


```

BBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      CCCCCCCC      RRRRRRRR      AAAAAA      TTTTTTTTTT      CC      CCCC
BBBBBBBB      AAAAAA      SSSSSSSS      SSSSSSSS      CCCCCCCC      RRRRRRRR      AAAAAA      TTTTTTTTTT      CC      CCCC
BB      BB      AA      AA      SS      SS      CC      RR      RR      AA      AA      TT      CC
BB      BB      AA      AA      SS      SS      CC      RR      RR      AA      AA      TT      CC
BB      BB      AA      AA      SS      SS      CC      RR      RR      AA      AA      TT      CC
BBBBBBBB      AA      AA      SSSSSS      SSSSSS      CC      RRRRRRRR      AA      AA      TT      CC
BBBBBBBB      AA      AA      SSSSSS      SSSSSS      CC      RRRRRRRR      AA      AA      TT      CC
BB      BB      AAAAAAAAAA      SS      SS      CC      RR      RR      AAAAAAAAAA      TT      CC
BB      BB      AAAAAAAAAA      SS      SS      CC      RR      RR      AAAAAAAAAA      TT      CC
BB      BB      AA      AA      SS      SS      CC      RR      RR      AA      AA      TT      CC
BB      BB      AA      AA      SS      SS      CC      RR      RR      AA      AA      TT      CC
BBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      CCCCCCCC      RR      RR      AA      AA      TT      CC
BBBBBBBB      AA      AA      SSSSSSSS      SSSSSSSS      CCCCCCCC      RR      RR      AA      AA      TT      CC

```

```

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

```

```

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

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

0001 0 MODULE BAS$SCRATCH (
0002 0 IDENT = '1-004'
0003 0 ) =
0004 1 BEGIN
0005 1
0006 1 *****
0007 1 *
0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0010 1 * ALL RIGHTS RESERVED.
0011 1 *
0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0017 1 * TRANSFERRED.
0018 1 *
0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0021 1 * CORPORATION.
0022 1 *
0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0025 1 *
0026 1 *****
0027 1
0028 1
0029 1
0030 1 **
0031 1 FACILITY:
0032 1 Basic support library - user callable
0033 1
0034 1 ABSTRACT:
0035 1 This module is the UPI level of the Basic SCRATCH construct. Initially,
0036 1 it contains only the code for sequential I/O. This module will set
0037 1 up the I/O data base for the LUN and go directly to the REC level.
0038 1
0039 1
0040 1 ENVIRONMENT:
0041 1 User access mode - AST reentrant.
0042 1
0043 1 AUTHOR: Donald G. Petersen, CREATION DATE: 27-Feb-79
0044 1
0045 1 MODIFIED BY:
0046 1
0047 1 DGP, 27-Feb-79 : VERSION 01
0048 1 1-001 - original. DGP 27-Feb-79
0049 1 1-002 - Include SWITCHES declaration. DGP 25-Jun-79
0050 1 1-003 - Set up ISB$A_USER_FP. JBS 25-JUL-1979
0051 1 1-004 - Check for virtual array use of this file. DGP 16-Oct-79
0052 1 --
0053 1
0054 1 !<BLF/PAGE>

```

```

56 0055 1 |
57 0056 1 | SWITCHES
58 0057 1 |
59 0058 1 |
60 0059 1 | SWITCHES ADDRESSING_MODE (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
61 0060 1 |
62 0061 1 |
63 0062 1 | LINKAGES
64 0063 1 |
65 0064 1 |
66 0065 1 | REQUIRE 'RTLIN:OTSLNK';           ! Define all linkages
67 0494 1 |
68 0495 1 |
69 0496 1 | TABLE OF CONTENTS:
70 0497 1 |
71 0498 1 |
72 0499 1 | FORWARD ROUTINE
73 0500 1 |     BASSCRATCH : NOVALUE;         ! UPI level Sequential SCRATCH
74 0501 1 |
75 0502 1 |
76 0503 1 | INCLUDE FILES:
77 0504 1 |
78 0505 1 |
79 0506 1 | REQUIRE 'RTLML:OTISISB';         ! ISB definitions
80 0674 1 |
81 0675 1 | REQUIRE 'RTLML:OTSLUB';         ! LUB definitions
82 0815 1 |
83 0816 1 | REQUIRE 'RTLIN:RTLPSECT';       ! Define DECLARE_PSECTS macro
84 0911 1 |
85 0912 1 | LIBRARY 'RTLS'ARLE';           ! Starlet system macros
86 0913 1 |
87 0914 1 |
88 0915 1 | MACROS:
89 0916 1 |
90 0917 1 |     NONE
91 0918 1 |
92 0919 1 | EQUATED SYMBOLS:
93 0920 1 |     NONE
94 0921 1 |
95 0922 1 |
96 0923 1 | PSECT DECLARATIONS:
97 0924 1 |
98 0925 1 | DECLARE_PSECTS (BAS);
99 0926 1 |
100 0927 1 | OWN STORAGE:
101 0928 1 |
102 0929 1 |     NONE
103 0930 1 |
104 0931 1 | EXTERNAL REFERENCES:
105 0932 1 |
106 0933 1 |
107 0934 1 | EXTERNAL ROUTINE
108 0935 1 |     BASS$STOP_IO : NOVALUE,       ! Signal fatal BASIC I/O error
109 0936 1 |     BASS$REC_SSE : JSB_REC0 NOVALUE, ! REC level processing - RMS interface
110 0937 1 |     SCRATCH sequential
111 0938 1 |     BASS$CB_PUSH . JSB_CB_PUSH NOVALUE, ! Load register CCB
112 0939 1 |     BASS$CB_POP : JSB_CB_POP NOVALUE; ! Done with register CCB

