F 1

```
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUU         UUU     GGGGGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEEEEEEEEEEE     BBBBBBBBBBBB     UUU         UUU  GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG   GGGGGGGGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDD         DDD  EEE              BBB         BBB  UUU         UUU  GGG         GGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
DDDDDDDDDDDD     EEEEEEEEEEEEEEE  BBBBBBBBBBBB     UUUUUUUUUUUUUUU     GGGGGGGGG
```

FILEID**DBGNCANCL

```
DDDDDDDD    BBBBBBBB      GGGGGGGG  NN        NN    CCCCCCCC    AAAAAA    NN        NN    CCCCCCCC  LL
DDDDDDDD    BBBBBBBB      GGGGGGGG  NN        NN    CCCCCCCC    AAAAAA    NN        NN    CCCCCCCC  LL
DD      DD  BB      BB  GG          NN        NN  CC          AA      AA  NN        NN  CC          LL
DD      DD  BB      BB  GG          NN        NN  CC          AA      AA  NN        NN  CC          LL
DD      DD  BB      BB  GG          NNNN      NN  CC          AA      AA  NNNN      NN  CC          LL
DD      DD  BB      BB  GG          NNNN      NN  CC          AA      AA  NNNN      NN  CC          LL
DD      DD  BBBBBBBB    GG          NN  NN    NN  CC          AA      AA  NN  NN    NN  CC          LL
DD      DD  BBBBBBBB    GG          NN  NN    NN  CC          AA      AA  NN  NN    NN  CC          LL
DD      DD  BB      BB  GG  GGGGGG  NN    NNNN    CC          AAAAAAAAAA  NN    NNNN    CC          LL
DD      DD  BB      BB  GG  GGGGGG  NN    NNNN    CC          AAAAAAAAAA  NN    NNNN    CC          LL
DD      DD  BB      BB  GG      GG  NN        NN  CC          AA      AA  NN        NN  CC          LL                ....
DD      DD  BB      BB  GG      GG  NN        NN  CC          AA      AA  NN        NN  CC          LL                ....
DDDDDDDD    BBBBBBBB      GGGGGG    NN        NN    CCCCCCCC  AA      AA  NN        NN    CCCCCCCC  LLLLLLLLLL        ....
DDDDDDDD    BBBBBBBB      GGGGGG    NN        NN    CCCCCCCC  AA      AA  NN        NN    CCCCCCCC  LLLLLLLLLL        ....


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
```

```
0 MODULE DBGNCANCL (IDENT = 'V04-000') =
0
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !
1 ! MODULE FUNCTION
1 !       This module contains the command parse and execution networks for
1 !       the CANCEL command.
1 !
1 ! AUTHOR:
1 !       David Plummer
1 !
1 ! CREATION DATE:
1 !       9-Jul-80
1 !
1 ! MODIFIED BY:
1 !       Richard Title   16-Sep-81
1 !
1 ! REVISION HISTORY:
1 !
1 ! 3.01  16-SEP-81       RT      Implemented CANCEL SOURCE command
1 ! 3.02  07-MAY-82       RT      Implemented CANCEL DEVELOPER
1 !
1
1 REQUIRE 'SRC$:DBGPROLOG.REQ';
1
1 LIBRARY 'LIB$:DBGGEN.L32';
1
1 FORWARD ROUTINE
1       DBG$NPARSE_CANCEL,              ! ATN parse network for CANCEL
1       DBG$NEXECUTE_CANCEL;            ! Command execution network for CANCEL
```

```
1 EXTERNAL ROUTINE
1      DBG$EVENT_SHOW_CANCEL_SYNTAX,            ! Syntax for SHOW|CANCEL BREAK|TRACE|WATCH
1      DBG$EVENT_SHOW_CANCEL_SEMANTICS,         ! Semantics for SHOW|CANCEL BREAK|TRACE.WATCH
1      DBG$EVENT_CANCEL_ALL,                    ! CANCEL/ALL eventpoints
1      DBG$RST_SETSCOPE: NOVALUE,               ! Cancels (and sets) user scope
1      DBG$RST_CANMOD,                          ! Cancels a module
1      DBG$NSAVE_STRING,                        ! Saves a string form the input stream
1      DBG$IS_IT_ENTRY,                         ! Returns true if address = entry point
1      DBG$GET_TEMPMEM,                         ! Allocates dynamic listed storage
1      DBG$SET_MOD_DEF,                         ! Resets mode level to default
1      DBG$NGET_TRANS_RADIX,                    ! Translate radix
1      DBG$NMATCH,                              ! Matches counted strings to input
1      DBG$SCR_EXECUTE_CANDISP_CMD:NOVALUE,     ! Execute the CANCEL DISPLAY command
1      DBG$SCR_EXECUTE_CANWIND_CMD:NOVALUE,     ! Execute the CANCEL WINDOW command
1      DBG$SCR_PARSE_CANDISP_CMD: NOVALUE,      ! Parse the CANCEL DISPLAY command
1      DBG$SCR_PARSE_CANWIND_CMD: NOVALUE,      ! Parse the CANCEL WINDOW command
1      DBG$SRC_CANCEL_SOURCE: NOVALUE,          ! Implements CANCEL SOURCE command
1      DBG$STA_GETSOURCEMOD,                    ! Looks up module rst pointer
1      DBG$SET_STP_DEF: NOVALUE,                ! Sets default step
1      DBG$NSYNTAX_ERROR,                       ! Formats a syntax error
1      DBG$NNEXT_WORD,                          ! Returns the next word of input
1      DBG$NPARSE_ADDRESS,                      ! Obtains an address expression descriptor
1      DBG$NSAVE_DECIMAL_INTEGER,               ! Parse an integer
1      DBG$NMAKE_ARG_VECT;                      ! Constructs a message argument vector
1
1 EXTERNAL
1      DBG$GB_RADIX: VECTOR[3, BYTE],           ! Radix settings
1      DBG$GL_DEVELOPER: BITVECTOR,             ! Developer switches
1      DBG$GL_GBLTYP,                           ! Override type
1      DBG$GW_GBLLNGTH: WORD,                   ! Override length
1      DBG$GL_DFLTTYP,                          ! Default type
1      DBG$GW_DFLTLENG: WORD,                   ! Default length
1      DBG$RUNFRAME: BLOCK [,BYTE],             ! User runframe
1      DBG$GB_RESIGNAL: BYTE,                   ! Flag for resignaling exceptions
1      DBG$GL_CONTEXT: BITVECTOR;               ! Context word
1
1 LITERAL
1
1      ! Legal verb composites
1      !
1      CANCEL_MINIMUM              = 1,
1      CANCEL_ALL                  = 1,
1      CANCEL_BREAK                = 2,    ! Also EVENT$K_CANCEL_BREAK
1      CANCEL_BREAK_ALL            = 3,
1      CANCEL_EXCEPTION_BREAK      = 4,    ! Also EVENT$K_CANCEL_BREAK_EXC
1      CANCEL_MODE                 = 5,
1      CANCEL_MODULE               = 6,
1      CANCEL_MODULE_ALL           = 7,
1      CANCEL_RADIX                = 20,
1      CANCEL_RADIX_OVERRIDE       = 21,
1      CANCEL_SCOPE                = 8,
1      CANCEL_TRACE                = 9,    ! Also EVENT$K_CANCEL_TRACE
1      CANCEL_TRACE_CALLS          = 10,
1      CANCEL_TRACE_BRANCH         = 11,
1      CANCEL_TRACE_ALL            = 12,
1      CANCEL_TYPE_OVERRIDE        = 13,
1      CANCEL_WATCH                = 14,   ! Also EVENT$K_CANCEL_WATCH
```

```
;  114         0246  1     CANCEL_WATCH_ALL          = 15,
;  115         0247  1     CANCEL_SOURCE             = 16,
;  116         0248  1     CANCEL_DEVELOPER          = 17,
;  117         0249  1     CANCEL_DISPLAY            = 18,
;  118         0250  1     CANCEL_WINDOW             = 19,
;  119         0251  1     CANCEL_MAXIMUM            = 21;
```

```
1 GLOBAL ROL .NE DBG$NPARSE_CANCEL (INPUT_DESC, VERB_NODE, MESSAGE_VECT) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine comprises the ATN parse network for the CANCEL verb.
1 !       A command execution tree is constructed during the parsing process
1 !       which is used as input to the command execution network following
1 !       a complete and successful parse. Upon detection a a syntax error,
1 !       a message argument vector is constructed and returned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       INPUT_DESC      - A longword containing the address of a standard
1 !                         ASCII string descriptor corresponding to the input
1 !                         command
1 !
1 !       VERB_NODE       - A longword containing the address of the command
1 !                         verb node which is the head node of the command
1 !                         execution tree
1 !
1 !       MESSAGE_VECT    - The address of a longword to contain the address of
1 !                         a standard message argument vector upon detection of
1 !                         errors
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       The command execution tree corresponding to the input command is constructed
1 !       on success.
1 !
1 !       On failure, a message argument vector is constructed and returned.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       An unsigned integer longword completion code
1 !
1 ! COMPLETION CODES:
1 !
1 !       STS$K_SUCCESS (1)       - Success. Command parsed and execution tree made.
1 !
1 !       STS$K_SEVERE  (4)       - Failure. No tree constructed. Message argument
1 !                                 vector constructed and returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BEGIN
2
2    MAP
2       VERB_NODE: REF DBG$VERB_NODE;   ! Pointer to command Verb Node
```

```
2
2     ! Define strings used at this level of parsing
2     !
