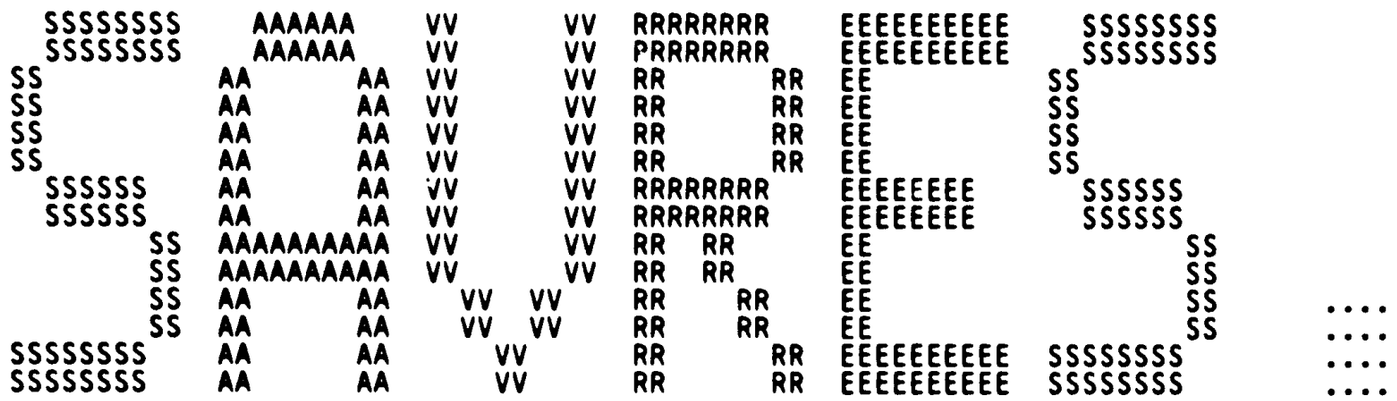
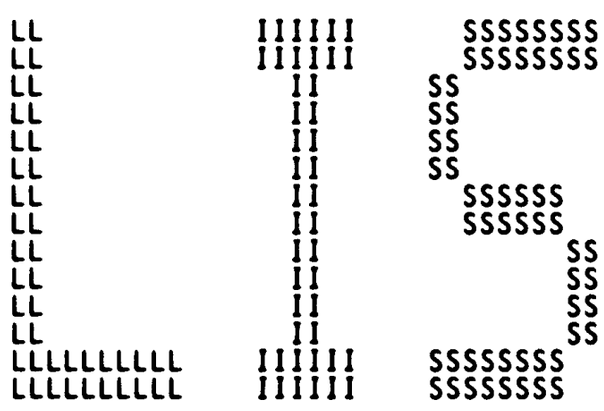
```
RRRRRRRRRRR      UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
RRRRRRRRRRRR     UUU          UUU  NNN         NNN     000000000     FFFFFFFFFFFFFFF   FFFFFFFFFFFFFFF
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
```