```

```
: 113      0940 1
: 114      0941 1 !+
: 115      0942 1 !: The following are the error codes used in this module.
: 116      0943 1 !-
: 117      0944 1
: 118      0945 1 EXTERNAL LITERAL
: 119      0946 1   BASSK_ILLILLACC : UNSIGNED (8),           ! Illegal or illogical access
: 120      0947 1   BASSK_IO_CHANOT : UNSIGNED (8);         ! I/O channel not open
: 121      0948 1
```

```

: 123 0949 1 GLOBAL ROUTINE BASSCRATCH (          ! SCRATCH sequential
: 124 0950 1     UNIT                          ! logical unit number
: 125 0951 1     ) : NOVALUE =
: 126 0952 1
: 127 0953 1 !++
: 128 0954 1 FUNCTIONAL DESCRIPTION:
: 129 0955 1
: 130 0956 1     This routine will set up the I/O data base for this LUN if necessary
: 131 0957 1     and then go directly to the REC level.  When control is returned to
: 132 0958 1     this routine, it pops the CCB off of the I/O system.  The actual inter-
: 133 0959 1     face to RMS is done at the REC level.  The file is truncated at the current record.
: 134 0960 1
: 135 0961 1 FORMAL PARAMETERS:
: 136 0962 1
: 137 0963 1     UNIT.rlu.v      logical unit number
: 138 0964 1
: 139 0965 1 IMPLICIT INPUTS:
: 140 0966 1
: 141 0967 1     LUB$V_VA USE      virtual array use on this file
: 142 0968 1     LUB$V_READ_ONLY   file is read only
: 143 0969 1
: 144 0970 1 IMPLICIT OUTPUTS:
: 145 0971 1
: 146 0972 1     ISB$B_STM TYPE    the statement type
: 147 0973 1     LUB$V_BLK_USE    block use on this file
: 148 0974 1
: 149 0975 1 COMPLETION CODES:
: 150 0976 1
: 151 0977 1     NONE
: 152 0978 1
: 153 0979 1 SIDE EFFECTS:
: 154 0980 1
: 155 0981 1     Signals:
: 156 0982 1     BASSK_IO_CHANOT (I/O channel not open)
: 157 0983 1     BASSK_ILCILLACC (illegal or illogical access)
: 158 0984 1
: 159 0985 1 --
: 160 0986 1
: 161 0987 2 BEGIN
: 162 0988 2
: 163 0989 2 BUILTIN
: 164 0990 2     FP;
: 165 0991 2
: 166 0992 2 GLOBAL REGISTER
: 167 0993 2     CCB = K_CCB_REG : REF BLOCK [, BYTE];
: 168 0994 2
: 169 0995 2 LOCAL
: 170 0996 2     FMP : REF BLOCK [, BYTE];
: 171 0997 2
: 172 0998 2     FMP = .FP;
: 173 0999 2 !+
: 174 1000 2 Allocate the LUB/ISB/RAB for this unit if necessary.  Store new CB (con-
: 175 1001 2 trol block) in OTS$$A_CUR_LUB.  Store signed unit number in LUB$W_LUN.
: 176 1002 2 --
: 177 1003 2 BASS$CB PUSH (.UNIT, LUB$K_ILUN MIN);
: 178 1004 2 CCB [ISB$A_USER_FP] = .FMP-[SF$C_SAVE_FP];
: 179 1005 2 !+

