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

```
_$2

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

RLI
RNO
RNO
RTY
SAV
STR
STR
STR
STR
STR
STR

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

```
  SSSSSSSS  TTTTTTTTTT  KK      KK  FFFFFFFFFF  RRRRRRRR    MM      MM
  SSSSSSSS  TTTTTTTTTT  KK      KK  FFFFFFFFFF  RRRRRRRR    MM      MM
SS              TT      KK      KK  FF          RR      RR  MMMM  MMMM
SS              TT      KK      KK  FF          RR      RR  MMMM  MMMM
SS              TT      KK    KK    FF          RR      RR  MM  MM  MM
SS              TT      KK    KK    FF          RR      RR  MM  MM  MM
  SSSSSS        TT      KKKKKK      FFFFFFFF    RRRRRRRR    MM      MM
  SSSSSS        TT      KKKKKK      FFFFFFFF    RRRRRRRR    MM      MM
        SS      TT      KK    KK    FF          RR  RR      MM      MM
        SS      TT      KK    KK    FF          RR  RR      MM      MM
        SS      TT      KK      KK  FF          RR    RR    MM      MM
        SS      TT      KK      KK  FF          RR    RR    MM      MM
SSSSSSSS        TT      KK      KK  FF          RR      RR  MM      MM
SSSSSSSS        TT      KK      KK  FF          RR      RR  MM      MM

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 'Misc save/restore for nesting directives'
;  2          0002  0 MODULE STKFRM (
;  3          0003  0                 IDENT = 'V04-000'
;  4        P 0004  0 %BLISS32[,
;  5        P 0005  0         ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE,NONEXTERNAL=LONG_RELATIVE)
;  6          0006  0         ]
;  7          0007  0                 ) =
;  8          0008  1 BEGIN
;  9          0009  1 !
; 10          0010  1 !*****************************************************************************
; 11          0011  1 !*                                                                           *
; 12          0012  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
; 13          0013  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
; 14          0014  1 !*  ALL RIGHTS RESERVED.                                                     *
; 15          0015  1 !*                                                                           *
; 16          0016  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
; 17          0017  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 18          0018  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE  THIS SOFTWARE OR  ANY  OTHER   *
; 19          0019  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
; 20          0020  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
; 21          0021  1 !*  TRANSFERRED.                                                             *
; 22          0022  1 !*                                                                           *
; 23          0023  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
; 24          0024  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
; 25          0025  1 !*  CORPORATION.                                                             *
; 26          0026  1 !*                                                                           *
; 27          0027  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
; 28          0028  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
; 29          0029  1 !*                                                                           *
; 30          0030  1 !*                                                                           *
; 31          0031  1 !*****************************************************************************
; 32          0032  1 !
; 33          0033  1 !++
; 34          0034  1 ! FACILITY:      DSR (Digital Standard RUNOFF) / DSRPLUS
; 35          0035  1 !
; 36          0036  1 ! ABSTRACT: Used by .NOTE, .LIST, .END NOTE and .END LIST to save/restore
; 37          0037  1 !                 various formatting information.
; 38          0038  1 !
; 39          0039  1 !
; 40          0040  1 ! ENVIRONMENT: Transportable
; 41          0041  1 !
; 42          0042  1 ! AUTHOR: R.W.Friday     CREATION DATE: June, 1978
; 43          0043  1 !
```

```
;    4          0044  1 %SBTTL 'Revision History'
;               0045  1 !
;               0046  1 ! MODIFIED BY:
;               0047  1 !
;    48         0048  1 !       002     REM00002        Ray Marshall    07-Mar-1983
;    49         0049  1 !               Global edit of all modules. Updated module names, idents,
;    50         0050  1 !               copyright dates. Changed require files to BLISS library.
;    51         0051  1 !
;    52         0052  1 !--
```

