```
                                                                            M  1




BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBBBBBBBBBBB        AAAAAAAAA        SSSSSSSSSSSS  RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBB         BBB  AAA         AAA  SSS              RRR         RRR        TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA     SSSSSSSSS     RRRRRRRRRRRR           TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAAAAAAAAAAAAAA              SSS  RRR   RRR              TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBB         BBB  AAA         AAA              SSS  RRR      RRR           TTT        LLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
BBBBBBBBBBBB     AAA         AAA  SSSSSSSSSSSS     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

G 16

FILEID**BASPURIOB

```
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP    UU      UU  RRRRRRRR      IIIIII      000000    BBBBBBBB
BBBBBBBB      AAAAAA      SSSSSSSS  PPPPPPPP    UU      UU  RRRRRRRR      IIIIII      000000    BBBBBBBB
BB      BB  AA      AA  SS          PP      PP  UU      UU  RR      RR      II      00      00  BB      BB
BB      BB  AA      AA  SS          PP      PP  UU      UU  RR      RR      II      00      00  BB      BB
BB      BB  AA      AA  SS          PP      PP  UU      UU  RR      RR      II      00      00  BB      BB
BB      BB  AA      AA  SS          PP      PP  UU      UU  RR      RR      II      00      00  BB      BB
BBBBBBBB    AA      AA    SSSSSS    PPPPPPPP    UU      UU  RRRRRRRR        II      00      00  BBBBBBBB
BBBBBBBB    AA      AA    SSSSSS    PPPPPPPP    UU      UU  RRRRRRRR        II      00      00  BBBBBBBB
BB      BB  AAAAAAAAAA          SS  PP          UU      UU  RR  RR          II      00      00  BB      BB
BB      BB  AAAAAAAAAA          SS  PP          UU      UU  RR  RR          II      00      00  BB      BB
BB      BB  AA      AA          SS  PP          UU      UU  RR    RR        II      00      00  BB      BB      ....
BB      BB  AA      AA          SS  PP          UU      UU  RR    RR        II      00      00  BB      BB      ....
BBBBBBBB    AA      AA  SSSSSSSS    PP          UUUUUUUUUU  RR      RR    IIIIII      000000    BBBBBBBB        ....
BBBBBBBB    AA      AA  SSSSSSSS    PP          UUUUUUUUUU  RR      RR    IIIIII      000000    BBBBBBBB        ....

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 BAS$$PUR_IO_BUF (                               ! Purge I/O buffer for a file.
0                 IDENT = '1-007'                        ! File: BASPURIOB.B32
0                 ) =
1 BEGIN
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 !++
1 ! FACILITY:
1 !
1 ! ABSTRACT:
1 !
1 !       This module contains routines which will check the LUN indicated by R11
1 !       and print the contents of the associated I/O buffer if there is valid
1 !       data in the buffer.  These routines are intended to be called before
1 !       a file is closed (explicitly or implicitly at end of program) or if
1 !       an error occurs during an output element transmit.
1 !
1 ! ENVIRONMENT: User mode - AST reentrant
1 !
1 ! AUTHOR: Donald G. Petersen, CREATION DATE: 22-Jan-79
1 !
1 ! MODIFIED BY:
1 !
1 !       DGP, : VERSION 1-01
1 ! 1-001 - original
1 ! 1-002 - Use 32-bit addresses for externals.  JBS 27-JAN-1979
1 ! 1-003 - Change entry point names so we have two: one for CLOSE
1 !          and one for error handling.  JBS for DGP 07-MAR-1979
1 ! 1-004 - Make PUR_IO_ERR purge the terminal on unit zero.  DGP 07-Mar-79
1 ! 1-005 - Remove references to BAS$ routines.  JBS 10-MAY-1979
1 ! 1-006 - Change from an OTS routine to a BAS routine, since FORTRAN
1 !         does not need to purge I/O buffers.  JBS 20-AUG-1979
1 ! 1-007 - Don't purge virtual arrays; it is quite complex, and CLOSE
1 !         will do it.  JBS 30-AUG-1979
```