2     BIND
2         DBG$CS_ALL              = UPLIT BYTE (%ASCIC 'ALL'),
2         DBG$CS_BREAK            = UPLIT BYTE (%ASCIC 'BREAK'),
2         DBG$CS_DEVELOPER        = UPLIT BYTE (%ASCIC 'DEVELOPER'),
2         DBG$CS_DISPLAY          = UPLIT BYTE (%ASCIC 'DISPLAY'),
2         DBG$CS_EXCEPTION        = UPLIT BYTE (%ASCIC 'EXCEPTION'),
2         DBG$CS_MODE             = UPLIT BYTE (%ASCIC 'MODE'),
2         DBG$CS_MODULE           = UPLIT BYTE (%ASCIC 'MODULE'),
2         DBG$CS_RADIX            = UPLIT BYTE (%ASCIC 'RADIX'),
2         DBG$CS_SCOPE            = UPLIT BYTE (%ASCIC 'SCOPE'),
2         DBG$CS_SOURCE           = UPLIT BYTE (%ASCIC 'SOURCE'),
2         DBG$CS_TRACE            = UPLIT BYTE (%ASCIC 'TRACE'),
2         DBG$CS_TYPE             = UPLIT BYTE (%ASCIC 'TYPE'),
2         DBG$CS_WATCH            = UPLIT BYTE (%ASCIC 'WATCH'),
2         DBG$CS_WINDOW           = UPLIT BYTE (%ASCIC 'WINDOW'),
2         DBG$CS_EQUAL            = UPLIT BYTE (%ASCIC '='),
2         DBG$CS_SLASH            = UPLIT BYTE (%ASCIC '/'),
2         DBG$CS_COMMA            = UPLIT BYTE (%ASCIC ','),
2         DBG$CS_CR               = UPLIT BYTE (1, dbg$k_car_return);
2
2     LOCAL
2         STATUS,                         ! Holds routine's return status
2         NOUN_NODE: REF DBG$NOUN_NODE;   ! Noun node of command execution tree
2
2
2
2     ! Create and link a noun node. Note that the noun node will not
2     ! be used for certain commands like CANCEL BREAK/ALL.
2     !
2     noun_node = dbg$get_tempmem (dbg$k_noun_node_size);
2     verb_node [dbg$l_verb_object_ptr] = .noun_node;
2
2
2     ! Parse the next keyword and transfer control to a subnetwork
2     !
2     SELECTONE TRUE OF
2         SET
2
2         [dbg$nmatch (.input_desc, dbg$cs_all, 1)] :     ! Cancel all
3             BEGIN
3             verb_node [dbg$b_verb_composite] = cancel_all;
2             END;
2
2         [dbg$nmatch (.input_desc, dbg$cs_break, 1)] :   ! CANCEL BREAK
3             BEGIN
3             VERB_NODE [DBG$B_VERB_COMPOSITE] = EVENT$K_CANCEL_BREAK;
3             RETURN DBG$EVENT_SHOW_CANCEL_SYNTAX (.INPUT_DESC,
3                                                  .VERB_NODE,
3                                                  .MESSAGE_VECT
3                                                 );
2             END;
2
2         [dbg$nmatch (.input_desc, dbg$cs_developer, 9)] :       ! Set Developer
3             BEGIN
```

```
;  3                LOCAL
;  3                    link;
;  3
;  3                verb_node [dbg$b_verb_composite] = cancel_developer;
;  3                link = verb_node[dbg$l_verb_object_ptr];
;  3                IF NOT dbg$nmatch(.input_desc, dbg$cs_cr, 1)
;  3                THEN
;  4                    BEGIN
;  4                    WHILE true DO
;  5                        BEGIN
;  5                        IF NOT dbg$nsave_decimal_integer(.input_desc, noun_node[dbg$l_noun_value],
;  5                                                          .message_vect)
;  5                        THEN
;  5                            RETURN sts$k_severe;
;  5
;  5                        IF (.noun_node[dbg$l_noun_value] LSS 0) OR
;  6                           (.noun_node[dbg$l_noun_value] GTR 31)
;  5                        THEN
;  6                            BEGIN
;  6                            .message_vect = dbg$nmake_arg_vect(dbg$_bitrange);
;  6                            RETURN sts$k_severe;
;  5                            END;
;  5
;  5                        link = noun_node[dbg$l_noun_link];
;  5                        IF NOT dbg$nmatch(.input_desc, dbg$cs_comma, 1)
;  5                        THEN
;  6                            BEGIN
;  6                            IF NOT dbg$nmatch(.input_desc, dbg$cs_cr, 1)
;  6                            THEN
;  7                                BEGIN
;  7                                .message_vect = dbg$nsyntax_error(dbg$nnext_word(.input_desc));
;  7                                RETURN sts$k_severe;
;  7                                END
;  7
;  6                            ELSE
;  6                                EXITLOOP;
;  6
;  5                            END;
;  5
;  5                        noun_node = dbg$get_tempmem (dbg$k_noun_node_size);
;  5                        .link = .noun_node;
;  4                        END;                    ! End of WHILE loop.
;  4
;  3                    END;
;  3
;  3                    .link = 0;
;  3
;  2                END;
;  2
;  2
;  2            ! Parse the CANCEL DISPLAY command.
;  2            !
;  2            [DBG$NMATCH(.INPUT_DESC, DBG$CS_DISPLAY, 3)]:
;  3                BEGIN
;  3                VERB_NODE[DBG$B_VERB_COMPOSITE] = CANCEL_DISPLAY;
;  3                DBG$SCR_PARSE_CANDISP_CMD(.INPUT_DESC, .VERB_NODE);
;  2                END;
```

```
2
2
2      ! Parse the CANCEL EXCEPTION BREAK command.
2      !
2      [dbg$nmatch (.input_desc, dbg$cs_exception, 1)] : ! CANCEL EXCEPTION BREAK
3          BEGIN
3
3          ! We look for BREAK
3          !
3          IF NOT dbg$nmatch (.input_desc, dbg$cs_break, 1)
3          THEN
4              BEGIN
4              .message_vect =
5              (
5               IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
5               THEN
5                   dbg$nmake_arg_vect (dbg$_needmore)
5               ELSE
5                   dbg$nsyntax_error (dbg$nnext_word (.input_desc))
4              );
4              RETURN sts$k_severe;
3              END;
3
3          verb_node [dbg$b_verb_composite] = cancel_exception_break;
3
3
3          ! Reset the noun and adverb pointers.
3          !
3          verb_node [dbg$l_verb_object_ptr] = 0;
3          verb_node [dbg$l_verb_adverb_ptr] = 0;
2          END;
2
2      [dbg$nmatch (.input_desc, dbg$cs_mode, 1)] :     ! CANCEL MODE
3          BEGIN
3          verb_node [dbg$b_verb_composite] = cancel_mode;
2          END;
2
2      [dbg$nmatch (.input_desc, dbg$cs_module, 4)] :  ! CANCEL MODULE
3          BEGIN
3
3          ! Check for CANCEL MODULE/ALL
3          !
3          IF dbg$nmatch (.input_desc, dbg$cs_slash, 1)
3          THEN
4              BEGIN
4              BIND
4                  DBG$CS_ALL = UPLIT BYTE (3, 'ALL');
4
4              IF NOT dbg$nmatch (.input_desc, dbg$cs_all, 1)
4              THEN
5                  BEGIN
5                  .message_vect =
6                  (IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
6                   THEN
6                       dbg$nmake_arg_vect (dbg$_needmore)
6                   ELSE
5                       dbg$nsyntax_error (dbg$nnext_word (.input_desc)));
```

```
5                       RETURN sts$k_severe;
4                       END;
4
4                   verb_node [dbg$b_verb_composite] = cancel_module_all;
4                   END
4
3               ELSE
4                   BEGIN
4
4                   ! We have a module name list to parse
4                   !
4                   BIND
4                       DBG$CS_COMMA = UPLIT BYTE (1, dbg$k_comma);
4                   LOCAL
4                       LINK;                   ! Temporary pointer
4
4                   ! Accept strings and commas
4                   !
4                   WHILE true
4                   DO
5                       BEGIN
5
5                       IF NOT DBG$NSAVE_STRING  (.input_desc,
5                                                 noun_node [dbg$l_noun_value],
5                                                 .message_vect)
5                       THEN
5                           RETURN sts$k_severe;
5
5
5                       ! Check for a comma
5                       !
5                       IF NOT dbg$nmatch (.input_desc, dbg$cs_comma, 1)
5                       THEN
5                           EXITLOOP;
5
5
5                       ! Create a new noun node to hold the next string
5                       !
5                       link = noun_node [dbg$l_noun_link];
5                       noun_node = dbg$get_tempmem (dbg$k_noun_node_size);
5                       .link = .noun_node;
5
4                       END;        ! End of loop
4
4
4                   ! Place a zero in the last link field
4                   !
4                   noun_node [dbg$l_noun_link] = 0;
4
4                   verb_node [dbg$b_verb_composite] = cancel_module;
4
3                   END;
3
2               END;
2
2           [dbg$nmatch (.input_desc, dbg$cs_radix, 1)]:
3               BEGIN
```

```
3                   BIND
3                       DBG$CS_OVERRIDE = UPLIT BYTE (8, 'OVERRIDE');
3
3                   verb_node[dbg$b_verb_composite] = cancel_radix;
3
3                   ! Look for the /
3                   !
3                   IF dbg$nmatch (.input_desc, dbg$cs_slash, 1)
3                   THEN
4                       BEGIN
4
4                       ! Look for 'override'
4                       !