```
;    55          0053  1 %SBTTL 'Module Level Declarations'
;    56          0054  1 !
;    57          0055  1 ! TABLE OF CONTENTS:
;    58          0056  1 !
;    59          0057  1 !
;    60          0058  1 ! INCLUDE FILES:
;    61          0059  1 !
;    62          0060  1
;    63          0061  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Li ·ary
;    64          0062  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    65          0193  1
;    66        U 0194  1 %IF DSRPLUS %THEN
;    67        U 0195  1 LIBRARY 'REQ:DPLLIB';                 ! DSRPLUS BLISS Library
;    68          0196  1 %ELSE
;    69          0197  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    70          0198  1 %FI
;    71          0199  1
;    72          0200  1 !
;    73          0201  1 ! MACROS:
;    74          0202  1 !
;    75          0203  1 !
;    76          0204  1 ! EQUATED SYMBOLS:
;    77          0205  1 !
;    78          0206  1 !
;    79          0207  1 ! OWN STORAGE:
;    80          0208  1 !
;    81          0209  1 !
;    82          0210  1 ! EXTERNAL REFERENCES:
;    83          0211  1 !
;    84          0212  1
;    85          0213  1 EXTERNAL
;    86          0214  1     FRMSTD,
;    87          0215  1     FRMSTK : FORM_STACK,
;    88          0216  1     GCA : GCA_DEFINITION,
;    89          0217  1     IRAC : IRAC_DEFINITION,
;    90          0218  1     SCA : SCA_DEFINITION;
;    91          0219  1
;    92          0220  1 EXTERNAL LITERAL                        !Error messages
;    93          0221  1     RNFNSF,
;    94          0222  1     RNFSTD,
;    95          0223  1     RNFUME;
;    96          0224  1
;    97          0225  1 EXTERNAL ROUTINE
;    98          0226  1     ERM,
;    99          0227  1     ERMA,
;   100          0228  1     PNTBAC;
;   101          0229  1
```

```
;  103          0230  1 %SBTTL 'STKFRM --'
;  104          0231  1 GLOBAL ROUTINE stkfrm (handler_code) : =
;  105          0232  1
;  106          0233  1 !++
;  107          0234  1 ! FUNCTIONAL DESCRIPTION:
;  108          0235  1 !
;  109          0236  1 !       See ABSTRACT, above.
;  110          0237  1 !
;  111          0238  1 ! FORMAL PARAMETERS:
;  112          0239  1 !
;  113          0240  1 !       Handler_code indicates which command requires formatting
;  114          0241  1 !       information to be stacked or unstacked.
;  115          0242  1 !       The value 0 means push without checking; the value -1
;  116          0243  1 !       means pop without checking.  These two values are provided
;  117          0244  1 !       for the indexing routines, which have to save this information.
;  118          0245  1 !       Values -1 and 0 are also used for handling literals.
;  119          0246  1 !       When FRMSTK is allocated, there will be one extra place allocated
;  120          0247  1 !       so that at least one entry will remain, even when .NOTES
;  121          0248  1 !       and .LISTS are nested to their limits.
;  122          0249  1 !
;  123          0250  1 ! IMPLICIT INPUTS:      None
;  124          0251  1 !
;  125          0252  1 ! IMPLICIT OUTPUTS:     None
;  126          0253  1 !
;  127          0254  1 ! ROUTINE VALUE:
;  128          0255  1 ! COMPLETION CODES:
;  129          0256  1 !
;  130          0257  1 !       Returns TRUE if no errors were detected.  Returns FALSE
;  131          0258  1 !       if the operation was not done.
;  132          0259  1 !
;  133          0260  1 ! SIDE EFFECTS:         None
;  134          0261  1 !
;  135          0262  1 !--
;  136          0263  1
;  137          0264  2     BEGIN
;  138          0265  2
;  139          0266  2     SELECT .handler_code OF
;  140          0267  2         SET
;  141          0268  2
;  142          0269  2         [h_note, h_list] :
;  143          0270  3             BEGIN
;  144          0271  3
;  145          0272  3             IF  .frmstd EQL frmstk_limit  THEN
;  146          0273  4                 BEGIN                           !Ignore command because of stack overflow
;  147          0274  4                 erma (rnfstd, false);
;  148          0275  4                 RETURN false;
;  149          0276  3                 END;
;  150          0277  3
;  151          0278  2             END;
;  152          0279  2
;  153          0280  2         [0, h_note, h_list] :
;  154          0281  3             BEGIN
;  155          0282  3             !Save items on stack.
;  156          0283  3             frmstd = .frmstd + 1;
;  157          0284  3             frmstk [.frmstd, frmstk_fill]       = .sca_fill;
;  158          0285  3             frmstk [.frmstd, frmstk_justify]    = .sca_justify;
;  159          0286  3             frmstk [.frmstd, frmstk_crock]      = .sca_crock;
```

