1025  2         setquick (false);
```

```
;  910     1026  2 %SBTTL 'DOOPTS -- Process /DEBUG switch'
;  911     1027  2
;  912     1028  2     !Process /DEBUG:INDEX switch
;  913     1029  2     gca_debug_index = .rno_cmd [rno$v_deb_index];
;  914     1030  2
;  915     1031  2     !Process /DEBUG:CONTENTS switch
;  916     1032  2     gca_debug_toc = .rno_cmd [rno$v_deb_cont];
;  917     1033  2
;  918     1034  2     !Process /DEBUG:FILES switch
;  919     1035  2     gca_debug_fil = .rno_cmd [rno$v_deb_files];
;  920     1036  2
;  921     1037  2     !Process /DEBUG:CONDITIONALS switch
;  922     1038  2     gca_debug_cnd = .rno_cmd [rno$v_deb_cond];
;  923     1039  2
;  924     1040  2     !Process /DEBUG:CROSS_REFERENCE switch
;  925     1041  2     gca_debug_cref = .rno_cmd [rno$v_deb_cros];
;  926     1042  2
;  927     1043  2     !Process /DEBUG:SAVE switch
;  928     1044  2     gca_debug_save = .rno_cmd [rno$v_deb_save];
```

```
;   930          1045  2 %SBTTL 'DOOPTS -- Process /MESSAGES switch'
;   931          1046  2
;   932          1047  2     !NOTE:  This code relies on the bit representations of REPORT_ERR_?????.
;   933          1048  2     !
;   934          1049  2 !!    IF .gca_cmd_quick
;   935          1050  2 !!    THEN
;   936          1051  2 !!      gca_cmd_msg = (1 ^ 1)           !If /QUICK, always report errors to user.
;   937          1052  2 !!    ELSE
;   938          1053  3     BEGIN
;   939          1054  3     gca_cmd_msg = .rno_cmd [rno$v_msg_out] + (.rno_cmd [rno$v_msg_user] ^ 1);
;   940          1055  3
;   941          1056  3     !If user didn't say /MESSAGES at all, direct them to everywhere
;   942          1057  3     IF .gca_cmd_msg EQL 0
;   943          1058  3     THEN                                !User didn't say /MESSAGES, so
;   944          1059  3         gca_cmd_msg = %B'11';   !direct messages everywhere.
;   945          1060  2     END;
;   946          1061  2
```

```
;  948          1062  2 %SBTTL 'DOOPTS -- Process change bar options.'
;  949          1063  2
;  950          1064  2     IF .rno_cmd [rno$v_chng_char]
;  951          1065  2     THEN                            !User specified character to be used as change bar.
;  952          1066  3         BEGIN
;  953          1067  3         sca_bar_char = .rno_cmd [rno$c_change];
;  954          1068  3
;  955          1069  3         IF .rno_cmd [rno$c_change] EQL 0
;  956          1070  3         THEN                        !User is forbidding change bars for entire document.
;  957          1071  3             gca_cmd_bar = false;
;  958          1072  2         END;
;  959          1073  2
;  960          1074  2     IF .rno_cmd [rno$v_change]
;  961          1075  2     THEN                            !User wants change bars enabled.
;  962          1076  2         bars (h_enable_bar);
```

```
;  964     1077  2 %SBTTL 'DOOPTS -- Process /RIGHT, /DOWN, /SIM, /PAUSE, & /SEQ'
;  965     1078  2
;  966     1079  2     !Process /RIGHT switch information
;  967     1080  2     phan_right = .rno_cmd [rno$h_right];
;  968     1081  2     IF .phan_right NEQ 0
;  969     1082  2     THEN
;  970     1083  2         gca_cmd_rit = true;
;  971     1084  2
;  972     1085  2     !Process /DOWN switch information
;  973     1086  2     phan_down = .rno_cmd [rno$h_down];
;  974     1087  2 !zzz
;  975     1088  2     !Process /SIMULATE switch information
;  976     1089  2     IF .rno_cmd [RNO$V_SIMULATE]
;  977     1090  2     THEN                          !User said /SIMULATE
;  978     1091  2         phan_simulate = true
;  979     1092  2     ELSE
;  980     1093  2         phan_simulate = false;
;  981     1094  2
;  982     1095  2     !Process /PAUSE switch information
;  983     1096  2     phan_pause = .rno_cmd [rno$v_pause];
;  984     1097  2
;  985     1098  2     !Process /SEQUENCE switch information
;  986     1099  2     gca_cmd_isq = .rno_cmd [rno$v_sequence];
```

```
;  988          1100  2 %SBTTL 'DOOPTS -- Process /LOG, /DEVICE, /DEC_INTERNAL switches'
;  989          1101  2
;  990          1102  2     ! /[NO]LOG switch
;  991          1103  2
;  992          1104  2         termination_log = .rno_cmd [rno$v_log];
;  993          1105  2
;  994          1106  2     ! /DEVICE switch
;  995          1107  2         !
;  996          1108  2         ! NOTE: the output device type was already picked up in RUNOFF.BLI:
;  997          1109  2         !       gca_op_dev = .rno_cmd [rno$v_4_out_format];
;  998          1110  2         !
;  999          1111  2         !Pick up LN01 output options.
; 1000          1112  2         IF (.gca_op_dev EQL op_dev_ln01
; 1001          1113  2           OR .gca_op_dev EQL op_dev_ln01e)
; 1002          1114  2         THEN
; 1003          1115  3             BEGIN                                       ! Options for LN01 output:
; 1004          1116  3             gca_ln01_ital_under = .rno_cmd [rno$v_ln01_ital_under]; ! set=italics, clear=underlining
; 1005          1117  3             gca_ln01_port_land = .rno_cmd [rno$v_ln01_port_land];   ! set=portrait, clear=landscape
; 1006          1118  3             !
; 1007          1119  3             ! Write initial escape sequences into the .LNI output file.
; 1008          1120  3             !
; 1009          1121  3             write_ln01_info (.rno_cmd);
; 1010          1122  3             END;
; 1011          1123  2
; 1012        U 1124  2 %IF FLIP %THEN
; 1013        U 1125  2         IF (.gca_op_dev EQL op_dev_flip)
; 1014        U 1126  2         THEN
; 1015        U 1127  2             BEGIN
; 1016        U 1128  2             !
; 1017        U 1129  2             ! Initialize the .BFL with a "new sequence" header.
; 1018        U 1130  2             !
; 1019        U 1131  2             putrty (maj_new_toc, toc_format);
; 1020        U 1132  2
; 1021        U 1133  2             ! Turn on flags to indicate that we want Index and Contents
; 1022        U 1134  2             ! information for FLIP output.
; 1023        U 1135  2             !
; 1024        U 1136  2             gca_bix = true;
; 1025        U 1137  2             gca_btc = true;
; 1026        U 1138  2             gca_cmd_btc = true;
; 1027        U 1139  2             END;
; 1028          1140  2 %FI
; 1029          1141  2     ! /DEC_INTERNAL switch
; 1030          1142  2
; 1031          1143  2         !Pick up the debugging flags
; 1032          1144  2         gca_diag1       =  .rno_cmd [rno$h_dbg1];
; 1033          1145  2         gca_diag2       =  .rno_cmd [rno$h_dbg2];
; 1034          1146  2
; 1035          1147  2         ! If the user said /DEC=OUTPUT_LINE_NUMBER, the CLI set bit 15 of
; 1036          1148  2         ! the second diagnostic word.  Use that word to initialize the flag
; 1037          1149  2         ! that controls the outputting of the output line numbers at the
; 1038          1150  2         ! front of every line of text.
; 1039          1151  2         gca_cmd_osq     =  .diag2_15;
```

```
;  1041          1152  2 %SBTTL 'DOOPTS -- Process /FORMSIZE switch information'
;  1042          1153  2
;  1043          1154  2     IF .rno_cmd [rno$h_form_size] GTR 0
;  1044          1155  2     THEN
;  1045          1156  2         !User said /FORMSIZE:n.  Use it either for /SIMULATE or
;  1046          1157  2         !/NOSIMULATE, as appropriate.
;  1047          1158  2         IF .phan_simulate
;  1048          1159  2         THEN
;  1049          1160  2             !User said /SIMULATE, so the specified form size is physical paper size
;  1050          1161  2             phan_plines = .rno_cmd [rno$h_form_size]
;  1051          1162  2         ELSE
;  1052          1163  2             !User is not simulating, so specified form size is
;  1053          1164  2             !number of lines allowed on the page by the spooler.
;  1054          1165  2             phan_slines = .rno_cmd [rno$h_form_size];
;  1055          1166  2
;  1056          1167  2     !If simulating formfeeds, or want to pause at top of each page, open
;  1057          1168  2     !the stream IOBs.
;  1058          1169  2     IF .phan_simulate OR .phan_pause
;  1059          1170  2     THEN
;  1060        P 1171  2         $XPO_OPEN (IOB = tsiiob,
;  1061        P 1172  2             FILE_SPEC = $XPO_INPUT,
;  1062        P 1173  2             OPTIONS = (INPUT, OUTPUT),
;  1063          1174  2             ATTRIBUTES = STREAM);
;  1064          1175  2
;  1065          1176  2     !Initial signals for /SIMULATE and /PAUSE
;  1066          1177  2
;  1067          1178  2     IF .phan_simulate
;  1068          1179  2     THEN
;  1069          1180  3         BEGIN                               ! Do not prompt in Batch (if
;  1070          1181  3         IF .tsiiob [iob$v_terminal]         ! controller is not a terminal).
;  1071          1182  3         THEN
;  1072          1183  4             BEGIN
;  1073        P 1184  4             $XPO_GET
;  1074        P 1185  4                 ( IOB = tsiiob
;  1075        P 1186  4                   , PROMPT= ( 32
;  1076        L 1187  4                             , CH$PTR(UPLIT (%STRING
;  1077        L 1188  4                                             (BELL, DEL, BELL, DEL,
;  1078        L 1189  4                                              'Position paper, type a space'
;  1079        P 1190  4                                              )
;  1080        P 1191  4                                             )
;  1081        P 1192  4                                       )
;  1082        P 1193  4                             )
;  1083        P 1194  4                   , CHARACTERS = 1
;  1084          1195  4                 );
;  1085          1196  4
;  1086          1197  4             !After getting the user's character, issue a carriage return.
;  1087        P 1198  4             $XPO_PUT
;  1088        P 1199  4                 ( IOB = tsiiob
;  1089        P 1200  4                 , STRING = (1, CH$PTR(UPLIT (%STRING (%CHAR(%O'15')))))
;  1090          1201  5                 )
;  1091          1202  3             END;
;  1092          1203  3         END
;  1093          1204  2     ELSE
;  1094          1205  2         IF .phan_pause
;  1095          1206  2         THEN
;  1096          1207  2             bwait ();
;  1097          1208  2
```

```
: 1098          1209  2     RETURN TRUE                                     !Command line was ok.
: 1099          1210  1     END;                                            !End of DOOPTS


                                                                            .TITLE  DOOPTS Digests and distributes command line inf
:                                                                                   ormatio
                                                                            .IDENT  \V04-000\

                                                                            .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    48  4E  52  2E  00000 P.AAA:  .ASCII  \.RNH\                                  ;
                                                    4E  52  42  2E  00004 P.AAB:  .ASCII  \.BRN\                                  ;
                                        00  00  00  44  4C  4F  42  2F  00008 P.AAC:  .ASCII  \/BOLD\<0><0><0>                     ;
                                        00  00  00  4E  57  4F  44  2F  00010 P.AAD:  .ASCII  \/DOWN\<0><0><0>                     ;
                                00  00  45  5A  49  53  5F  4D  52  4F  46  2F  00018 P.AAE:  .ASCII  \/FORM_SIZE\<0><0>             ;
                                        00  00  54  48  47  49  52  2F  00024 P.AAF:  .ASCII  \/RIGHT\<0><0>                       ;
                                00  00  45  4E  49  4C  52  45  44  4E  55  2F  0002C P.AAG:  .ASCII  \/UNDERLINE\<0><0>           ;
52  45  44  4E  55  5F  45  54  41  52  41  50  45  53  2F  00038 P.AAH:  .ASCII  \/SEPARATE_UNDERLINE\<0>                        ;
                                                        00  45  4E  49  4C  00047
44  4E  55  5F  47  4E  49  43  41  50  53  4E  4F  4E  2F  0004C P.AAI:  .ASCII  \/NONSPACING_UNDERLINE\<0><0><0>                ;
                                    00  00  00  45  4E  49  4C  52  45  0005B
                                    54  55  50  4E  49  24  53  59  53  00064 P.AAJ:  .ASCII  \SYS$INPUT\                           ;
                                                                        0006D       .BLKB   3
61  70  20  6E  6F  69  74  69  73  6F  50  7F  07  7F  07  00070 P.AAO:  .ASCII  <7><127><7><127>\Position paper, \               ;
                                                20  2C  72  65  70  0007F
                    65  63  61  70  73  20  61  20  65  70  79  74  00084       .ASCII  \type a space\                              ;
                                                            00  00  00  0D  00090 P.AAS:  .ASCII  <13><0><0><0>                     ;

                                                                            .PSECT  $OWN$,NOEXE,2

                                                                        00000 U_EXT:    .BLKB   4
                                                                        00004 U_EXT_PTR:
                                                                                .BLKB   4
                                                                        00008 TYPE_LENGTH:
                                                                                .BLKB   4
                                                                        0000C TYPE_PTR:
                                                                                .BLKB   4
                                                                        00010 PRSE_SPEC_BLOCK:
                                                                                .BLKB   72
                                                                        00058 RANGE_ERROR_FLAG:
                                                                                .BLKB   4
                                                                    0004  0005C $IOB$DEFAULT:
                                                                                .WORD   4                                           ;
                                                                  01  0E  0005E         .BYTE   14, 1                               ;
                                                                00000000' 00060         .ADDRESS P.AAB                              ;
                                                                    0009  00064 $IOB$FILE_SPEC:
                                                                                .WORD   9                                           ;
                                                                  01  0E  00066         .BYTE   14, 1                               ;
                                                                00000000' 00068         .ADDRESS P.AAJ                              ;

                                                            TEMP=                               PRSE_SPEC_BLOCK+40
                                                                                .EXTRN  RINTES, RNFCEM, RNFCOB
                                                                                .EXTRN  RNFINM, RNFIVS, RNFTMP
                                                                                .EXTRN  RNFTMV, FS01, FRA
                                                                                .EXTRN  GCA, HCT, KHAR, SPAGER
                                                                                .EXTRN  TPAGER, RNOIOB, RNIIOB
                                                                                .EXTRN  BRNIOB, BRNOOB, FFNAME
```

```
                                                        .EXTRN  SEMCOD, OUTOPT, PHAN
                                                        .EXTRN  SCA, TSIIOB, VRCNT
                                                        .EXTRN  BARS, BWAIT, CLH
                                                        .EXTRN  ERM, ERME, GRAB_RESULTANT
                                                        .EXTRN  GNAME, PARSEP, PUTRTY
                                                        .EXTRN  RGH, VRENTR, VRFIND
                                                        .EXTRN  XPO$PARSE_SPEC, XPO$FAILURE
                                                        .EXTRN  XPO$OPEN, XPO$PUT
                                                        .EXTRN  XST$FORMAT, XST$FREE_TEMP
                                                        .EXTRN  XPO$GET

