

```
BBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSS      RRRRRRRRRRR      TTTTTTTTTTTTTT   LLL
BBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSS      RRRRRRRRRRR      TTTTTTTTTTTTTT   LLL
BBBBBBBBBBBBB      AAAAAAAAAA      SSSSSSSSSSS      RRRRRRRRRRR      TTTTTTTTTTTTTT   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
BBB      BBB      AAA      AAA      SSS      RRR      RRR      TTT      LLL
BBB      BBB      AAA      AAA      SSS      RRR      RRR      TTT      LLL
BBBBBBBBBBBBB      AAA      AAA      SSSSSSSSS      RRRRRRRRRRR      TTT      LLL
BBBBBBBBBBBBB      AAA      AAA      SSSSSSSSS      RRRRRRRRRRR      TTT      LLL
BBBBBBBBBBBBB      AAA      AAA      SSSSSSSSS      RRRRRRRRRRR      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAA      SSS      RRR      RRR      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAA      SSS      RRR      RRR      TTT      LLL
BBB      BBB      AAAAAAAAAAAAAAAAA      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
BBBBBBBBBBBBB      AAA      AAA      SSSSSSSSS      RRR      RRR      TTT      LLL
BBBBBBBBBBBBB      AAA      AAA      SSSSSSSSS      RRR      RRR      TTT      LLL
BBBBBBB      BBB      AAA      AAA      SSSSSSSSS      RRR      RRR      TTT      LLL
LLLLLLLLLLLLLLLLLL
```

```

BBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      SSSSSSSS      TTTTTTTTTT      AAAAAA      RRRRRRRR
BBBBBBBB      AAAAAA      SSSSSSSS      RRRRRRRR      EEEEEEEEEE      SSSSSSSS      TTTTTTTTTT      AAAAAA      RRRRRRRR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BBBBBBBB      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      SSSSSS      TT      AA      AA      RRRRRRRR
BBBBBBBB      AA      AA      SSSSSS      RRRRRRRR      EEEEEEEE      SSSSSS      TT      AA      AA      RRRRRRRR
BB      BB      AAAAAAAAAA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AAAAAAAAAA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BB      BB      AA      AA      SS      RR      RR      EE      SS      TT      AA      AA      RR      RR
BBBBBBBB      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      SSSSSSSS      TT      AA      AA      RR      RR
BBBBBBBB      AA      AA      SSSSSSSS      RR      RR      EEEEEEEEEE      SSSSSSSS      TT      AA      AA      RR      RR

```

```

....
....
....
....

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS

```

```

1 0001 0 MODULE BAS$$RESTART_IO ( ! Restart BASIC I/O statement
2 0002 0 IDENT = '1-002' ! File: BASRESTAR.B32
3 0003 0 ) =
4 0004 1 BEGIN
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
10 0010 1 * ALL RIGHTS RESERVED. *
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
17 0017 1 * TRANSFERRED. *
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
21 0021 1 * CORPORATION. *
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1 ++
30 0030 1 FACILITY: BASIC + 2 Support Library - User callable
31 0031 1
32 0032 1 ABSTRACT:
33 0033 1
34 0034 1 This module effects the restart of a BASIC I/O statement.
35 0035 1 This only happens on certain typing errors from a terminal.
36 0036 1
37 0037 1 ENVIRONMENT: VAX-11 User mode
38 0038 1
39 0039 1 AUTHOR: John Sauter, CREATION DATE: 07-MAY-1979
40 0040 1
41 0041 1 MODIFIED BY:
42 0042 1
43 0043 1 1-001 - Original. JBS 07-MAY-1979
44 0044 1 1-002 - Conform to new BAS$$IO_BEG. JBS 24-JUL-1979
45 0045 1 --
46 0046 1
47 0047 1 !<BLF/PAGE>