```
;  160     0287  3            frmstk [.frmstd, frmstk_rm]          = .sca_rm;
;  161     0288  3            frmstk [.frmstd, frmstk_lm]          = .sca_lm;
;  162     0289  3            frmstk [.frmstd, frmstk_spacing]     = .sca_spacing;
;  163     0290  3            frmstk [.frmstd, frmstk_command]     = .handler_code;
;  164     0291  3            frmstk [.frmstd, frmstk_iseqn]       = .irac_iseqn;
;  165     0292  3            frmstk [.frmstd, frmstk_ipagen]      = .irac_ipagen;
;  166     0293  3            frmstk [.frmstd, frmstk_fspecp]      = .irac_fspecp;
;  167     0294  3            frmstk [.frmstd, frmstk_fspecc]      = .irac_fspecc;
;  168     0295  3            frmstk [.frmstd, frmstk_req_d]       = .gca_req_depth;
;  169     0296  3            frmstk [.frmstd, frmstk_do_nbits]    = .sca_do_nbits;
;  170     0297  3            frmstk [.frmstd, frmstk_ker]         = .sca_ker;
;  171     0298  3            frmstk [ frmstd, frmstk_index]       = .sca_index;
;  172     0299  3            RETURN true;
;  173     0300  2            END;
;  174     0301  2 !
;  175     0302  2        !Some general error checking before popping the stack
;  176     0303  2
;  177     0304  2        [h_end_note, h_end_list] :
;  178     0305  3            BEGIN
;  179     0306  3
;  180     0307  3            IF  .frmstd EQL 0  THEN
;  181     0308  4                BEGIN                              !The stack is empty
;  182     0309  4                erma (rnfume, false);
;  183     0310  4                RETURN false;
;  184     0311  3                END;
;  185     0312  3
;  186     0313  2            END;
;  187     0314  2
;  188     0315  2        [h_end_note] :
;  189     0316  3            BEGIN
;  190     0317  3
;  191     0318  3            IF  .frmstk [.frmstd, frmstk_command] NEQ h_note  THEN
;  192     0319  4                BEGIN                              !The entry was not made by a .NOTE command.
;  193     0320  4                erma (rnfume, false);
;  194     0321  4                pntbac ();
;  195     0322  4                RETURN false;
;  196     0323  3                END;
;  197     0324  3
;  198     0325  2            END;
;  199     0326  2
;  200     0327  2        [h_end_list] :
;  201     0328  3            BEGIN
;  202     0329  3
;  203     0330  3            IF  .frmstk [.frmstd, frmstk_command] NEQ h_list  THEN
;  204     0331  4                BEGIN                              !The entry was not made by a .LIST command.
;  205     0332  4                erma (rnfume, false);
;  206     0333  4                pntbac ();
;  207     0334  4                RETURN false;
;  208     0335  3                END;
;  209     0336  3
;  210     0337  2            END;
;  211     0338  2 !
;  212     0339  2        !Pop the stack.  Give a warning message if the .END command is not
;  213     0340  2        !in the same file as the command that started the list or note.
;  214     0341  2
;  215     0342  2        [h_end_note, -1, h_end_list] :
;  216     0343  3            BEGIN
```

```
;  217          0344  3
;  218          0345  3              IF  .frmstk [.frmstd, frmstk_req_d] NEQ .gca_req_depth  THEN
;  219          0346  4                  BEGIN
;  220          0347  4                  erma (rnfnsf, false);
;  221          0348  4                  pntbac ();
;  222          0349  3                  END;
;  223          0350  3
;  224          0351  3              !Restore items from stack.
;  225          0352  3              sca_fill    = .frmstk [.frmstd, frmstk_fill];
;  226          0353  3              sca_justify = .frmstk [.frmstd, frmstk_justify];
;  227          0354  3              sca_crock   = .frmstk [.frmstd, frmstk_crock];
;  228          0355  3              sca_rm      = .frmstk [.frmstd, frmstk_rm];
;  229          0356  3              sca_lm      = .frmstk [.frmstd, frmstk_lm];
;  230          0357  3              sca_spacing = .frmstk [.frmstd, frmstk_spacing];
;  231          0358  3              sca_do_nbits= .frmstk [.frmstd, frmstk_do_nbits];
;  232          0359  3              sca_ker     = .frmstk [.frmstd, frmstk_ker];
;  233          0360  3              sca_index   = .frmstk [.frmstd, frmstk_index];
;  234          0361  3              frmstd      = .frmstd - 1;
;  235          0362  3              RETURN true;
;  236          0363  2              END;
;  237          0364  2          TES;
;  238          0365  2          true
;  239          0366  1      END;                                    !End of STKFRM


                                                .TITLE  STKFRM Misc save/restore for nesting directives
                                                .IDENT  \V04-000\

                                                .EXTRN  FRMSTD, FRMSTK, GCA
                                                .EXTRN  IRAC, SCA, RNFNSF
                                                .EXTRN  RNFSTD, RNFUME, ERM
                                                .EXTRN  ERMA, PNTBAC

