


```

PPPPPPPP      AAAAAA      SSSSSSSS      000000      CCCCCCCC      TTTTTTTTTT
PPPPPPPP      AAAAAA      SSSSSSSS      000000      CCCCCCCC      TTTTTTTTTT
PP      PP      AA      AA      SS      00      00      CC      TT
PP      PP      AA      AA      SS      00      00      CC      TT
PP      PP      AA      AA      SS      00      00      CC      TT
PP      PP      AA      AA      SS      00      00      CC      TT
PPPPPPPP      AA      AA      SSSSSS      00      00      CC      TT
PPPPPPPP      AA      AA      SSSSSS      00      00      CC      TT
PP      AAAAAAAAAA      SS      00      00      CC      TT
PP      AAAAAAAAAA      SS      00      00      CC      TT
PP      AA      AA      SS      00      00      CC      TT
PP      AA      AA      SS      00      00      CC      TT
PP      AA      AA      SSSSSSSS      000000      CCCCCCCC      TT
PP      AA      AA      SSSSSSSS      000000      CCCCCCCC      TT

```

```

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

```

```

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 PASSOCT ( %TITLE 'Convert value in base 8 to string'
2 0002 0 IDENT = '1-001' ! File: PASSOCT.B32 Edit: SBL1001
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 ++
31 0031 1 FACILITY: Pascal Language Support
32 0032 1
33 0033 1 ABSTRACT:
34 0034 1
35 0035 1 This module contains PASSOCT which implements the
36 0036 1 VAX-11 Pascal OCT procedure.
37 0037 1
38 0038 1 ENVIRONMENT: User mode - AST reentrant
39 0039 1
40 0040 1 AUTHOR: Steven B. Lionel, CREATION DATE: 1-April-1981
41 0041 1
42 0042 1 MODIFIED BY:
43 0043 1
44 0044 1 1-001 - Original. SBL 1-April-1981
45 0045 1 --
46 0046 1

```

```

: 48      0047 1 %SBTTL 'Declarations'
: 49      0048 1 |
: 50      0049 1 | PROLOGUE DEFINITIONS:
: 51      0050 1 |
: 52      0051 1 |
: 53      0052 1 REQUIRE 'RTLIN:PASPROLOG';           ! Externals, linkages, PSECTs, structures
: 54      0116 1 |
: 55      0117 1 |
: 56      0118 1 | TABLE OF CONTENTS:
: 57      0119 1 |
: 58      0120 1 |
: 59      0121 1 FORWARD ROUTINE
: 60      0122 1     PASSOCT: NOVALUE;                ! Convert value in base 8
: 61      0123 1 |
: 62      0124 1 |
: 63      0125 1 | MACROS:
: 64      0126 1 |
: 65      0127 1 |     NONE
: 66      0128 1 |
: 67      0129 1 | EQUATED SYMBOLS:
: 68      0130 1 |
: 69      0131 1 |     NONE
: 70      0132 1 |
: 71      0133 1 | FIELDS:
: 72      0134 1 |
: 73      0135 1 |     NONE
: 74      0136 1 |
: 75      0137 1 | OWN STORAGE:
: 76      0138 1 |
: 77      0139 1 |     NONE
: 78      0140 1 |
: 79      0141 1 |
: 80      0142 1 !! If this is for a V2 system, redefine OTSS$CVT_L_TO as PASS$CVT_L_TO.
: 81      L 0143 1 %IF %VARIANT
: 82      U 0144 1 %THEN
: 83      U 0145 1 UNDECLARE
: 84      U 0146 1     OTSS$CVT_L_TO;
: 85      U 0147 1 EXTERNAL ROUTINE
: 86      U 0148 1     PASS$CVT_L_TO;
: 87      U 0149 1 BIND ROUTINE
: 88      U 0150 1     OTSS$CVT_L_TO = PASS$CVT_L_TO;
: 89      0151 1 %FI