```
;  58          0058  1 !--
;  59          0059  1
;  60          0060  1 !<BLF/PAGE>
```

```
1 !
1 ! SWITCHES:
1 !
1
1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
1
1 !
1 ! LINKAGES:
1 !
1
1 REQUIRE 'RTLIN:OTSLNK';                        ! define all linkages
1
1 !
1 ! TABLE OF CONTENTS:
1 !
1
1 FORWARD ROUTINE
1     BAS$$PUR_IO_ERR : NOVALUE,                 ! Purge outstanding output buffer
1                                                ! following an error
1     BAS$$PUR_IO_CLO : CALL_CCB NOVALUE;        ! Purge outstanding output buffer
1
1                                                ! during a close
1 !
1 ! INCLUDE FILES:
1 !
1
1 REQUIRE 'RTLML:OTSLUB';                        ! I/O statement block
1
1 REQUIRE 'RTLIN:RTLPSECT';                      ! Define DECLARE_PSECTS macro
1
1 REQUIRE 'RTLIN:BASIOERR';                      ! Define I/O error symbols
1
1 LIBRARY 'RTLSTARLE';                           ! STARLET library
1
1 !
1 ! MACROS:
1 !
1 !       NONE
1 !
1 ! EQUATED SYMBOLS:
1 !
1
1 LITERAL
1     K_CR = %X'0D',                             ! ASCII CR
1     K_NULL = %X'00',                           ! ASCII NUL
1     K_LF = %X'0B';                             ! ASCII LF
1
1 !
1 ! PSECT declarations
1 !
1 DECLARE_PSECTS (BAS);                          ! Put this in BAS psect
1 !
1 ! OWN STORAGE:
1 !
1 !       NONE
1 !
1 ! EXTERNAL REFERENCES:
```

```
;  119          0831  1 !
;  120          0832  1
;  121          0833  1 EXTERNAL ROUTINE
;  122          0834  1     BAS$$CB_PUSH : JSB_CB_PUSH,                 ! Load register CCB
;  123          0835  1     BAS$$CB_POP : JSB_CB_POP NOVALUE,           ! Done with register CCB
;  124          0836  1     BAS$$NEXT_LUN : NOVALUE,                    ! Get next logical unit number
;  125          0837  1     BAS$$STOP_IO : NOVALUE;                     ! Signal fatal BASIC I/O error
;  126          0838  1
```

```
1 GLOBAL ROUTINE BAS$$PUR_IO_CLO                        ! Purge the contents of an I/O buffer
1     : CALL_CCB NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !       This routine will PUT the contents of the buffer associated with the LUN
1 !       passed to it if the buffer has valid data in it.  This routine is expected
1 !       to aid in closing a file either implicitly or explicitly if the last lan-
1 !       guage to access the file has the Basic semantics where an output record
1 !       may be continued across several output statements.  Therefore, an output
1 !       buffer may exist which has valid data in it which should be printed before
1 !       a program finishes executing.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !       NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !       LUB$V_OUTBUF_DR         Flag to indicate the output buffer has valid
1 !                               data in it
1 !       LUB$W_LUN               LUN number
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !       LUB$V_OUTBUF_DR         Flag to indicate the output buffer has valid
1 !       LUB$B_BAS_VFC2          'post' carriage control for PRN file format
1 !                               data in it
1 !
1 ! ROUTINE VALUE:
1 !
1 !       NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !       NONE
1 !
1 !--
1
2    BE  N
2
2    EXTERNAL REGISTER
2       CCB : REF BLOCK [, BYTE];
2
3    IF (.CCB [LUB$V_OUTBUF_DR] AND (.CCB [LUB$B_ORGAN] NEQ LUB$K_ORG_VIRTU))
2    THEN
3       BEGIN
3 !+
3 ! Write out the buffer.  This invloves a call to RMS.
3 !-
3       CCB [RAB$W_RSZ] = .CCB [LUB$A_BUF_PTR] - .CCB [LUB$A_RBUF_ADR];
3       CCB [RAB$L_RBF] = .CCB [LUB$A_RBUF_ADR];
3       CCB [LUB$V_OUTBUF_DR] = 0;
3
4       IF ( NOT $PUT (RAB = .CCB))
3       THEN
```

```
;   185         0896  4             BEGIN
;   186         0897  4
;   187         0898  4             WHILE (.CCB [RAB$L_STS] EQL RMS$_RSA) DO
;   188         0899  5                 BEGIN
;   189         0900  5                 $WAIT (RAB = .CCB);
;   190         0901  5                 $PUT (RAB = .CCB);
;   191         0902  4                 END;
;   192         0903  4
;   193         0904  4             IF ( NOT .CCB [RAB$L_STS]) THEN BAS$$STOP_IO (BAS$K_IOERR_REC);
;   194         0905  4
;   195         0906  3             END;
;   196         0907  3
;   197         0908  3         CCB [LUB$B_BAS_VFC1] = K_LF;
;   198         0909  3         CCB [LUB$B_BAS_VFC2] = K_NULL;
;   199         0910  3 !+
;   200         0911  3 ! Initialize the record buffer.
;   201         0912  3 !-
;   202         0913  3         CCB [LUB$A_BUF_PTR] = .CCB [LUB$A_RBUF_ADR];
;   203         0914  3         CCB [LUB$A_BUF_END] = .CCB [LUB$A_RBUF_ADR] + .CCB [LUB$W_RBUF_SIZE];
;   204         0915  3         CCB [LUB$L_PRINT_POS] = 0;
;   205         0916  2         END;
;   206         0917  2
;   207         0918  2     RETURN;
;   208         0919  1     END;                                        !End of BAS$$PUR_IO_CLO