                                                        .PSECT  $CODE$,NOWRT,2

                                        0FFC 00000      .ENTRY  DOOPTS, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-      ; 0418
                                                                R11
                5B 00000000G   EF   9E 00002            MOVAB   FS01, R11                                       ;
                5A 00000000G   EF   9E 00009            MOVAB   IOB$+44, R10                                    ;
                59 00000000'   EF   9E 00010            MOVAB   RANGE_ERROR_FLAG, R9                            ;
                58 00000000G   EF   9E 00017            MOVAB   GCA+208, R8                                     ;
                5E                  10  C2 0001E       SUBL2   #16, SP                                         ;
                                    69  D4 00021        CLRL    RANGE_ERROR_FLAG                                ; 0455
      50 00000000G   EF             1C  C1 00023        ADDL3   #28, RNIIOB, R0                                 ; 0462
                        00000000G   EF  9F 0002B        PUSHAB  XPO$FAILURE                                     ;
                                    7E  D4 00031        CLRL    -(SP)                                           ;
                7E                  01  CE 00033        MNEGL   #1, -(SP)                                       ;
                           B8       A9  9F 00036        PUSHAB  PRSE_SPEC_BLOCK                                 ;
                                    50  DD 00039        PUSHL   R0                                              ;
      00000000G   EF                05  FB 0003B        CALLS   #5, XPO$PARSE_SPEC                              ;
            B0    A9         E0    A9  3C 00042        MOVZWL  TEMP, TYPE_LENGTH                               ; 0468
            B4    A9         E4    A9  D0 00047        MOVL    TEMP+4, TYPE_PTR                                ; 0469
                  04         B0    A9  D1 0004C        CMPL    TYPE_LENGTH, #4                                 ; 0472
                                    7E  12 00050        BNEQ    6$                                              ;
                  AC   A9         A8  A9 9E 00052       MOVAB   U_EXT, U_EXT_PTR                                ; 0477
                       53         01  D0 00057        MOVL    #1, I                                           ; 0479
                       51         B4  B9 9A 0005A 1$:   MOVZBL  @TYPE_PTR, KAHR                                 ; 0485
                                  B4  A9 D6 0005E       INCL    TYPE_PTR
            00000061   8F             51  D1 00061        CMPL    KAHR, #97                                       ; 0487
                                      2D  19 00068        BLSS    4$                                              ;
            0000007A   8F             51  D1 0006A        CMPL    KAHR, #122                                      ;
                                      24  14 00071        BGTR    4$                                              ;
                                      52  D4 00073        CLRL    R2                                              ; 0489
            00000041   8F             51  D1 00075        CMPL    KAHR, #65                                       ;
                                      02  19 0007C        BLSS    2$                                              ;
                                      52  D6 0007E        INCL    R2                                              ;
                                      50  D4 00080 2$:     CLRL    R0                                              ;
            0000005A   8F             51  D1 00082        CMPL    KAHR, #90                                       ;
                                      02  14 00089        BGTR    3$                                              ;
                                      50  D6 0008B        INCL    R0                                              ;
                       54             52  D2 0008D 3$:     MCOML   R2, R4                                          ;
      AC    B9         50             54  8B 00090        BICB3   R4, R0, @U_EXT_PTR                              ;
                                      04  11 00095        BRB     5$                                              ; 0491
                  AC   B9             51  90 00097 4$:     MOVB    KAHR, @U_EXT_PTR                                ;
                              AC      A9  D6 0009B 5$:     INCL    U_EXT_PTR                                       ; 0489
            B8                53      04  F3 0009E        AOBLEQ  #4, I, 1$                                       ; 0491
                  AC   A9         A8  A9 9E 000A2       MOVAB   U_EXT, U_EXT_PTR                                ; 0494
            00000000'  EF         AC  B9 D1 000A7       CMPL    @U_EXT_PTR, P.AAA                               ; 0496
                                      1F  12 000AF        BNEQ    6$                                              ;
```

```
                                00000000G   FF   D4 000B1          CLRL    @HCT+8                                  ; 0500
                                00000000G   FF   D4 000B7          CLRL    @PHAN+40                                ; 0501
                                00000000G   EF   D4 000BD          CLRL    PHAN+44                                 ; 0502
                     00000000G  FF         48   8F 9A 000C3       MOVZBL  #72, @SCA+120                           ; 0503
                           BC   B8         48   8F 9A 000CB       MOVZBL  #72, @GCA+140                           ; 0504
                                54         04   AC D0 000D0 6$:    MOVL    RNO_CMD, R4                             ; 0510
                                                55 D4 000D4       CLRL    R5                                      ;
                                           30   A4 B5 000D6       TSTW    48(R4)                                  ;
                                                03 12 000D9       BNEQ    7$                                      ;
                                             013F 31 000DB        BRW     26$                                     ;
                                                55 D6 000DE 7$:    INCL    R5                                      ;
                                           6E   34   A4 D0 000E0  MOVL    52(R4), IRA                             ; 0523
                                     04    AE         6E D0 000E4 MOVL    IRA, IRA+4                              ; 0524
                                   FF30    C8         6E D0 000E8 MOVL    IRA, GCA                                ; 0525
                                     08    AE         30 A4 3C 000ED MOVZWL  48(R4), IRA+8                        ; 0528
                                     0C    AE         08 AE D0 000F2 MOVL    IRA+8, IRA+12                        ; 0529
                                                      0C AB D4 000F7 CLRL    FS01+12                              ; 0533
                                           6B         10 AB 9E 000FA MOVAB   FS01+16, FS01                        ;
                                     04    AB         6B D0 000FE MOVL    FS01, FS01+4                            ;
                                                      0C AE D5 00102 TSTL    IRA+12                               ; 0534
                                                      0E 14 00105 BGTR    8$                                      ;
                     00000000G     EF         00G    8F 9A 00107 MOVZBL  #RINTES, KHAR                          ;
                           0C      AE                01 CE 0010F MNEGL   #1, IRA+12                              ;
                                                      0E 11 00113 BRB     9$                                      ;
                     00000000G     EF         04     BE 9A 00115 8$:    MOVZBL  @IRA+4, KHAR                    ;
                                              04     AE D6 0011D       INCL    IRA+4                               ;
                                              0C     AE D7 00120       DECL    IRA+12                              ;
                                                      52 D4 00123 9$:    CLRL    I                                   ; 0537
                  53                  52              04 78 00125 10$:   ASHL    #4, I, R3                           ; 0540
                                00000000GEF43         9F 00129          PUSHAB  SPAGER[R3]                          ;
                                              04     AE 9F 00130       PUSHAB  IRA                                 ;
                     00000000G  EF                     02 FB 00133       CALLS   #2, PARSEP                          ;
                                66                     50 E9 0013A       BLBC    R0, 15$                             ;
                                51 00000000GEF43       9E 0013D          MOVAB   TPAGER[R3], R1                      ; 0548
                                                      50 D4 00145       CLRL    I                                   ; 0550
                                                   6140 D4 00147 11$:    CLRL    (R1)[I]                             ; 0551
                  F9                  50              03 F3 0014A       AOBLEQ  #3, I, 11$                          ;
                                9C    A8         01     A2 9E 0014E       MOVAB   1(R2), GCA+108                      ; 0553
                                50 00000000G     EF     D0 00153          MOVL    KHAR, R0                            ; 0555
                     00000000G  8F                     50 D1 0015A       CMPL    R0, #RINTES                         ;
                                                      54 13 00161       BEQL    17$                                 ;
                                3A                    50 D1 00163       CMPL    R0, #58                             ; 0564
                                                      09 13 00166       BEQL    12$                                 ;
                     0000007C   8F                     50 D1 00168       CMPL    R0, #124                            ;
                                                      38 12 0016F       BNEQ    16$                                 ;
                                              0C     AE D5 00171 12$:   TSTL    IRA+12                              ; 0569
                                              0E     14 00174          BGTR    13$                                 ;
                     00000000G  EF         00G    8F 9A 00176       MOVZBL  #RINTES, KHAR                       ;
                           0C   AE                01 CE 0017E       MNEGL   #1, IRA+12                          ;
                                                      0E 11 00182       BRB     14$                                 ;
                     00000000G  EF         04     BE 9A 00184 13$:   MOVZBL  @IRA+4, KHAR                    ;
                                              04     AE D6 0018C       INCL    IRA+4                               ;
                                              0C     AE D7 0018F       DECL    IRA+12                              ;
                                00000000GEF43         9F 00192 14$:     PUSHAB  TPAGER[R3]                          ; 0571
                                              04     AE 9F 00199       PUSHAB  IRA                                 ;
                     00000000G  EF                     02 FB 0019C       CALLS   #2, PARSEP                          ;
                                03                     50 E8 001A3 15$:   BLBS    R0, 16$                             ;
```

```
                                          06E4   31 001A6          BRW      88$                        ;
                      50 00000000G   EF          D0 001A9  16$:    MOVL     KHAR, R0                   ; 0578
             00000000G   8F               50     D1 001B0          CMPL     R0, #RINTES                ;
                                          41     13 001B7  17$:    BEQL     22$                        ;
                         2C               50     D1 001B9          CMPL     R0, #44                    ; 0583
                                          05     13 001BC          BEQL     18$                        ;
                         3B               50     D1 001BE          CMPL     R0, #59                    ;
                                          23     12 001C1          BNEQ     20$                        ;
                                   0C     AE     D5 001C3  18$:    TSTL     IRA+12                     ; 0587
                                          0E     14 001C6          BGTR     19$                        ;
             00000000G   EF       00G     8F     9A 001C8          MOVZBL   #RINTES, KHAR              ;
                    0C   AE               01     CE 001D0          MNEGL    #1, IRA+12                 ;
                                          1E     11 001D4          BRB      21$                        ;
             00000000G   EF       04      BE     9A 001D6  19$:    MOVZBL   @IRA+4, KHAR               ;
                                  04      AE     D6 001DE          INCL     IRA+4                      ;
                                  0C      AE     D7 001E1          DECL     IRA+12                     ;
                                  0E      11        001E4          BRB      21$                        ;
                                  08      AE     DD 001E6  20$:    PUSHL    IRA+8                      ; 0591
                                  04      AE     DD 001E9          PUSHL    IRA                        ;
                         00000000G        8F     DD 001EC          PUSHL    #RNFCEM                    ;
                                          1B     11 001F2          BRB      23$                        ;
      FF2B           52           01      04     F1 001F4  21$:    ACBL     #4, #1, I, 10$             ; 0540
             00000000G   8F 00000000G     EF     D1 001FA  22$:    CMPL     KHAR, #RINTES              ; 0604
                                          0B     13 00205          BEQL     24$                        ;
                                          7E     7C 00207          CLRQ     -(SP)                      ; 0608
                         00000000G        8F     DD 00209          PUSHL    #RNFTMP                    ;
                                        023A     31 0020F  23$:    BRW      44$                        ;
                         04               55     E9 00212  24$:    BLBC     R5, 25$                    ; 0612
                    01   A8               04     88 00215          BISB2    #4, GCA+209                ; 0614
                    A0   A8               01     D0 00219  25$:    MOVL     #1, GCA+112                ; 0615
                         55       4C      A4     9E 0021D  26$:    MOVAB    76(R4), R5                 ; 0724
                         03       02      A5     E8 00221          BLBS     2(R5), 28$                 ;
                                        014E     31 00225  27$:    BRW      30$                        ;
       02            65           04      16     ED 00228  28$:    CMPZV    #22, #4, (R5), #2          ; 0726
                                          F6     13 0022D          BEQL     27$                        ;
                      50 00000000G   EF   1C     C1 0022F          ADDL3    #28, RNIIOB, R0            ; 0744
                    D8   AA       10      A4     9E 00237          MOVAB    16(R4), ICB$+4             ;
                    DC   AA       04      A9     9E 0023C          MOVAB    $IOB$DEFAULT, IOB$+8       ;
                    E0   AA               50     D0 00241          MOVL     R0, IOB$+12                ;
                    02   AA 00010002      8F     C8 00245          BISL2    #65538, IOB$+46            ;
                         6A               01     90 0024D          MOVB     #1, IOB$+44                ;
                         00000000G        EF     9F 00250          PUSHAB   GRAB_RESULTANT             ;
                                          7E     D4 00256          CLRL     -(SP)                      ;
                                  D4      AA     9F 00258          PUSHAB   IOB$                       ;
             00000000G   EF               03     FB 0025B          CALLS    #3, XPO$OPEN               ;
                         23               50     E8 00262          BLBS     STATUS, 29$                ; 0746
                         00000000G        EF     DD 00265          PUSHL    SEMCOD                     ; 0748
                      7E 00000000G        EF     3C 0026B          MOVZWL   TEMP, -(SP)                ;
                         00000000G        EF     DD 00272          PUSHL    TEMP+4                     ;
                         00000000G        8F     DD 00278          PUSHL    #RNFCOB                    ;
             00000000G   EF               04     FB 0027E          CALLS    #4, ERME                   ;
                                        0605     31 00285          BRW      88$                        ; 0749
                    08   AE               01     CE 00288  29$:    MNEGL    #1, TEMP                   ; 0763
                    0C   AE               02     D0 0028C          MOVL     #2, TEMP+4                 ; 0764
                         6E               08     B0 00290          MOVW     #8, $IOB$OUTPUT            ; 0767
                    02   AE               02     90 00293          MOVB     #2, $IOB$OUTPUT+2          ;
                    03   AE               01     90 00297          MOVB     #1, $IOB$OUTPUT+3          ;
```