4                       IF NOT dbg$nmatch (.input_desc, dbg$cs_override, 1)
4                       THEN
5                           BEGIN
5                           .message_vect =
6                               (IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
6                                THEN
6                                    dbg$nmake_arg_vect (dbg$_needmore)
6                                ELSE
5                           dbg$nsyntax_error (dbg$nnext_word (.input_desc)));
5                           RETURN sts$k_severe;
4                           END;
4
4                       verb_node [dbg$b_verb_composite] = cancel_radix_override;
3                       END;
2                   END;
2
2               [dbg$nmatch (.input_desc, dbg$cs_scope, 1)] :
3                   BEGIN
3                   verb_node [dbg$b_verb_composite] = cancel_scope;
2                   END;
2
2               [dbg$nmatch (.input_desc, dbg$cs_source, 2)] :
3                   BEGIN
3                   verb_node[dbg$b_verb_composite] = cancel_source;
3
3                   ! Check for CANCEL SOURCE/MODULE=modname
3
3                   IF dbg$nmatch (.input_desc, dbg$cs_slash, 1)
3                   THEN
4                       BEGIN
4                       LOCAL
4                           modnameptr;
4                       BIND
4                           dbg$cs_module = UPLIT BYTE (6, 'MODULE');
4
4                       ! Read the string MODULE
4
4                       IF NOT dbg$nmatch (.input_desc, dbg$cs_module, 4)
4                       THEN
5                           BEGIN
5                           .message_vect = dbg$nsyntax_error(
5                               dbg$nnext_word(.input_desc));
5                           RETURN sts$k_severe;
4                           END;
```

```

        ! Read the = sign

        IF NOT dbg$nmatch (.input_desc, dbg$cs_equal, 1)
        THEN
            BEGIN
            .message_vect = dbg$nsyntax_error(
                dbg$nnext_word(.input_desc));
            RETURN sts$k_severe;
            END;

        ! Read the module name

        IF NOT dbg$nsave_string (.input_desc,
            modnameptr, .message_vect)
        THEN
            RETURN sts$k_severe;

        ! Convert the module name into an rst pointer

        noun_node[dbg$l_noun_value] =
            dbg$sta_getsourcemod(.modnameptr);

        ! If the above routine returns zero then the user has
        ! entered an invalid module name.

        IF .noun_node[dbg$l_noun_value] EQL 0
        THEN
            BEGIN
            .message_vect = dbg$nmake_arg_vect(
                dbg$_nosuchmodu, 1, .modnameptr);
            RETURN sts$k_severe;
            END;

        END ! CANCEL SOURCE/MODULE=modname

    ELSE ! the user has just entered CANCEL SOURCE

        noun_node[dbg$l_noun_value] = 0;

    END; ! CANCEL SOURCE

[dbg$nmatch (.input_desc, dbg$cs_trace, 1)] :    ! CANCEL TRACE
    BEGIN
    VERB_NODE [DBG$B_VERB_COMPOSITE] = EVENT$K_CANCEL_TRACE;
    RETURN DBG$EVENT_SHOW_CANCEL_SYNTAX (.INPUT_DESC,
                                         .VERB_NODE,
                                         .MESSAGE_VECT
                                        );
    END;

[dbg$nmatch (.input_desc, dbg$cs_type, 2)] :     ! CANCEL TYPE/OVERRIDE
    BEGIN
    BIND
        DBG$CS_OVERRIDE = UPLIT BYTE (8, 'OVERRIDE');

    ! Look for the /
```

```
;  3                  !
;  3                  IF NOT dbg$nmatch (.input_desc, dbg$cs_slash, 1)
;  3                  THEN
;  4                      BEGIN
;  4                      .message_vect =
;  5                      (IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
;  5                      THEN
;  5                          dbg$nmake_arg_vect (dbg$_needmore)
;  5                      ELSE
;  4                          dbg$nsyntax_error (dbg$nnext_word (.input_desc)));
;  4                      RETURN sts$k_severe;
;  3                      END;
;  3
;  3
;  3                  ! Look for 'override'
;  3                  !
;  3                  IF NOT dbg$nmatch (.input_desc, dbg$cs_override, 1)
;  3                  THEN
;  4                      BEGIN
;  4                      .message_vect =
;  5                      (IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
;  5                      THEN
;  5                          dbg$nmake_arg_vect (dbg$_needmore)
;  5                      ELSE
;  4                          dbg$nsyntax_error (dbg$nnext_word (.input_desc)));
;  4                      RETURN sts$k_severe;
;  3                      END;
;  3
;  3                  verb_node [dbg$b_verb_composite] = cancel_type_override;
;  2                  END;
;  2
;  2              [dbg$nmatch (.input_desc, dbg$cs_watch, 1)] :    ! CANCEL WATCH
;  3                  BEGIN
;  3                  VERB_NODE [DBG$B_VERB_COMPOSITE] = EVENT$K_CANCEL_WATCH;
;  3                  RETURN DBG$EVENT_SHOW_CANCEL_SYNTAX (.INPUT_DESC,
;  3                                                       .VERB_NODE,
;  3                                                       .MESSAGE_VECT
;  3                                                      );
;  2                  END;
;  2
;  2              ! Parse the CANCEL WINDOW command.
;  2              !
;  2              [DBG$NMATCH(.INPUT_DESC, DBG$CS_WINDOW, 3)]:
;  3                  BEGIN
;  3                  VERB_NODE[DBG$B_VERB_COMPOSITE] = CANCEL_WINDOW;
;  3                  DBG$SCR_PARSE_CANWIND_CMD(.INPUT_DESC, .VERB_NODE);
;  2                  END;
;  2
;  2
;  2              ! Any other CANCEL command constitutes a syntax error.
;  2              !
;  2              [OTHERWISE] :    ! Syntax error
;  3                  BEGIN
;  3                  .message_vect =
;  4                  (
;  4                   IF dbg$nmatch (.input_desc, dbg$cs_cr, 1)
;  4                   THEN
;  577          0708  4                 dbg$nmake_arg_vect (dbg$_needmore)
;  578          0709  4             ELSE
;  579          0710  4                 dbg$nsyntax_error (dbg$nnext_word (.input_desc))
;  580          0711  3             );
;  581          0712  3             RETURN sts$k_severe;
;  582          0713  2             END;
;  583          0714  2
;  584          0715  2         TES;
;  585          0716  2
;  586          0717  2     RETURN STS$K_SUCCESS;
;  587          0718  2
;  588          0719  1     END;


                                                                        .TITLE  DBGNCANCL
                                                                        .IDENT  \V04-000\

                                                                        .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

                                      4C  4C  41  03  00000 P.AAA:    .ASCII  <3>\ALL\                          ;
                              4B  41  45  52  42  05  00004 P.AAB:    .ASCII  <5>\BREAK\                        ;
          52  45  50  4F  4C  45  56  45  44  09  0000A P.AAC:    .ASCII  <9>\DEVELOPER\                    ;
                  59  41  4C  50  53  49  44  07  00014 P.AAD:    .ASCII  <7>\DISPLAY\                      ;
          4E  4F  49  54  50  45  43  58  45  09  0001C P.AAE:    .ASCII  <9>\EXCEPTION\                    ;
                              45  44  4F  4D  04  00026 P.AAF:    .ASCII  <4>\MODE\                         ;
                      45  4C  55  44  4F  4D  06  0002B P.AAG:    .ASCII  <6>\MODULE\                       ;
                          58  49  44  41  52  05  00032 P.AAH:    .ASCII  <5>\RADIX\                        ;
                          45  50  4F  43  53  05  00038 P.AAI:    .ASCII  <5>\SCOPE\                        ;
                      45  43  52  55  4F  53  06  0003E P.AAJ:    .ASCII  <6>\SOURCE\                       ;
                          45  43  41  52  54  05  00045 P.AAK:    .ASCII  <5>\TRACE\                        ;
                              45  50  59  54  04  0004B P.AAL:    .ASCII  <4>\TYPE\                         ;
                          48  43  54  41  57  05  00050 P.AAM:    .ASCII  <5>\WATCH\                        ;
                      57  4F  44  4E  49  57  06  00056 P.AAN:    .ASCII  <6>\WINDOW\                       ;
                                          3D  01  0005D P.AAO:    .ASCII  <1>\=\                            ;
                                          2F  01  0005F P.AAP:    .ASCII  <1>\/\                            ;
                                          2C  01  00061 P.AAQ:    .ASCII  <1>\,\                            ;
                                          0D  01  00063 P.AAR:    .BYTE   1, 13                             ;
                                              03  00065 P.AAS:    .BYTE   3                                 ;
                                      4C  4C  41  00066           .ASCII  \ALL\                             ;
                                          2C  01  00069 P.AAT:    .BYTE   1, 44                             ;
                                              08  0006B P.AAU:    .BYTE   8                                 ;
                  45  44  49  52  52  45  56  4F  0006C           .ASCII  \OVERRIDE\                        ;
                                              06  00074 P.AAV:    .BYTE   6                                 ;
                          45  4C  55  44  4F  4D  00075           .ASCII  \MODULE\                          ;
                                              08  0007B P.AAW:    .BYTE   8                                 ;
                  45  44  49  52  52  45  56  4F  0007C           .ASCII  \OVERRIDE\                        ;

                                                        DBG$CS_ALL=             P.AAA
                                                        DBG$CS_BREAK=           P.AAB
                                                        DBG$CS_DEVELOPER=       P.AAC
                                                        DBG$CS_DISPLAY=         P.AAD
                                                        DBG$CS_EXCEPTION=       P.AAE
                                                        DBG$CS_MODE=            P.AAF
                                                        DBG$CS_MODULE=          P.AAG
                                                        DBG$CS_RADIX=           P.AAH
                                                        DBG$CS_SCOPE=           P.AAI
                                                        DBG$CS_SOURCE=          P.AAJ
```

```
                                        DBG$CS_TRACE=         P.AAK
                                        DBG$CS_TYPE=          P.AAL
                                        DBG$CS_WATCH=         P.AAM
                                        DBG$CS_WINDOW=        P.AAN
                                        DBG$CS_EQUAL=         P.AAO
                                        DBG$CS_SLASH=         P.AAP
                                        DBG$CS_COMMA=         P.AAQ
                                        DBG$CS_CR=            P.AAR
                                        DBG$CS_ALL=           P.AAS
                                        DBG$CS_COMMA=         P.AAT
                                        DBG$CS_OVERRIDE=      P.AAU
                                        DBG$CS_MODULE=        P.AAV
                                        DBG$CS_OVERRIDE=      P.AAW
                                                .EXTRN  DBG$EVENT_SHOW_CANCEL_SYNTAX
                                                .EXTRN  DBG$EVENT_SHOW_CANCEL_SEMANTICS
                                                .EXTRN  DBG$EVENT_CANCEL_ALL
                                                .EXTRN  DBG$RST_SETSCOPE
                                                .EXTRN  DBG$RST_CANMOD, DBG$NSAVE_STRING