                                                                .TITLE  BAS$$PUR_IO_BUF
                                                                .IDENT  \1-007\

                                                                .EXTRN  BAS$$CB_PUSH, BAS$$CB_POP
                                                                .EXTRN  BAS$$NEXT_LUN, BAS$$STOP_IO
                                                                .EXTRN  SYS$PUT, SYS$WAIT

                                                                .PSECT  _BAS$CODE,NOWRT,  SHR,  PIC,2

                                                 0004 00000     .ENTRY  BAS$$PUR_IO_CLO, Save R2                          ; 0839
                          52 00000000G     00  9E 00002         MOVAB   SYS$PUT, R2                                       ;
            5C        FE  AB               03  E1 00009         BBC     #3, -2(CCB), 4$                                   ; 0884
                          05        C4     AB  91 0000E         CMPB    -60(CCB), #5                                      ;
                                           56  13 00012         BEQL    4$                                                ;
      22    AB        B0  AB        EC     AB  A3 00014         SUBW3   -20(CCB), -80(CCB), 34(CCB)                       ; 0890
                      28  AB        EC     AB  D0 0001B         MOVL    -20(CCB), 40(CCB)                                 ; 0891
                      FE  AB               08  8A 00020         BICB2   #8, -2(CCB)                                       ; 0892
                                           5B  DD 00024         PUSHL   CCB                                               ; 0894
                          62               01  FB 00026         CALLS   #1, SYS$PUT                                       ;
                          28               50  E8 00029         BLBS    R0, 3$                                            ;
            000182DA  8F            08     AB  D1 0002C 1$:     CMPL    8(CCB), #99034                                    ; 0898
                          10               12 00034             BNEQ    2$                                                ;
                                           5B  DD 00036         PUSHL   CCB                                               ; 0900
            00000000G 00               01  FB 00038             CALLS   #1, SYS$WAIT                                      ;
                                           5B  DD 0003F         PUSHL   CCB                                               ; 0901
                          62               01  FB 00041         CALLS   #1, SYS$PUT                                       ;
                          E6               11 00044             BRB     1$                                                ; 0898
                          0A        08     AB  E8 00046 2$:     BLBS    8(CCB), 3$                                        ; 0904
                          7E               01  CE 0004A         MNEGL   #1, -(SP)                                         ;
            00000000G 00               01  FB 0004D             CALLS   #1, BAS$$STOP_IO                                  ;
                      DA  AB               0B  B0 00054 3$:     MOVW    #11, -38(CCB)                                     ; 0908
```

```
                               B0   AB       EC    AB  D0 00058          MOVL    -20(CCB), -80(CCB)                              ; 0913
                                    50       D2    AB  3C 0005D          MOVZWL  -46(CCB), R0                                    ; 0914
                               B4   AB       EC  BB40  9E 00061          MOVAB   @-20(CCB)[R0], -76(CCB)                         ;
                                             C8    AB  D4 00067          CLRL    -56(CCB)                                        ; 0915
                                                       04 0006A 4$:      RET                                                     ; 0919

; Routine Size:  107 bytes,    Routine Base:  _BAS$CODE + 0000


;  209          0920  1
```

```
1 GLOBAL ROUTINE BAS$$PUR_IO_ERR                        ! Purge terminal I/O
1     : NOVALUE =
1
1 !++
1 ! FUNCTIONAL DESCRIPTION:
1 !
1 !         This routine will PUT the contents of any terminal buffer.
1 !         It is used just before printing an error message to be sure that
1 !         the message appears after any output produced before the error
1 !         condition.
1 !
1 ! FORMAL PARAMETERS:
1 !
1 !         NONE
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !         LUB$V_OUTBUF_DR         Flag to indicate the output buffer has valid
1 !                                 data in it
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !         LUB$V_OUTBUF_DR         Flag to indicate the output buffer has valid
1 !         LUB$B_BAS_VFC2          'post' carriage control for PRN file format
1 !                                 data in it
1 !
1 ! ROUTINE VALUE:
1 !
1 !         NONE
1 !
1 ! SIDE EFFECTS:
1 !
1 !         NONE
1 !
1 !--
1
2     BEGIN
2
2     GLOBAL REGISTER
2         CCB = K_CCB_REG : REF BLOCK [, BYTE];
2
2     LOCAL
2         FLAG,
2         LUN;
2
2 !+
2 ! Scan through all logical units, purging the ones OPEN to a terminal.
2 !-
2     FLAG = 0;
2
2     DO
3         BEGIN
3 !+
3 ! Get the next logical unit number.
3 !-
3         BAS$$NEXT_LUN (FLAG, LUN);
3
```