FILEID**SAVRES





```
;     1          0001  0 %TITLE 'SAVRES -- Save & Restore'
;     2          0002  0 MODULE SAVRES   (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 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    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:   SAVE & RESTORE handling and initialization routines
;    37          0037  1 !
;    38          0038  1 ! ENVIRONMENT:   Transportable
;    39          0039  1 !
;    40          0040  1 ! AUTHOR: Ken Alden  CREATION DATE: January, 1983
;    41          0041  1 !
```

```
;  43          0042  1 %SBTTL 'Revision History'
;  44          0043  1 !
;  45          0044  1 ! MODIFIED BY:
;  46          0045  1 !
;  47          0046  1 !       010     REM00010        Ray Marshall    16-November-1983
;  48          0047  1 !               Changed numeric offsets used in routine SAVINI to using macros
;  49          0048  1 !                 so that future changes to GCA, HCT, PHDEF, and/or SCA will
;  50          0049  1 !                 have less chance of resulting in failures of the SAVE/RESTORE
;  51          0050  1 !                 directives.
;  52          0051  1 !
;  53          0052  1 !       009     KFA00009        Ken Alden       30-Sep-1983
;  54          0053  1 !               Moved some the the variables from save all to save and
;  55          0054  1 !                 vice versa.
;  56          0055  1 !
;  57          0056  1 !       008     KFA00008        Ken Alden       27-Jul-1983
;  58          0057  1 !               Modified SAVINI with the addition of gca_flag_cmd.
;  59          0058  1 !
;  60          0059  1 !       007     KFA00007        Ken Alden       5-Jul-1983
;  61          0060  1 !               Modified SAVINI with the addition of sca_flags.
;  62          0061  1 !
;  63          0062  1 !       006     KAD00006        Keith Dawson    12-May-1983
;  64          0063  1 !               Modified SAVINI slightly -- having adjusted some of the
;  65          0064  1 !                 storage locations in GCA.REQ.
;  66          0065  1 !
;  67          0066  1 !       005     KFA00005        Ken Alden       3-May-1983
;  68          0067  1 !               Added conditional for debugging so XPO_PUTs will only
;  69          0068  1 !                 be put if the output file if open.
;  70          0069  1 !
;  71          0070  1 !       004     KFA00004        Ken Alden       25-Apr-1983
;  72          0071  1 !               Added support for /DEBUG=SAVE.
;  73          0072  1 !
;  74          0073  1 !       003     KFA00003        Ken Alden       25-Apr-1983
;  75          0074  1 !               Fixed the block_copy macro so it will pick up the first
;  76          0075  1 !               word in the save block properly.
;  77          0076  1 !
;  78          0077  1 !       002     REM00002        Ray Marshall    07-Mar-1983
;  79          0078  1 !               Global edit of all modules. Updated module names, idents,
;  80          0079  1 !               copyright dates. Changed require files to BLISS library.
;  81          0080  1 !
;  82          0081  1 !--
```

```
;    84          0082  1 %SBTTL 'Module Level Declarations'
;    85          0083  1 !
;    86          0084  1 ! TABLE OF CONTENTS:
;    87          0085  1 !
;    88          0086  1 !
;    89          0087  1 ! INCLUDE FILES:
;    90          0088  1 !
;    91          0089  1
;    92          0090  1 LIBRARY 'NXPORT:XPORT';                 ! XPORT Library
;    93          0091  1 REQUIRE 'REQ:RNODEF';                   ! RUNOFF variant definitions
;    94          0222  1
;    95        U 0223  1 %IF DSRPLUS %THEN
;    96        U 0224  1 LIBRARY 'REQ:DPLLIB';                   ! DSRPLUS BLISS Library
;    97          0225  1 %ELSE
;    98          0226  1 LIBRARY 'REQ:DSRLIB';                   ! DSR BLISS Library
;    99          0227  1 %FI
;   100          0228  1
;   101          0229  1 !
;   102          0230  1 ! MACROS:
;   103          0231  1 !
;   104          0232  1   MACRO
;   105          0233  1     block_copy (in_adr, length, out_adr)
;   106        M 0234  1       =
;   107        M 0235  1           BEGIN
;   108        M 0236  1           BIND FOO = in_adr : vector;
;   109        M 0237  1           BIND BAR = out_adr :vector;
;   110        M 0238  1
;   111        M 0239  1           INCR I FROM 0 to (length - 1) DO
;   112        M 0240  1              BAR [.I] = .FOO [.I];
;   113        M 0241  1
;   114        M 0242  1           END
;   115          0243  1           %;
;   116          0244  1
;   117          0245  1 ! EQUATED SYMBOLS:
;   118          0246  1 !
;   119          0247  1 ! OWN STORAGE:
;   120          0248  1 !
;   121          0249  1 !
;   122          0250  1 ! EXTERNAL REFERENCES:
;   123          0251  1 !
;   124          0252  1 EXTERNAL LITERAL
;   125          0253  1     rnfloc,
;   126          0254  1     rnftms,                                                    ; R
;   127          0255  1     rnfwfr,
;   128          0256  1     rnfncs;
;   129          0257  1
;   130          0258  1 EXTERNAL
;   131          0259  1     rnoiob : REF $XPO_IOB(),
;   132          0260  1     irac : irac_definition,
;   133          0261  1     savstk : savstack,
;   134          0262  1     flgt : flgt_definition,
;   135          0263  1     sca : sca_definition,
;   136          0264  1     hct : hct_definition,
;   137          0265  1     gca : gca_definition,
;   138          0266  1     phan : phan_definition,
;   139          0267  1     save : $save_block,
;   140          0268  1     saveall : $saveall_block;
```

```
;  141          0269  1
;  142          0270  1  EXTERNAL ROUTINE
;  143          0271  1          erm,
;  144          0272  1          erma;
```

```
;  146          0273  1 %SBTTL 'SAVRES -- Save & restore handler module'
;  147          0274  1 GLOBAL ROUTINE savres (handler_code,internal_code) : NOVALUE =
;  148          0275  1
;  149          0276  1 !++
;  150          0277  1 ! FUNCTIONAL DESCRIPTION:
;  151          0278  1 !
;  152          0279  1 !       This routine performs the SAVEs and RESTOREs.
;  153          0280  1 !       This routine will perform up to 20 save/save alls and after that it
;  154          0281  1 !       will remember ten more if the user happened to accidently try a .SAVE
;  155          0282  1 !       The reason for remembering is to more informative to the user by
;  156          0283  1 !       pointing back to where the illegal save occured.  After ten illegal
;  157          0284  1 !       SAVEs, this routine will just ignore the whole command. This of course
;  158          0285  1 !       will skew the restores but we figured, who cares at that point?
;  159          0286  1 !
;  160          0287  1 !       One additional item I would like to point out is that the maximum
;  161          0288  1 !       number of saves(20) is SAVSTK_SIZE.
;  162          0289  1 !
;  163          0290  1 !       The maiximum number of recorded save commands(30) is STKSAV_SAVE.
;  164          0291  1 !
;  165          0292  1 ! FORMAL PARAMETERS:
;  166          0293  1 !
;  167          0294  1 !       HANDLER_CODE indicates which command is to be processed.
;  168          0295  1 !       INTERNAL_CODE indicates where this routine was called: utilities(-1),
;  169          0296  1 !       RSDERM (=2), DOCM(1).
;  170          0297  1 !
;  171          0298  1 ! IMPLICIT INPUTS:      none
;  172          0299  1 !
;  173          0300  1 ! IMPLICIT OUTPUTS:     none
;  174          0301  1 !
;  175          0302  1 ! ROUTINE VALUE and
;  176          0303  1 ! COMPLETION CODES:     none
;  177          0304  1 !
;  178          0305  1 ! SIDE EFFECTS:         none discovered yet
;  179          0306  1 !
;  180          0307  1 !--
;  181          0308  1
;  182          0309  2     BEGIN
;  183          0310  2     OWN
;  184          0311  2         x,                                          !Just a copy of the SAVSTACK depth
;  185          0312  2         savptr;
;  186          0313  2
;  187          0314  2     SELECT .handler_code OF
;  188          0315  2         SET
;  189          0316  2
;  190          0317  2         [h_save] :
;  191          0318  3            BEGIN
;  192          0319  3            !a .SAVE command has been hit.
;  193          0320  3            !This section will save all the necessary values of items the utility
;  194          0321  3            !programs may alter.  Tab stops have also been saved since it was thought
;  195          0322  3            !they may be saved often without the need for .SAVE ALL.
;  196          0323  3
;  197          0324  3            !Let's check to see if there is any more room for one more save.
;  198          0325  3             x = .savstk [0, savstk_depth] + 1;
;  199          0326  3
;  200          0327  3             IF  .x  GTR  savstk_size                                                    ; R
;  201          0328  3             THEN
;  202          0329  3                erma (rnftms, false);!Already at maximum save depth, ignore and issue error.   ;
```

```
;  203          0330  3
;  204          0331  3              !Remember how the .SAVE was started.(even if we're 'ignoring' it)
;  205          0332  3              !Only if the number of SAVEs is fewer than 30(STKSAV_SIZE).
;  206          0333  3              IF .x GTR stksav_size
;  207          0334  3              THEN
;  208          0335  3                  RETURN;                                     !Forget this command, user has really extend
;  209          0336  3              !
;  210          0337  3              ! Show whether this is a save or save_all and output the nesting
;  211          0338  3              ! level if user asked for /debug=save.
;  212          0339  3              !
;  213          0340  3              IF .gca_debug_save and NOT .gca_skip_out
;  214          0341  3              THEN
;  215          0342  4                  BEGIN
;  216        P 0343  4                  $XPO_PUT ( IOB    = .rnoiob
;  217        P 0344  4                        ,STRING = $STR_CONCAT( '.SAVE '
;  218        P 0345  4                                              ,'('
;  219        P 0346  4                                              ,$STR_ASCII(.x)
;  220        P 0347  4                                              ,')'
;  221          0348  4                                             ));
;  222          0349  3                  END;
;  223          0350  3