                                                .PSECT  $CODE$,NOWRT,2

                                    07FC 00000          .ENTRY  STKFRM, Save R2,R3,R4,R5,R6,R7,R8,R9,R10    ; 0231
                5A 00000000G    EF    9E 00002          MOVAB   PNTBAC, R10                                 ;
                59 00000000G    8F    D0 00009          MOVL    #RNFUME, R9                                 ;
                58 00000000G    EF    9E 00010          MOVAB   GCA+188, R8                                 ;
                57 00000000G    EF    9E 00017          MOVAB   ERMA, R7                                    ;
                56 00000000G    EF    9E 0001E          MOVAB   IRAC+8, R6                                  ;
                55 00000000G    EF    9E 00025          MOVAB   FRMSTD, R5                                  ;
                54 00000000G    EF    9E 0002C          MOVAB   SCA+104, R4                                 ;
                53 00000000G    EF    9E 00033          MOVAB   FRMSTK-40, R3                               ;
                52          04  AC    D0 0003A          MOVL    HANDLER_CODE, R2                            ; 0266
0000006E        8F              52    D1 0003E          CMPL    R2, #110                                    ; 0269
                                09    13 00045          BEQL    1$                                          ;
00000072        8F              52    D1 00047          CMPL    R2, #114                                    ;
                                10    12 0004E          BNEQ    2$                                          ;
                0F              65    D1 00050 1$:      CMPL    FRMSTD, #15                                 ; 0272
                                0B    12 00053          BNEQ    2$                                          ;
                                7E    D4 00055          CLRL    -(SP)                                       ; 0274
                    00000000G   8F    DD 00057          PUSHL   #RNFSTD                                     ;
                          0087  31    31 0005D          BRW     6$                                          ;
                                52    D5 00060 2$:      TSTL    R2                                          ; 0280
                                12    13 00062          BEQL    3$                                          ;
0000006E        8F              52    D1 00064          CMPL    R2, #110                                    ;
```

```
                              09  13 0006B       BEQL    3$                            :
   00000072  8F               52  D1 0006D       CMPL    R2, #114                      :
                              5F  12 00074       BNEQ    4$                            :
                              65  D6 00076 3$:   INCL    FRMSTD                        : 0283
50           65               0F  C5 00078       MULL3   #15, FRMSTD, R0               : 0284
        EC A340        00     B4  D0 0007C       MOVL    @SCA+104, FRMSTK-60[R0]       :
        F0 A340        FC     B4  D0 00082       MOVL    @SCA+100, FRMSTK-56[R0]       : 0285
        18 A340        08     B4  D0 00088       MOVL    @SCA+112, FRMSTK-16[R0]       : 0286
        F4 A340        10     B4  D0 0008E       MOVL    @SCA+120, FRMSTK-52[R0]       : 0287
        F8 A340        0C     B4  D0 00094       MOVL    @SCA+116, FRMSTK-48[R0]       : 0288
        FC A340        14     B4  D0 0009A       MOVL    @SCA+124, FRMSTK-44[R0]       : 0289
           6340               52  D0 000A0       MOVL    R2, FRMSTK-40[R0]             : 0290
        04 A340               66  D0 000A4       MOVL    IRAC+8, FRMSTK-36[R0]         : 0291
        08 A340        04     A6  D0 000A9       MOVL    IRAC+12, FRMSTK-32[R0]        : 0292
        0C A340        08     A6  D0 000AF       MOVL    IRAC+16, FRMSTK-28[R0]        : 0293
        10 A340        0C     A6  D0 000B5       MOVL    IRAC+20, FRMSTK-24[R0]        : 0294
        14 A340               68  D0 000BB       MOVL    GCA+188, FRMSTK-20[R0]        : 0295
        1C A340        40     A4  D0 000C0       MOVL    SCA+168, FRMSTK-12[R0]        : 0296
        20 A340        1C     B4  D0 000C6       MOVL    @SCA+132, FRMSTK-8[R0]        : 0297
        24 A340        6C     A4  D0 000CC       MOVL    SCA+212, FRMSTK-4[R0]         : 0298
                            00B2  31 000D2       BRW     14$                           : 0299
             3B               52  D1 000D5 4$:   CMPL    R2, #59                       : 0304
                              05  13 000D8       BEQL    5$                            :
             3D               52  D1 000DA       CMPL    R2, #61                       :
                              0D  12 000DD       BNEQ    7$                            :
                              65  D5 000DF 5$:   TSTL    FRMSTD                        : 0307
                              09  12 000E1       BNEQ    7$                            :
                              7E  D4 000E3       CLRL    -(SP)                         : 0309
                              59  DD 000E5       PUSHL   R9                            :
             67               02  FB 000E7 6$:   CALLS   #2, ERMA                      :
                              30  11 000EA       BRB     10$                           : 0310
             3D               52  D1 000EC 7$:   CMPL    R2, #61                       : 0315
                              0E  12 000EF       BNEQ    8$                            :
50           65               0F  C5 000F1       MULL3   #15, FRMSTD, R0               : 0318
   00000072  8F             6340  D1 000F5       CMPL    FRMSTK-40[R0], #114           :
                              13  12 000FD       BNEQ    9$                            :
             3B               52  D1 000FF 8$:   CMPL    R2, #59                       : 0327
                              1B  12 00102       BNEQ    11$                           :
50           65               0F  C5 00104       MULL3   #15, FRMSTD, R0               : 0330
   0000006E  8F             6340  D1 00108       CMPL    FRMSTK-40[R0], #110           :
                              0D  13 00110       BEQL    11$                           :
                              7E  D4 00112 9$:   CLRL    -(SP)                         : 0332
                              59  DD 00114       PUSHL   R9                            :
             67               02  FB 00116       CALLS   #2, ERMA                      :
             6A               00  FB 00119       CALLS   #0, PNTBAC                    : 0333
                              50  D4 0011C 10$:  CLRL    R0                            : 0334
                                  04 0011E       RET                                   :
   FFFFFFFF  8F               52  D1 0011F 11$:  CMPL    R2, #-1                       : 0342
                              0A  13 00126       BEQL    12$                           :
             3B               52  D1 00128       CMPL    R2, #59                       :
                              05  13 0012B       BEQL    12$                           :
             3D               52  D1 0012D       CMPL    R2, #61                       :
                              55  12 00130       BNEQ    14$                           :
50           65               0F  C5 00132 12$:  MULL3   #15, FRMSTD, R0               : 0345
             68          14 A340  D1 00136       CMPL    FRMSTK-20[R0], GCA+188        :
                              0E  13 0013B       BEQL    13$                           :
                              7E  D4 0013D       CLRL    -(SP)                         : 0347
```