```

```

180 1006 ! If the channel is not open, give an error message.
181 1007 SCRATCH is not valid on unit 0.
182 1008
183 1009
184 1010 IF ( NOT .CCB [LUB$V_OPENED]) THEN BAS$$STOP_IO (BAS$K_IO_CHANOT);
185 1011
186 1012
187 1013 ! Now that the data base is in place, store the statement type and go
188 1014 directly to the REC level.
189 1015
190 1016 CCB [ISB$B_STTM_TYPE] = ISB$K_ST_TY_SCR;
191 1017
192 1018 ! Check for virtual array usage and set block usage
193 1019
194 1020 IF .CCB [LUB$V_VA_USE] OR .CCB [LUB$V_READ_ONLY] THEN BAS$$STOP_IO(BAS$K_ILLILLACC);
195 1021 CCB [LUB$V_BLK_USE] = 1;
196 1022 BAS$$REC_SSE (J);
197 1023
198 1024 ! Now that the SCRATCH has been done, pop the CCB off the I/O system.
199 1025
200 1026 BAS$$CB_POP ();
201 1027 END;

```

!End of BAS\$\$SCRATCH

```

.TITLE BAS$$SCRATCH
.IDENT \1-004\

.EXTRN BAS$$STOP_IO, BAS$$REC_SSE
.EXTRN BAS$$CB_PUSH, BAS$$CB_POP
.EXTRN BAS$K_ILLILLACC
.EXTRN BAS$K_IO_CHANOT

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

```

			083C	00000	.ENTRY	BAS\$\$SCRATCH, Save R2, R3, R4, R5, R11	: 0949
	54	00000000G	00	9E 00002	MOVAB	BAS\$\$STOP_IO, R4	
	53		5D	DO 00009	MOVL	FP, FMP	: 0998
	50		08	CE 0000C	MNEGL	#8, R0	: 1003
	52	04	AC	DO 0000F	MOVL	UNIT, R2	
		00000000G	00	16 00013	JSB	BAS\$\$CB_PUSH	
	FF4C		0C	A3 DO 00019	MOVL	12(FMP), -180(CCB)	: 1004
	07		FC	AB E8 0001F	BLBS	-4(CCB), 1\$: 1010
	7E	00G	8F	9A 00023	MOVZBL	#BAS\$K_IO_CHANOT, -(SP)	
	64		01	FB 00027	CALLS	#1, BAS\$\$STOP_IO	
	FF71		26	90 0002A 1\$:	MOVB	#38, -143(CCB)	: 1016
	05	FF	AB	E8 0002F	BLBS	-1(CCB), 2\$: 1020
07	FC		02	E1 00033	BBC	#2, -4(CCB), 3\$	
	7E	00G	8F	9A 00038 2\$:	MOVZBL	#BAS\$K_ILLILLACC, -(SP)	
	64		01	FB 0003C	CALLS	#1, BAS\$\$STOP_IO	
	FF		02	88 0003F 3\$:	BISB2	#2, -1(CCB)	: 1021
		00000000G	00	16 00043	JSB	BAS\$\$REC_SSE	: 1022
		00000000G	00	16 00049	JSB	BAS\$\$CB_POP	: 1026
			04	0004F	RET		: 1027

; Routine Size: 80 bytes. Routine Base: _BAS\$CODE + 0000

BAS\$SCRATCH
1-004

L 12
16-Sep-1984 01:12:41
14-Sep-1984 11:56:39

VAX-11 Bliss-32 V4.0-742
[BASRTL.SRC]BAS\$SCRATC.B32;1

Page 6
(3)

```
: 202      1028  1
: 203      1029  1 END
: 204      1030  1
: 205      1031  0 ELUDOM
```

!End of module - BAS\$SCRATC

PSECT SUMMARY

```
: Name                Bytes                Attributes
: _BAS$CODE           80 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)
```

Library Statistics

```
: File                Total  Symbols  Percent  Pages  Processing
:                   -----  Loaded  -----  Mapped  Time
: _$255$DUA28:[SYSLIB]STARLET.L32;1  9776      1      0      581    00:01.1
```

COMMAND QUALIFIERS

```
: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/NOTRACE/LIS=LIS$:BAS$SCRATC/OBJ=OBJ$:BAS$SCRATC MSRC$:BAS$SCRATC/UPDATE=(ENH$:BAS$SCRATC
: )
```

```
: Size:                80 code + 0 data bytes
: Run Time:            00:08.2
: Elapsed Time:       00:21.1
: Lines/CPU Min:      7525
: Lexemes/CPU-Min:   45386
: Memory Used:        113 pages
: Compilation Complete
```


0031 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

The image displays a grid of 144 terminal windows, arranged in 12 rows and 12 columns. Each window shows a different terminal session on a VAX/VMS system. The sessions vary in content, including:

- System messages and boot logs.
- Command-line interactions with various system utilities.
- Output from specific commands such as:
 - `BASRTDIM LIS`
 - `BASSARITH LIS`
 - `BASSCALE LIS`
 - `BASIGNAL LIS`
 - `BASRUNIMI LIS`
 - `BASCRATC LIS`
 - `BASRSTSFI LIS`
 - `BASLEEP LIS`
 - `BASSTOP LIS`
 - `BASSEG LIS`
- Data listings and reports.
- Error messages and diagnostic information.