                                                .EXTRN  DBG$IS_IT_ENTRY
                                                .EXTRN  DBG$GET_TEMPMEM
                                                .EXTRN  DBG$SET_MOD_DEF
                                                .EXTRN  DBG$NGET_TRANS_RADIX
                                                .EXTRN  DBG$NMATCH, DBG$SCR_EXECUTE_CANDISP_CMD
                                                .EXTRN  DBG$SCR_EXECUTE_CANWIND_CMD
                                                .EXTRN  DBG$SCR_PARSE_CANDISP_CMD
                                                .EXTRN  DBG$SCR_PARSE_CANWIND_CMD
                                                .EXTRN  DBG$SRC_CANCEL_SOURCE
                                                .EXTRN  DBG$STA_GETSOURCEMOD
                                                .EXTRN  DBG$SET_STP_DEF
                                                .EXTRN  DBG$NSYNTAX_ERROR
                                                .EXTRN  DBG$NNEXT_WORD, DBG$NPARSE_ADDRESS
                                                .EXTRN  DBG$NSAVE_DECIMAL_INTEGER
                                                .EXTRN  DBG$NMAKE_ARG_VECT
                                                .EXTRN  DBG$GB_RADIX, DBG$GL_DEVELOPER
                                                .EXTRN  DBG$GL_GBLTYP, DBG$GQ_GBLLNGTH
                                                .EXTRN  DBG$GL_DFLTTYP, DBG$GQ_DFLTLENG
                                                .EXTRN  DBG$RUNFRAME, DBG$GB_RESIGNAL
                                                .EXTRN  DBG$GL_CONTEXT

                                                .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                            07FC 00000          .ENTRY  DBG$NPARSE_CANCEL, Save R2,R3,R4,R5,R6,R7,- ; 0252
                                                        R8,R9,R10                                    ;
        5A 00000000G    00   9E 00002           MOVAB   DBG$NMAKE_ARG_VECT, R10                      ;
        59 00000000G    00   9E 00009           MOVAB   DBG$NSAVE_STRING, R9                         ;
        58 00000000G    00   9E 00010           MOVAB   DBG$GET_TEMPMEM, R8                          ;
        57 00000000G    00   9E 00017           MOVAB   DBG$NMATCH, R7                               ;
        56 00000000'    EF   9E 0001E           MOVAB   DBG$CS_SLASH, R6                             ;
        5E              04   C2 00025           SUBL2   #4, SP                                       ;
                        04   DD 00028           PUSHL   #4                                           ; 0341
        68              01   FB 0002A           CALLS   #1, DBG$GET_TEMPMEM                          ;
        54              50   D0 0002D           MOVL    R0, NOUN_NODE                                ;
        52         04   AC   7D 00030           MOVQ    INPUT_DESC, R2                               ; 0350
  08    A3              54   D0 00034           MOVL    NOUN_NODE, 8(R3)                             ; 0342
                        01   DD 00038           PUSHL   #1                                           ; 0350
                   A1   A6   9F 0003A           PUSHAB  DBG$CS_ALL                                   ;
                        52   DD 0003D           PUSHL   R2                                           ;
```

```
                    67         03 FB 0003F        CALLS   #3, DBG$NMATCH                              ;
                    01         50 D1 00042        CMPL    R0, #1                                      ;
                               07 12 00045        BNEQ    1$                                          ;
          01        A3         01 90 00047        MOVB    #1, 1(R3)                                   ; 0352
                             0287 31 0004B        BRW     43$                                         ; 0347
                               01 DD 0004E 1$:    PUSHL   #1                                          ; 0355
                          A5   A6 9F 00050        PUSHAB  DBG$CS_BREAK                                ;
                               52 DD 00053        PUSHL   R2                                          ;
                    67         03 FB 00055        CALLS   #3, DBG$NMATCH                              ;
                    01         50 D1 00058        CMPL    R0, #1                                      ;
                               07 12 0005B        BNEQ    2$                                          ;
          01        A3         02 90 0005D        MOVB    #2, 1(R3)                                   ; 0357
                             0244 31 00061        BRW     35$                                         ; 0360
                               09 DD 00064 2$:    PUSHL   #9                                          ; 0364
                          AB   A6 9F 00066        PUSHAB  DBG$CS_DEVELOPER                            ;
                               52 DD 00069        PUSHL   R2                                          ;
                    67         03 FB 0006B        CALLS   #3, DBG$NMATCH                              ;
                    01         50 D1 0006E        CMPL    R0, #1                                      ;
                               6B 12 00071        BNEQ    9$                                          ;
          01        A3         11 90 00073        MOVB    #17, 1(R3)                                  ; 0369
                    55    08   A3 9E 00077        MOVAB   8(R3), LINK                                 ; 0370
                               01 DD 0007B        PUSHL   #1                                          ; 0371
                          04   A6 9F 0007D        PUSHAB  DBG$CS_CR                                   ;
                               52 DD 00080        PUSHL   R2                                          ;
                    67         03 FB 00082        CALLS   #3, DBG$NMATCH                              ;
                    52         50 E8 00085        BLBS    R0, 8$                                      ;
                          0C   AC DD 00088 3$:    PUSHL   MESSAGE_VECT                                ; 0377
                               14 BB 0008B        PUSHR   #^M<R2,R4>                                  ; 0376
   00000000G        00         03 FB 0008D        CALLS   #3, DBG$NSAVE_DECIMAL_INTEGER               ;
                    03         50 E8 00094        BLBS    R0, 4$                                      ;
                             0267 31 00097        BRW     42$                                         ;
                               64 D5 0009A 4$:    TSTL    (NOUN_NODE)                                 ; 0381
                               05 19 0009C        BLSS    5$                                          ;
                    1F         64 D1 0009E        CMPL    (NOUN_NODE), #31                            ; 0382
                               09 15 000A1        BLEQ    6$                                          ;
                    00028248   8F DD 000A3 5$:    PUSHL   #164424                                     ; 0385
                             023A 31 000A9        BRW     39$                                         ;
                    55    08   A4 9E 000AC 6$:    MOVAB   8(R4), LINK                                 ; 0389
                               01 DD 000B0        PUSHL   #1                                          ; 0390
                          02   A6 9F 000B2        PUSHAB  DBG$CS_COMMA                                ;
                               52 DD 000B5        PUSHL   R2                                          ;
                    67         03 FB 000B7        CALLS   #3, DBG$NMATCH                              ;
                    10         50 E8 000BA        BLBS    R0, 7$                                      ;
                               01 DD 000BD        PUSHL   #1                                          ; 0393
                          04   A6 9F 000BF        PUSHAB  DBG$CS_CR                                   ;
                               52 DD 000C2        PUSHL   R2                                          ;
                    67         03 FB 000C4        CALLS   #3, DBG$NMATCH                              ;
                    10         50 E8 000C7        BLBS    R0, 8$                                      ;
                             021E 31 000CA        BRW     40$                                         ; 0396
                               04 DD 000CD 7$:    PUSHL   #4                                          ; 0405
                    68         01 FB 000CF        CALLS   #1, DBG$GET_TEMPMEM                         ;
                    54         50 D0 000D2        MOVL    R0, NOUN_NODE                               ;
                    65         54 D0 000D5        MOVL    NOUN_NODE, (LINK)                           ; 0406
                               AE 11 000D8        BRB     3$                                          ; 0374
                               65 D4 000DA 8$:    CLRL    (LINK)                                      ; 0411
                               56 11 000DC        BRB     12$                                         ; 0347
                               03 DD 000DE 9$:    PUSHL   #3                                          ; 0418
```

```
                              B5   A6   9F 000E0          PUSHAB  DBG$CS_DISPLAY                        ;
                                   52   DD 000E3          PUSHL   R2                                    ;
                   67              03   FB 000E5          CALLS   #3, DBG$NMATCH                        ;
                   01              50   D1 000E8          CMPL    R0, #1                                ;
                                   0F   12 000EB          BNEQ    10$                                   ;
           01      A3              12   90 000ED          MOVB    #18, 1(R3)                            ; 0420
                                   0C   BB 000F1          PUSHR   #^M<R2,R3>                            ; 0421
   00000000G       00              02   FB 000F3          CALLS   #2, DBG$SCR_PARSE_CANDISP_CMD         ;
                                   66   11 000FA          BRB     15$                                   ; 0347
                                   01   DD 000FC  10$:    PUSHL   #1                                    ; 0427
                              BD   A6   9F 000FE          PUSHAB  DBG$CS_EXCEPTION                      ;
                                   52   DD 00101          PUSHL   R2                                    ;
                   67              03   FB 00103          CALLS   #3, DBG$NMATCH                        ;
                   01              50   D1 00106          CMPL    R0, #1                                ;
                                   16   12 00109          BNEQ    11$                                   ;
                                   01   DD 0010B          PUSHL   #1                                    ; 0432
                              A5   A6   9F 0010D          PUSHAB  DBG$CS_BREAK                          ;
                                   52   DD 00110          PUSHL   R2                                    ;
                   67              03   FB 00112          CALLS   #3, DBG$NMATCH                        ;
                   43              50   E9 00115          BLBC    R0, 14$                               ;
           01      A3              04   90 00118          MOVB    #4, 1(R3)                             ; 0446
                              04   A3   7C 0011C          CLRQ    4(R3)                                 ; 0452