```
                04   AE        08   AE   9E 0029B          MOVAB   TEMP, $IOB$OUTPUT+4                                  ;
                18   AA             6E   9E 002A0          MOVAB   $IOB$OUTPUT, IOB$+68                                 ;
                     6A             07   90 002A4          MOVB    #7, IOB$+44                                          ;
                          00000000G  EF   9F 002A7          PUSHAB  XPO$FAILURE                                          ;
                                     7E   D4 002AD          CLRL    -(SP)                                                ;
                                D4   AA   9F 002AF          PUSHAB  IOB$                                                 ;
          00000000G  EF              03   FB 002B2          CALLS   #3, XPO$PUT                                          ;
                     7E              01   7D 002B9          MOVQ    #1, -(SP)                                            ; 0771
                                     01   DD 002BC          PUSHL   #1                                                   ;
          00000000G  EF              03   FB 002BE          CALLS   #3, RGH                                              ;
                08   AE  00020001    8F   D0 002C5          MOVL    #131073, TEMP                                        ; 0774
                     6E              04   B0 002CD          MOVW    #4, $IOB$OUTPUT                                      ; 0776
                02   AE              02   90 002D0          MOVB    #2, $IOB$OUTPUT+2                                    ;
                03   AE              01   90 002D4          MOVB    #1, $IOB$OUTPUT+3                                    ;
                04   AE        08   AE   9E 002D8          MOVAB   TEMP, $IOB$OUTPUT+4                                  ;
                18   AA             6E   9E 002DD          MOVAB   $IOB$OUTPUT, IOB$+68                                 ;
                     6A             07   90 002E1          MOVB    #7, IOB$+44                                          ;
                          00000000G  EF   9F 002E4          PUSHAB  XPO$FAILURE                                          ;
                                     7E   D4 002EA          CLRL    -(SP)                                                ;
                                D4   AA   9F 002EC          PUSHAB  IOB$                                                 ;
          00000000G  EF              03   FB 002EF          CALLS   #3, XPO$PUT                                          ;
                     7E              02   7D 002F6          MOVQ    #2, -(SP)                                            ; 0780
                                     02   DD 002F9          PUSHL   #2                                                   ;
          00000000G  EF              03   FB 002FB          CALLS   #3, RGH                                              ;
                08   AE              01   D0 00302          MOVL    #1, TEMP                                             ; 0783
                0C   AE              03   D0 00306          MOVL    #3, TEMP+4                                           ; 0784
                     6E              08   B0 0030A          MOVW    #8, $IOB$OUTPUT                                      ; 0786
                02   AE              02   90 0030D          MOVB    #2, $IOB$OUTPUT+2                                    ;
                03   AE              01   90 00311          MOVB    #1, $IOB$OUTPUT+3                                    ;
                04   AE        08   AE   9E 00315          MOVAB   TEMP, $IOB$OUTPUT+4                                  ;
                18   AA             6E   9E 0031A          MOVAB   $IOB$OUTPUT, IOB$+68                                 ;
                     6A             07   90 0031E          MOVB    #7, IOB$+44                                          ;
                          00000000G  EF   9F 00321          PUSHAB  XPO$FAILURE                                          ;
                                     7E   D4 00327          CLRL    -(SP)                                                ;
                                D4   AA   9F 00329          PUSHAB  IOB$                                                 ;
          00000000G  EF              03   FB 0032C          CALLS   #3, XPO$PUT                                          ;
                     7E              02   7D 00333          MOVQ    #2, -(SP)                                            ; 0790
                                     03   DD 00336          PUSHL   #3                                                   ;
          00000000G  EF              03   FB 00338          CALLS   #3, RGH                                              ;
                08   AE              01   D0 0033F          MOVL    #1, TEMP                                             ; 0793
                0C   AE              01   D0 00343          MOVL    #1, TEMP+4                                           ; 0794
                     6E              08   B0 00347          MOVW    #8, $IOB$OUTPUT                                      ; 0796
                02   AE              02   90 0034A          MOVB    #2, $IOB$OUTPUT+2                                    ;
                03   AE              01   90 0034E          MOVB    #1, $IOB$OUTPUT+3                                    ;
                04   AE        08   AE   9E 00352          MOVAB   TEMP, $IOB$OUTPUT+4                                  ;
                18   AA             6E   9E 00357          MOVAB   $IOB$OUTPUT, IOB$+68                                 ;
                     6A             07   90 0035B          MOVB    #7, IOB$+44                                          ;
                          00000000G  EF   9F 0035E          PUSHAB  XPO$FAILURE                                          ;
                                     7E   D4 00364          CLRL    -(SP)                                                ;
                                D4   AA   9F 00366          PUSHAB  IOB$                                                 ;
          00000000G  EF              03   FB 00369          CALLS   #3, XPO$PUT                                          ;
                AC   A8              07   88 00370          BISB2   #7, GCA+124                                          ; 0802
                                     04   11 00374          BRB     31$                                                  ; 0724
                AC   A8              07   8A 00376 30$:     BICB2   #7, GCA+124                                          ; 0809
                          38         A4   B5 0037A 31$:     TSTW    56(R4)                                               ; 0834
                                     38   13 0037D          BEQL    34$                                                  ;
                     6E        3C   A4   D0 0037F          MOVL    60(R4), IRA                                          ; 0846
```

```
                04   AE            6E  D0 00383          MOVL    IRA, IRA+4                          ; 0847
                08   AE        38  A4  3C 00387          MOVZWL  56(R4), IRA+8                       ; 0848
                0C   AE        08  AE  D0 0038C          MOVL    IRA+8, IRA+12                       ; 0849
                                   0B  14 00391          BGTR    32$                                 ; 0850
        00000000G   EF        00G  8F  9A 00393          MOVZBL  #RINTES, KHAR
                                0096  31 0039B          BRW     40$
        00000000G   EF        04   BE  9A 0039E 32$:     MOVZBL  @IRA+4, KHAR
                              04   AE  D6 003A6          INCL    IRA+4
                              0C   AE  D7 003A9          DECL    IRA+12
        00000000G   8F 00000000G   EF  D1 003AC 33$:     CMPL    KHAR, #RINTES                       ; 0852
                                   03  12 003B7 34$:     BNEQ    35$
                                 009A  31 003B9          BRW     45$
                              0C   AB  D4 003BC 35$:     CLRL    FS01+12                             ; 0854
                        6B    10   AB  9E 003BF          MOVAB   FS01+16, FS01
                04   AB            6B  D0 003C3          MOVL    FS01, FS01+4
                                   5B  DD 003C7          PUSHL   R11                                 ; 0856
                              04   AE  9F 003C9          PUSHAB  IRA
        00000000G   EF             02  FB 003CC          CALLS   #2, GNAME
                    01             50  D1 003D3          CMPL    R0, #1
                                   6C  12 003D6          BNEQ    43$
                    14 00000000G   EF  D1 003D8          CMPL    VRCNT, #20                          ; 0863
                                   0A  19 003DF          BLSS    36$
                                   7E  7C 003E1          CLRQ    -(SP)                               ; 0866
                       00000000G   8F  DD 003E3          PUSHL   #RNFTMV
                                   61  11 003E9          BRB     44$
                              0C   AB  DD 003EB 36$:     PUSHL   FS01+12                             ; 0870
                                   6B  DD 003EE          PUSHL   FS01
        00000000G   EF             02  FB 003F0          CALLS   #2, VRFIND
        FFFFFFFF    8F             50  D1 003F7          CMPL    R0, #-1
                                   13  12 003FE          BNEQ    37$
                                   01  DD 00400          PUSHL   #1                                  ; 0872
                    7E             20  7D 00402          MOVQ    #32, -(SP)
                                   20  DD 00405          PUSHL   #32
                              0C   AB  DD 00407          PUSHL   FS01+12
                                   6B  DD 0040A          PUSHL   FS01
        00000000G   EF             06  FB 0040C          CALLS   #6, VRENTR
                    50 00000000G   EF  D0 00413 37$:     MOVL    KHAR, R0                            ; 0877
                    2C             50  D1 0041A          CMPL    R0, #44
                                   05  13 0041D          BEQL    38$
                    3B             50  D1 0041F          CMPL    R0, #59
                                   17  12 00422          BNEQ    42$
                              0C   AE  D5 00424 38$:     TSTL    IRA+12                              ; 0879
                                   03  15 00427          BLEQ    39$
                                 FF72  31 00429          BRW     32$
        00000000G   EF        00G  8F  9A 0042C 39$:     MOVZBL  #RINTES, KHAR
                04   AE            01  CE 00434 40$:     MNEGL   #1, IRA+12
                                 FF71  31 00438 41$:     BRW     33$
        00000000G   8F             50  D1 0043B 42$:     CMPL    R0, #RINTES                         ; 0881
                                   F4  13 00442          BEQL    41$
                                   7E  7C 00444 43$:     CLRQ    -(SP)                               ; 0885
                       00000000G   8F  DD 00446          PUSHL   #RNFIVS
        00000000G   EF             03  FB 0044C 44$:     CALLS   #3, ERM
                                 0437  31 00453          BRW     88$                                 ; 0886
                    52        54   A4  9E 00456 45$:     MOVAB   84(R4), R2                          ; 0892
                              02   A2  B5 0045A          TSTW    2(R2)
                                   06  19 0045D          BLSS    46$
                    0A        02   A2  B1 0045F          CMPW    2(R2), #10                          ; 0893
```

```
                                  18 15 00463        BLEQ    47$                          ;
                                  05 DD 00465 46$:   PUSHL   #5                           ; 0896
                       00000000'  EF 9F 00467        PUSHAB  P.AAC                        ;
                       00000000G  8F DD 0046D        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 00473        CALLS   #3, ERM                      ;
                     69           01 D0 0047A        MOVL    #1, RANGE_ERROR_FLAG         ; 0897
                     57       5C  A4 32 0047D 47$:   CVTWL   92(R4), R7                   ; 0900
                                  07 19 00481        BLSS    48$                          ;
               00C8  8F           57 B1 00483        CMPW    R7, #200                     ; 0901
                                  18 15 00488        BLEQ    49$                          ;
                                  05 DD 0048A 48$:   PUSHL   #5                           ; 0904
                       00000000'  EF 9F 0048C        PUSHAB  P.AAD                        ;
                       00000000G  8F DD 00492        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 00498        CALLS   #3, ERM                      ;
                     69           01 D0 0049F        MOVL    #1, RANGE_ERROR_FLAG         ; 0905
                              60  A4 B5 004A2 49$:   TSTW    96(R4)                       ; 0908
                                  08 19 004A5        BLSS    50$                          ;
               00C8  8F       60  A4 B1 004A7        CMPW    96(R4), #200                 ; 0909
                                  18 15 004AD        BLEQ    51$                          ;
                                  0A DD 004AF 50$:   PUSHL   #10                          ; 0912
                       00000000'  EF 9F 004B1        PUSHAB  P.AAE                        ;
                       00000000G  8F DD 004B7        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 004BD        CALLS   #3, ERM                      ;
                     69           01 D0 004C4        MOVL    #1, RANGE_ERROR_FLAG         ; 0913
                     56       5E  A4 32 004C7 51$:   CVTWL   94(R4), R6                   ; 0916
                                  07 19 004CB        BLSS    52$                          ;
               0096  8F           56 B1 004CD        CMPW    R6, #150                     ; 0917
                                  18 15 004D2        BLEQ    53$                          ;
                                  06 DD 004D4 52$:   PUSHL   #6                           ; 0920
                       00000000'  EF 9F 004D6        PUSHAB  P.AAF                        ;
                       00000000G  8F DD 004DC        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 004E2        CALLS   #3, ERM                      ;
                     69           01 D0 004E9        MOVL    #1, RANGE_ERROR_FLAG         ; 0921
                     53       50  A4 9E 004EC 53$:   MOVAB   80(R4), R3                   ; 0924
          2A         63           06 E1 004F0        BBC     #6, (R3), 55$                ;
                              01  A2 95 004F4        TSTB    1(R2)                        ;
                                  25 13 004F7        BEQL    55$                          ;
                     20       01  A2 91 004F9        CMPB    1(R2), #32                   ; 0925
                                  07 1F 004FD        BLSSU   54$                          ;
                 7D  8F       01  A2 91 004FF        CMPB    1(R2), #125                  ; 0926
                                  18 1B 00504        BLEQU   55$                          ;
                                  0A DD 00506 54$:   PUSHL   #10                          ; 0929
                       00000000'  EF 9F 00508        PUSHAB  P.AAG                        ;
                       00000000G  8F DD 0050E        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 00514        CALLS   #3, ERM                      ;
                     69           01 D0 0051B        MOVL    #1, RANGE_ERROR_FLAG         ; 0930
                     2A       01  A3 E9 0051E 55$:   BLBC    1(R3), 57$                   ; 0933
                              01  A2 95 00522        TSTB    1(R2)                        ;
                                  25 13 00525        BEQL    57$                          ;
                     20       01  A2 91 00527        CMPB    1(R2), #32                   ; 0934
                                  07 1F 0052B        BLSSU   56$                          ;
                 7D  8F       01  A2 91 0052D        CMPB    1(R2), #125                  ; 0935
                                  18 1B 00532        BLEQU   57$                          ;
                                  13 DD 00534 56$:   PUSHL   #19                          ; 0938
                       00000000'  EF 9F 00536        PUSHAB  P.AAH                        ;
                       00000000G  8F DD 0053C        PUSHL   #RNFINM                      ;
          00000000G  EF           03 FB 00542        CALLS   #3, ERM                      ;
```

```
                                      69         01  D0 00549        MOVL    #1, RANGE_ERROR_FLAG                ; 0939
                    23                63         0A  E1 0054C  57$:  BBC     #10, (R3), 58$                      ; 0942
                                           01    A2  95 00550        TSTB    1(R2)
                                                 1E  13 00553        BEQL    58$
                    20                     01    A2  91 00555        CMPB    1(R2), #32                          ; 0944
                                                 18  1B 00559        BLEQU   58$
                                                 15  DD 0055B        PUSHL   #21                                 ; 0947
                                    00000000'    EF  9F 0055D        PUSHAB  P.AAI
                                    00000000G    8F  DD 00563        PUSHL   #RNFINM
                         00000000G    EF         03  FB 00569        CALLS   #3, ERM
                                      69         01  D0 00570        MOVL    #1, RANGE_ERROR_FLAG                ; 0948
                                      03         69  E9 00573  58$:  BLBC    RANGE_ERROR_FLAG, 59$               ; 0951
                                          0314   31 00576            BRW     88$
00000000G  EF       65                01         02  EF 00579  59$:  EXTZV   #2, #1, (R5), OUTOPT+12             ; 0957
                         00000000G    EF         01  D0 00582        MOVL    #1, OUTOPT+16                       ; 0958
                                                 50  D4 00589        CLRL    R0                                  ; 0961
                                           02    A2  B5 0058B        TSTW    2(R2)
                                                 02  15 0058E        BLEQ    60$
                                                 50  D6 00590        INCL    R0
     FF74  C8       01                00         50  F0 00592  60$:  INSV    R0, #0, #1, GCA+68
00000000G  EF       01                00   FF74  C8  F0 00599        INSV    GCA+68, #0, #1, SCA+168             ; 0962
                         00000000G    EF   02    A2  32 005A4        CVTWL   2(R2), OUTOPT+20                    ; 0963
           50       63                01         05  EF 005AC        EXTZV   #5, #1, (R3), R0                    ; 0966
     FF74  C8       01                01         50  F0 005B1        INSV    R0, #1, #1, GCA+68
           50       63                01         05  EF 005B8        EXTZV   #5, #1, (R3), R0                    ; 0967
00000000G  EF       01                01         50  F0 005BD        INSV    R0, #1, #1, SCA+168
                                           01    A2  95 005C6        TSTB    1(R2)                               ; 0969
                                                 10  12 005C9        BNEQ    61$
                    0C                63         0B  E1 005CB        BBC     #11, (R3), 61$                      ; 0970
                              FF74    C8         02  8A 005CF        BICB2   #2, GCA+68                          ; 0973
                         00000000G    EF         02  8A 005D4        BICB2   #2, SCA+168                         ; 0974
           5A       63                05         E1 005DB      61$:  BBC     #5, (R3), 67$                       ; 0977
00000000G  EF       63                01         07  EF 005DF        EXTZV   #7, #1, (R3), OUTOPT+8              ; 0981
00000000G  EF       63                01         09  EF 005E8        EXTZV   #9, #1, (R3), OUTOPT+4              ; 0982
                    06                63         0B  E1 005F1        BBC     #11, (R3), 62$                      ; 0986
                                      50   01    A2  9A 005F5        MOVZBL  1(R2), R0                           ; 0989
                                                 37  11 005F9        BRB     66$
                                      01 00000000G   EF  D1 005FB  62$:  CMPL    OUTOPT+8, #1                    ; 0996
                                                 05  12 00602        BNEQ    63$
                                      50         2D  D0 00604        MOVL    #45, R0
                                                 29  11 00607        BRB     66$
                                      01 00000000G   EF  D1 00609  63$:  CMPL    OUTOPT+4, #1                    ; 0997
                                                 05  12 00610        BNEQ    64$
                                      50         07  D0 00612        MOVL    #7, R0
                                                 1B  11 00615        BRB     66$
                                      01 00000000G   EF  D1 00617  64$:  CMPL    OUTOPT+12, #1                   ; 0998
                                                 0E  13 0061E        BEQL    65$
                                      01 00000000G   EF  D1 00620        CMPL    OUTOPT+16, #1
                                                 05  13 00627        BEQL    65$
                                      50         01  CE 00629        MNEGL   #1, R0
                                                 04  11 0062C        BRB     66$
                                      50   5F    8F  9A 0062E  65$:  MOVZBL  #95, R0
                         00000000G    EF         50  D0 00632  66$:  MOVL    R0, OUTOPT                          ; 0985
           50       68                04         04  EF 00639  67$:  EXTZV   #4, #4, GCA+208, R0                 ; 1010
                                      04         50  D1 0063E        CMPL    R0, #4
                                                 0A  13 00641        BEQL    68$
                                      05         50  D1 00643        CMPL    R0, #5                              ; 1011
```