```
                    00000000G  8F   DD 0013F          PUSHL   #RNFNSF                                  ;
                           67      02   FB 00145          CALLS   #2, ERMA                                 ;
                           6A      00   FB 00148          CALLS   #0, PNTBAC                               ; 0348
                           50      65   0F   C5 0014B 13$:    MULL3   #15, FRMSTD, R0                          ; 0352
                        00   B4      EC A340   D0 0014F          MOVL    FRMSTK-60[R0], @SCA+104                  ;
                        FC   B4      F0 A340   D0 00155          MOVL    FRMSTK-56[R0], @SCA+100                  ; 0353
                        08   B4      18 A340   D0 0015B          MOVL    FRMSTK-16[R0], @SCA+112                  ; 0354
                        10   B4      F4 A340   D0 00161          MOVL    FRMSTK-52[R0], @SCA+120                  ; 0355
                        0C   B4      F8 A340   D0 00167          MOVL    FRMSTK-48[R0], @SCA+116                  ; 0356
                        14   B4      FC A340   D0 0016D          MOVL    FRMSTK-44[R0], @SCA+124                  ; 0357
                        40   A4      1C A340   D0 00173          MOVL    FRMSTK-12[R0], SCA+168                   ; 0358
                        1C   B4      20 A340   D0 00179          MOVL    FRMSTK-8[R0], @SCA+132                   ; 0359
                        6C   A4      24 A340   D0 0017F          MOVL    FRMSTK-4[R0], SCA+212                    ; 0360
                                        65   D7 00185          DECL    FRMSTD                                   ; 0361
                                50      01   D0 00187 14$:    MOVL    #1, R0                                   ; 0366
                                             04 0018A          RET                                              ;

; Routine Size:  395 bytes,    Routine Base:  $CODE$ + 0000

;  240          0367  1
;  241          0368  1 END                                                 !End of module
;  242          0369  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                            Bytes                           Attributes
;
;  $CODE$                                395  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]XPORT.L32;1           590         0         0         252       00:00.1
;  _$255$DUA28:[RUNOFF.SRC]DSRLIB.L32;1     1248        43         3          86       00:00.3



;                                      COMMAND QUALIFIERS
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:STKFRM/OBJ=OBJ$:STKFRM MSRC$:STKFRM/UPDATE=(ENH$:STKFRM)
; Size:          395 code + 0 data bytes
```

```
; Run Time:          00:09.6
; Elapsed Time:      00:29.6
; Lines/CPU Min:      2311
; Lexemes/CPU-Min: 15526
; Memory Used:  129 pages
; Compilation Complete
```