```
;  268          0978  4         IF (.FLAG NEQ 0)
;  269          0979  3         THEN
;  270          0980  4             BEGIN
;  271          0981  4 !+
;  272          0982  4 ! LUN is the next logical unit number.  If the file it represents is
;  273          0983  4 ! open to a terminal, purge it.
;  274          0984  4 !-
;  275          0985  4             BAS$$CB_PUSH (.LUN, LUB$K_ILUN_MIN);
;  276          0986  4
;  277          0987  5             IF (.CCB [LUB$V_OPENED] AND .CCB [LUB$V_UNIT_0] AND .CCB [LUB$V_FORCIBLE])
;  278          0988  4             THEN
;  279          0989  4                 BAS$$PUR_IO_CLO ();
;  280          0990  4
;  281          0991  4             BAS$$CB_POP ();
;  282          0992  3             END;
;  283          0993  3
;  284          0994  3         END
;  285          0995  2     UNTIL (.FLAG EQL 0);
;  286          0996  2
;  287          0997  2     RETURN;
;  288          0998  1     END;                                        !End of BAS$$PUR_IO_ERR



                                        0804 00000          .ENTRY  BAS$$PUR_IO_ERR, Save R2,R11                  ; 0921
                        5E                08  C2 00002          SUBL2   #8, SP                                        ;
                                 04       AE  D4 00005          CLRL    FLAG                                          ; 0969
                                          5E  DD 00008 1$:      PUSHL   SP                                            ; 0976
                                 08       AE  9F 0000A          PUSHAB  FLAG                                          ;
            00000000G   00                02  FB 0000D          CALLS   #2, BAS$$NEXT_LUN                             ;
                                 04       AE  D5 00014          TSTL    FLAG                                          ; 0978
                                          25  13 00017          BEQL    3$                                            ;
                        50                08  CE 00019          MNEGL   #8, R0                                        ; 0985
                        52                6E  D0 0001C          MOVL    LUN, R2                                       ;
                           00000000G      00  16 0001F          JSB     BAS$$CB_PUSH                                  ;
                        0F       FC       AB  E9 00025          BLBC    -4(CCB), 2$                                   ; 0987
                                 FE       AB  95 00029          TSTB    -2(CCB)                                       ;
                                          0A  18 0002C          BGEQ    2$                                            ;
        05       FE     AB                06  E1 0002E          BBC     #6, -2(CCB), 2$                               ;
              FF5D      CF                00  FB 00033          CALLS   #0, BAS$$PUR_IO_CLO                           ; 0989
                           00000000G      00  16 00038 2$:      JSB     BAS$$CB_POP                                   ; 0991
                                 04       AE  D5 0003E 3$:      TSTL    FLAG                                          ; 0995
                                          C5  12 00041          BNEQ    1$                                            ;
                                              04 00043          RET                                                   ; 0998

; Routine Size:  68 bytes,    Routine Base:  _BAS$CODE + 006B


;  289          0999  1
;  290          1000  1 END                                             !End of module - BAS$$PUR_IO_BUF
;  291          1001  1
;  292          1002  0 ELUDOM
;                                     PSECT SUMMARY
;
;      Name                      Bytes                           Attributes
;
;   _BAS$CODE                       175  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                              Library Statistics
;
;                                             -------- Symbols --------       Pages        Processing
;      File                                   Total    Loaded   Percent       Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1          9776        9         0         581           00:01.2