```

```

: 91 0152 1 %SBTTL 'PASSOCT - Convert value in base 8 to string'
: 92 0153 1 GLOBAL ROUTINE PASSOCT (
: 93 0154 1     RESULT: REF VECTOR [, BYTE],           ! Result string
: 94 0155 1     TOTAL_WIDTH: WORD SIGNED,           ! Total field width
: 95 0156 1     NBITS,                               ! Size of value in bits
: 96 0157 1     VALUE,                               ! Address of value
: 97 0158 1     MIN_DIGITS: SIGNED                  ! Minimum number of digits
: 98 0159 1     ): NOVALUE =
: 99 0160 1
100 0161 1 ++
101 0162 1 | FUNCTIONAL DESCRIPTION:
102 0163 1 |
103 0164 1 |     This procedure implements the VAX-11 Pascal OCT function. It
104 0165 1 |     converts a value to an ASCII representation in base 8 and stores
105 0166 1 |     that result in a string.
106 0167 1 |
107 0168 1 | CALLING SEQUENCE:
108 0169 1 |
109 0170 1 |     CALL PASSOCT (RESULT.wt.r, TOTAL_WIDTH.rw.v, NBITS.rl.v, VALUE.rz.r
110 0171 1 |     [, MIN_DIGITS.rl.v])
111 0172 1 |
112 0173 1 | FORMAL PARAMETERS:
113 0174 1 |
114 0175 1 |     RESULT           - The string into which the result will be placed.
115 0176 1 |
116 0177 1 |     TOTAL_WIDTH     - Total field width.
117 0178 1 |
118 0179 1 |     NBITS           - The size of VALUE in bits.
119 0180 1 |
120 0181 1 |     VALUE           - The address of the value to write.
121 0182 1 |
122 0183 1 |     MIN_DIGITS     - Optional. The minimum number of digits to appear
123 0184 1 |     in the result. Defaults to the minimum necessary to
124 0185 1 |     represent every bit of the value.
125 0186 1 |
126 0187 1 | IMPLICIT INPUTS:
127 0188 1 |
128 0189 1 |     NONE
129 0190 1 |
130 0191 1 | IMPLICIT OUTPUTS:
131 0192 1 |
132 0193 1 |     NONE
133 0194 1 |
134 0195 1 | ROUTINE VALUE:
135 0196 1 |
136 0197 1 |     NONE
137 0198 1 |
138 0199 1 | SIDE EFFECTS:
139 0200 1 |
140 0201 1 |     NONE
141 0202 1 |
142 0203 1 | SIGNALLED ERRORS:
143 0204 1 |
144 0205 1 |     NEGDIGARG - negative "digits" argument to BIN, HEX or OCT is not allowed
145 0206 1 |
146 0207 1 | --
147 0208 1