;  224          0351  3              savstk [0, savstk_depth] = .x;                  !Save new stack depth.
;  225          0352  3              savstk [.x, savstk_source] = .handler_code;     !Remember command.
;  226          0353  3              savstk [.x, savstk_req_d] = .gca_req_depth;     !Remember .REQUIRE depth.
;  227          0354  3              savstk [.x, savstk_ipagen] = .irac_ipagen;      !Remember input page number.
;  228          0355  3              savstk [.x, savstk_iseqn] = .irac_iseqn;        !Remember input line number.
;  229          0356  3              savstk [.x, savstk_fspecp] = .irac_fspecp;      !Remember file spec.
;  230          0357  3              savstk [.x, savstk_fspecc] = .irac_fspecc;      !File spec length
;  231          0358  3
;  232          0359  3               IF  .x GTR  savstk_size
;  233          0360  3               THEN
;  234          0361  3                  RETURN; ! We've make a record of the save but don't do anything.
;  235          0362  3
;  236          0363  3              !Now, get some memory to store the save values.
;  237          0364  3              $XPO_GET_MEM( FULLWORDS = save_length, RESULT = savptr);
;  238          0365  3               savstk [.x, savstk_address] = .savptr;         !Save pointer to starting address of save bl
;  239          0366  3
;  240          0367  3              !Finally, we perform the save by copying the SAVEALL storage area to our dynamic stack.
;  241          0368  3              block_copy (flgt, save_length, .savptr);
;  242          0369  3
;  243          0370  2              END;
;  244          0371  2
;  245          0372  2          [h_save_all] :
;  246          0373  3              BEGIN
;  247          0374  3              !a Save All command has been hit.
;  248          0375  3              !A .SAVE ALL saves all that a .SAVE command saves in addition to several
;  249          0376  3              !other flags and action bits.
;  250          0377  3
;  251          0378  3              !Let's check to see if there is any more room for one more save all.
;  252          0379  3               x = .savstk [0, savstk_depth] + 1;
;  253          0380  3
;  254          0381  3               IF  .x GTR  savstk_size  THEN
;  255          0382  3                  erma (rnftms, false);!Already at maximum save depth, ignore and issue error.
;  256          0383  3
;  257          0384  3               !Remember how the .SAVE ALL was started.
;  258          0385  3               !Only if the number of SAVE ALLs is fewer than 30(STKSAV_SIZE).
;  259          0386  3               !
```

```
;  260     0387  3            IF .x GTR stksav_size  THEN
;  261     0388  3                RETURN;                                  !Forget this command, user has really extend
;  262     0389  3                !
;  263     0390  3                ! Show whether this is a save or save_all and output the nesting
;  264     0391  3                ! level if user asked for /debug=save.
;  265     0392  3                !
;  266     0393  3                IF .gca_debug_save and NOT .gca_skip_out
;  267     0394  3                THEN
;  268     0395  4                    BEGIN
;  269   P 0396  4                    $XPO_PUT ( IOB    = .rnoiob
;  270   P 0397  4                          ,STRING = $STR_CONCAT( '.SAVE ALL '
;  271   P 0398  4                                                ,'('
;  272   P 0399  4                                                ,$STR_ASCII(.x)
;  273   P 0400  4                                                ,')'
;  274     0401  4                                               ));
;  275     0402  4                    END;
;  276     0403  3
;  277     0404  3                savstk [0, savstk_depth] = .x;                !Save new stack depth.
;  278     0405  3                savstk [.x, savstk_source] = .handler_code;   !Remember command.
;  279     0406  3                savstk [.x, savstk_req_d] = .gca_req_depth;   !Remember .REQUIRE depth.
;  280     0407  3                savstk [.x, savstk_ipagen] = .irac_ipagen;    !Remember input page number.
;  281     0408  3                savstk [.x, savstk_iseqn] = .irac_iseqn;      !Remember input line number.
;  282     0409  3                savstk [.x, savstk_fspecp] = .irac_fspecp;    !Remember file spec.
;  283     0410  3                savstk [.x, savstk_fspecc] = .irac_fspecc;    !File spec length
;  284     0411  3
;  285     0412  3                IF  .x GTR  savstk_size  THEN
;  286     0413  3                   RETURN; ! We've make a record of the save but don't do anything.
;  287     0414  3
;  288     0415  3                !Now, get some memory to store the save values.
;  289     0416  3                $XPO_GET_MEM( FULLWORDS = saveall_length, RESULT = savptr);
;  290     0417  3                 savstk [.x, savstk_address] = .savptr;       !Save pointer to starting address of save bl
;  291     0418  3
;  292     0419  3                !Finally, we perform the save by copying the SAVEALL storage area to our dynamic stack.
;  293     0420  3                block_copy (flgt, saveall_length, .savptr);
;  294     0421  3
;  295     0422  2                END;
;  296     0423  2
;  297     0424  2            [h_restore] :
;  298     0425  3               BEGIN
;  299     0426  3               !a Restore command has been hit. There may have been some save commands
;  300     0427  3               !that were ignored because of too many saves.  Checking for that..
;  301     0428  3
;  302     0429  3               x = .savstk [0, savstk_depth];       !What's the current stack depth?
;  303     0430  3
;  304     0431  3               IF .x GTR savstk_size
;  305     0432  3               THEN !User has asked for more saves than possible.
;  306     0433  4                  BEGIN
;  307     0434  4                  savstk [0, savstk_depth] = .savstk [0, savstk_depth] - 1;!Ignore this save.
;  308     0435  4                  IF .internal_code EQL 1             !User requested this restore, issue warning.
;  309     0436  4                  THEN                                !Otherwise do it silently.
;  310     0437  4                    erma (rnftms, false);
;  311     0438  4                  RETURN;
;  312     0439  3                  END;
;  313     0440  3               !
;  314     0441  3               ! Show that a restore is being done and output the nesting
;  315     0442  3               ! level if user asked for /debug=save.
;  316     0443  3               !
```

```
;  317          0444  3            IF .gca_debug_save and NOT .gca_skip_out
;  318          0445  3            THEN
;  319          0446  4                BEGIN
;  320        P 0447  4                $XPO_PUT ( IOB    = .rnoiob
;  321        P 0448  4                   ,STRING = $STR_CONCAT( '.RESTORE  '
;  322        P 0449  4                                         ,'('
;  323        P 0450  4                                         ,$STR_ASCII(.x)
;  324        P 0451  4                                         ,')'
;  325          0452  4                                         ));
;  326          0453  4                IF .savstk [.x, savstk_source] EQL h_save
;  327          0454  4                THEN
;  328        P 0455  4                    $XPO_PUT ( IOB    = .rnoiob
;  329        P 0456  4                       ,STRING = $STR_CONCAT( 'Corresponds to SAVE on line '
;  330        P 0457  4                                         ,$STR_ASCII(.savstk [.x, savstk_iseqn])
;  331        P 0458  4                                         ,' page '
;  332        P 0459  4                                         ,$STR_ASCII(.savstk [.x, savstk_ipagen])
;  333        P 0460  4                                         ,' of input file '
;  334        P 0461  4                                         ,(.savstk [.x, savstk_fspecc]
;  335        P 0462  4                                         ,.savstk [.x, savstk_fspecp])
;  336          0463  5                                          ))
;  337          0464  4                 ELSE
;  338        P 0465  4                    $XPO_PUT ( IOB    = .rnoiob
;  339        P 0466  4                       ,STRING = $STR_CONCAT( 'Corresponds to SAVE ALL on line '
;  340        P 0467  4                                         ,$STR_ASCII(.savstk [.x, savstk_iseqn])
;  341        P 0468  4                                         ,' page '
;  342        P 0469  4                                         ,$STR_ASCII(.savstk [.x, savstk_ipagen])
;  343        P 0470  4                                         ,' of input file '
;  344        P 0471  4                                         ,(.savstk [.x, savstk_fspecc]
;  345        P 0472  4                                         ,.savstk [.x, savstk_fspecp])
;  346          0473  4                                          ));
;  347          0474  4
;  348          0475  3                END;
;  349          0476  3
;  350          0477  3            !Now stored in the SAVSTKis the list of all the saves.  Each
;  351          0478  3            !item is the type of save (SAVE or SAVE ALL). So below we..
;  352          0479  3            SELECT .savstk [.x, savstk_source] OF
;  353          0480  3              SET                   !Determine what to restore. (if anything)
;  354          0481  3
;  355          0482  3              [h_save] :
;  356          0483  4                 BEGIN
;  357          0484  4                 !The last SAVE was just the small save.
;  358          0485  4
;  359          0486  4                 !Is this restore in same file as corresponding SAVE/SAVEALL?
;  360          0487  4                 IF  .SAVSTK [.X, SAVSTK_REQ_D] NEQ .GCA_REQ_DEPTH
;  361          0488  4                 THEN
;  362          0489  5                    BEGIN
;  363          0490  5                    ERMA (RNFWFR, FALSE);              !Wrong File for Restore
;  364          0491  5                    RETURN;
;  365          0492  4                    END;
;  366          0493  4
;  367          0494  4                 !Update the address counted-list with a starting address of zero for debugging.
;  368          0495  4                 SAVPTR = .SAVSTK [.X, SAVSTK_ADDRESS];
;  369          0496  4                 SAVSTK [.X, SAVSTK_ADDRESS] = 0;
;  370          0497  4
;  371          0498  4                 !We may now copy the SAVEd block over to their 'active' area.
;  372          0499  4                 block_copy (.SAVPTR, SAVE_LENGTH, FLGT);
;  373          0500  4