                                   76   11 0011F          BRB     19$                                   ; 0347
                                   01   DD 00121  11$:    PUSHL   #1                                    ; 0455
                              C7   A6   9F 00123          PUSHAB  DBG$CS_MODE                           ;
                                   52   DD 00126          PUSHL   R2                                    ;
                   67              03   FB 00128          CALLS   #3, DBG$NMATCH                        ;
                   01              50   D1 0012B          CMPL    R0, #1                                ;
                                   06   12 0012E          BNEQ    13$                                   ;
           01      A3              05   90 00130          MOVB    #5, 1(R3)                             ; 0457
                                   61   11 00134  12$:    BRB     19$                                   ; 0347
                                   04   DD 00136  13$:    PUSHL   #4                                    ; 0460
                              CC   A6   9F 00138          PUSHAB  DBG$CS_MODULE                         ;
                                   52   DD 0013B          PUSHL   R2                                    ;
                   67              03   FB 0013D          CALLS   #3, DBG$NMATCH                        ;
                   01              50   D1 00140          CMPL    R0, #1                                ;
                                   54   12 00143          BNEQ    20$                                   ;
                                   01   DD 00145          PUSHL   #1                                    ; 0465
                            0044   8F   BB 00147          PUSHR   #^M<R2,R6>                            ;
                   67              03   FB 0014B          CALLS   #3, DBG$NMATCH                        ;
                   13              50   E9 0014E          BLBC    R0, 16$                               ;
                                   01   DD 00151          PUSHL   #1                                    ; 0471
                              06   A6   9F 00153          PUSHAB  DBG$CS_ALL                            ;
                                   52   DD 00156          PUSHL   R2                                    ;
                   67              03   FB 00158          CALLS   #3, DBG$NMATCH                        ;
                   64              50   E9 0015B  14$:    BLBC    R0, 21$                               ;
           01      A3              07   90 0015E          MOVB    #7, 1(R3)                             ; 0483
                                   7D   11 00162  15$:    BRB     24$                                   ; 0465
                              0C   AC   DD 00164  16$:    PUSHL   MESSAGE_VECT                          ; 0504
                                   14   BB 00167          PUSHR   #^M<R2,R4>                            ; 0503
                   69              03   FB 00169          CALLS   #3, DBG$NSAVE_STRING                  ;
                   03              50   E8 0016C          BLBS    R0, 17$                               ;
                                 018F   31 0016F          BRW     42$                                   ;
                                   01   DD 00172  17$:    PUSHL   #1                                    ; 0511
                              0A   A6   9F 00174          PUSHAB  DBG$CS_COMMA                          ;
                                   52   DD 00177          PUSHL   R2                                    ;
                   67              03   FB 00179          CALLS   #3, DBG$NMATCH                        ;
```

```
              11              50  E9 0017C          BLBC    R0, 18$                              ;
              55        08    A4  9E 0017F          MOVAB   8(R4), LINK                          ; 0518
                              04  DD 00183          PUSHL   #4                                   ; 0519
              68              01  FB 00185          CALLS   #1, DBG$GET_TEMPMEM                  ;
              54              50  D0 00188          MOVL    R0, NOUN_NODE                        ;
              65              54  D0 0018B          MOVL    NOUN_NODE, (LINK)                    ; 0520
                              D4  11 0018E          BRB     16$                                  ; 0498
                        08    A4  D4 00190 18$:     CLRL    8(NOUN_NODE)                         ; 0527
        01    A3              06  90 00193          MOVB    #6, 1(R3)                            ; 0529
                              48  11 00197 19$:     BRB     24$                                  ; 0347
                              01  DD 00199 20$:     PUSHL   #1                                   ; 0535
                        D3    A6  9F 0019B          PUSHAB  DBG$CS_RADIX                         ;
                              52  DD 0019E          PUSHL   R2                                   ;
              67              03  FB 001A0          CALLS   #3, DBG$NMATCH                       ;
              01              50  D1 001A3          CMPL    R0, #1                               ;
                              26  12 001A6          BNEQ    23$                                  ;
        01    A3              14  90 001A8          MOVB    #20, 1(R3)                           ; 0540
                              01  DD 001AC          PUSHL   #1                                   ; 0544
                      0044    8F  BB 001AE          PUSHR   #^M<R2,R6>                           ;
              67              03  FB 001B2          CALLS   #3, DBG$NMATCH                       ;
              29              50  E9 001B5          BLBC    R0, 24$                              ;
                              01  DD 001B8          PUSHL   #1                                   ; 0550
                        0C    A6  9F 001BA          PUSHAB  DBG$CS_OVERRIDE                      ;
                              52  DD 001BD          PUSHL   R2                                   ;
              67              03  FB 001BF          CALLS   #3, DBG$NMATCH                       ;
              03              50  E8 001C2 21$:     BLBS    R0, 22$                              ;
                            010B  31 001C5          BRW     38$                                  ;
        01    A3              15  90 001C8 22$:     MOVB    #21, 1(R3)                           ; 0562
                              13  11 001CC          BRB     24$                                  ; 0347
                              01  DD 001CE 23$:     PUSHL   #1                                   ; 0566
                        D9    A6  9F 001D0          PUSHAB  DBG$CS_SCOPE                         ;
                              52  DD 001D3          PUSHL   R2                                   ;
              67              03  FB 001D5          CALLS   #3, DBG$NMATCH                       ;
              01              50  D1 001D8          CMPL    R0, #1                               ;
                              06  12 001DB          BNEQ    25$                                  ;
        01    A3              08  90 001DD          MOVB    #8, 1(R3)                            ; 0568
                              6D  11 001E1 24$:     BRB     30$                                  ; 0347
                              02  DD 001E3 25$:     PUSHL   #2                                   ; 0571
                        DF    A6  9F 001E5          PUSHAB  DBG$CS_SOURCE                        ;
                              52  DD 001E8          PUSHL   R2                                   ;
              67              03  FB 001EA          CALLS   #3, DBG$NMATCH                       ;
              01              50  D1 001ED          CMPL    R0, #1                               ;
                              60  12 001F0          BNEQ    31$                                  ;
        01    A3              10  90 001F2          MOVB    #16, 1(R3)                           ; 0573
                              01  DD 001F6          PUSHL   #1                                   ; 0577
                      0044    8F  BB 001F8          PUSHR   #^M<R2,R6>                           ;
              67              03  FB 001FC          CALLS   #3, DBG$NMATCH                       ;
              4C              50  E9 001FF          BLBC    R0, 29$                              ;
                              04  DD 00202          PUSHL   #4                                   ; 0587
                        15    A6  9F 00204          PUSHAB  DBG$CS_MODULE                        ;
                              52  DD 00207          PUSHL   R2                                   ;
              67              03  FB 00209          CALLS   #3, DBG$NMATCH                       ;
              0A              50  E9 0020C          BLBC    R0, 26$                              ;
                              01  DD 0020F          PUSHL   #1                                   ; 0597
                        FE    A6  9F 00211          PUSHAB  DBG$CS_EQUAL                         ;
                              52  DD 00214          PUSHL   R2                                   ;
              67              03  FB 00216          CALLS   #3, DBG$NMATCH                       ;
```

```
                         03            50  E8 00219 26$:    BLBS    R0, 27$                              ;
                                     00CC  31 0021C         BRW     40$                                  ;
                                0C     AC  DD 0021F 27$:    PUSHL   MESSAGE_VECT                         ; 0608
                                04     AE  9F 00222         PUSHAB  MODNAMEPTR                           ; 0607
                                       52  DD 00225         PUSHL   R2                                   ;
                         69            03  FB 00227         CALLS   #3, DBG$NSAVE_STRING                 ;
                         03            50  E8 0022A         BLBS    R0, 28$                              ;
                                     00D1  31 0022D         BRW     42$                                  ;
                                       6E  DD 00230 28$:    PUSHL   MODNAMEPTR                           ; 0615
               00000000G 00            01  FB 00232         CALLS   #1, DBG$STA_GETSOURCEMOD             ;
                         64            50  D0 00239         MOVL    R0, (NOUN_NODE)                      ;
                                       55  12 0023C         BNEQ    33$                                  ; 0620
                                       6E  DD 0023E         PUSHL   MODNAMEPTR                           ; 0624
                                       01  DD 00240         PUSHL   #1                                   ; 0623
                           000281E8    8F  DD 00242         PUSHL   #164328                              ;
                         6A            03  FB 00248         CALLS   #3, DBG$NMAKE_ARG_VECT               ;
                                     00AF  31 0024B         BRW     41$                                  ;
                                       64  D4 0024E 29$:    CLRL    (NOUN_NODE)                          ; 0632
                                       7F  11 00250 30$:    BRB     37$                                  ; 0347
                                       01  DD 00252 31$:    PUSHL   #1                                   ; 0636
                                E6     A6  9F 00254         PUSHAB  DBG$CS_TRACE                         ;