```
                                               05  13 00646      BEQL    68$                              ;
                              03               50  D1 00648      CMPL    R0, #3                           ; 1012
                                               12  12 0064B      BNEQ    69$                              ;
                                   00000000G   EF  7C 0064D 68$: CLRQ    OUTOPT                           ; 1015
                                   00000000G   EF  7C 00653      CLRQ    OUTOPT+8                         ; 1017
                                   00000000G   EF  D4 00659      CLRL    OUTOPT+20                        ; 1019
                              09               63  E9 0065F 69$: BLBC    (R3), 70$                        ; 1023
                                               7E  D4 00662      CLRL    -(SP)                            ; 1025
                      00000000V   EF           01  FB 00664      CALLS   #1, SETQUICK                     ;
          50          65          01           0D  EF 0066B 70$: EXTZV   #13, #1, (R5), R0                ; 1029
   A4     A8          01          00           50  F0 00670      INSV    R0, #0, #1, GCA+116              ;
          50          65          01           0A  EF 00676      EXTZV   #10, #1, (R5), R0                ; 1032
   A4     A8          01          01           50  F0 0067B      INSV    R0, #1, #1, GCA+116              ;
          50          65          01           0C  EF 00681      EXTZV   #12, #1, (R5), R0                ; 1035
   A4     A8          01          03           50  F0 00686      INSV    R0, #3, #1, GCA+116              ;
          50          65          01           09  EF 0068C      EXTZV   #9, #1, (R5), R0                 ; 1038
   A4     A8          01          02           50  F0 00691      INSV    R0, #2, #1, GCA+116              ;
          50          65          01           0B  EF 00697      EXTZV   #11, #1, (R5), R0                ; 1041
   A4     A8          01          04           50  F0 0069C      INSV    R0, #4, #1, GCA+116              ;
          50          65          01           0E  EF 006A2      EXTZV   #14, #1, (R5), R0                ; 1044
   A4     A8          01          05           50  F0 006A7      INSV    R0, #5, #1, GCA+116              ;
          51          65          01           14  EF 006AD      EXTZV   #20, #1, (R5), R1                ; 1054
          50          65          01           15  EF 006B2      EXTZV   #21, #1, (R5), R0                ;
                           FF64   C8      6140     3E 006B7      MOVAW   (R1)[R0], GCA+52                 ;
                                               05  12 006BD      BNEQ    71$                              ; 1057
                           FF64   C8           03  D0 006BF      MOVL    #3, GCA+52                       ; 1059
                      0E          65           06  E1 006C4 71$: BBC     #6, (R5), 72$                    ; 1064
                      00000000G   FF           62  9A 006C8      MOVZBL  (R2), @SCA+136                   ; 1067
                                               05  12 006CF      BNEQ    72$                              ; 1069
                           FF7C   C8           01  8A 006D1      BICB2   #1, GCA+76                       ; 1071
                      09          65           04  E1 006D6 72$: BBC     #4, (R5), 73$                    ; 1074
                                               2F  DD 006DA      PUSHL   #47                              ; 1076
                      00000000G   EF           01  FB 006DC      CALLS   #1, BARS                         ;
                      00000000G   EF           56  D0 006E3 73$: MOVL    R6, PHAN+20                      ; 1080
                                               05  13 006EA      BEQL    74$                              ; 1081
                           FF7C   C8           08  88 006EC      BISB2   #8, GCA+76                       ; 1083
                      00000000G   EF           57  D0 006F1 74$: MOVL    R7, PHAN+48                      ; 1086
                      09          63           03  E1 006F8      BBC     #3, (R3), 75$                    ; 1089
                      00000000G   EF           01  D0 006FC      MOVL    #1, PHAN+52                      ; 1091
                                               06  11 00703      BRB     76$                              ;
                                   00000000G   EF  D4 00705 75$: CLRL    PHAN+52                          ; 1093
00000000G EF          65          01           1E  EF 0070B 76$: EXTZV   #30, #1, (R5), PHAN+60           ; 1096
          50          63          01           01  EF 00714      EXTZV   #1, #1, (R3), R0                 ; 1099
 FF7C     C8          01          02           50  F0 00719      INSV    R0, #2, #1, GCA+76               ;
          50          65          01           12  EF 00720      EXTZV   #18, #1, (R5), R0                ; 1104
          68          01          01           50  F0 00725      INSV    R0, #1, #1, GCA+208              ;
          04          68          04           04  ED 0072A      CMPZV   #4, #4, GCA+208, #4              ; 1112
                                               07  13 0072F      BEQL    77$                              ;
          05          68          04           04  ED 00731      CMPZV   #4, #4, GCA+208, #5              ; 1113
                                               1B  12 00736      BNEQ    78$                              ;
   01     A8          01          00     02    A3  F0 00738 77$: INSV    2(R3), #0, #1, GCA+209           ; 1116
          50          63          01           11  EF 0073F      EXTZV   #17, #1, (R3), R0                ; 1117
   01     A8          01          01           50  F0 00744      INSV    R0, #1, #1, GCA+209              ;
                                               54  DD 0074A      PUSHL   R4                               ; 1121
                      00000000V   EF           01  FB 0074C      CALLS   #1, WRITE_LN01_INFO              ;
                             04   A8     58    A4  32 00753 78$: CVTWL   88(R4), GCA+212                  ; 1144
                             08   A8     5A    A4  32 00758      CVTWL   90(R4), GCA+216                  ; 1145
```

```
                50      09  A8              01              07  EF 0075D          EXTZV   #7, #1, GCA+217, R0                     ; 1151
          FF7C  C8          01              04              50  F0 00763          INSV    R0, #4, #1, GCA+76
                                            50        60    A4  32 0076A          CVTWL   96(R4), R0                              ; 1154
                                                              17  15 0076E          BLEQ    80$
                                            09 00000000G      EF  E9 00770          BLBC    PHAN+52, 79$                            ; 1158
                          00000000G         EF                50  D0 00777          MOVL    R0, PHAN+8                              ; 1161
                                                              07  11 0077E          BRB     80$
                          00000000G         EF                50  D0 00780 79$:     MOVL    R0, PHAN+36                             ; 1165
                                            07 00000000G      EF  E8 00787 80$:     BLBS    PHAN+52, 81$                            ; 1169
                                            2F 00000000G      EF  E9 0078E          BLBC    PHAN+60, 82$
                          00000000G         EF          0C    A9  9E 00795 81$:     MOVAB   $IOB$FILE_SPEC, IOB$+4                  ; 1174
                          00000000G         EF  00040003      8F  C8 0079D          BISL2   #262147, IOB$+46
                          00000000G         EF                01  90 007A8          MOVB    #1, IOB$+44
                                                00000000G   EF  9F 007AF          PUSHAB  XPO$FAILURE
                                                            7E  D4 007B5          CLRL    -(SP)
                                                00000000G   EF  9F 007B7          PUSHAB  IOB$
                          00000000G         EF                03  FB 007BD          CALLS   #3, XPO$OPEN
                                            03 00000000G      EF  E8 007C4 82$:     BLBS    PHAN+52, 83$                            ; 1178
                                                        00AD    31 007CB          BRW     86$
                    03 00000000G            EF                04  E0 007CE 83$:     BBS     #4, TSIIOB+50, 84$                      ; 1181
                                                        00B0    31 007D6          BRW     87$
                          08    AE                      20  B0 007D9 84$:     MOVW    #32, $STR$STRING                        ; 1195
                          0A    AE                      0E  90 007DD          MOVB    #14, $STR$STRING+2
                          0B    AE                      01  90 007E1          MOVB    #1, $STR$STRING+3
                          0C    AE  00000000'           EF  9E 007E5          MOVAB   P.AAO, $STR$STRING+4
                                                            7E  D4 007ED          CLRL    -(SP)
                                                      0C    AE  9F 007EF          PUSHAB  $STR$STRING
                                                            7E  D4 007F2          CLRL    -(SP)
                          00000000G         EF                03  FB 007F4          CALLS   #3, XST$FORMAT
                                            52                50  D0 007FB          MOVL    R0, R2
                                            50 00000000G      EF  D0 007FE          MOVL    IOB$+36, R0
                                                              09  13 00805          BEQL    85$
                                                              50  DD 00807          PUSHL   R0
                          00000000G         EF                01  FB 00809          CALLS   #1, XST$FREE_TEMP
                          00000000G         EF                52  D0 00810 85$:     MOVL    R2, IOB$+36
                          00000000G         EF                01  B0 00817          MOVW    #1, IOB$+52
                          00000000G         EF                0E  90 0081E          MOVB    #14, IOB$+54
                          00000000G         EF                06  90 00825          MOVB    #6, IOB$+44
                                                00000000G   EF  9F 0082C          PUSHAB  XPO$FAILURE
                                                            7E  D4 00832          CLRL    -(SP)
                                                00000000G   EF  9F 00834          PUSHAB  IOB$
                          00000000G         EF                03  FB 0083A          CALLS   #3, XPO$GET
                                08          AE                01  B0 00841          MOVW    #1, $IOB$OUTPUT                         ; 1201
                                0A          AE                0E  90 00845          MOVB    #14, $IOB$OUTPUT+2
                                0B          AE                01  90 00849          MOVB    #1, $IOB$OUTPUT+3
                                0C          AE  00000000'     EF  9E 0084D          MOVAB   P.AAS, $IOB$OUTPUT+4
                          00000000G         EF          08    AE  9E 00855          MOVAB   $IOB$OUTPUT, IOB$+68
                          00000000G         EF                07  90 0085D          MOVB    #7, IOB$+44
                                                00000000G   EF  9F 00864          PUSHAB  XPO$FAILURE
                                                            7E  D4 0086A          CLRL    -(SP)
                                                00000000G   EF  9F 0086C          PUSHAB  IOB$
                          00000000G         EF                03  FB 00872          CALLS   #3, XPO$PUT
                                                              0E  11 00879          BRB     87$                                     ; 1178
                                            07 00000000G      EF  E9 0087B 86$:     BLBC    PHAN+60, 87$                            ; 1205
                          00000000G         EF                00  FB 00882          CALLS   #0, BWAIT                               ; 1207
                                            50                01  D0 00889 87$:     MOVL    #1, R0                                  ; 1209
                                                              04  0088C          RET                                             ;
```

```
                                                G 10
                                50  D4 0088D 88$:    CLRL    R0                      ; 1210
                                    04 0088F         RET                             ;

; Routine Size:  2192 bytes,    Routine Base:  $CODE$ + 0000

; 1100          1211  1
```

```
;  1102          1212  1 %SBTTL 'Disable output and enable "quick" processing'
;  1103          1213  1 GLOBAL ROUTINE SETQUICK (close_output) : NOVALUE =
;  1104          1214  1 !++
;  1105          1215  1 !
;  1106          1216  1 ! FUNCTIONAL DESCRIPTION:
;  1107          1217  1 !
;  1108          1218  1 !       This routine is called to close the output file and set up the GCA
;  1109          1219  1 !       for "quick" output processing.
;  1110          1220  1 !
;  1111          1221  1 ! FORMAL PARAMETERS:
;  1112          1222  1 !
;  1113          1223  1 !       close_output    - If true, the output file is closed.
;  1114          1224  1 !
;  1115          1225  1 ! IMPLICIT INPUTS:
;  1116          1226  1 !
;  1117          1227  1 !       gca_cmd_quick   - If true, no processing is done
;  1118          1228  1 !
;  1119          1229  1 ! IMPLICIT OUTPUTS:
;  1120          1230  1 !
;  1121          1231  1 !       gca_cmd_quick   - Set to true
;  1122          1232  1 !       gca_cmd_bar     - Set to false
;  1123          1233  1 !       gca_skip_out    - Set to true
;  1124          1234  1 !       gca_cmd_msg     - Set to (1 ^ 1)
;  1125          1235  1 !       output file is closed                   !If close_output is true.
;  1126          1236  1 !
;  1127          1237  1 ! ROUTINE VALUE:
;  1128          1238  1 ! COMPLETION CODES:     None
;  1129          1239  1 !
;  1130          1240  1 ! SIDE EFFECTS:         None
;  1131          1241  1 !
;  1132          1242  1 !--
;  1133          1243  2     BEGIN
;  1134          1244  2
;  1135          1245  2     IF NOT .gca_cmd_quick
;  1136          1246  2     THEN
;  1137          1247  3         BEGIN
;  1138          1248  3         gca_cmd_quick   = true;
;  1139          1249  3         gca_cmd_bar     = false;
;  1140          1250  3         gca_skip_out    = true;
;  1141          1251  3         gca_cmd_msg     = (1 ^ 1);
;  1142          1252  3         IF .close_output
;  1143          1253  3         THEN
;  1144          1254  3             clh (clh_close_del_out);
;  1145          1255  2         END;
;  1146          1256  2
;  1147          1257  1     END;                    ! End of SETQUICK


                                        0004 00000         .ENTRY  SETQUICK, Save R2                                       ; 1213
                52 00000000G    EF   9E 00002         MOVAB   GCA+208, R2                                             ;
                1E              62   E8 00009         BLBS    GCA+208, 1$                                             ; 1245
                62              01   88 0000C         BISB2   #1, GCA+208                                             ; 1248
            FF7C    C2          01   8A 0000F         BICB2   #1, GCA+76                                              ; 1249
              A0    A2          01   D0 00014         MOVL    #1, GCA+112                                             ; 1250
```

```
                          FF64   C2              02  D0 00018          MOVL    #2, GCA+52                              ; 1251
                                 09        04    AC  E9 0001D          BLBC    CLOSE_OUTPUT, 1$                        ; 1252
                                                 0C  DD 00021          PUSHL   #12                                     ; 1254
                                                                                                                       ;
                     00000000G   EF              01  FB 00023          CALLS   #1, CLH                                 ;
                                                     04 0002A 1$:      RET                                             ; 1257