```

```
;  374     0501  4                !Return the memory since we don't need it any more
;  375     0502  4                $XPO_FREE_MEM( BINARY_DATA = (SAVE_LENGTH, .SAVPTR ));
;  376     0503  4
;  377     0504  4                !And finally decrement the number of SAVEs by one.
;  378     0505  4                SAVSTK [0, SAVSTK_DEPTH] = .SAVSTK [0, SAVSTK_DEPTH] - 1;
;  379     0506  3                END;
;  380     0507  3
;  381     0508  3             [h_save_all] :
;  382     0509  4                BEGIN
;  383     0510  4                !This restore is for a Save All.
;  384     0511  4                !The code in this section is just a reverse of the .SAVE ALL area.
;  385     0512  4
;  386     0513  4                !Is this restore in same file as corresponding SAVE/SAVEALL?
;  387     0514  4                IF  .SAVSTK [.X, SAVSTK_REQ_D] NEQ .GCA_REQ_DEPTH
;  388     0515  4                THEN
;  389     0516  5                   BEGIN
;  390     0517  5                   ERMA (RNFWFR, FALSE);              !Wrong File for Restore
;  391     0518  5                   RETURN;
;  392     0519  4                   END;
;  393     0520  4
;  394     0521  4                !Update the address counted-list with a starting address of zero for debugging.
;  395     0522  4                SAVPTR = .SAVSTK [.X, SAVSTK_ADDRESS];
;  396     0523  4                SAVSTK [.X, SAVSTK_ADDRESS] = 0;
;  397     0524  4
;  398     0525  4                !We may now copy the SAVEd block over to their 'active' area.
;  399     0526  4                block_copy (.SAVPTR, SAVEALL_LENGTH, FLGT);
;  400     0527  4
;  401     0528  4                !Return the memory since we don't need it any more
;  402     0529  4                $XPO_FREE_MEM( BINARY_DATA = (SAVEALL_LENGTH, .SAVPTR ));
;  403     0530  4
;  404     0531  4                !And finally decrement the number of SAVEs by one.
;  405     0532  4                SAVSTK [0, SAVSTK_DEPTH] = .SAVSTK [0, SAVSTK_DEPTH] - 1;
;  406     0533  4
;  407     0534  3                END;
;  408     0535  3
;  409     0536  3             [OTHERWISE] :
;  410     0537  4                BEGIN
;  411     0538  4                !Nothing more has been 'SAVEd' in this file; issue error and return.
;  412     0539  4                ERMA (RNFNCS, FALSE);               !No corresponding saves.
;  413     0540  4                RETURN;
;  414     0541  3                END;
;  415     0542  3
;  416     0543  3             TES;                                   !End of RESTORE set.
;  417     0544  2          END;                           !End of RESTORE
;  418     0545  2
;  419     0546  2          TES;                           !End of SAVRES set.
;  420     0547  1      END;                               ! end of routine SAVRES


                                                        .TITLE  SAVRES SAVRES -- Save & Restore
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                20  45  56  41  53  2E  00000 P.AAD:  .ASCII  \.SAVE \                    ;
                                                    28  00006 P.AAE:  .ASCII  \(\                       ;
                                                    29  00007 P.AAF:  .ASCII  \)\                       ;
```

```
                                                20  4C  4C  41  20  45  56  41  53  2E  00008 P.AAM:   .ASCII  \.SAVE ALL \                ;
                                                                                    28  00012 P.AAN:   .ASCII  \(\                          ;
                                                                                    29  00013 P.AAO:   .ASCII  \)\                          ;
                                                20  20  45  52  4F  54  53  45  52  2E  00014 P.AAV:   .ASCII  \.RESTORE  \               ;
                                                                                    28  0001E P.AAW:   .ASCII  \(\                          ;
                                                                                    29  0001F P.AAX:   .ASCII  \)\                          ;
20  6F  74  20  73  64  6E  6F  70  73  65  72  72  6F  43  00020 P.ABE:   .ASCII  \Corresponds to SAVE on line \                ;
        20  65  6E  69  6C  20  6E  6F  20  45  56  41  53  0002F                                                                             ;
                                            20  65  67  61  70  20  0003C P.ABF:   .ASCII  \ page \                                        ;
20  65  6C  69  66  20  74  75  70  6E  69  20  66  6F  20  00042 P.ABG:   .ASCII  \ of input file \                                    ;
20  6F  74  20  73  64  6E  6F  70  73  65  72  72  6F  43  00051 P.ABN:   .ASCII  \Corresponds to SAVE ALL on line \                 ;
6E  69  6C  20  6E  6F  20  4C  4C  41  20  45  56  41  53  00060                                                                             ;
                                                                    20  65  0006F                                                           ;
                                            20  65  67  61  70  20  00071 P.ABO:   .ASCII  \ page \                                        ;
20  65  6C  69  66  20  74  75  70  6E  69  20  66  6F  20  00077 P.ABP:   .ASCII  \ of input file \                                    ;

                                                                                            .PSECT  $OWN$,NOEXE,2

                                                                                    00000 X:        .BLKB   4
                                                                                    00004 SAVPTR:   .BLKB   4
                                                                              0006  00008 $STR$STRING0:
                                                                                            .WORD   6                                       ;
                                                                           01  0E   0000A           .BYTE   14, 1                           ;
                                                                      00000000'     0000C           .ADDRESS P.AAD                          ;
                                                                              0001  00010 $STR$STRING1:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   00012           .BYTE   14, 1                           ;
                                                                      00000000'     00014           .ADDRESS P.AAE                          ;
                                                                              0001  00018 $STR$STRING3:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   0001A           .BYTE   14, 1                           ;
                                                                      00000000'     0001C           .ADDRESS P.AAF                          ;
                                                                              000A  00020 $STR$STRING0:
                                                                                            .WORD   10                                      ;
                                                                           01  0E   00022           .BYTE   14, 1                           ;
                                                                      00000000'     00024           .ADDRESS P.AAM                          ;
                                                                              0001  00028 $STR$STRING1:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   0002A           .BYTE   14, 1                           ;
                                                                      00000000'     0002C           .ADDRESS P.AAN                          ;
                                                                              0001  00030 $STR$STRING3:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   00032           .BYTE   14, 1                           ;
                                                                      00000000'     00034           .ADDRESS P.AAO                          ;
                                                                              000A  00038 $STR$STRING0:
                                                                                            .WORD   10                                      ;
                                                                           01  0E   0003A           .BYTE   14, 1                           ;
                                                                      00000000'     0003C           .ADDRESS P.AAV                          ;
                                                                              0001  00040 $STR$STRING1:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   00042           .BYTE   14, 1                           ;
                                                                      00000000'     00044           .ADDRESS P.AAW                          ;
                                                                              0001  00048 $STR$STRING3:
                                                                                            .WORD   1                                       ;
                                                                           01  0E   0004A           .BYTE   14, 1                           ;
                                                                      00000000'     0004C           .ADDRESS P.AAX                          ;
                                                                              001C  00050 $STR$STRING0:
```

```
                                                        .WORD   28                                       ;
                      01  0E  00052                     .BYTE   14, 1                                    ;
                00000000'     00054                     .ADDRESS P.ABE                                   ;
                    0006      00058 $STR$STRING2:
                                                        .WORD   6                                        ;
                      01  0E  0005A                     .BYTE   14, 1                                    ;
                00000000'     0005C                     .ADDRESS P.ABF                                   ;
                    000F      00060 $STR$STRING4:
                                                        .WORD   15                                       ;