```

```

49 0048 1 |
50 0049 1 | SWITCHES:
51 0050 1 |
52 0051 1 |
53 0052 1 SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
54 0053 1 |
55 0054 1 |
56 0055 1 | LINKAGES
57 0056 1 |
58 0057 1 |
59 0058 1 REQUIRE 'RTLIN:OTSLNK'; ! Define all linkages
60 0487 1 |
61 0488 1 |
62 0489 1 | TABLE OF CONTENTS:
63 0490 1 |
64 0491 1 |
65 0492 1 FORWARD ROUTINE
66 0493 1 BAS$$RESTART_IO; ! Restart an I/O statement
67 0494 1 |
68 0495 1 |
69 0496 1 | INCLUDE FILES:
70 0497 1 |
71 0498 1 |
72 0499 1 REQUIRE 'RTLML:OTSLUB'; ! logical unit block (LUB) offsets
73 0639 1 |
74 0640 1 REQUIRE 'RTLML:OTSISB'; ! I/O statement block (ISB) offsets
75 0808 1 |
76 0809 1 REQUIRE 'RTLIN:RTLPSECT'; ! Define DECLARE_PSECTS macro
77 0904 1 |
78 0905 1 LIBRARY 'RTLSTARLE'; ! STARLET macros and symbols
79 0906 1 |
80 0907 1 |
81 0908 1 | MACROS:
82 0909 1 |
83 0910 1 | NONE
84 0911 1 |
85 0912 1 | EQUATED SYMBOLS:
86 0913 1 |
87 0914 1 | NONE
88 0915 1 |
89 0916 1 | PSECT DECLARATIONS:
90 0917 1 |
91 0918 1 DECLARE_PSECTS (BAS);
92 0919 1 |
93 0920 1 | OWN STORAGE:
94 0921 1 |
95 0922 1 | NONE
96 0923 1 |
97 0924 1 |
98 0925 1 | EXTERNAL REFERENCES:
99 0926 1 |
100 0927 1 |
101 0928 1 EXTERNAL ROUTINE
102 0929 1 BAS$IO_END : NOVALUE, ! End of I/O statement
103 0930 1 BAS$$IO_BEG : NOVALUE; ! Start of I/O statement
104 0931 1 |
105 0932 1 EXTERNAL

```

BAS\$\$RESTART\_10  
1-002

: 106           0933 1     OTSS\$\$A\_CUR\_LUB;  
: 107           0934 1

I 13  
16-Sep-1984 01:04:44     VAX-11 Bliss-32 V4.0-742  
14-Sep-1984 11:56:35     [BASRTL.SRC]BASRESTAR.B32;1

! Address of current logical unit block

```

109 0935 1 GLOBAL ROUTINE BAS$$RESTART_IO = ! Restart an I/O statement
110 0936 1
111 0937 1 !++
112 0938 1 ABSTRACT:
113 0939 1
114 0940 1 Restart a BASIC I/O statement after a typing error.
115 0941 1 Information about the current statement is gathered, and then
116 0942 1 the statement is put through a normal end by calling BAS$IO_END.
117 0943 1 Using the gathered information the statement is initialized
118 0944 1 again by calling BAS$$IO_BEG with its original parameters.
119 0945 1 Return is made to the RESTART_IO subroutine in the error module,
120 0946 1 which will branch to the start of the I/O list.
121 0947 1
122 0948 1 FORMAL PAREMETERS:
123 0949 1
124 0950 1 NONE
125 0951 1
126 0952 1 IMPLICIT INPUTS:
127 0953 1
128 0954 1 OTSS$A_CUR_LUB.ra Pointer to the current LUB/ISB/RAB.
129 0955 1
130 0956 1 IMPLICIT OUTPUTS:
131 0957 1
132 0958 1 The LUB/ISB/RAB is reinitialized for the I/O statement.
133 0959 1
134 0960 1 ROUTINE VALUE:
135 0961 1
136 0962 1 The address of the beginning of the I/O list.
137 0963 1
138 0964 1 SIDE EFFECTS:
139 0965 1
140 0966 1 NONE
141 0967 1 --
142 0968 1
143 0969 2 BEGIN
144 0970 2
145 0971 2 GLOBAL REGISTER
146 0972 2 CCB = K_CCB_REG : REF BLOCK [, BYTE];
147 0973 2
148 0974 2 LOCAL
149 0975 2 RESTART_PC, ! Address of the start of the I/O list
150 0976 2 LUN, ! Logical unit number
151 0977 2 STMT_TYPE, ! Statement type
152 0978 2 FMP; ! User's frame pointer
153 0979 2
154 0980 2 !+
155 0981 2 Save the information needed to call BAS$$IO_BEG.
156 0982 2 --
157 0983 2 CCB = .OTSS$A_CUR_LUB;
158 0984 2 RESTART_PC = .CCB [ISB$A_RESTARTPC];
159 0985 2 LUN = .CCB [LUB$W_LUN];
160 0986 2 STMT_TYPE = .CCB [ISB$B_STMT_TYPE];
161 0987 2 FMP = .CCB [ISB$A_USER_FP];
162 0988 2 !+
163 0989 2 Suppress any "dangling prompt". The prompt will be produced again
164 0990 2 when we restart the I/O statement.
165 0991 2 --