                                       52  DD 00257         PUSHL   R2                                   ;
                         67            03  FB 00259         CALLS   #3, DBG$NMATCH                       ;
                         01            50  D1 0025C         CMPL    R0, #1                               ;
                                       06  12 0025F         BNEQ    32$                                  ;
                      01 A3            09  90 00261         MOVB    #9, 1(R3)                            ; 0638
                                       41  11 00265         BRB     35$                                  ; 0641
                                       02  DD 00267 32$:    PUSHL   #2                                   ; 0645
                                EC     A6  9F 00269         PUSHAB  DBG$CS_TYPE                          ;
                                       52  DD 0026C         PUSHL   R2                                   ;
                         67            03  FB 0026E         CALLS   #3, DBG$NMATCH                       ;
                         01            50  D1 00271         CMPL    R0, #1                               ;
                                       1F  12 00274         BNEQ    34$                                  ;
                                       01  DD 00276         PUSHL   #1                                   ; 0652
                              0044     8F  BB 00278         PUSHR   #^M<R2,R6>                           ;
                         67            03  FB 0027C         CALLS   #3, DBG$NMATCH                       ;
                         51            50  E9 0027F         BLBC    R0, 38$                              ;
                                       01  DD 00282         PUSHL   #1                                   ; 0667
                                1C     A6  9F 00284         PUSHAB  DBG$CS_OVERRIDE                      ;
                                       52  DD 00287         PUSHL   R2                                   ;
                         67            03  FB 00289         CALLS   #3, DBG$NMATCH                       ;
                         44            50  E9 0028C         BLBC    R0, 38$                              ;
                      01 A3            0D  90 0028F         MOVB    #13, 1(R3)                           ; 0679
                                       70  11 00293 33$:    BRB     43$                                  ; 0347
                                       01  DD 00295 34$:    PUSHL   #1                                   ; 0682
                                F1     A6  9F 00297         PUSHAB  DBG$CS_WATCH                         ;
                                       52  DD 0029A         PUSHL   R2                                   ;
                         67            03  FB 0029C         CALLS   #3, DBG$NMATCH                       ;
                         01            50  D1 0029F         CMPL    R0, #1                               ;
                                       11  12 002A2         BNEQ    36$                                  ;
                      01 A3            0E  90 002A4         MOVB    #14, 1(R3)                           ; 0684
                                0C     AC  DD 002A8 35$:    PUSHL   MESSAGE_VECT                         ; 0687
                                       0C  BB 002AB         PUSHR   #^M<R2,R3>                           ; 0685
               00000000G 00            03  FB 002AD         CALLS   #3, DBG$EVENT_SHOW_CANCEL_SYNTAX     ;
                                       04  002B4            RET                                          ;
                                       03  DD 002B5 36$:    PUSHL   #3                                   ; 0693
                                F7     A6  9F 002B7         PUSHAB  DBG$CS_WINDOW                        ;
```

```
                                          52  DD 002BA          PUSHL   R2                                        ;
                          67              03  FB 002BC          CALLS   #3, DBG$NMATCH                            ;
                          01              50  D1 002BF          CMPL    R0, #1                                    ;
                                          0F  12 002C2          BNEQ    38$                                       ;
                     01   A3              13  90 002C4          MOVB    #19, 1(R3)                                ; 0695
                                          0C  BB 002C8          PUSHR   #^M<R2,R3>                                ; 0696
              00000000G   00              02  FB 002CA          CALLS   #2, DBG$SCR_PARSE_CANWIND_CMD             ;
                                          32  11 002D1 37$:     BRB     43$                                       ; 0347
                                          01  DD 002D3 38$:     PUSHL   #1                                        ; 0706
                                  04      A6  9F 002D5          PUSHAB  DBG$CS_CR                                 ;
                                          52  DD 002D8          PUSHL   R2                                        ;
                          67              03  FB 002DA          CALLS   #3, DBG$NMATCH                            ;
                          0B              50  E9 002DD          BLBC    R0, 40$                                   ;
                              000280D0    8F  DD 002E0          PUSHL   #164048                                   ; 0708
                          6A              01  FB 002E6 39$:     CALLS   #1, DBG$NMAKE_ARG_VECT                    ;
                                          12  11 002E9          BRB     41$                                       ;
                                          52  DD 002EB 40$:     PUSHL   R2                                        ; 0710
              00000000G   00              01  FB 002ED          CALLS   #1, DBG$NNEXT_WORD                        ;
                                          50  DD 002F4          PUSHL   R0                                        ;
              00000000G   00              01  FB 002F6          CALLS   #1, DBG$NSYNTAX_ERROR                     ;
                     0C   BC              50  D0 002FD 41$:     MOVL    R0, @MESSAGE_VECT                         ; 0705
                          50              04  D0 00301 42$:     MOVL    #4, R0                                    ; 0712
                                              04 00304          RET                                               ;
                          50              01  D0 00305 43$:     MOVL    #1, R0                                    ; 0717
                                              04 00308          RET                                               ; 0719

; Routine Size:  777 bytes,    Routine Base:  DBG$CODE + 0000
```

```
1 GLOBAL ROUTINE DBG$NEXECUTE_CANCEL (VERB_NODE, MESSAGE_VECT) =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine uses the command execution tree constructed by the parse
1 !       network as input and performs the semantic actions associated with
1 !       the given input corresponding to the CANCEL xxx command. If the command
1 !       cannot be executed, a message argument vector is constructed and returned.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       VERB_NODE       - A longword containing the address of the head node
1 !                         of the command execution tree. This corresponds to
1 !                         the veb node.
1 !
1 !       MESSAGE_VECT    - The address of a longword to contain the address of
1 !                         a standard message argument vector upon detection of
1 !                         errors.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       The linked list command execution tree pointed to by verb_node.
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       On failure, a message argument vector is constructed and returned.
1 !
1 ! ROUTINE VALUE:
1 !
1 !       An unsigned integer longword completion code
1 !
1 ! COMPLETION CODES:
1 !
1 !       STS$K_SUCCESS (1)       - Success. Command executed.
1 !
1 !       STS$K_SEVERE  (4)       - Failure. Command not executed. Message argument
1 !                                 vector constructed and returned.
1 !
1 ! SIDE EFFECTS:
1 !
1 !       Various semantic actions corresponding to the CANCEL xxx command are
1 !       performed.
1 !
1 !--
2     BEGIN
2
2     MAP
2       VERB_NODE: REF DBG$VERB_NODE;   ! Pointer to command Verb Node
2
2     LOCAL
2       NOUN_NODE: REF DBG$NOUN_NODE,   ! Pointer to a command Noun Node
2       ADDR_EXP_DESC,                  ! Address expression descriptor
2       ADDRESS: VECTOR [2],            ! Address and bit offset
2       TYPE;                           ! Type of AED described object
2
2
```

```

    ! Recover the noun node
    !
    NOUN_NODE = .VERB_NODE [DBG$L_VERB_OBJECT_PTR];


    ! Perform the indicated action base on the verb composite
    !
    CASE .VERB_NODE[DBG$B_VERB_COMPOSITE] FROM CANCEL_MINIMUM TO CANCEL_MAXIMUM OF
        SET


        ! Execute the CANCEL ALL command.
        !
        [CANCEL_ALL]:
            BEGIN
            LOCAL
                SCOPE_LIST,
                DUMMY;

            ! Just cancel everything in sight
            !
            scope_list = 0;
            dbg$gl_context [dbg$k_all] = true;

            DBG$EVENT_CANCEL_ALL ();

            dbg$runframe [dbg$v_trace_all] = false;     ! For next two calls
            dbg$gb_resignal = true;                     ! Exception break
            dbg$set_mod_def ();                         ! Set mode defaults
            dbg$set_stp_def ();                         ! Set step defaults
            dbg$rst_setscope (scope_list, dummy);       ! Scopes (new debugger)
            dbg$gl_gbltyp = -1;                         ! Override type
            dbg$gw_gbllngth = 0;                        ! Override length
            dbg$gl_dflttyp = dsc$k_dtype_l;             ! Default type
            dbg$gw_dfltleng = 4;                        ! Default length
            END;

        [cancel_break] :          ! CANCEL BREAK <ADDR_EXP>
            RETURN DBG$EVENT_SHOW_CANCEL_SEMANTICS (.VERB_NODE,
                                                    .MESSAGE_VECT
                                                   );


        ! Execute the CANCEL DEVELOPER 0, 1, ..., n command.  Cancel all bits
        ! in DBG$GL_DEVELOPER indicated on the command.  If no bits are speci-
        ! fied, clear all developer bits.
        !
        [CANCEL_DEVELOPER]:
            BEGIN
            NOUN_NODE = .VERB_NODE[DBG$L_VERB_OBJECT_PTR];
            IF .NOUN_NODE EQL 0 THEN DBG$GL_DEVELOPER = 0;
            WHILE .NOUN_NODE NEQ 0 DO
                BEGIN
                DBG$GL_DEVELOPER[.NOUN_NODE[DBG$L_NOUN_VALUE]] = FALSE;
                NOUN_NODE = .NOUN_NODE[DBG$L_NOUN_LINK];
                END;

    END;


! Execute the CANCEL DISPLAY command.
!
[CANCEL_DISPLAY]:
    DBG$SCR_EXECUTE_CANDISP_CMD(.VERB_NODE);


! Execute the CANCEL EXCEPTION BREAK command.
!
[CANCEL_EXCEPTION_BREAK]:
    BEGIN
    DBG$GB_RESIGNAL = TRUE;
    RETURN DBG$EVENT_SHOW_CANCEL_SEMANTICS( .VERB_NODE,.MESSAGE_VECT);
    END;


! Execute the CANCEL MODE command.
!
[CANCEL_MODE]:
    BEGIN
    dbg$gb_radix[dbg$b_radix_input] = dbg$nget_trans_radix(dbg$k_default);
    dbg$gb_radix[dbg$b_radix_output] = dbg$nget_trans_radix(dbg$k_default);
    dbg$gb_radix[dbg$b_radix_output_over] = dbg$k_default;
    DBG$SET_MOD_DEF();
    END;

[cancel_module] :       ! CANCEL MODULE or CANCEL MODULE/ALL
    BEGIN

    ! Module names are stored away as counted strings
    !