                      01  0E  00062                     .BYTE   14, 1                                    ;
                00000000'     00064                     .ADDRESS P.ABG                                   ;
                    0020      00068 $STR$STRING0:
                                                        .WORD   32                                       ;
                      01  0E  0006A                     .BYTE   14, 1                                    ;
                00000000'     0006C                     .ADDRESS P.ABN                                   ;
                    0006      00070 $STR$STRING2:
                                                        .WORD   6                                        ;
                      01  0E  00072                     .BYTE   14, 1                                    ;
                00000000'     00074                     .ADDRESS P.ABO                                   ;
                    000F      00078 $STR$STRING4:
                                                        .WORD   15                                       ;
                      01  0E  0007A                     .BYTE   14, 1                                    ;
                00000000'     0007C                     .ADDRESS P.ABP                                   ;

                                                        .EXTRN  RNFLOC, RNFTMS, RNFWFR
                                                        .EXTRN  RNFNCS, RNOIOB, IRAC
                                                        .EXTRN  SAVSTK, FLGT, SCA
                                                        .EXTRN  HCT, GCA, PHAN, SAVE
                                                        .EXTRN  SAVEALL, ERM, ERMA
                                                        .EXTRN  XST$JOIN, XST$ASCII
                                                        .EXTRN  XPO$PUT, XPO$FAILURE
                                                        .EXTRN  XPO$ALLOC_MEM, XPO$FREE_MEM

                                                        .PSECT  $CODE$,NOWRT,2

                          0FFC 00000                    .ENTRY  SAVRES, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-  ; 0274
                                                                R11                                      ;
               5B 00000000G  EF  9E 00002               MOVAB   XST$JOIN, R11                            ;
               5A 00000000G  EF  9E 00009               MOVAB   RNOIOB, R10                              ;
               59 00000000G  EF  9E 00010               MOVAB   XST$ASCII, R9                            ;
               58 00000000G  EF  9E 00017               MOVAB   IRAC+12, R8                              ;
               57 00000000G  EF  9E 0001E               MOVAB   XPO$FAILURE, R7                          ;
               56 00000000G  EF  9E 00025               MOVAB   GCA+188, R6                              ;
               55 00000000G  EF  9E 0002C               MOVAB   SAVSTK, R5                               ;
               54 00000000'  EF  9E 00033               MOVAB   X, R4                                    ;
               5E            08  C2 0003A               SUBL2   #8, SP                                   ;
               53        04  AC  D0 0003D               MOVL    HANDLER_CODE, R3                         ; 0314
         000000B2  8F        53  D1 00041               CMPL    R3, #178                                 ; 0317
                             03  13 00048               BEQL    1$                                       ;
                           00DC  31 0004A               BRW     7$                                       ;
      64           65        01  C1 0004D 1$:           ADDL3   #1, SAVSTK, X                            ; 0325
                   14        64  D1 00051               CMPL    X, #20                                   ; 0327
                             0F  15 00054               BLEQ    2$                                       ;
                             7E  D4 00056               CLRL    -(SP)                                    ; 0329
                   00000000G 8F  DD 00058               PUSHL   #RNFTMS                                  ;
         00000000G EF        02  FB 0005E               CALLS   #2, ERMA                                 ;
                   1E        64  D1 00065 2$:           CMPL    X, #30                                   ; 0333
```

```
                                        75 14 00068      BGTR    4$                              :
36       B8   A6                        05 E1 0006A      BBC     #5, GCA+116, 3$                 : 0340
              32            B4          A6 E8 0006F      BLBS    GCA+112, 3$                     :
              52                        6A D0 00073      MOVL    RNGIOB, R2                      : 0348
                                        7E D4 00076      CLRL    -(SP)                           :
                                        64 DD 00078      PUSHL   X                               :
              7E          0903          8F 3C 0007A      MOVZWL  #2307, -(SP)                    :
              69                        03 FB 0007F      CALLS   #3, XST$ASCII                   :
                            18          A4 9F 00082      PUSHAB  $STR$STRING3                    :
                                        50 DD 00085      PUSHL   R0                              :
                            10          A4 9F 00087      PUSHAB  $STR$STRING1                    :
                            08          A4 9F 0008A      PUSHAB  $STR$STRING0                    :
              6B                        04 FB 0008D      CALLS   #4, XST$JOIN                    :
         44   A2                        50 D0 00090      MOVL    R0, 68(R2)                      :
         2C   A2                        07 90 00094      MOVB    #7, 44(R2)                      :
                                        57 DD 00098      PUSHL   R7                              :
                                        7E D4 0009A      CLRL    -(SP)                           :
                                        52 DD 0009C      PUSHL   R2                              :
   00000000G  EF                        03 FB 0009E      CALLS   #3, XPO$PUT                     :
              51                        64 D0 000A5 3$:  MOVL    X, R1                           : 0351
              65                        51 D0 000A8      MOVL    R1, SAVSTK                      :
50            51                        1C C5 000AB      MULL3   #28, R1, R0                     : 0352
                            08 A540        9F 000AF      PUSHAB  SAVSTK+8[R0]                    :
              9E                        53 D0 000B3      MOVL    R3, @(SP)+                      :
                            0C A540        9F 000B6      PUSHAB  SAVSTK+12[R0]                   : 0353
              9E                        66 D0 000BA      MOVL    GCA+188, @(SP)+                 :
                            10 A540        9F 000BD      PUSHAB  SAVSTK+16[R0]                   : 0354
              9E                        68 D0 000C1      MOVL    IRAC+12, @(SP)+                 :
                            14 A540        9F 000C4      PUSHAB  SAVSTK+20[R0]                   : 0355
              9E            FC          A8 D0 000C8      MOVL    IRAC+8, @(SP)+                  :
                            18 A540        9F 000CC      PUSHAB  SAVSTK+24[R0]                   : 0356
              9E            04          A8 D0 000D0      MOVL    IRAC+16, @(SP)+                 :
                            1C A540        9F 000D4      PUSHAB  SAVSTK+28[R0]                   : 0357
              9E            08          A8 D0 000D8      MOVL    IRAC+20, @(SP)+                 :
              14                        51 D1 000DC      CMPL    R1, #20                         : 0359
                                        6F 14 000DF 4$:  BGTR    10$                             :
              6E 02020000               8F D0 000E1      MOVL    #33685504, $XPO$DESC            : 0364
                            04          AE D4 000E8      CLRL    $XPO$DESC+4                     :
                                        57 DD 000EB      PUSHL   R7                              :
                                        7E 7C 000ED      CLRQ    -(SP)                           :
              7E                        01 CE 000EF      MNEGL   #1, -(SP)                       :
                            10          AE 9F 000F2      PUSHAB  $XPO$DESC                       :
              7E          0294          8F 3C 000F5      MOVZWL  #660, -(SP)                     :
   00000000G  EF                        06 FB 000FA      CALLS   #6, XPO$ALLOC_MEM               :
              05                        50 E9 00101      BLBC    $XPO$STATUS, 5$                 :
         04   A4            04          AE D0 00104      MOVL    $XPO$DESC+4, SAVPTR             :
50            64                        1C C5 00109 5$:  MULL3   #28, X, R0                      : 0365
                            04 A540        9F 0010D      PUSHAB  SAVSTK+4[R0]                    :
              9E            04          A4 D0 00111      MOVL    SAVPTR, @(SP)+                  :
                                        50 D4 00115      CLRL    I                               : 0368
         04 B440 00000000GEF            40 D0 00117 6$:  MOVL    FOO[I], @SAVPTR[I]              :
EE            50 000000A4               8F F3 00121      AOBLEQ  #164, I, 6$                     :
   000000B3   8F                        53 D1 00129 7$:  CMPL    R3, #179                        : 0372
                                        03 13 00130      BEQL    8$                              :
                                      00DC 31 00132      BRW     15$                             :
64            65                        01 C1 00135 8$:  ADDL3   #1, SAVSTK, X                   : 0379
              14                        64 D1 00139      CMPL    X, #20                          : 0381
```

```
                                                0F  15 0013C          BLEQ    9$                                          ;
                                                7E  D4 0013E          CLRL    -(SP)                                       ; 0382
                                  00000000G     8F  DD 00140          PUSHL   #RNFTMS
                        00000000G EF            02  FB 00146          CALLS   #2, ERMA