```

```

: 166      0992 2      CCB [RAB$B_PSZ] = 0;
: 167      0993 2      +
: 168      0994 2      Now end the I/O statement.
: 169      0995 2      -
: 170      0996 2      BASSIO_END ();
: 171      0997 2      CCB = 0;
: 172      0998 2      +
: 173      0999 2      Now start a new I/O statement, just like the one we ended.
: 174      1000 2      -
: 175      1001 2      BASS$IO_BEG (.STMT_TYPE, .LUN, .FMP, .RESTART_PC);
: 176      1002 2      +
: 177      1003 2      Return the address of the I/O list to our caller, who will branch to
: 178      1004 2      it, thereby completing the restart of the I/O statement.
: 179      1005 2      -
: 180      1006 2      RETURN (.RESTART_PC);
: 181      1007 1      END.

```

! End of BAS\$\$RESTART\_IO routine

```

.TITLE  BAS$$RESTART_IO
.IDENT  \1-002\

.EXTRN  BASSIO_END, BASS$IO_BEG
.EXTRN  OTSS$A_CUR_LUB

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

.ENTRY  BAS$$RESTART_IO, Save R2,R3,R4,R5,R11
: 0935
: 0983  MOVL  OTSS$A_CUR_LUB, CCB
: 0984  MOVL  -188(CCB), _RESTART_PC
: 0985  CVTWL -58(CCB), LUN
: 0986  MOVZBL -143(CCB), STMT_TYPE
: 0987  MOVL  -180(CCB), FMP
: 0992  CLRB  52(CCB)
: 0996  CALLS #0, BASSIO_END
: 0997  CLRL  CCB
: 1001  PUSHR #^M<R2,R4>
:
:
: 1006  PUSHR #^M<R3,R5>
: 1007  CALLS #4, BASS$IO_BEG
:      MOVL  RESTART_PC, _R0
:      RET

```

		083C	00000
5B	00000000G	00	D0 00002
54	FF44	CB	D0 00009
55	C6	AB	32 0000E
53	FF71	CB	9A 00012
52	FF4C	CB	D0 00017
	34	AB	94 0001C
00000000G	00	00	FB 0001F
		5B	D4 00026
		14	BB 00028
00000000G	00	28	BB 0002A
		04	FB 0002C
	50	54	D0 00033
		04	00036

; Routine Size: 55 bytes, Routine Base: \_BAS\$CODE + 0000

```

: 182      1008 1
: 183      1009 1 END
: 184      1010 1
: 185      1011 0 ELUDOM

```

! End of BAS\$\$RESTART\_IO module

PSECT SUMMARY

Name	Bytes	Attributes
------	-------	------------

:  
: \_BAS\$CODE 55 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

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

COMMAND QUALIFIERS

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

: Size: 55 code + 0 data bytes  
: Run Time: 00:08.2  
: Elapsed Time: 00:24.1  
: Lines/CPU Min: 7388  
: Lexemes/CPU-Min: 44492  
: Memory Used: 110 pages  
: Compilation Complete



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

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 100 small terminal window screenshots, arranged in a 10x10 pattern. Each window shows a different software application or system utility. Several windows are clearly labeled with titles such as:

- BASRAD50 LIS
- BASRSET LIS
- BASRECLPRO LIS
- BASRESTAR LIS
- BASRANDOM LIS
- BASREMAP LIS
- BASRESTOR LIS
- BASRIGHT LIS

The content within the windows includes various data tables, command-line interfaces, and graphical elements typical of early computer systems. The overall appearance is that of a dense array of active workstations or a multi-user terminal environment.