    LOCAL
        NAME_BUFF : REF VECTOR [,BYTE]; ! Module name buffer

    WHILE .noun_node NEQA 0
    DO
        BEGIN

        ! Retrieve the name buffer and call the symbol table
        !
        name_buff = .noun_node [dbg$l_noun_value];
        IF NOT dbg$rst_canmod (name_buff [1], .name_buff [0])
        THEN
            BEGIN
            .message_vect = dbg$nmake_arg_vect (dbg$_nosuchmodu,
                                                1,
                                                name_buff [0]);
            RETURN sts$k_severe;
            END;


        ! Obtain the next noun node
        !
        noun_node = .noun_node [dbg$l_noun_link];

        END;    . End of Loop

    END;

[cancel_module_all] :
    BEGIN
    dbg$rst_canmod (0, 0);
    END;

[cancel_radix] :
    BEGIN
    dbg$gb_radix[dbg$b_radix_input] = dbg$nget_trans_radix(dbg$k_default);
    dbg$gb_radix[dbg$b_radix_output] = dbg$nget_trans_radix(dbg$k_default);
    dbg$gb_radix[dbg$b_radix_output_over] = dbg$k_default;
    END;

[cancel_radix_override]:
    dbg$gb_radix[dbg$b_radix_output_over] = dbg$k_default;

[cancel_scope] :
    BEGIN
    LOCAL
        DUMMY,
        SCOPE_LIST;

    scope_list = 0;
    dbg$rst_setscope (scope_list, dummy);
    END;

[cancel_source] :       ! CANCEL SOURCE[/MODULE=modname]
    BEGIN
    dbg$src_cancel_source(.noun_node[dbg$l_noun_value]);
    END;

[cancel_trace] :        ! CANCEL TRACE <ADDR_EXP>
    RETURN DBG$EVENT_SHOW_CANCEL_SEMANTICS (.VERB_NODE,
                                            .MESSAGE_VECT
                                           );

! Execute the CANCEL TYPE/OVERRIDE command.
!
[CANCEL_TYPE_OVERRIDE]:
    BEGIN
    DBG$GL_GBLTYP = -1;
    DBG$GW_GBLLNGTH = 0;
    END;


! Execute the CANCEL WATCH <addr-expr> command.
!
[CANCEL_WATCH]:
    RETURN DBG$EVENT_SHOW_CANCEL_SEMANTICS (.VERB_NODE,
                                            .MESSAGE_VECT
                                           );
! Execute the CANCEL WINDOW command.
!
```

```
:  818          0948  2          [CANCEL_WINDOW]:
:  819          0949  2              DBG$SCR_EXECUTE_CANWIND_CMD(.VERB_NODE);
:  820          0950  2
:  821          0951  2
:  822          0952  2          ! Any other CASE index constitutes and internal DEBUG error.
:  823          0953  2          !
:  824          0954  2          [INRANGE,OUTRANGE]:
:  825          0955  2              $DBG_ERROR('DBGNCANCL\NEXECUTE_CANCEL');
:  826          0956  2
:  827          0957  2          TES;
:  828          0958  2
:  829          0959  2      RETURN STS$K_SUCCESS;
:  830          0960  2
:  831          0961  1      END;


                                                                             .PSECT  DBG$PLIT,NOWRT,  SHR,  PIC,0

45  58  45  4E  5C  4C  43  4E  41  43  4E  47  42  44  19  00084 P.AAX:  .ASCII  <25>\DBGNCANCL\<92>\NEXECUTE_CANCEL\          ;
                4C  45  43  4E  41  43  5F  45  54  55  43  00093                                                               ;


                                                                             .PSECT  DBG$CODE,NOWRT,  SHR,  PIC,0

                                                  0FFC 00000          .ENTRY  DBG$NEXECUTE_CANCEL, Save R2,R3,R4,R5,R6,-  ; 0720
                                                                              R7,R8,R9,R10,R11                           ;
                                 5B 00000000G  00  9E 00002          MOVAB   DBG$GW_GBLLNGTH, R11                       ;
                                 5A 00000000G  00  9E 00009          MOVAB   DBG$GL_GBLTYP, R10                         ;
                                 59 00000000G  00  9E 00010          MOVAB   DBG$RST_SETSCOPE, R9                       ;
                                 58 00000000G  00  9E 00017          MOVAB   DBG$SET_MOD_DEF, R8                        ;
                                 57 00000000G  00  9E 0001E          MOVAB   DBG$GB_RESIGNAL, R7                        ;
                                 56 00000000G  00  9E 00025          MOVAB   DBG$NGET_TRANS_RADIX, R6                   ;
                                 55 00000000G  00  9E 0002C          MOVAB   DBG$GB_RADIX, R5                           ;
                                 5E            18  C2 00033          SUBL2   #24, SP                                    ;
                                 52        04  AC  D0 00036          MOVL    VERB_NODE, R2                              ; 0780
                                 53        08  A2  D0 0003A          MOVL    8(R2), NOUN_NODE                           ;
                   14            01        01  A2  8F 0003E          CASEB   1(R2), #1, #20                             ; 0785
       00AC      002A          0144          0041     00043 1$:      .WORD   3$-1$,-                                    ;
       0124      0102          00CC          00B2     0004B                  23$-1$,-                                   ;
       002A      002A          002A          0144     00053                  2$-1$,-                                    ;
       0132      002A          0144          013D     0005B                  9$-1$,-                                    ;
       010D      0151          00A1          0083     00063                  10$-1$,-                                   ;
                                             011E     0006B                  12$-1$,-                                   ;
                                                                             14$-1$,-                                   ;
                                                                             19$-1$,-                                   ;
                                                                             23$-1$,-                                   ;
                                                                             2$-1$,-                                    ;
                                                                             2$-1$,-                                    ;
                                                                             2$-1$,-                                    ;
                                                                             21$-1$,-                                   ;
                                                                             23$-1$,-                                   ;
                                                                             2$-1$,-                                    ;
                                                                             20$-1$,-                                   ;
                                                                             4$-1$,-                                    ;
                                                                             7$-1$,-                                    ;
```

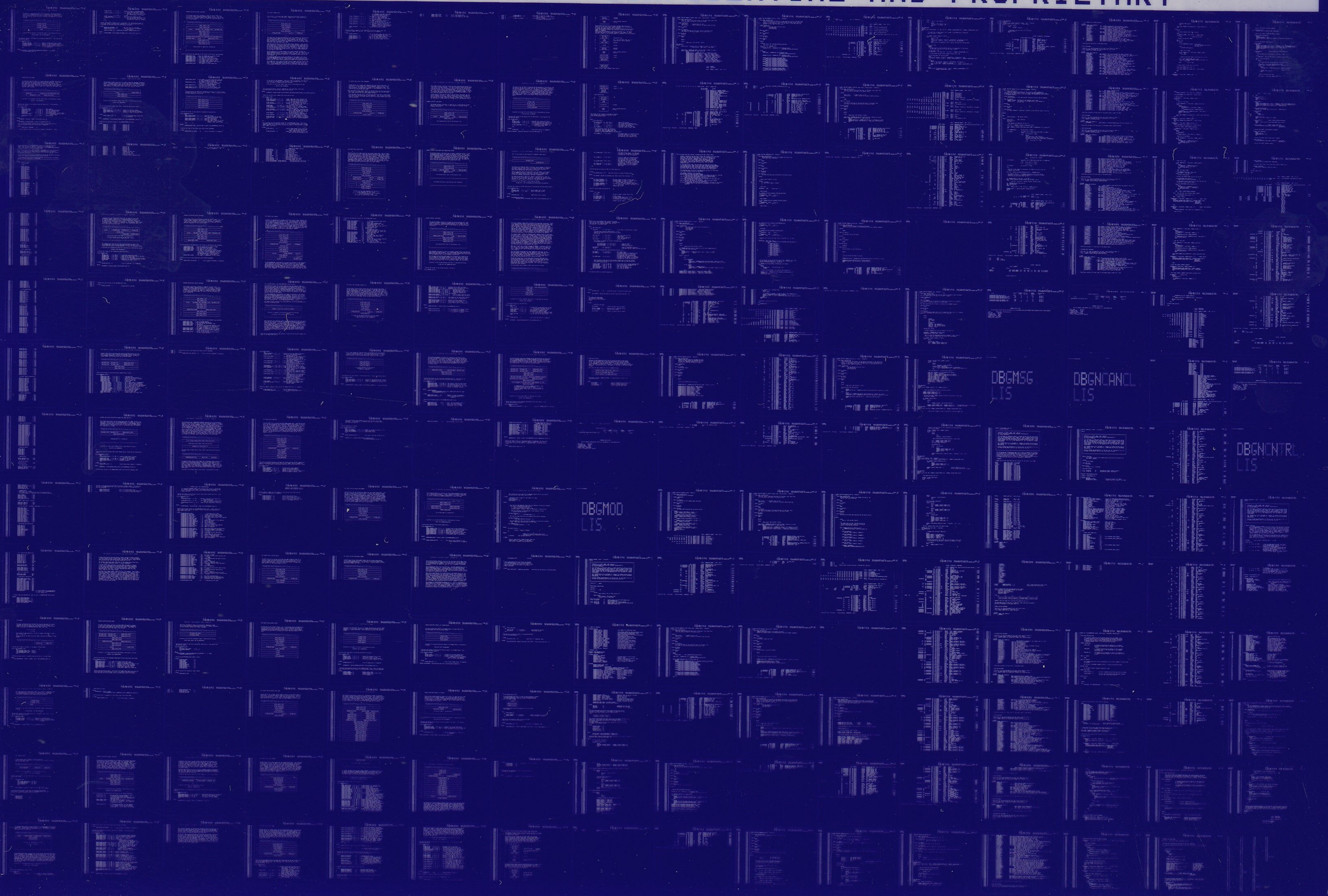
```
                                                                 24$-1$,-                                      ;
                                                                 16$-1$,-                                      ;
                                                                 17$-1$                                        ;
                       00000000'  EF  9F 0006D 2$:     PUSHAB    P.AAX                                         ; 0955
                                  01  DD 00073         PUSHL     #1                                            ;
                       00028362   8F  DD 00075         PUSHL     #164706                                       ;
           00000000G  00          03  FB 0007B         CALLS     #3, LIB$SIGNAL                                ;
                                  69  11 00082         BRB       8$                                            ;
                             04   AE  D4 00084 3$:     CLRL      SCOPE_LIST                                    ; 0799
           00000000G  00          02  88 00087         BISB2     #2, DBG$GL_CONTEXT+1                          ; 0800
           00000000G  00          00  FB 0008E         CALLS     #0, DBG$EVENT_CANCEL_ALL                      ; 0802
           00000000G  00          04  8A 00095         BICB2     #4, DBG$RUNFRAME+72                           ; 0804
                      67          01  90 0009C         MOVB      #1, DBG$GB_RESIGNAL                           ; 0805
                      68          00  FB 0009F         CALLS     #0, DBG$SET_MOD_DEF                           ; 0806
           00000000G  00          00  FB 000A2         CALLS     #0, DBG$SET_STP_DEF                           ; 0807
                                  5E  DD 000A9         PUSHL     SP                                            ; 0808
                             08   AE  9F 000AB         PUSHAB    SCOPE_LIST                                    ;
                      69          02  FB 000AE         CALLS     #2, DBG$RST_SETSCOPE                          ;
                      6A          01  CE 000B1         MNEGL     #1, DBG$GL_GBLTYP                             ; 0809
                                  6B  B4 000B4         CLRW      DBG$GW_GBLLNGTH                               ; 0810
           00000000G  00          08  D0 000B6         MOVL      #8, DBG$GL_DFLTTYP                            ; 0811
           00000000G  00          04  B0 000BD         MOVW      #4, DBG$GW_DFLTLENG                           ; 0812