                                  1E            64  D1 00148 9$:      CMPL    X, #30                                      ; 0387
                                                75  14 00150 10$:     BGTR    12$                                         ;
                     36       B8  A6            05  E1 00152          BBC     #5, GCA+116, 11$                            ; 0393
                                  32       B4   A6  E8 00157          BLBS    GCA+112, 11$                                ;
                                  52            6A  D0 0015B          MOVL    RNOIOB, R2                                  ; 0401
                                                7E  D4 0015E          CLRL    -(SP)                                       ;
                                                64  DD 00160          PUSHL   X                                           ;
                                  7E      0903  8F  3C 00162          MOVZWL  #2307, -(SP)                                ;
                                  69            03  FB 00167          CALLS   #3, XST$ASCII                               ;
                                           30   A4  9F 0016A          PUSHAB  $STR$STRING3                                ;
                                                50  DD 0016D          PUSHL   R0                                          ;
                                           28   A4  9F 0016F          PUSHAB  $STR$STRING1                                ;
                                           20   A4  9F 00172          PUSHAB  $STR$STRING0                                ;
                                  6B            04  FB 00175          CALLS   #4, XST$JOIN                                ;
                              44  A2            50  D0 00178          MOVL    R0, 68(R2)                                  ;
                              2C  A2            07  90 0017C          MOVB    #7, 44(R2)                                  ;
                                                57  DD 00180          PUSHL   R7                                          ;
                                                7E  D4 00182          CLRL    -(SP)                                       ;
                                                52  DD 00184          PUSHL   R2                                          ;
                        00000000G EF            03  FB 00186          CALLS   #3, XPO$PUT                                 ;
                                  51            64  D0 0018D 11$:     MOVL    X, R1                                       ; 0404
                                  65            51  D0 00190          MOVL    R1, SAVSTK                                  ;
                     50           51            1C  C5 00193          MULL3   #28, R1, R0                                 ; 0405
                                           08 A540  9F 00197          PUSHAB  SAVSTK+8[R0]                                ;
                                  9E            53  D0 0019B          MOVL    R3, @(SP)+                                  ;
                                           0C A540  9F 0019E          PUSHAB  SAVSTK+12[R0]                               ; 0406
                                  9E            66  D0 001A2          MOVL    GCA+188, @(SP)+                             ;
                                           10 A540  9F 001A5          PUSHAB  SAVSTK+16[R0]                               ; 0407
                                  9E            68  D0 001A9          MOVL    IRAC+12, @(SP)+                             ;
                                           14 A540  9F 001AC          PUSHAB  SAVSTK+20[R0]                               ; 0408
                                  9E        FC  A8  D0 001B0          MOVL    IRAC+8, @(SP)+                              ;
                                           18 A540  9F 001B4          PUSHAB  SAVSTK+24[R0]                               ; 0409
                                  9E        04  A8  D0 001B8          MOVL    IRAC+16, @(SP)+                             ;
                                           1C A540  9F 001BC          PUSHAB  SAVSTK+28[R0]                               ; 0410
                                  9E        08  A8  D0 001C0          MOVL    IRAC+20, @(SP)+                             ;
                                  14            51  D1 001C4          CMPL    R1, #20                                     ; 0412
                                                5F  14 001C7 12$:     BGTR    16$                                         ;
                                  6E 02020000   8F  D0 001C9          MOVL    #33685504, $XPO$DESC                        ; 0416
                                           04   AE  D4 001D0          CLRL    $XPO$DESC+4                                 ;
                                                57  DD 001D3          PUSHL   R7                                          ;
                                                7E  7C 001D5          CLRQ    -(SP)                                       ;
                                  7E            01  CE 001D7          MNEGL   #1, -(SP)                                   ;
                                           10   AE  9F 001DA          PUSHAB  $XPO$DESC                                   ;
                                  7E      04A8  8F  3C 001DD          MOVZWL  #1192, -(SP)                                ;
                        00000000G EF            06  FB 001E2          CALLS   #6, XPO$ALLOC_MEM                           ;
                                  05            50  E9 001E9          BLBC    $XPO$STATUS, T3$                            ;
                           04   A4       04   AE  D0 001EC          MOVL    $XPO$DESC+4, SAVPTR                         ;
                     50           64            1C  C5 001F1 13$:     MULL3   #28, X, R0                                  ; 0417
                                           04 A540  9F 001F5          PUSHAB  SAVSTK+4[R0]                                ;
                                  9E        04   A4  D0 001F9          MOVL    SAVPTR, @(SP)+                              ;
                                                50  D4 001FD          CLRL    I                                           ; 0420
                           04 B440 00000000GEF40  D0 001FF 14$:     MOVL    FOO[I], @SAVPTR[I]                          ;
                     EE           50 00000129   8F  F3 00209          AOBLEQ  #297, I, 14$                                ;
```

```
                 000000AF   8F              53   D1 00211 15$:    CMPL    R3, #175                        ; 0424
                                            0E   12 00218         BNEQ    16$
                            64              65   D0 0021A         MOVL    SAVSTK, X                       ; 0429
                            14              64   D1 0021D         CMPL    X, #20                          ; 0431
                                            14   15 00220         BLEQ    18$
                                            65   D7 00222         DECL    SAVSTK                          ; 0434
                            01        08    AC   D1 00224         CMPL    INTERNAL_CODE, #1               ; 0435
                                            01   13 00228 16$:    BEQL    17$
                                            04   0022A           RET
                                            7E   D4 0022B 17$:    CLRL    -(SP)                           ; 0437
                       00000000G        8F   DD 0022D           PUSHL   #RNFTMS
                                      01F3   31 00233           BRW     30$
                 03         B8        A6        05   E0 00236 18$:    BBS     #5, GCA+116, 20$                ; 0444
                                      0106   31 0023B 19$:    BRW     23$
                            F9        B4    A6   E8 0023E 20$:    BLBS    GCA+112, 19$
                            52              6A   D0 00242         MOVL    RNOIOB, R2                      ; 0452
                                            7E   D4 00245         CLRL    -(SP)
                                            64   DD 00247         PUSHL   X
                            7E      0903    8F   3C 00249         MOVZWL  #2307, -(SP)
                            69              03   FB 0024E         CALLS   #3, XST$ASCII
                                    48      A4   9F 00251         PUSHAB  $STR$STRING3
                                            50   DD 00254         PUSHL   R0
                                    40      A4   9F 00256         PUSHAB  $STR$STRING1
                                    38      A4   9F 00259         PUSHAB  $STR$STRING0
                            6B              04   FB 0025C         CALLS   #4, XST$JOIN
                         44 A2              50   D0 0025F         MOVL    R0, 68(R2)
                         2C A2              07   90 00263         MOVB    #7, 44(R2)
                                            57   DD 00267         PUSHL   R7
                                            7E   D4 00269         CLRL    -(SP)
                                            52   DD 0026B         PUSHL   R2
                    00000000G   EF          03   FB 0026D         CALLS   #3, XPO$PUT
                                52          6A   D0 00274         MOVL    RNOIOB, R2                      ; 0463
                 50             64          1C   C5 00277         MULL3   #28, X, R0                      ; 0433
                                      08 A540   9F 0027B         PUSHAB  SAVSTK+8[R0]
                    000000B2    8F          9E   D1 0027F         CMPL    @(SP)+, #178
                                            53   12 00286         BNEQ    21$
                                            7E   D4 00288         CLRL    -(SP)                           ; 0463
                                      14 A540   9F 0028A         PUSHAB  SAVSTK+20[R0]
                                            9E   DD 0028E         PUSHL   @(SP)+
                            7E      0903    8F   3C 00290         MOVZWL  #2307, -(SP)
                            69              03   FB 00295         CALLS   #3, XST$ASCII
                            53              50   D0 00298         MOVL    R0, R3
                                            7E   D4 0029B         CLRL    -(SP)
                 50             64          1C   C5 0029D         MULL3   #28, X, R0