; Routine Size:  43 bytes,    Routine Base:  $CODE$ + 0890

; 1148          1258  1
```

```
;  1150          1259  1 ROUTINE write_ln01_info (rno_cmd : ref $rno_cmd) : NOVALUE =
;  1151          1260  1
;  1152          1261  1 !++
;  1153          1262  1 ! FUNCTIONAL DESCRIPTION:
;  1154          1263  1 !
;  1155          1264  1 !       This routine writes LN01-specific escape codes into the output file.
;  1156          1265  1 !
;  1157          1266  1 ! FORMAL PARAMETERS:    None
;  1158          1267  1 !
;  1159          1268  1 !       RNO_CMD is the command-block structure address, passed from DOOPTS.
;  1160          1269  1 !
;  1161          1270  1 ! IMPLICIT INPUTS:
;  1162          1271  1 !
;  1163          1272  1 !       The following GCA bits are checked to control which escape
;  1164          1273  1 !       sequences are written:
;  1165          1274  1 !
;  1166          1275  1 !         rno_cmd [rno$v_ln01_header]
;  1167          1276  1 !         rno_cmd [rno$v_ln01_load]
;  1168          1277  1 !         gca_op_dev
;  1169          1278  1 !         gca_ln01_ital_under
;  1170          1279  1 !         gca_ln01_port_land
;  1171          1280  1 !
;  1172          1281  1 ! IMPLICIT OUTPUTS:
;  1173          1282  1 !
;  1174          1283  1 !       Initial escape sequences are written to the output file.
;  1175          1284  1 !
;  1176          1285  1 ! ROUTINE VALUE:
;  1177          1286  1 ! COMPLETION CODES:     None
;  1178          1287  1 !
;  1179          1288  1 ! SIDE EFFECTS:        None
;  1180          1289  1 !
;  1181          1290  1 !--
;  1182          1291  1
;  1183          1292  2     BEGIN
;  1184          1293  2
;  1185          1294  2     MACRO
;  1186        M 1295  2         write_escape_sequence (string) =
;  1187        M 1296  2             BEGIN
;  1188        M 1297  2             LOCAL
;  1189        M 1298  2                 len,
;  1190        M 1299  2                 ptr;
;  1191        M 1300  2             fs_init (fra);
;  1192        M 1301  2             len = .string[STR$H_LENGTH];
;  1193        M 1302  2             ptr = .string[STR$A_POINTER];
;  1194        M 1303  2             fs_wchar (fra, escape);             !Write initial escape character.
;  1195        M 1304  2             INCR i FROM 1 TO .len DO            !Append formal.
;  1196        M 1305  2                 fs_wchar (fra, CH$RCHAR_A(ptr) );
;  1197        M 1306  2             clh (clh_out_nocr[f);
;  1198        M 1307  2             END
;  1199          1308  2         %;
;  1200          1309  2
;  1201          1310  2     LOCAL
;  1202          1311  2         decslpp,                            ! Lines (actually, pixels) per physical page.
;  1203          1312  2         pixels_per_line_spacing,            ! Used to calculate decslpp.
;  1204          1313  2         twice_paper_size,                   ! Accounts for portrait/landscape orientation.
;  1205          1314  2         right_shift,                        ! Pretend the user said /RIGHT= this value.
;  1206          1315  2         top_margin,
```

```
;  1207          1316  2          bottom_margin,
;  1208          1317  2          left_margin,
;  1209          1318  2          right_margin,
;  1210          1319  2          text_font : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1211          1320  2          bold_font : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1212          1321  2          italic_font : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1213          1322  2          bold_italic_font : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1214          1323  2          text_def : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1215          1324  2          bold_def : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1216          1325  2          italic_def : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1217          1326  2          bold_italic_def : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) ),
;  1218          1327  2          work_string : $STR_DESCRIPTOR (CLASS=DYNAMIC, STRING=(0,0) );
;  1219          1328  2
;  1220          1329  2      !
;  1221          1330  2      ! Assign parameters that depend on portrait/landscape orientation.
;  1222          1331  2      !
;  1223          1332  2      IF .gca_ln01_port_land EQL 1        !Portrait orientation.
;  1224          1333  2      THEN
;  1225          1334  3          BEGIN
;  1226          1335  3          right_shift = 2;
;  1227          1336  3          pixels_per_line_spacing = 50;
;  1228          1337  3          END
;  1229          1338  2      ELSE                                !Landscape orientation.
;  1230          1339  3          BEGIN
;  1231          1340  3          IF .gca_op_dev EQL op_dev_ln01e
;  1232          1341  3          THEN                            !European-style LN01.
;  1233          1342  3              right_shift = 13
;  1234          1343  3          ELSE                            !American-style LN01.
;  1235          1344  3              right_shift = 9;
;  1236          1345  3
;  1237          1346  3          pixels_per_line_spacing = 35;
;  1238          1347  2          END;
;  1239          1348  2
;  1240          1349  2      !
;  1241          1350  2      ! Calculate value to use for decslpp (pixels per page).
;  1242          1351  2      !
;  1243          1352  2      decslpp =
;  1244          1353  3          (IF .phan_simulate
;  1245          1354  3          THEN
;  1246          1355  4              (.phan_plines * .pixels_per_line_spacing)
;  1247          1356  3          ELSE
;  1248          1357  4              (.phan_slines * .pixels_per_line_spacing)
;  1249          1358  2          );
;  1250          1359  2
;  1251          1360  2      !
;  1252          1361  2      ! Calculate values for margins.
;  1253          1362  2      !
;  1254          1363  2      top_margin = 1;
;  1255          1364  2      bottom_margin = .decslpp;
;  1256          1365  2      left_margin = 1;                !DEC rom revision #14 left must be 1.
;  1257          1366  2      right_margin = 65536;           !Largest unsigned number.
;  1258          1367  2
;  1259          1368  2      !
;  1260          1369  2      ! Set up right shift if user did not say /RIGHT himself.
;  1261          1370  2      !
;  1262          1371  2      IF NOT .rno_cmd [rno$v_s_right]
;  1263          1372  2      THEN                            !User did not specify /RIGHT.
```

```
;  1264     1373  3          BEGIN
;  1265     1374  3          phan_right = .right_shift;
;  1266     1375  3          gca_cmd_rit = true;
;  1267     1376  2          END;
;  1268     1377  2      !
;  1269     1378  2      ! Set up vertical shift if user did not say /DOWN himself.
;  1270     1379  2      !
;  1271     1380  2      IF NOT .rno_cmd [rno$v_s_down]
;  1272     1381  2      THEN                              !User did not specify /DOWN.
;  1273     1382  2          phan_down = 2;
;  1274     1383  2
```

```
;  1276        1384  2
;  1277        1385  2      IF .rno_cmd [rno$v_ln01_header]      ! Only process & output header if it
;  1278        1386  2      THEN                                !  was not suppressed by the user.
;  1279        1387  3          BEGIN
;  1280        1388  3          !
;  1281        1389  3          ! Build up a suffix string to determine which fonts get used.
;  1282        1390  3          !
;  1283        1391  3          IF .gca_ln01_port_land EQL 1    ! Determine orientation
;  1284        1392  3          THEN
;  1285        1393  4              $STR_COPY (TARGET= work_string, STRING= 'P' )       ! Portrait
;  1286        1394  3          ELSE
;  1287        1395  3              $STR_COPY (TARGET= work_string, STRING= 'L' );      ! Landscape
;  1288        1396  3
;  1289        1397  3          IF .gca_op_dev EQL op_dev_ln01e
;  1290        1398  3          THEN
;  1291        1399  3              $STR_APPEND (TARGET= work_string, STRING= 'E' );    ! European-style
;  1292        1400  3
;  1293        1401  3          ! If font loading is specified (by the user not saying NOLOAD), we
;  1294        1402  3          ! will need the text strings for their module names built:
;  1295        1403  3          !
;  1296        1404  3          IF .rno_cmd [rno$v_ln01_load]   ! Only load fonts if needed
;  1297        1405  3          THEN
;  1298        1406  4              BEGIN
;  1299        1407  4              !
;  1300        1408  4              ! Build each font module name string by adding the suffix string
;  1301        1409  4              ! to the base name.
;  1302        1410  4              !
;  1303      P 1411  4              $STR_COPY  ( TARGET = text_font
;  1304        1412  4                          ,STRING = $STR_CONCAT ( 'DSR$FONT_T', work_string ) );
;  1305        1413  4
;  1306      P 1414  4              $STR_COPY  ( TARGET = bold_font
;  1307        1415  4                          ,STRING = $STR_CONCAT ( 'DSR$FONT_B', work_string ) );
;  1308        1416  4
;  1309      P 1417  4              $STR_COPY  ( TARGET = italic_font
;  1310        1418  4                          ,STRING = $STR_CONCAT ( 'DSR$FONT_I', work_string ) );
;  1311        1419  4
;  1312      P 1420  4              $STR_COPY  ( TARGET = bold_italic_font
;  1313        1421  5                          ,STRING = $STR_CONCAT ( 'DSR$FONT_BI', work_string ) )
;  1314        1422  3              END;
;  1315        1423  3
;  1316        1424  3          ! Build each font definition file name string by adding the suffix
;  1317        1425  3          ! string to the base name.
;  1318        1426  3          !
;  1319      P 1427  3          $STR_COPY  ( TARGET = text_def
;  1320        1428  3                      ,STRING = $STR_CONCAT( 'DSR$FONT_DEFINE_T', work_string ));
;  1321        1429  3
;  1322      P 1430  3          $STR_COPY  ( TARGET = bold_def
;  1323        1431  3                      ,STRING = $STR_CONCAT( 'DSR$FONT_DEFINE_B', work_string ));
;  1324        1432  3
;  1325      P 1433  3          $STR_COPY  ( TARGET = italic_def
;  1326        1434  3                      ,STRING = $STR_CONCAT( 'DSR$FONT_DEFINE_I', work_string ));
;  1327        1435  3
;  1328      P 1436  3          $STR_COPY  ( TARGET = bold_italic_def
;  1329        1437  3                      ,STRING = $STR_CONCAT( 'DSR$FONT_DEFINE_BI', work_string ));
;  1330        1438  3
```

```
;  1332          1439  3 !+
;  1333          1440  3 ! The preliminaries are taken care of. Now write the initializing escape
;  1334          1441  3 ! sequences to the output (.LNI) file. The sequences are written in the
;  1335          1442  3 ! following order.  The first five are written only if the user didn't
;  1336          1443  3 ! issue a NOHEADER parameter:
;  1337          1444  3 !
;  1338          1445  3 !       Font Load
;  1339          1446  3 !       Font Assignment (Text)
;  1340          1447  3 !       Font Assignment (Bold)
;  1341          1448  3 !       Font Assignment (Italic)        [if Italic specified or defaulted]
;  1342          1449  3 !       Font Assignment (Bold Italic)   [if Italic specified or defaulted]
;  1343          1450  3 !
;  1344          1451  3 !       Set Lines Per Physical Page
;  1345          1452  3 !       Top and Bottom Margins
;  1346          1453  3 !       Left and Right Margins
;  1347          1454  3 !       Default Font Invocation
;  1348          1455  3 !       Vertical Position Absolute
;  1349          1456  3 !
;  1350          1457  3 ! The CLH routine is used to write output. The strings to write are built
;  1351          1458  3 ! up in the FRA fixed-string, and CLH called to write output with no ter-
;  1352          1459  3 ! minating <CR> and <LF>.
;  1353          1460  3 !-
;  1354          1461  3
;  1355          1462  3         !
;  1356          1463  3         ! Write Font Load escape sequence unless the user said not to. Format:
;  1357          1464  3         !
;  1358          1465  3         !   <ESC>]VMS;1;DSR$FONT_LOAD,fn1,fn2,...,ANSI$ST,def1,def2,...<ESC>\
;  1359          1466  3         !
;  1360          1467  3         !   fn1, fn2,...    = names of library modules containing fonts
;  1361          1468  3         !   def1, def2,...  = names of lib. modules containing font definitions
;  1362          1469  3         !
;  1363          1470  3         ! The above sequence is known as an OSC ESCAPE SEQUENCE.
;  1364          1471  3
;  1365          1472  3         $STR_COPY (TARGET = work_string, STRING= ']VMS;1;' ); ! OSC control seq.
;  1366          1473  3
;  1367          1474  3         IF .rno_cmd [rno$v_ln01_load]   ! Only load fonts if needed
;  1368          1475  4         THEN
;  1369          1476  4             BEGIN
;  1370          1477  4
;  1371          1478  4                 ! load the Text and Bold fonts.
;  1372        P 1479  4
;  1373        P 1480  4             $STR_APPEND ( TARGET = work_string
;  1374        P 1481  4                          ,STRING= $STR_CONCAT   ( 'DSR$FONT_LOAD,'
;  1375        P 1482  4                                                  ,text_font
;  1376        P 1483  4                                                  ,','
;  1377          1484  4                                                  ,bold_font
;  1378          1485  4                                                  ,','              )       );
;  1379          1486  4
;  1380          1487  4             ! If italics were requested, also load the Italic and Bold Italic
;  1381          1488  4             ! fonts.
;  1382          1489  4             !
;  1383          1490  4             IF .gca_ln01_ital_under EQL 1       !Italics requested.
;  1384        P 1491  4             THEN
;  1385        P 1492  4                 $STR_APPEND ( TARGET = work_string
;  1386        P 1493  4                              ,STRING = $STR_CONCAT ( italic_font
;  1387        P 1494  4                                                      ,','
;  1388          1495  4                                                      ,bold_italic_font
                                                                             ,','              )       );
```

```
;  1389          1496  4
;  1390          1497  4               ! Finish the font-load escape sequence.
;  1391          1498  4               !
;  1392          1499  5               $STR_APPEND ( TARGET = work_string ,STRING = 'ANSI$ST,' )
;  1393          1500  5
;  1394          1501  5               END;  ! of font loading controls
;  1395          1502  3
;  1396          1503  3           ! This section specifies the font definition module names
;  1397          1504  3           !
;  1398          1505  3           ! Add text and bold definition module names:
;  1399          1506  3           !
;  1400        P 1507  3           $STR_APPEND ( TARGET = work_string
;  1401        P 1508  3                       ,STRING = $STR_CONCAT ( text_def
;  1402        P 1509  3                                               ,','
;  1403          1510  3                                               ,bold_def   )       );
;  1404          1511  3
;  1405          1512  3           ! If italics were requested, also load the definition modules for the
;  1406          1513  3           ! Italic and Bold Italic fonts.
;  1407          1514  3           !
;  1408          1515  3           IF .gca_ln01_ital_under EQL 1   !Italics requested.
;  1409          1516  3           THEN
;  1410        P 1517  3               $STR_APPEND ( TARGET = work_string
;  1411        P 1518  3                           ,STRING = $STR_CONCAT ( ','
;  1412        P 1519  3                                                   ,italic_def
;  1413        P 1520  3                                                   ,','
;  1414          1521  3                                                   ,bold_italic_def  )  );
;  1415          1522  3
;  1416          1523  3           ! Finish the font load and definition escape sequence.
;  1417          1524  3           !
;  1418        P 1525  3           $STR_APPEND ( TARGET = work_string
;  1419          1526  3                       ,STRING = $STR_CONCAT ( %CHAR(escape), '\' )  );
;  1420          1527  3
;  1421          1528  3           ! Write the string.
;  1422          1529  3           !
;  1423          1530  4           write_escape_sequence (work_string)
;  1424          1531  2           END;
;  1425          1532  2
```

```
;  1427          1533  2      !
;  1428          1534  2      ! Write Set Lines Per Physical Page escape sequence. Format:
;  1429          1535  2      !
;  1430          1536  2      !   <ESC> [ Pn t                        !Pn = form length (pixels)
;  1431          1537  2      !
;  1432        P 1538  2      $STR_COPY  ( TARGET = work_string
;  1433        P 1539  2                  ,STRING = $STR_CONCAT ( '[', $STR_ASCII(.decslpp), 't' )  );
;  1434          1540  2
;  1435          1541  2      write_escape_sequence (work_string);
;  1436          1542  2
;  1437          1543  2      !
;  1438          1544  2      ! Write Top and Bottom Margins escape sequence. Format:
;  1439          1545  2      !
;  1440          1546  2      !  <ESC> [ Pn ; Pm r
;  1441          1547  2      !
;  1442          1548  2      !  Pn = top margin (pixels)
;  1443          1549  2      !  Pm  = bottom margin (pixels)
;  1444          1550  2      !
;  1445        P 1551  2      $STR_COPY  ( TARGET = work_string
;  1446        P 1552  2                  ,STRING = $STR_CONCAT ( '['
;  1447        P 1553  2                                         ,$STR_ASCII(.top_margin)
;  1448        P 1554  2                                         ,';'
;  1449        P 1555  2                                         ,$STR_ASCII(.bottom_margin)
;  1450          1556  2                                         ,'r'                    )  );
;  1451          1557  2
;  1452          1558  2      write_escape_sequence (work_string);
;  1453          1559  2
;  1454          1560  2      !
;  1455          1561  2      ! Write Left and Right Margins escape sequence. Format:
;  1456          1562  2      !
;  1457          1563  2      !  <ESC> [ Pn ; Pm s
;  1458          1564  2      !
;  1459          1565  2      !  Pn = left margin (pixels)
;  1460          1566  2      !  Pm  = right margin (pixels)
;  1461          1567  2      !
;  1462        P 1568  2      $STR_COPY  ( TARGET = work_string
;  1463        P 1569  2                  ,STRING = $STR_CONCAT ( '['
;  1464        P 1570  2                                         ,$STR_ASCII(.left_margin)
;  1465        P 1571  2                                         ,';'
;  1466        P 1572  2                                         ,$STR_ASCII(.right_margin)
;  1467          1573  2                                         ,'s'    )  );
;  1468          1574  2
;  1469          1575  2      write_escape_sequence (work_string);
```