;                                      COMMAND QUALIFIERS

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

; Size:         175 code + 0 data bytes
; Run Time:        00:09.1
; Elapsed Time:    00:22.4
; Lines/CPU Min:    6577
; Lexemes/CPU-Min: 35960
; Memory Used:  103 pages
; Compilation Complete
```

0029 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

BASOPEN
LIS

BASPOWJJ
LIS

BASPOS
LIS

BASPOWDJ
LIS

BASOPENDE
LIS

BASPOWGG
LIS

BASPOWHH
LIS

BASPOWRJ
LIS

BASPOWII
LIS

BASPURIOB
LIS

BASPOWDD
LIS

BASOPENZE
LIS

BASPOWDR
LIS

BASPOWGJ
LIS

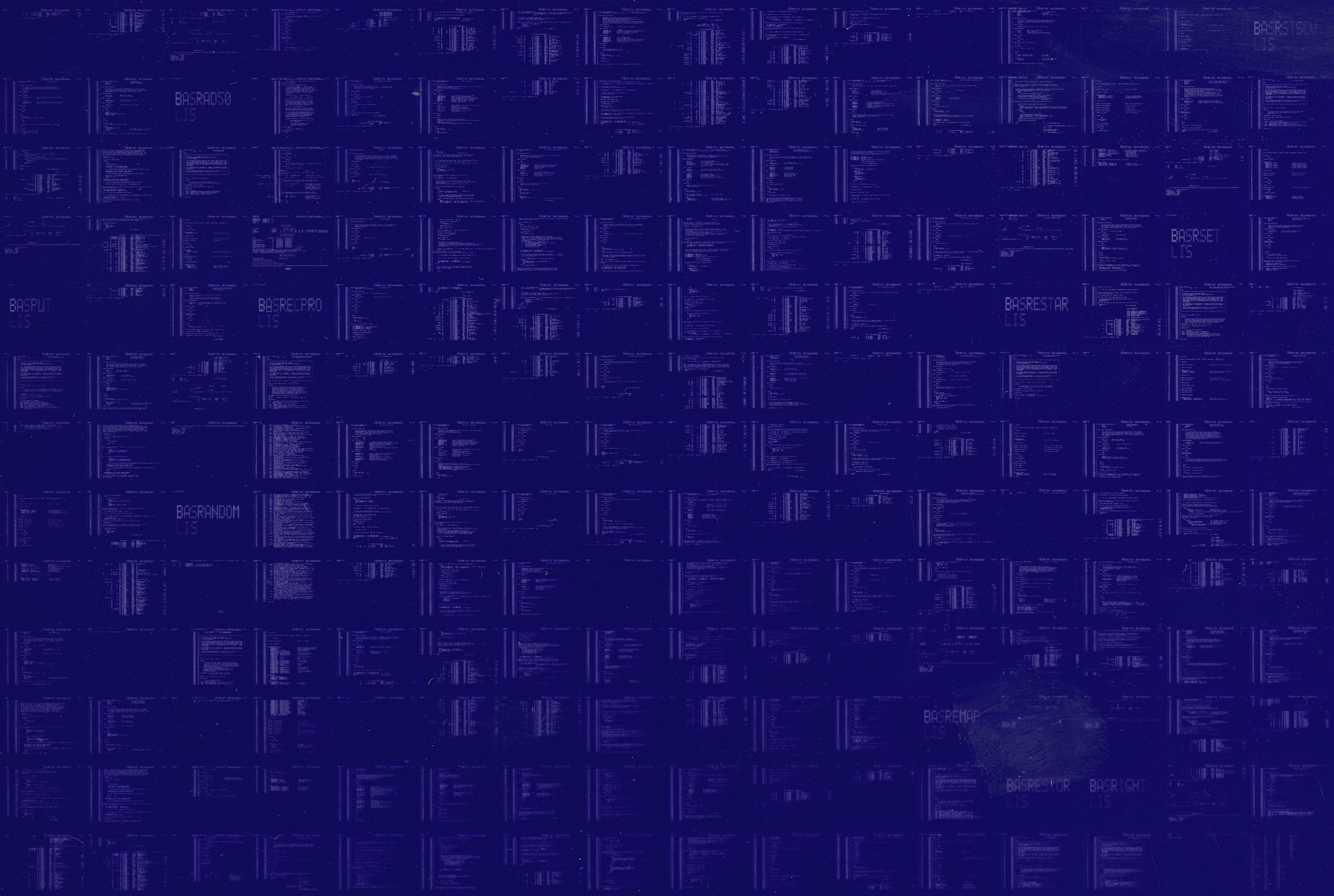
BASPOWRD
LIS

BASPOWHJ
LIS

BASPOWRR
LIS

0030 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY


BASRSTSCV
LIS

BASRAD50
LIS

BASRSET
LIS

BASPUT
LIS

BASRECPRO
LIS

BASRESTAR
LIS

BASRANDOM
LIS

BASREMAP
LIS

BASRESTOR
LIS

BASRIGHT
LIS