                                      10 A540   9F 002A1         PUSHAB  SAVSTK+16[R0]
                                            9E   DD 002A5         PUSHL   @(SP)+
                            7E      0903    8F   3C 002A7         MOVZWL  #2307, -(SP)
                            69              03   FB 002AC         CALLS   #3, XST$ASCII
                 51             64          1C   C5 002AF         MULL3   #28, X, R1
                                      1C A541   9F 002B3         PUSHAB  SAVSTK+28[R1]
                            6E              9E   B0 002B7         MOVW    @(SP)+, $STR$STRING5
                         02 AE              0E   90 002BA         MOVB    #14, $STR$STRING5+2
                         03 AE              01   90 002BE         MOVB    #1, $STR$STRING5+3
                                      18 A541   9F 002C2         PUSHAB  SAVSTK+24[R1]
                         04 AE              9E   D0 002C6         MOVL    @(SP)+, $STR$STRING5+4
                                            5E   DD 002CA         PUSHL   SP
                                    60      A4   9F 002CC         PUSHAB  $STR$STRING4
```

```
                                                50   DD 002CF          PUSHL   R0
                                             58 A4   9F 002D1          PUSHAB  $STR$STRING2
                                                53   DD 002D4          PUSHL   R3
                                             50 A4   9F 002D6          PUSHAB  $STR$STRING0
                                                51   11 002D9          BRB     22$
                                                7E   D4 002DB 21$:     CLRL    -(SP)                                   0473
                                             14 A540 9F 002DD          PUSHAB  SAVSTK+20[R0]
                                                9E   DD 002E1          PUSHL   @(SP)+
                                          7E    0903 8F 3C 002E3       MOVZWL  #2307, -(SP)
                                          69       03 FB 002E8       CALLS   #3, XST$ASCII
                                          53       50 D0 002EB       MOVL    R0, R3
                                                7E   D4 002EE          CLRL    -(SP)
                                    50    64       1C C5 002F0       MULL3   #28, X, R0
                                             10 A540 9F 002F4          PUSHAB  SAVSTK+16[R0]
                                                9E   DD 002F8          PUSHL   @(SP)+
                                          7E    0903 8F 3C 002FA       MOVZWL  #2307, -(SP)
                                          69       03 FB 002FF       CALLS   #3, XST$ASCII
                                    51    64       1C C5 00302       MULL3   #28, X, R1
                                             1C A541 9F 00306          PUSHAB  SAVSTK+28[R1]
                                          6E       9E B0 0030A       MOVW    @(SP)+, $STR$STRING5
                                    02    AE       0E 90 0030D       MOVB    #14, $STR$STRING5+2
                                    03    AE       01 90 00311       MOVB    #1, $STR$STRING5+3
                                             18 A541 9F 00315          PUSHAB  SAVSTK+24[R1]
                                    04    AE       9E D0 00319       MOVL    @(SP)+, $STR$STRING5+4
                                                5E   DD 0031D          PUSHL   SP
                                             78 A4   9F 0031F          PUSHAB  $STR$STRING4
                                                50   DD 00322          PUSHL   R0
                                             70 A4   9F 00324          PUSHAB  $STR$STRING2
                                                53   DD 00327          PUSHL   R3
                                             68 A4   9F 00329          PUSHAB  $STR$STRING0
                                          6B       06 FB 0032C 22$:     CALLS   #6, XST$JOIN
                                    44    A2       50 D0 0032F       MOVL    R0, 68(R2)
                                    2C    A2       07 90 00333       MOVB    #7, 44(R2)
                                                57   DD 00337          PUSHL   R7
                                                7E   D4 00339          CLRL    -(SP)
                                                52   DD 0033B          PUSHL   R2
                              00000000G         EF   03 FB 0033D       CALLS   #3, XPO$PUT
                          50                    64   1C C5 00344 23$:     MULL3   #28, X, R0                          0479
                                             08 A540 9F 00348          PUSHAB  SAVSTK+8[R0]
                                          53       9E D0 0034C       MOVL    @(SP)+, R3
                                          52       01 D0 0034F       MOVL    #1, R2
                              000000B2    8F       53 D1 00352       CMPL    R3, #178                                0482
                                                56   12 00359          BNEQ    25$
                                                52   D4 0035B          CLRL    R2
                                             0C A540 9F 0035D          PUSHAB  SAVSTK+12[R0]                           0487
                                          66       9E D1 00361       CMPL    @(SP)+, GCA+188
                                                63   12 00364          BNEQ    26$
                          50                    64   1C C5 00366       MULL3   #28, X, R0                          0495
                                             04 A540 9F 0036A          PUSHAB  SAVSTK+4[R0]
                                    04    A4       9E D0 0036E       MOVL    @(SP)+, SAVPTR
                                             04 A540 9F 00372          PUSHAB  SAVSTK+4[R0]                            0496
                                                9E   D4 00376          CLRL    @(SP)+
                                                50   D4 00378          CLRL    I                                       0499
                              00000000GEF       40   04 B440 D0 0037A 24$:     MOVL    @SAVPTR[I], BAR[I]
                      EE                50 000000A4   8F F3 00384       AOBLEQ  #164, I, 24$
                                          6E    0294 8F B0 0038C       MOVW    #660, $XPO$DESC                         0502
                                    02    AE       02 90 00391       MOVB    #2, $XPO$DESC+2
```

```
                03   AE              02   90 00395          MOVB    #2, $XPO$DESC+3                                  ;
                04   AE        04    A4   D0 00399          MOVL    SAVPTR, $XPO$DESC+4                              ;
                                     57   DD 0039E          PUSHL   R7                                               ;
                                     7E   7C 003A0          CLRQ    -(SP)                                            ;
                     7E              02   CE 003A2          MNEGL   #2, -(SP)                                        ;
                                 10  AE   9F 003A5          PUSHAB  $XPO$DESC                                        ;
    00000000G    EF                  05   FB 003A8          CALLS   #5, XPO$FREE_MEM                                 ;
                                     65   D7 003AF          DECL    SAVSTK                                           ; 0505
    000000B3     8F                  53   D1 003B1 25$:     CMPL    R3, #179                                         ; 0508
                                     64   12 003B8          BNEQ    29$                                              ;
                                     52   D4 003BA          CLRL    R2                                               ;
50                   64              1C   C5 003BC          MULL3   #28, X, R0                                       ; 0514
                                 0C A540  9F 003C0          PUSHAB  SAVSTK+12[R0]                                    ;
                     66              9E   D1 003C4          CMPL    @(SP)+, GCA+188                                  ;
                                     0A   13 003C7          BEQL    27$                                              ;
                                     7E   D4 003C9 26$:     CLRL    -(SP)                                            ; 0517
                         00000000G   8F   DD 003CB          PUSHL   #RNFWFR                                          ;
                                     56   11 003D1          BRB     30$                                              ;
50                   64              1C   C5 003D3 27$:     MULL3   #28, X, R0                                       ; 0522
                                 04 A540  9F 003D7          PUSHAB  SAVSTK+4[R0]                                     ;
                04   A4              9E   D0 003DB          MOVL    @(SP)+, SAVPTR                                   ;
                                 04 A540  9F 003DF          PUSHAB  SAVSTK+4[R0]                                     ; 0523
                                     9E   D4 003E3          CLRL    @(SP)+                                           ;
                                     50   D4 003E5          CLRL    I                                                ; 0526
    00000000GEF40                04 B440  D0 003E7 28$:     MOVL    @SAVPTR[I], BAR[I]                               ;
EE                   50 00000129     8F   F3 003F1          AOBLEQ  #297, I, 28$                                     ;
                     6E     04A8     8F   B0 003F9          MOVW    #1192, $XPO$DESC                                 ; 0529
                02   AE              02   90 003FE          MOVB    #2, $XPO$DESC+2                                  ;
                03   AE              02   90 00402          MOVB    #2, $XPO$DESC+3                                  ;
                04   AE        04    A4   D0 00406          MOVL    SAVPTR, $XPO$DESC+4                              ;
                                     57   DD 0040B          PUSHL   R7                                               ;
                                     7E   7C 0040D          CLRQ    -(SP)                                            ;
                     7E              02   CE 0040F          MNEGL   #2, -(SP)                                        ;
                                 10  AE   9F 00412          PUSHAB  $XPO$DESC                                        ;