```
;  1471          1576  2
;  1472          1577  2      ! Write the sequence to load the DSR$PAGE_SIZE module to allow the user to
;  1473          1578  2      ! redefine the parameters just set.  It is assumed that initially, for the
;  1474          1579  2      ! normal user, this module will be a null module; i.e. it will not contain
;  1475          1580  2      ! anything.  However, its reference here assumes that it will be present in
;  1476          1581  2      ! the library.
;  1477          1582  2      !
;  1478          1583  2      ! This sequence is known as an OSC ESCAPE SEQUENCE.
;  1479          1584  2
;  1480          1585  2      IF .rno_cmd [rno$v_ln01_header]     ! Only process & output header if it
;  1481          1586  2      THEN                               !  was not suppressed by the user.
;  1482          1587  3          BEGIN
;  1483        P 1588  3          $STR_COPY  ( TARGET = work_string
;  1484        P 1589  3                      ,STRING = $STR_CONCAT ( ']VMS;1;DSR$PAGE_SIZE'
;  1485        P 1590  3                                             ,%CHAR(escape)
;  1486          1591  3                                             ,'\'  )  );
;  1487          1592  3
;  1488          1593  4          write_escape_sequence (work_string)
;  1489          1594  2          END;
```

```
;  1491          1595  2     !
;  1492          1596  2     ! Write Default Font Invocation escape sequence. Format:
;  1493          1597  2     !
;  1494          1598  2     !  <ESC> [ 12 m
;  1495          1599  2     !
;  1496          1600  2     $STR_COPY (TARGET = work_string, STRING = '[12m' );
;  1497          1601  2     write_escape_sequence (work_string);
;  1498          1602  2
;  1499          1603  2     !
;  1500          1604  2     ! Write Vertical Position Absolute escape sequence. Format:
;  1501          1605  2     !
;  1502          1606  2     !   <ESC> [ 0 d
;  1503          1607  2     !
;  1504          1608  2     $STR_COPY (TARGET=work_string, STRING = '[0d' );
;  1505          1609  2     write_escape_sequence (work_string);
;  1506          1610  2
;  1507          1611  2     ! Force a formfeed record to tell the print symbiont to reset its line
;  1508          1612  2     ! counters.  This is a HACK made necessary because VMS-land won't provide
;  1509          1613  2     ! a better method to accomplish the same end.
;  1510          1614  2
;  1511          1615  2     phan_lines_tp = .phan_lines_tp + 1  ! Count one line of output
;  1512          1616  2
;  1513          1617  1     END;                                        !End of write_ln01_info


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                50  00094 P.AAT:   .ASCII  \P\                                     ;
                                                4C  00095 P.AAU:   .ASCII  \L\                                     ;
                                                45  00096 P.AAV:   .ASCII  \E\                                     ;
                                54  5F  54  4E  4F  46  24  52  53  44  00097 P.AAX:   .ASCII  \DSR$FONT_T\                          ;
                                42  5F  54  4E  4F  46  24  52  53  44  000A1 P.ABA:   .ASCII  \DSR$FONT_B\                          ;
                                49  5F  54  4E  4F  46  24  52  53  44  000AB P.ABD:   .ASCII  \DSR$FONT_I\                          ;
                            49  42  5F  54  4E  4F  46  24  52  53  44  000B5 P.ABG:   .ASCII  \DSR$FONT_BI\                         ;
45  4E  49  46  45  44  5F  54  4E  4F  46  24  52  53  44  000C0 P.ABJ:   .ASCII  \DSR$FONT_DEFINE_T\                       ;
                                                54  5F  000CF
45  4E  49  46  45  44  5F  54  4E  4F  46  24  52  53  44  000D1 P.ABM:   .ASCII  \DSR$FONT_DEFINE_B\                       ;
                                                42  5F  000E0
45  4E  49  46  45  44  5F  54  4E  4F  46  24  52  53  44  000E2 P.ABP:   .ASCII  \DSR$FONT_DEFINE_I\                       ;
                                                49  5F  000F1
45  4E  49  46  45  44  5F  54  4E  4F  46  24  52  53  44  000F3 P.ABS:   .ASCII  \DSR$FONT_DEFINE_BI\                      ;
                                            49  42  5F  00102
                                3B  31  3B  53  4D  56  5D  00105 P.ABU:   .ASCII  \]VMS;1;\                               ;
    2C  44  41  4F  4C  5F  54  4E  4F  46  24  52  53  44  0010C P.ABY:   .ASCII  \DSR$FONT_LOAD,\                         ;
                                                2C  0011A P.ABZ:   .ASCII  \,\                                     ;
                                                2C  0011B P.ACA:   .ASCII  \,\                                     ;
                                                2C  0011C P.ACG:   .ASCII  \,\                                     ;
                                                2C  0011D P.ACH:   .ASCII  \,\                                     ;
                            2C  54  53  24  49  53  4E  41  0011E P.ACK:   .ASCII  \ANSI$ST,\                           ;
                                                2C  00126 P.ACM:   .ASCII  \,\                                     ;
                                                2C  00127 P.ACQ:   .ASCII  \,\                                     ;
                                                2C  00128 P.ACR:   .ASCII  \,\                                     ;
                                                1B  00129 P.ACW:   .ASCII  <27>                                    ;
                                                5C  0012A P.ACX:   .ASCII  <92>                                    ;
                                                5B  0012B P.ADC:   .ASCII  \[\                                     ;
                                                74  0012C P.ADD:   .ASCII  \t\                                     ;
```

```
                                                  5B 0012D P.ADJ:  .ASCII  \[\                              ;
                                                  3B 0012E P.ADK:  .ASCII  \;\                              ;
                                                  72 0012F P.ADL:  .ASCII  \r\                              ;
                                                  5B 00130 P.ADS:  .ASCII  \[\                              ;
                                                  3B 00131 P.ADT:  .ASCII  \;\                              ;
                                                  73 00132 P.ADU:  .ASCII  \s\                              ;
45 47 41 50 24 52 53 44 3B 31 3B 53 4D 56 5D 00133 P.AEB:  .ASCII  \]VMS;1;DSR$PAGE_SIZE\            ;
                                   45 5A 49 53 5F 00142                                               ;
                                                  1B 00147 P.AEC:  .ASCII  <27>                             ;
                                                  5C 00148 P.AED:  .ASCII  <92>                             ;
                                         6D 32 31 5B 00149 P.AEH:  .ASCII  \[12m\                           ;
                                            64 30 5B 0014D P.AEI:  .ASCII  \[0d\                            ;

                                                                   .PSECT  $OWN$,NOEXE,2

                                         0001  0006C $STR$STRING:
                                                                   .WORD   1                                ;
                                       01  0E  0006E               .BYTE   14, 1                            ;
                                   00000000'  00070               .ADDRESS P.AAT                           ;
                                         0001  00074 $STR$STRING:
                                                                   .WORD   1                                ;
                                       01  0E  00076               .BYTE   14, 1                            ;
                                   00000000'  00078               .ADDRESS P.AAU                           ;
                                         0001  0007C $STR$STRING:
                                                                   .WORD   1                                ;
                                       01  0E  0007E               .BYTE   14, 1                            ;
                                   00000000'  00080               .ADDRESS P.AAV                           ;
                                         000A  00084 $STR$STRING0:
                                                                   .WORD   10                               ;
                                       01  0E  00086               .BYTE   14, 1                            ;
                                   00000000'  00088               .ADDRESS P.AAX                           ;
                                         000A  0008C $STR$STRING0:
                                                                   .WORD   10                               ;
                                       01  0E  0008E               .BYTE   14, 1                            ;
                                   00000000'  00090               .ADDRESS P.ABA                           ;
                                         000A  00094 $STR$STRING0:
                                                                   .WORD   10                               ;
                                       01  0E  00096               .BYTE   14, 1                            ;
                                   00000000'  00098               .ADDRESS P.ABD                           ;
                                         000B  0009C $STR$STRING0:
                                                                   .WORD   11                               ;
                                       01  0E  0009E               .BYTE   14, 1                            ;
                                   00000000'  000A0               .ADDRESS P.ABG                           ;
                                         0011  000A4 $STR$STRING0:
                                                                   .WORD   17                               ;
                                       01  0E  000A6               .BYTE   14, 1                            ;
                                   00000000'  000A8               .ADDRESS P.ABJ                           ;
                                         0011  000AC $STR$STRING0:
                                                                   .WORD   17                               ;
                                       01  0E  000AE               .BYTE   14, 1                            ;
                                   00000000'  000B0               .ADDRESS P.ABM                           ;
                                         0011  000B4 $STR$STRING0:
                                                                   .WORD   17                               ;
                                       01  0E  000B6               .BYTE   14, 1                            ;
                                   00000000'  000B8               .ADDRESS P.ABP                           ;
                                         0012  000BC $STR$STRING0:
                                                                   .WORD   18                               ;
```

```
      01  0E  000BE                  .BYTE    14, 1
  00000000'   000C0                  .ADDRESS P.ABS
        0007  000C4  $STR$STRING:
                                     .WORD    7
      01  0E  000C6                  .BYTE    14, 1
  00000000'   000C8                  .ADDRESS P.ABU
        000E  000CC  $STR$STRING0:
                                     .WORD    14
      01  0E  000CE                  .BYTE    14, 1
  00000000'   000D0                  .ADDRESS P.ABY
        0001  000D4  $STR$STRING2:
                                     .WORD    1
      01  0E  000D6                  .BYTE    14, 1
  00000000'   000D8                  .ADDRESS P.ABZ
        0001  000DC  $STR$STRING4:
                                     .WORD    1
      01  0E  000DE                  .BYTE    14, 1
  00000000'   000E0                  .ADDRESS P.ACA
        0001  000E4  $STR$STRING1:
                                     .WORD    1
      01  0E  000E6                  .BYTE    14, 1
  00000000'   000E8                  .ADDRESS P.ACG
        0001  000EC  $STR$STRING3:
                                     .WORD    1
      01  0E  000EE                  .BYTE    14, 1
  00000000'   000F0                  .ADDRESS P.ACH
        0008  000F4  $STR$STRING:
                                     .WORD    8
      01  0E  000F6                  .BYTE    14, 1
  00000000'   000F8                  .ADDRESS P.ACK
        0001  000FC  $STR$STRING1:
                                     .WORD    1
      01  0E  000FE                  .BYTE    14, 1
  00000000'   00100                  .ADDRESS P.ACM
        0001  00104  $STR$STRING0:
                                     .WORD    1
      01  0E  00106                  .BYTE    14, 1
  00000000'   00108                  .ADDRESS P.ACQ
        0001  0010C  $STR$STRING2:
                                     .WORD    1
      01  0E  0010E                  .BYTE    14, 1
  00000000'   00110                  .ADDRESS P.ACR
        0001  00114  $STR$STRING0:
                                     .WORD    1
      01  0E  00116                  .BYTE    14, 1
  00000000'   00118                  .ADDRESS P.ACW
        0001  0011C  $STR$STRING1:
                                     .WORD    1
      01  0E  0011E                  .BYTE    14, 1
  00000000'   00120                  .ADDRESS P.ACX
        0001  00124  $STR$STRING0:
                                     .WORD    1
      01  0E  00126                  .BYTE    14, 1
  00000000'   00128                  .ADDRESS P.ADC
        0001  0012C  $STR$STRING2:
                                     .WORD    1
      01  0E  0012E                  .BYTE    14, 1
```

```
                00000000'  00130          .ADDRESS P.ADD                                          ;
                    0001   00134 $STR$STRING0:
                                          .WORD   1                                               ;
                  01  0E   00136          .BYTE   14, 1                                           ;
                00000000'  00138          .ADDRESS P.ADJ                                          ;
                    0001   0013C $STR$STRING2:
                                          .WORD   1                                               ;
                  01  0E   0013E          .BYTE   14, 1                                           ;
                00000000'  00140          .ADDRESS P.ADK                                          ;
                    0001   00144 $STR$STRING4:
                                          .WORD   1                                               ;
                  01  0E   00146          .BYTE   14, 1                                           ;
                00000000'  00148          .ADDRESS P.ADL                                          ;
                    0001   0014C $STR$STRING0:
                                          .WORD   1                                               ;
                  01  0E   0014E          .BYTE   14, 1                                           ;
                00000000'  00150          .ADDRESS P.ADS                                          ;
                    0001   00154 $STR$STRING2:
                                          .WORD   1                                               ;
                  01  0E   00156          .BYTE   14, 1                                           ;
                00000000'  00158          .ADDRESS P.ADT                                          ;
                    0001   0015C $STR$STRING4:
                                          .WORD   1                                               ;
                  01  0E   0015E          .BYTE   14, 1                                           ;
                00000000'  00160          .ADDRESS P.ADU                                          ;
                    0014   00164 $STR$STRING0:
                                          .WORD   20                                              ;
                  01  0E   00166          .BYTE   14, 1                                           ;
                00000000'  00168          .ADDRESS P.AEB                                          ;
                    0001   0016C $STR$STRING1:
                                          .WORD   1                                               ;
                  01  0E   0016E          .BYTE   14, 1                                           ;
                00000000'  00170          .ADDRESS P.AEC                                          ;
                    0001   00174 $STR$STRING2:
                                          .WORD   1                                               ;
                  01  0E   00176          .BYTE   14, 1                                           ;
                00000000'  00178          .ADDRESS P.AED                                          ;
                    0004   0017C $STR$STRING:
                                          .WORD   4                                               ;
                  01  0E   0017E          .BYTE   14, 1                                           ;
                00000000'  00180          .ADDRESS P.AEH                                          ;
                    0003   00184 $STR$STRING:
                                          .WORD   3                                               ;
                  01  0E   00186          .BYTE   14, 1                                           ;
                00000000'  00188          .ADDRESS P.AEI                                          ;

                                          .EXTRN  XST$COPY, STR$FAILURE
                                          .EXTRN  XST$APPEND, XST$JOIN
                                          .EXTRN  XST$ASCII

                                          .PSECT  $CODE$,NOWRT,2

                    0FFC 00000 WRITE_LN01_INFO:
                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 1259
5B 00000000G   EF   9E 00002              MOVAB   STR$FAILURE, R11                                ;
5A 00000000'   EF   9E 00009              MOVAB   $STR$STRING, R10                                ;
59 00000000G   EF   9E 00010              MOVAB   FRA+4, R9                                       ;
```