                                  47  11 000C4         BRB       11$                                           ; 0785
                      53     08   A2  D0 000C6 4$:     MOVL      8(R2), NOUN_NODE                              ; 0827
                                  06  12 000CA         BNEQ      5$                                            ; 0828
                       00000000G  00  D4 000CC         CLRL      DBG$GL_DEVELOPER                              ;
                                  53  D5 000D2 5$:     TSTL      NOUN_NODE                                     ; 0829
                                  78  13 000D4         BEQL      15$                                           ;
        00 00000000G  00          63  E5 000D6         BBCC      (NOUN_NODE), DBG$GL_DEVELOPER, 6$             ; 0831
                      53     08   A3  D0 000DE 6$:     MOVL      8(NOUN_NODE), NOUN_NODE                       ; 0832
                                  EE  11 000E2         BRB       5$                                            ; 0829
                                  52  DD 000E4 7$:     PUSHL     R2                                            ; 0841
           00000000G  00          01  FB 000E6         CALLS     #1, DBG$SCR_EXECUTE_CANDISP_CMD               ;
                                  76  11 000ED 8$:     BRB       18$                                           ;
                      67          01  90 000EF 9$:     MOVB      #1, DBG$GB_RESIGNAL                           ; 0848
                                0092  31 000F2         BRW       23$                                           ; 0849
                                  01  DD 000F5 10$:    PUSHL     #1                                            ; 0857
                      66          01  FB 000F7         CALLS     #1, DBG$NGET_TRANS_RADIX                      ;
                      65          50  90 000FA         MOVB      R0, DBG$GB_RADIX                              ;
                                  01  DD 000FD         PUSHL     #1                                            ; 0858
                      66          01  FB 000FF         CALLS     #1, DBG$NGET_TRANS_RADIX                      ;
                 01   A5          50  90 00102         MOVB      R0, DBG$GB_RADIX+1                            ;
                 02   A5          01  90 00106         MOVB      #1, DBG$GB_RADIX+2                            ; 0859
                      68          00  FB 0010A         CALLS     #0, DBG$SET_MOD_DEF                           ; 0860
                                  76  11 0010D 11$:    BRB       22$                                           ; 0785
                                  53  D5 0010F 12$.    TSTL      NOUN_NODE                                     ; 0871
                                  72  13 00111         BEQL      22$                                           ;
                      54          63  D0 00113         MOVL      (NOUN_NODE), NAME_BUFF                        ; 0877
                      7E          64  9A 00116         MOVZBL    (NAME_BUFF), -(SP)                            ; 0878
                             01   A4  9F 00119         PUSHAB    1(NAME_BUFF)                                  ;
           00000000G  00          02  FB 0011C         CALLS     #2, DBG$RST_CANMOD                            ;
                      19          50  E8 00123         BLBS      R0, 13$                                       ;
                                  54  DD 00126         PUSHL     NAME_BUFF                                     ; 0883
                                  01  DD 00128         PUSHL     #1                                            ;
                       000281E8   8F  DD 0012A         PUSHL     #164328                                       ;
           00000000G  00          03  FB 00130         CALLS     #3, DBG$NMAKE_ARG_VECT                        ;
                 08   BC          50  D0 00137         MOVL      R0, @MESSAGE_VECT                             ;
```

```
                        50        04  D0 0013B        MOVL    #4, R0                                   ; 0884
                                      04 0013E        RET                                              ;
                        53    08  A3  D0 0013F 13$:   MOVL    8(NOUN_NODE), NOUN_NODE                  ; 0890
                                  CA  11 00143        BRB     12$                                      ; 0871
                                  7E  7C 00145 14$:   CLRQ    -(SP)                                    ; 0898
              00000000G  00       02  FB 00147        CALLS   #2, DBG$RST_CANMOD                       ;
                                  4D  11 0014E 15$:   BRB     25$                                      ; 0785
                                  01  DD 00150 16$:   PUSHL   #1                                       ; 0903
                        66        01  FB 00152        CALLS   #1, DBG$NGET_TRANS_RADIX                 ;
                        65        50  90 00155        MOVB    R0, DBG$GB_RADIX                         ;
                                  01  DD 00158        PUSHL   #1                                       ; 0904
                        66        01  FB 0015A        CALLS   #1, DBG$NGET_TRANS_RADIX                 ;
                   01   A5        50  90 0015D        MOVB    R0, DBG$GB_RADIX+1                       ;
                   02   A5        01  90 00161 17$:   MOVB    #1, DBG$GB_RADIX+2                       ; 0909
                                  36  11 00165 18$:   BRB     25$                                      ;
                              0C  AE  D4 00167 19$:   CLRL    SCOPE_LIST                               ; 0917
                              08  AE  9F 0016A        PUSHAB  DUMMY                                    ; 0918
                              10  AE  9F 0016D        PUSHAB  SCOPE_LIST                               ;
                        69        02  FB 00170        CALLS   #2, DBG$RST_SETSCOPE                     ;
                                  28  11 00173        BRB     25$                                      ; 0785
                                  63  DD 00175 20$:   PUSHL   (NOUN_NODE)                              ; 0923
              00000000G  00       01  FB 00177        CALLS   #1, DBG$SRC_CANCEL_SOURCE                ;
                                  1D  11 0017E        BRB     25$                                      ; 0785
                        6A        01  CE 00180 21$:   MNEGL   #1, DBG$GL_GBLTYP                        ; 0935
                                  6B  B4 00183        CLRW    DBG$GW_GBLLNGTH                          ; 0936
                                  16  11 00185 22$:   BRB     25$                                      ; 0785
                              08  AC  DD 00187 23$:   PUSHL   MESSAGE_VECT                             ; 0944
                                  52  DD 0018A        PUSHL   R2                                       ; 0943
              00000000G  00       02  FB 0018C        CALLS   #2, DBG$EVENT_SHOW_CANCEL_SEMANTICS      ;
                                      04 00193        RET                                              ;
                                  52  DD 00194 24$:   PUSHL   R2                                       ; 0949
              00000000G  00       01  FB 00196        CALLS   #1, DBG$SCR_EXECUTE_CANWIND_CMD          ;
                        50        01  D0 0019D 25$:   MOVL    #1, R0                                   ; 0959
                                      04 001A0        RET                                              ; 0961

; Routine Size:  417 bytes,    Routine Base:  DBG$CODE + 0309


;  832          0962  1
;  833          0963  0 END ELUDOM




                                                                   .EXTRN  LIB$SIGNAL


;                                  PSECT SUMMARY
;
;       Name                     Bytes                        Attributes
;
;  DBG$PLIT                        158  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)
;  DBG$CODE                       1194  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(0)



;                              Library Statistics
```

```
;
;                                          -------- Symbols --------        Pages      Processing
;      File                                Total    Loaded   Percent        Mapped     Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619        3         0         1000        00:01.8
;  _$255$DUA28:[DEBUG.OBJ]STRUCDEF.L32;1      32        0         0            7        00:00.1
;  _$255$DUA28:[DEBUG.OBJ]DBGLIB.L32;1      1545       28         1           97        00:02.0
;  _$255$DUA28:[DEBUG.OBJ]DSTRECRDS.L32;1
;                                            418        0         0           31        00:00.3
;  _$255$DUA28:[DEBUG.OBJ]DBGMSG.L32;1       386        6         1           22        00:00.3
;  _$255$DUA28:[DEBUG.OBJ]DBGGEN.L32;1       150        0         0           12        00:00.3



;                                 COMMAND QUALIFIERS

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

; Size.          1194 code + 158 data bytes
; Run Time:         00:27.3
; Elapsed Time:     01:28.7
; Lines/CPU Min:     2119
; Lexemes/CPU-Min:  9625
; Memory Used:  320 pages
; Compilation Complete
```

0086 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



DBGMSG
LIS

DBGNCANCL
LIS

DBGNCNTRL
LIS

DBGMOD
LIS