    00000000G    EF                  05   FB 00415          CALLS   #5, XPO$FREE_MEM                                 ;
                                     65   D7 0041C          DECL    SAVSTK                                           ; 0532
                     0F              52   E9 0041E 29$:     BLBC    R2, 31$                                          ; 0536
                                     7E   D4 00421          CLRL    -(SP)                                            ; 0539
                         00000000G   8F   DD 00423          PUSHL   #RNFNCS                                          ;
    00000000G    EF                  02   FB 00429 30$:     CALLS   #2, ERMA                                         ;
                                     04   00430 31$:     RET                                                      ; 0547

; Routine Size:  1073 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  422          0548  1 %SBTTL 'SAVRES -- Save & restore handler module'
;  423          0549  1 GLOBAL ROUTINE savini : NOVALUE =
;  424          0550  1
;  425          0551  1 !++
;  426          0552  1 ! FUNCTIONAL DESCRIPTION:
;  427          0553  1 !
;  428          0554  1 !       This routine is called from RINIT and changes the storage locations of several variables
;  429          0555  1 !       that need to be SAVEd but were previously stored in other block data areas.
;  430          0556  1 !
;  431          0557  1 ! FORMAL PARAMETERS:    none
;  432          0558  1 !
;  433          0559  1 ! IMPLICIT INPUTS:      none
;  434          0560  1 !
;  435          0561  1 ! IMPLICIT OUTPUTS:     none
;  436          0562  1 !
;  437          0563  1 ! ROUTINE VALUE and
;  438          0564  1 ! COMPLETION CODES:     none
;  439          0565  1 !
;  440          0566  1 ! SIDE EFFECTS:         none discovered yet
;  441          0567  1 !
;  442          0568  1 !--
;  443          0569  1
;  444          0570  2     BEGIN
;  445          0571  2
;  446          0572  2     BIND
;  447          0573  2     s = save : $save_block,
;  448          0574  2     sa = saveall : $saveall_block;
;  449          0575  2
;  450          0576  2 !We now reassign the storage locations for the following variables so that
;  451          0577  2 !they wil all be stored in the save or saveall block memory.
;  452          0578  2
;  453          0579  2         sca_f_justify   = s  [save$g_sca_justify];
;  454          0580  2         sca_f_fill      = s  [save$g_sca_fill];
;  455          0581  2         sca_f_cc_ok     = sa [saveal[$g_sca_cc_ok];
;  456          0582  2         sca_f_crock     = s  [save$g_sca_crock];
;  457          0583  2         sca_f_lm        = s  [save$g_sca_lm];
;  458          0584  2         sca_f_rm        = s  [save$g_sca_rm];
;  459          0585  2         sca_f_spacing   = s  [save$g_sca_spacing];
;  460          0586  2         sca_f_period    = sa [saveal[$g_sca_period];
;  461          0587  2         sca_f_ker       = s  [save$g_sca_ker];
;  462          0588  2         sca_f_bar_char  = sa [saveal[$g_sca_bar_char];
;  463          0589  2         sca_f_autotitle = s  [save$g_sca_autotitle];
;  464          0590  2         sca_f_flags     = s  [save$g_sca_flags];
;  465          0591  2
;  466          0592  2         gca_f_autopara  = sa [saveall$g_gca_autopara];
;  467          0593  2         gca_f_autosubt  = sa [saveall$g_gca_autosubt];
;  468          0594  2         gca_f_autotabl  = sa [saveall$g_gca_autotabl];
;  469          0595  2         gca_f_autojust  = sa [saveall$g_gca_autojust];
;  470          0596  2         gca_f_xcase     = sa [saveall$g_gca_xcase];
;  471          0597  2         gca_f_case      = sa [saveall$g_gca_case];
;  472          0598  2         gca_f_flag_cmd  = s  [save$g_gca_flag_cmd];
;  473          0599  2         gca_f_lwidth    = s  [save$g_gca_width];
;  474          0600  2         gca_f_keep      = s  [save$g_gca_keep];
;  475          0601  2
;  476          0602  2         phan_f_llines   = s  [save$g_phan_llines];
;  477          0603  2         phan_f_header   = sa [saveall$g_phan_header];
;  478          0604  2         phan_f_paging   = sa [saveall$g_phan_paging];
```

```
;  479          0605  2
;  480          0606  2          hct_f_date      = s  [save$g_hct_date];
;  481          0607  2          hct_f_headers   = s  [save$g_hct_headers];
;  482          0608  2          hct_f_number_page= s  [save$g_hct_number_pag];
;  483          0609  2          hct_f_hd_case   = s  [save$g_hct_hd_case];
;  484          0610  2          hct_f_subtitle  = s  [save$g_hct_subtitle];
;  485          0611  2          hct_f_nmpg_np   = s  [save$g_hct_nmpg_np];
;  486          0612  2
;  487          0613  1      END;                          ! End of SAVINI


                                                00FC 00000          .ENTRY  SAVINI, Save R2,R3,R4,R5,R6,R7                  ; 0549
                                    57 00000000G    EF   9E 00002   MOVAB   PHAN+4, R7                                      ;
                                    56 00000000G    EF   9E 00009   MOVAB   HCT+4, R6                                       ;
                                    55 00000000G    EF   9E 00010   MOVAB   GCA+4, R5                                       ;
                                    54 00000000G    EF   9E 00017   MOVAB   SA+32, R4                                       ;
                                    53 00000000G    EF   9E 0001E   MOVAB   SCA+100, R3                                     ;
                                    52 00000000G    EF   9E 00025   MOVAB   S+4, R2                                         ;
                                    63              62   9E 0002C   MOVAB   S+4, SCA+100                                   ; 0579
                                04  A3      04      A2   9E 0002F   MOVAB   S+8, SCA+104                                   ; 0580
                                08  A3              64   9E 00034   MOVAB   SA+32, SCA+108                                ; 0581
                                0C  A3      08      A2   9E 00038   MOVAB   S+12, SCA+112                                  ; 0582
                                10  A3      0C      A2   9E 0003D   MOVAB   S+16, SCA+116                                  ; 0583
                                14  A3      10      A2   9E 00042   MOVAB   S+20, SCA+120                                  ; 0584
                                18  A3      14      A2   9E 00047   MOVAB   S+24, SCA+124                                  ; 0585
                                1C  A3      08      A4   9E 0004C   MOVAB   SA+40, SCA+128                                ; 0586
                                20  A3      18      A2   9E 00051   MOVAB   S+28, SCA+132                                  ; 0587
                                24  A3      04      A4   9E 00056   MOVAB   SA+36, SCA+136                                ; 0588
                                28  A3      1C      A2   9E 0005B   MOVAB   S+32, SCA+140                                  ; 0589
                                2C  A3      20      A2   9E 00060   MOVAB   S+36, SCA+144                                  ; 0590
                                    65      E4      A4   9E 00065   MOVAB   SA+4, GCA+4                                    ; 0592
                                04  A5      E8      A4   9E 00069   MOVAB   SA+8, GCA+8                                    ; 0593
                                08  A5      EC      A4   9E 0006E   MOVAB   SA+12, GCA+12                                  ; 0594
                                0C  A5      E0      A4   9E 00073   MOVAB   SA, GCA+16                                     ; 0595
                                50  A5      F4      A4   9E 00078   MOVAB   SA+20, GCA+84                                  ; 0596
                                7C  A5      F0      A4   9E 0007D   MOVAB   SA+16, GCA+128                                 ; 0597
                              0080  C5      28      A2   9E 00082   MOVAB   S+44, GCA+132                                  ; 0598
                              0088  C5      24      A2   9E 00088   MOVAB   S+40, GCA+140                                  ; 0599
                              00C8  C5      2C      A2   9E 0008E   MOVAB   S+48, GCA+204                                  ; 0600
                                    67      48      A2   9E 00094   MOVAB   S+76, PHAN+4                                   ; 0602
                                18  A7      F8      A4   9E 00098   MOVAB   SA+24, PHAN+28                                 ; 0603
                                24  A7      FC      A4   9E 0009D   MOVAB   SA+28, PHAN+40                                 ; 0604
                                    66      34      A2   9E 000A2   MOVAB   S+56, HCT+4                                     ; 0606
                                04  A6      38      A2   9E 000A6   MOVAB   S+60, HCT+8                                     ; 0607
                                08  A6      40      A2   9E 000AB   MOVAB   S+68, HCT+12                                    ; 0608
                                10  A6      30      A2   9E 000B0   MOVAB   S+52, HCT+20                                    ; 0609
                                14  A6      44      A2   9E 000B5   MOVAB   S+72, HCT+24                                    ; 0610
                                28  A6      3C      A2   9E 000BA   MOVAB   S+64, HCT+44                                    ; 0611
                                                        04 000BF    RET                                                     ; 0613

; Routine Size:  192 bytes,    Routine Base:  $CODE$ + 0431

;  488          0614  1 END                           ! End of module
```

; 489 0615 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
OWN	128	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$PLIT$	134	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$CODE$	1265	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time
_255DUA28:[SYSLIB]XPORT.L32;1	590	128	21	252	00:00.1
_255DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	103	8	86	00:00.3

COMMAND QUALIFIERS

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

; Size: 1265 code + 262 data bytes
; Run Time: 01:06.4
; Elapsed Time: 01:57.8
; Lines/CPU Min: 556
; Lexemes/CPU-Min:101574
; Memory Used: 417 pages
; Compilation Complete

RSDERM
LIS

RTERM
LIS

RNODEF
LIS

RUNOFF
LIS

RNOVMS
LIS

RSKIPS
LIS

RNODAT
LIS

SAVRES
LIS

RNOURS
LIS

RNFERM
LIS