```
                     5E        BC    AE  9E 00017          MOVAB   -68(SP), SP                                     :
          3C         AE  020E0000    8F  D0 0001B          MOVL    #34471936, TEXT_FONT                            : 1319
                                40   AE  D4 00023          CLRL    TEXT_FONT+4                                     :
          34         AE  020E0000    8F  D0 00026          MOVL    #34471936, BOLD_FONT                            : 1320
                                38   AE  D4 0002E          CLRL    BOLD_FONT+4                                     :
          2C         AE  020E0000    8F  D0 00031          MOVL    #34471936, ITALIC_FONT                          : 1321
                                30   AE  D4 00039          CLRL    ITALIC_FONT+4                                   :
          24         AE  020E0000    8F  D0 0003C          MOVL    #34471936, BOLD_ITALIC_FONT                     : 1322
                                28   AE  D4 00044          CLRL    BOLD_ITALIC_FONT+4                              :
          1C         AE  020E0000    8F  D0 00047          MOVL    #34471936, TEXT_DEF                             : 1323
                                20   AE  D4 0004F          CLRL    TEXT_DEF+4                                      :
          14         AE  020E0000    8F  D0 00052          MOVL    #34471936, BOLD_DEF                             : 1324
                                18   AE  D4 0005A          CLRL    BOLD_DEF+4                                      :
          0C         AE  020E0000    8F  D0 0005D          MOVL    #34471936, ITALIC_DEF                           : 1325
                                10   AE  D4 00065          CLRL    ITALIC_DEF+4                                    :
          04         AE  020E0000    8F  D0 00068          MOVL    #34471936, BOLD_ITALIC_DEF                      : 1326
                                08   AE  D4 00070          CLRL    BOLD_ITALIC_DEF+4                               :
                         020E0000    8F  DD 00073          PUSHL   #34471936                                       : 1327
                                04   AE  D4 00079          CLRL    WORK_STRING+4                                   :
          08 00000000G  EF           01  E1 0007C          BBC     #1, GCA+209, 1$                                 : 1332
                        51           02  D0 00084          MOVL    #2, RIGHT_SHIFT                                 : 1335
                        50           32  D0 00087          MOVL    #50, PIXELS_PER_LINE_SPACING                    : 1336
                                     16  11 0008A          BRB     4$                                              : 1332
 05 00000000G  EF           04       04  ED 0008C 1$:      CMPZV   #4, #4, GCA+208, #5                             : 1340
                                     05  12 00095          BNEQ    2$                                              :
                            51       0D  D0 00097          MOVL    #13, RIGHT_SHIFT                                : 1342
                                     03  11 0009A          BRB     3$                                              :
                            51       09  D0 0009C 2$:      MOVL    #9, RIGHT_SHIFT                                 : 1344
                            50       23  D0 0009F 3$:      MOVL    #35, PIXELS_PER_LINE_SPACING                    : 1346
                            0A 00000000G EF  E9 000A2 4$:  BLBC    PHAN+52, 5$                                     : 1353
          54 00000000G  EF           50  C5 000A9          MULL3   PIXELS_PER_LINE_SPACING, PHAN+8, DECSLPP        : 1355
                                     08  11 000B1          BRB     6$                                              :
          54 00000000G  EF           50  C5 000B3 5$:      MULL3   PIXELS_PER_LINE_SPACING, PHAN+36, DECSLPP       : 1357
                        58           01  D0 000BB 6$:      MOVL    #1, TOP_MARGIN                                  : 1363
                        57           54  D0 000BE          MOVL    DECSLPP, BOTTOM_MARGIN                          : 1364
                        56           01  D0 000C1          MOVL    #1, LEFT_MARGIN                                 : 1365
                        55  00010000 8F  D0 000C4          MOVL    #65536, RIGHT_MARGIN                            : 1366
                        52        04 AC  D0 000CB          MOVL    RNO_CMD, R2                                     : 1371
          0E            52   A2      05  E0 000CF          BBS     #5, 82(R2), 7$                                  :
            00000000G   EF           51  D0 000D4          MOVL    RIGHT_SHIFT, PHAN+20                            : 1374
            00000000G   EF           08  88 000DB          BISB2   #8, GCA+76                                      : 1375
          07            52   A2      04  E0 000E2 7$:      BBS     #4, 82(R2), 8$                                  : 1380
            00000000G   EF           02  D0 000E7          MOVL    #2, PHAN+48                                     : 1382
          03            52   A2      02  E0 000EE 8$:      BBS     #2, 82(R2), 9$                                  : 1385
                                   0262  31 000F3          BRW     19$                                             :
          0B 00000000G  EF           01  E1 000F6 9$:      BBC     #1, GCA+209, 10$                                : 1391
                                     5B  DD 000FE          PUSHL   R11                                             : 1393
                                     7E  D4 00100          CLRL    -(SP)                                           :
                                08   AE  9F 00102          PUSHAB  $STR$TARGET                                     :
                                     5A  DD 00105          PUSHL   R10                                             :
                                     0A  11 00107          BRB     11$                                             :
                                     5B  DD 00109 10$:     PUSHL   R11                                             : 1395
                                     7E  D4 0010B          CLRL    -(SP)                                           :
                                08   AE  9F 0010D          PUSHAB  $STR$TARGET                                     :
                                08   AA  9F 00110          PUSHAB  $STR$STRING                                     :
                                     7E  D4 00113 11$:     CLRL    -(SP)                                           :
            00000000G   EF           05  FB 00115          CALLS   #5, XST$COPY                                    :
```

```
                05 00000000G  EF            04                04  ED 0011C          CMPZV   #4, #4, GCA+208, #5                    ; 1397
                                                                13  12 00125          BNEQ    12$
                                                                5B  DD 00127          PUSHL   R11                                    ; 1399
                                                                7E  D4 00129          CLRL    -(SP)
                                                          08    AE  9F 0012B          PUSHAB  WORK_STRING
                                                          10    AA  9F 0012E          PUSHAB  $STR$STRING
                                                                7E  D4 00131          CLRL    -(SP)
                                    00000000G   EF              05  FB 00133          CALLS   #5, XST$APPEND
                                 78         52  A2              03  E1 0013A 12$:     BBC     #3, 82(R2), 13$                        ; 1404
                                                                5E  DD 0013F          PUSHL   SP                                     ; 1412
                                                          18    AA  9F 00141          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 00144          CALLS   #2, XST$JOIN
                                                                5B  DD 0014B          PUSHL   R11
                                                                7E  D4 0014D          CLRL    -(SP)
                                                          48    AE  9F 0014F          PUSHAB  $STR$TARGET
                                                                50  DD 00152          PUSHL   R0
                                                                7E  D4 00154          CLRL    -(SP)
                                    00000000G   EF              05  FB 00156          CALLS   #5, XST$COPY
                                                                5E  DD 0015D          PUSHL   SP                                     ; 1415
                                                          20    AA  9F 0015F          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 00162          CALLS   #2, XST$JOIN
                                                                5B  DD 00169          PUSHL   R11
                                                                7E  D4 0016B          CLRL    -(SP)
                                                          40    AE  9F 0016D          PUSHAB  $STR$TARGET
                                                                50  DD 00170          PUSHL   R0
                                                                7E  D4 00172          CLRL    -(SP)
                                    00000000G   EF              05  FB 00174          CALLS   #5, XST$COPY
                                                                5E  DD 0017B          PUSHL   SP                                     ; 1418
                                                          28    AA  9F 0017D          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 00180          CALLS   #2, XST$JOIN
                                                                5B  DD 00187          PUSHL   R11
                                                                7E  D4 00189          CLRL    -(SP)
                                                          38    AE  9F 0018B          PUSHAB  $STR$TARGET
                                                                50  DD 0018E          PUSHL   R0
                                                                7E  D4 00190          CLRL    -(SP)
                                    00000000G   EF              05  FB 00192          CALLS   #5, XST$COPY
                                                                5E  DD 00199          PUSHL   SP                                     ; 1421
                                                          30    AA  9F 0019B          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 0019E          CALLS   #2, XST$JOIN
                                                                5B  DD 001A5          PUSHL   R11
                                                                7E  D4 001A7          CLRL    -(SP)
                                                          30    AE  9F 001A9          PUSHAB  $STR$TARGET
                                                                50  DD 001AC          PUSHL   R0
                                                                7E  D4 001AE          CLRL    -(SP)
                                    00000000G   EF              05  FB 001B0          CALLS   #5, XST$COPY
                                                                5E  DD 001B7 13$:     PUSHL   SP                                     ; 1428
                                                          38    AA  9F 001B9          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 001BC          CALLS   #2, XST$JOIN
                                                                5B  DD 001C3          PUSHL   R11
                                                                7E  D4 001C5          CLRL    -(SP)
                                                          28    AE  9F 001C7          PUSHAB  $STR$TARGET
                                                                50  DD 001CA          PUSHL   R0
                                                                7E  D4 001CC          CLRL    -(SP)
                                    00000000G   EF              05  FB 001CE          CALLS   #5, XST$COPY
                                                                5E  DD 001D5          PUSHL   SP                                     ; 1431
                                                          40    AA  9F 001D7          PUSHAB  $STR$STRINGO
                                    00000000G   EF              02  FB 001DA          CALLS   #2, XST$JOIN                           ;
```

```
                                 5B   DD 001E1          PUSHL   R11                                                    ;
                                 7E   D4 001E3          CLRL    -(SP)                                                  ;
                              20 AE   9F 001E5          PUSHAB  $STR$TARGET                                            ;
                                 50   DD 001E8          PUSHL   R0                                                     ;
                                 7E   D4 001EA          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 001EC     CALLS   #5, XST$COPY                                           ;
                                 5E   DD 001F3          PUSHL   SP                                                     ; 1434
                              48 AA   9F 001F5          PUSHAB  $STR$STRING0                                           ;
                00000000G     EF       02 FB 001F8     CALLS   #2, XST$JOIN                                           ;
                                 5B   DD 001FF          PUSHL   R11                                                    ;
                                 7E   D4 00201          CLRL    -(SP)                                                  ;
                              18 AE   9F 00203          PUSHAB  $STR$TARGET                                            ;
                                 50   DD 00206          PUSHL   R0                                                     ;
                                 7E   D4 00208          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 0020A     CALLS   #5, XST$COPY                                           ;
                                 5E   DD 00211          PUSHL   SP                                                     ; 1437
                              50 AA   9F 00213          PUSHAB  $STR$STRING0                                           ;
                00000000G     EF       02 FB 00216     CALLS   #2, XST$JOIN                                           ;
                                 5B   DD 0021D          PUSHL   R11                                                    ;
                                 7E   D4 0021F          CLRL    -(SP)                                                  ;
                              10 AE   9F 00221          PUSHAB  $STR$TARGET                                            ;
                                 50   DD 00224          PUSHL   R0                                                     ;
                                 7E   D4 00226          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 00228     CALLS   #5, XST$COPY                                           ;
                                 5B   DD 0022F          PUSHL   R11                                                    ; 1471
                                 7E   D4 00231          CLRL    -(SP)                                                  ;
                              08 AE   9F 00233          PUSHAB  $STR$TARGET                                            ;
                              58 AA   9F 00236          PUSHAB  $STR$STRING                                            ;
                                 7E   D4 00239          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 0023B     CALLS   #5, XST$COPY                                           ;
             69       52   A2          03 E1 00242     BBC     #3, 82(R2), 15$                                        ; 1473
                              70 AA   9F 00247          PUSHAB  $STR$STRING4                                           ; 1484
                              3C AE   9F 0024A          PUSHAB  $STR$STRING3                                           ;
                              68 AA   9F 0024D          PUSHAB  $STR$STRING2                                           ;
                              4C AE   9F 00250          PUSHAB  $STR$STRING1                                           ;
                              60 AA   9F 00253          PUSHAB  $STR$STRING0                                           ;
                00000000G     EF       05 FB 00256     CALLS   #5, XST$JOIN                                           ;
                                 5B   DD 0025D          PUSHL   R11                                                    ;
                                 7E   D4 0025F          CLRL    -(SP)                                                  ;
                              08 AE   9F 00261          PUSHAB  WORK_STRING                                            ;
                                 50   DD 00264          PUSHL   R0                                                     ;
                                 7E   D4 00266          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 00268     CALLS   #5, XST$APPEND                                         ;
                           26 00000000G  EF E9 0026F     BLBC    GCA+209, 14$                                           ; 1489
                                 0080    CA 9F 00276     PUSHAB  $STR$STRING3                                           ; 1495
                                 2C   AE 9F 0027A        PUSHAB  $STR$STRING2                                           ;
                                 78   AA 9F 0027D        PUSHAB  $STR$STRING1                                           ;
                                 3C   AE 9F 00280        PUSHAB  $STR$STRING0                                           ;
                00000000G     EF       04 FB 00283     CALLS   #4, XST$JOIN                                           ;
                                 5B   DD 0028A          PUSHL   R11                                                    ;
                                 7E   D4 0028C          CLRL    -(SP)                                                  ;
                              08 AE   9F 0028E          PUSHAB  WORK_STRING                                            ;
                                 50   DD 00291          PUSHL   R0                                                     ;
                                 7E   D4 00293          CLRL    -(SP)                                                  ;
                00000000G     EF       05 FB 00295     CALLS   #5, XST$APPEND                                         ;
                                 5B   DD 0029C 14$:     PUSHL   R11                                                    ; 1499
                                 7E   D4 0029E          CLRL    -(SP)                                                  ;
```

```
                        08   AE  9F 002A0        PUSHAB  WORK_STRING                 ;
                      0088   CA  9F 002A3        PUSHAB  $STR$STRING                 ;
                             7E  D4 002A7        CLRL    -(SP)                       ;
    00000000G  EF            05  FB 002A9        CALLS   #5, XST$APPEND              ;
                        18   AE  9F 002B0  15$:  PUSHAB  $STR$STRING2                ; 1510
                      0090   CA  9F 002B3        PUSHAB  $STR$STRING1                ;
                        28   AE  9F 002B7        PUSHAB  $STR$STRING0                ;
    00000000G  EF            03  FB 002BA        CALLS   #3, XST$JOIN                ;
                             5B  DD 002C1        PUSHL   R11                         ;
                             7E  D4 002C3        CLRL    -(SP)                       ;
                        08   AE  9F 002C5        PUSHAB  WORK_STRING                 ;
                             50  DD 002C8        PUSHL   R0                          ;
                             7E  D4 002CA        CLRL    -(SP)                       ;
    00000000G  EF            05  FB 002CC        CALLS   #5, XST$APPEND              ;
               27 00000000G  EF  E9 002D3        BLBC    GCA+209, 16$                ; 1515
                        08   AE  9F 002DA        PUSHAB  $STR$STRING3                ; 1521
                      00A0   CA  9F 002DD        PUSHAB  $STR$STRING2                ;
                        18   AE  9F 002E1        PUSHAB  $STR$STRING1                ;
                      0098   CA  9F 002E4        PUSHAB  $STR$STRING0                ;
    00000000G  EF            04  FB 002E8        CALLS   #4, XST$JOIN                ;
                             5B  DD 002EF        PUSHL   R11                         ;
                             7E  D4 002F1        CLRL    -(SP)                       ;
                        08   AE  9F 002F3        PUSHAB  WORK_STRING                 ;
                             50  DD 002F6        PUSHL   R0                          ;
                             7E  D4 002F8        CLRL    -(SP)                       ;
    00000000G  EF            05  FB 002FA        CALLS   #5, XST$APPEND              ;
                      00B0   CA  9F 00301  16$:  PUSHAB  $STR$STRING1                ; 1526
                      00A8   CA  9F 00305        PUSHAB  $STR$STRING0                ;
    00000000G  EF            02  FB 00309        CALLS   #2, XST$JOIN                ;
                             5B  DD 00310        PUSHL   R11                         ;
                             7E  D4 00312        CLRL    -(SP)                       ;
                        08   AE  9F 00314        PUSHAB  WORK_STRING                 ;
                             50  DD 00317        PUSHL   R0                          ;
                             7E  D4 00319        CLRL    -(SP)                       ;
    00000000G  EF            05  FB 0031B        CALLS   #5, XST$APPEND              ;
                        08   A9  D4 00322        CLRL    FRA+12                      ; 1530
          FC   A9       0C   A9  9E 00325        MOVAB   FRA+16, FRA                 ;
               69       FC   A9  D0 0032A        MOVL    FRA, FRA+4                  ;
               53            6E  3C 0032E        MOVZWL  WORK_STRING, LEN            ;
               51       04   AE  D0 00331        MOVL    WORK_STRING+4, PTR          ;
          00   B9            1B  90 00335        MOVB    #27, @FRA+4                 ;
                             69  D6 00339        INCL    FRA+4                       ;
                        08   A9  D6 0033B        INCL    FRA+12                      ;
                             50  D4 0033E        CLRL    I                           ;
                             09  11 00340        BRB     18$                         ;
          00   B9            81  90 00342  17$:  MOVB    (PTR)+, @FRA+4              ;
                             69  D6 00346        INCL    FRA+4                       ;
                        08   A9  D6 00348        INCL    FRA+12                      ;
     F3        50            53  F3 0034B  18$:  AOBLEQ  LEN, I, 17$                 ;
                             0B  DD 0034F        PUSHL   #11                         ;
    00000000G  EF            01  FB 00351        CALLS   #1, CLH                     ;
                             7E  D4 00358  19$:  CLRL    -(SP)                       ; 1539
                             54  DD 0035A        PUSHL   DECSLPP                     ;
               7E     0903   8F  3C 0035C        MOVZWL  #2307, -(SP)                ;
    00000000G  EF            03  FB 00361        CALLS   #3, XST$ASCII               ;
                      00C0   CA  9F 00368        PUSHAB  $STR$STRING2                ;
                             50  DD 0036C        PUSHL   R0                          ;
```

```
                                00B8  CA  9F 0036E          PUSHAB  $STR$STRINGO                                    ;
                00000000G       EF        03  FB 00372          CALLS   #3, XST$JOIN                                ;
                                          5B  DD 00379          PUSHL   R11                                         ;
                                          7E  D4 0037B          CLRL    -(SP)                                       ;
                                    08    AE  9F 0037D          PUSHAB  $STR$TARGET                                 ;
                                          50  DD 00380          PUSHL   R0                                          ;
                                          7E  D4 00382          CLRL    -(SP)                                       ;
                00000000G       EF        05  FB 00384          CALLS   #5, XST$COPY                                ;
                                    08    A9  D4 0038B          CLRL    FRA+12                                      ; 1541
                      FC        A9    0C  A9  9E 0038E          MOVAB   FRA+16, FRA                                 ;
                                69        FC  A9  D0 00393          MOVL    FRA, FRA+4                                  ;
                                53            6E  3C 00397          MOVZWL  WORK_STRING, LEN                            ;
                                51        04  AE  D0 0039A          MOVL    WORK_STRING+4, PTR                          ;
                         00     B9            1B  90 0039E          MOVB    #27, @FRA+4                                 ;
                                          69  D6 003A2          INCL    FRA+4                                       ;
                                    08    A9  D6 003A4          INCL    FRA+12                                      ;
                                          50  D4 003A7          CLRL    I                                           ;
                                          09  11 003A9          BRB     21$                                         ;
                         00     B9            81  90 003AB 20$:      MOVB    (PTR)+, @FRA+4                              ;
                                          69  D6 003AF          INCL    FRA+4                                       ;
                                    08    A9  D6 003B1          INCL    FRA+12                                      ;
                F3                50            53  F3 003B4 21$:      AOBLEQ  LEN, I, 20$                         ;
                                          0B  DD 003B8          PUSHL   #11                                         ;
                00000000G       EF        01  FB 003BA          CALLS   #1, CLH                                     ;
                                          7E  D4 003C1          CLRL    -(SP)                                       ; 1556
                                          58  DD 003C3          PUSHL   TOP_MARGIN                                  ;
                                7E        0903  8F  3C 003C5          MOVZWL  #2307, -(SP)                                ;
                00000000G       EF        03  FB 003CA          CALLS   #3, XST$ASCII                               ;
                                54            50  D0 003D1          MOVL    R0, R4                                      ;
                                          7E  D4 003D4          CLRL    -(SP)                                       ;
                                          57  DD 003D6          PUSHL   BOTTOM_MARGIN                               ;
                                7E        0903  8F  3C 003D8          MOVZWL  #2307, -(SP)                                ;
                00000000G       EF        03  FB 003DD          CALLS   #3, XST$ASCII                               ;
                                    00D8  CA  9F 003E4          PUSHAB  $STR$STRING4                                ;
                                          50  DD 003E8          PUSHL   R0                                          ;
                                    00D0  CA  9F 003EA          PUSHAB  $STR$STRING2                                ;
                                          54  DD 003EE          PUSHL   R4                                          ;
                                    00C8  CA  9F 003F0          PUSHAB  $STR$STRINGO                                ;
                00000000G       EF        05  FB 003F4          CALLS   #5, XST$JOIN                                ;
                                          5B  DD 003FB          PUSHL   R11                                         ;
                                          7E  D4 003FD          CLRL    -(SP)                                       ;
                                    08    AE  9F 003FF          PUSHAB  $STR$TARGET                                 ;
                                          50  DD 00402          PUSHL   R0                                          ;
                                          7E  D4 00404          CLRL    -(SP)                                       ;
                00000000G       EF        05  FB 00406          CALLS   #5, XST$COPY                                ;
                                    08    A9  D4 0040D          CLRL    FRA+12                                      ; 1558
                      FC        A9    0C  A9  9E 00410          MOVAB   FRA+16, FRA                                 ;
                                69        FC  A9  D0 00415          MOVL    FRA, FRA+4                                  ;
                                53            6E  3C 00419          MOVZWL  WORK_STRING, LEN                            ;
                                51        04  AE  D0 0041C          MOVL    WORK_STRING+4, PTR                          ;
                         00     B9            1B  90 00420          MOVB    #27, @FRA+4                                 ;
                                          69  D6 00424          INCL    FRA+4                                       ;
                                    08    A9  D6 00426          INCL    FRA+12                                      ;
                                          50  D4 00429          CLRL    I                                           ;
                                          09  11 0042B          BRB     23$                                         ;
                         00     B9            81  90 0042D 22$:      MOVB    (PTR)+, @FRA+4                              ;
                                          69  D6 00431          INCL    FRA+4                                       ;
```

```
                          08   A9  D6 00433        INCL    FRA+12                  :
 F3            50              53  F3 00436  23$:  AOBLEQ  LEN, I, 22$             :
                               0B  DD 0043A        PUSHL   #11                     :
      00000000G  EF            01  FB 0043C        CALLS   #1, CLH                 :
                               7E  D4 00443        CLRL    -(SP)                   : 1573
                               56  DD 00445        PUSHL   LEFT_MARGIN             :
               7E       0903   8F  3C 00447        MOVZWL  #2307, -(SP)            :
      00000000G  EF            03  FB 0044C        CALLS   #3, XST$ASCII           :
               53              50  D0 00453        MOVL    R0, R3                  :
                               7E  D4 00456        CLRL    -(SP)                   :
                               55  DD 00458        PUSHL   RIGHT_MARGIN            :
               7E       0903   8F  3C 0045A        MOVZWL  #2307, -(SP)            :
      00000000G  EF            03  FB 0045F        CALLS   #3, XST$ASCII           :
                        00F0   CA  9F 00466        PUSHAB  $STR$STRING4            :
                               50  DD 0046A        PUSHL   R0                      :
                        00E8   CA  9F 0046C        PUSHAB  $STR$STRING2            :
                               53  DD 00470        PUSHL   R3                      :
                        00E0   CA  9F 00472        PUSHAB  $STR$STRING0            :
      00000000G  EF            05  FB 00476        CALLS   #5, XST$JOIN            :
                               5B  DD 0047D        PUSHL   R11                     :
                               7E  D4 0047F        CLRL    -(SP)                   :
                          08   AE  9F 00481        PUSHAB  $STR$TARGET             :
                               50  DD 00484        PUSHL   R0                      :
                               7E  D4 00486        CLRL    -(SP)                   :
      00000000G  EF            05  FB 00488        CALLS   #5, XST$COPY            :
                          08   A9  D4 0048F        CLRL    FRA+12                  : 1575
            FC   A9       0C   A9  9E 00492        MOVAB   FRA+16, FRA             :
                 69       FC   A9  D0 00497        MOVL    FRA, FRA+4              :
                 53            6E  3C 0049B        MOVZWL  WORK_STRING, LEN        :
                 51       04   AE  D0 0049E        MOVL    WORK_STRING+4, PTR      :
            00   B9            1B  90 004A2        MOVB    #27, @FRA+4             :
                               69  D6 004A6        INCL    FRA+4                   :
                          08   A9  D6 004A8        INCL    FRA+12                  :
                               50  D4 004AB        CLRL    I                       :
                               09  11 004AD        BRB     25$                     :
            00   B9            81  90 004AF  24$:  MOVB    (PTR)+, @FRA+4          :
                               69  D6 004B3        INCL    FRA+4                   :
                          08   A9  D6 004B5        INCL    FRA+12                  :
 F3            50              53  F3 004B8  25$:  AOBLEQ  LEN, I, 24$             :
                               0B  DD 004BC        PUSHL   #11                     :
      00000000G  EF            01  FB 004BE        CALLS   #1, CLH                 :
 5B         52   A2            02  E1 004C5        BBC     #2, 82(R2), 28$         : 1585
                        0108   CA  9F 004CA        PUSHAB  $STR$STRING2            : 1591
                        0100   CA  9F 004CE        PUSHAB  $STR$STRING1            :
                        00F8   CA  9F 004D2        PUSHAB  $STR$STRING0            :
      00000000G  EF            03  FB 004D6        CALLS   #3, XST$JOIN            :
                               5B  DD 004DD        PUSHL   R11                     :
                               7E  D4 004DF        CLRL    -(SP)                   :
                          08   AE  9F 004E1        PUSHAB  $STR$TARGET             :
                               50  DD 004E4        PUSHL   R0                      :
                               7E  D4 004E6        CLRL    -(SP)                   :
      00000000G  EF            05  FB 004E8        CALLS   #5, XST$COPY            :
                          08   A9  D4 004EF        CLRL    FRA+12                  : 1593
            FC   A9       0C   A9  9E 004F2        MOVAB   FRA+16, FRA             :
                 69       FC   A9  D0 004F7        MOVL    FRA, FRA+4              :
                 52            6E  3C 004FB        MOVZWL  WORK_STRING, LEN        :
                 51       04   AE  D0 004FE        MOVL    WORK_STRING+4, PTR      :
```

```
          00  B9          1B  90 00502        MOVB    #27, @FRA+4              ;
                          69  D6 00506        INCL    FRA+4                    ;
                    08    A9  D6 00508        INCL    FRA+12                   ;
                          50  D4 0050B        CLRL    I                        ;
                          09  11 0050D        BRB     27$                      ;
          00  B9          81  90 0050F 26$:   MOVB    (PTR)+, @FRA+4           ;
                          69  D6 00513        INCL    FRA+4                    ;
                    08    A9  D6 00515        INCL    FRA+12                   ;
    F3            50      52  F3 00518 27$:   AOBLEQ  LEN, I, 26$              ;
                          0B  DD 0051C        PUSHL   #11                      ;
   00000000G  EF          01  FB 0051E        CALLS   #1, CLH                  ;
                          5B  DD 00525 28$:   PUSHL   R11                      ; 1600
                          7E  D4 00527        CLRL    -(SP)                    ;
                    08    AE  9F 00529        PUSHAB  $STR$TARGET              ;
                  0110    CA  9F 0052C        PUSHAB  $STR$STRING              ;
                          7E  D4 00530        CLRL    -(SP)                    ;
   00000000G  EF          05  FB 00532        CALLS   #5, XST$COPY             ;
                    08    A9  D4 00539        CLRL    FRA+12                   ; 1601
          FC  A9    0C    A9  9E 0053C        MOVAB   FRA+16, FRA              ;
              69    FC    A9  D0 00541        MOVL    FRA, FRA+4               ;
              52          6E  3C 00545        MOVZWL  WORK_STRING, LEN         ;
              51    04    AE  D0 00548        MOVL    WORK_STRING+4, PTR       ;
          00  B9          1B  90 0054C        MOVB    #27, @FRA+4              ;
                          69  D6 00550        INCL    FRA+4                    ;
                    08    A9  D6 00552        INCL    FRA+12                   ;
                          50  D4 00555        CLRL    I                        ;
                          09  11 00557        BRB     30$                      ;
          00  B9          81  90 00559 29$:   MOVB    (PTR)+, @FRA+4           ;
                          69  D6 0055D        INCL    FRA+4                    ;
                    08    A9  D6 0055F        INCL    FRA+12                   ;
    F3            50      52  F3 00562 30$:   AOBLEQ  LEN, I, 29$              ;
                          0B  DD 00566        PUSHL   #11                      ;
   00000000G  EF          01  FB 00568        CALLS   #1, CLH                  ;
                          5B  DD 0056F        PUSHL   R11                      ; 1608
                          7E  D4 00571        CLRL    -(SP)                    ;
                    08    AE  9F 00573        PUSHAB  $STR$TARGET              ;
                  0118    CA  9F 00576        PUSHAB  $STR$STRING              ;
                          7E  D4 0057A        CLRL    -(SP)                    ;
   00000000G  EF          05  FB 0057C        CALLS   #5, XST$COPY             ;
                    08    A9  D4 00583        CLRL    FRA+12                   ; 1609
          FC  A9    0C    A9  9E 00586        MOVAB   FRA+16, FRA              ;
              69    FC    A9  D0 0058B        MOVL    FRA, FRA+4               ;
              52          6E  3C 0058F        MOVZWL  WORK_STRING, LEN         ;
              51    04    AE  D0 00592        MOVL    WORK_STRING+4, PTR       ;
          00  B9          1B  90 00596        MOVB    #27, @FRA+4              ;
                          69  D6 0059A        INCL    FRA+4                    ;
                    08    A9  D6 0059C        INCL    FRA+12                   ;
                          50  D4 0059F        CLRL    I                        ;
                          09  11 005A1        BRB     32$                      ;
          00  B9          81  90 005A3 31$:   MOVB    (PTR)+, @FRA+4           ;
                          69  D6 005A7        INCL    FRA+4                    ;
                    08    A9  D6 005A9        INCL    FRA+12                   ;
    F3            50      52  F3 005AC 32$:   AOBLEQ  LEN, I, 31$              ;
                          0B  DD 005B0        PUSHL   #11                      ;
   00000000G  EF          01  FB 005B2        CALLS   #1, CLH                  ;
              00000000G   EF  D6 005B9        INCL    PHAN+12                  ; 1615
                          04  005BF           RET                              ; 1617
```

**FI

```
; Routine Size:  1472 bytes,    Routine Base:  $CODE$ + 08BB

; 1514          1618  1 END                                        !End of module
; 1515          1619  0 ELUDOM
```

```
;                                PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $OWN$                          396  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                         336  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        3707  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       183        31         252        00:00.1
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1   1248       204        16          86        00:00.3
```

```
;                                COMMAND QUALIFIERS

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

; Size:          3707 code + 732 data bytes
; Run Time:        02:43.8
; Elapsed Time:    05:40.7
; Lines/CPU Min:     593
; Lexemes/CPU-Min: 97016
; Memory Used:  734 pages
; Compilation Complete
```