```

```

: 148 0209 2 BEGIN
: 149 0210 2
: 150 0211 2 LOCAL
: 151 0212 2     ACTUAL_DIGITS,           ! Number of digits actually used
: 152 0213 2     ACTUAL_NBITS,         ! Value size actually used
: 153 0214 2     DESCR: "BLOCK" [8, BYTE]; ! String descriptor
: 154 0215 2
: 155 0216 2 LITERAL
: 156 0217 2     M_SIZE_IN_BITS = %X'04'; ! Flags argument for
: 157 0218 2                                           ! convert routine
: 158 0219 2 BUILTIN
: 159 0220 2     ACTUALCOUNT;
: 160 0221 2
: 161 0222 2     !+
: 162 0223 2     ! Set initial values for conversion.
: 163 0224 2     !-
: 164 0225 2
: 165 0226 2     ACTUAL_NBITS = .NBITS;
: 166 0227 2     ACTUAL_DIGITS = (.ACTUAL_NBITS+2)/3;
: 167 0228 2
: 168 0229 2     !+
: 169 0230 2     ! Create result string descriptor with actual width.
: 170 0231 2     !-
: 171 0232 2
: 172 0233 2     DESCR [DSC$B_CLASS] = DSC$K_CLASS_S;
: 173 0234 2     DESCR [DSC$B_DTYPE] = DSC$K_DTYPE_T;
: 174 0235 2     DESCR [DSC$A_POINTER] = RESULT [0];
: 175 0236 2
: 176 0237 2     IF ACTUALCOUNT () GEQU 5
: 177 0238 2     THEN
: 178 0239 2         BEGIN
: 179 0240 2             ACTUAL_DIGITS = .MIN_DIGITS;
: 180 0241 2             IF .ACTUAL_DIGITS LSS 0
: 181 0242 2             THEN
: 182 0243 2                 SIGNAL_STOP (PASS$NEGDIGARG); ! Negative "digits" argument to BIN, HEX or OCT is not allow
: 183 0244 2             END;
: 184 0245 2
: 185 0246 2     DESCR [DSC$W_LENGTH] = .TOTAL_WIDTH;
: 186 0247 2     !+
: 187 0248 2     ! Will TOTAL_WIDTH truncate the value?
: 188 0249 2     !+
: 189 0250 2     IF .TOTAL_WIDTH LSSU (.ACTUAL_NBITS+2)/3
: 190 0251 2     THEN
: 191 0252 2         ACTUAL_NBITS = .TOTAL_WIDTH * 3;
: 192 0253 2
: 193 0254 2     IF .ACTUAL_DIGITS GTRU .TOTAL_WIDTH
: 194 0255 2     THEN
: 195 0256 2         ACTUAL_DIGITS = .TOTAL_WIDTH;
: 196 0257 2
: 197 0258 2     !+
: 198 0259 2     ! Do the conversion. We assume it won't fail.
: 199 0260 2     !-
: 200 0261 2
: 201 0262 2     OTSS$CVT_L_TO (.VALUE, DESCR, .ACTUAL_DIGITS, .ACTUAL_NBITS,
: 202 0263 2         M_SIZE_IN_BITS);
: 203 0264 2
: 204 0265 2     RETURN;

```

PASSOCT
1-001

Convert value in base 8 to string
PASSOCT - Convert value in base 8 to string

N 9
16-Sep-1984 01:45:34
14-Sep-1984 12:51:40

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASSOCT.B32;1

Page 5
(3)

: 205
: 206

0266 2
0267 1 END;

! End of routine PASSOCT

			001C	00000		.ENTRY	PASSOCT, Save R2,R3,R4		: 0153
	5E		08	C2	00002	SUBL2	#8, SP		
	52	OC	AC	D0	00005	MOVL	NBITS, ACTUAL_NBITS		: 0226
	53	02	A2	9E	00009	MOVAB	2(R2), R3		: 0227
	53		03	C6	0000D	DIVL2	#3, R3		
	54		53	D0	00010	MOVL	R3, ACTUAL_DIGITS		
	02	AE	010E	8F	B0	MOVW	#270, DESCR+2		: 0234
	04	AE	04	AC	D0	MOVL	RESULT, DESCR+4		: 0235
	05			6C	91	CMPB	(AP), #5		: 0237
				13	1F	BLSSU	1\$		
	54	14	AC	D0	00023	MOVL	MIN_DIGITS, ACTUAL_DIGITS		: 0240
				0D	18	BGEQ	1\$: 0241
				8F	DD	PUSHL	#PASS_NEGDIGARG		: 0243
	00000000G		00	01	FB	CALLS	#1, LIB\$STOP		
			50	AC	32	CVTWL	TOTAL_WIDTH, R0		: 0246
			6E	50	B0	MOVW	R0, DESCR		
			53	50	D1	CML	R0, R3		: 0250
				04	1E	BGEQU	2\$		
	52		50	03	C5	MULL3	#3, R0, ACTUAL_NBITS		: 0252
			50	54	D1	CML	ACTUAL_DIGITS, R0		: 0254
				03	1B	BLEQU	3\$		
			54	50	D0	MOVL	R0, ACTUAL_DIGITS		: 0256
				04	DD	PUSHL	#4		: 0262
				52	DD	PUSHL	ACTUAL_NBITS		
				54	DD	PUSHL	ACTUAL_DIGITS		
		OC	AE	9F	00054	PUSHAB	DESCR		
		10	AC	DD	00057	PUSHL	VALUE		
	00000000G		00	05	FB	CALLS	#5, OT\$CVT_L_TO		: 0267
				04	00061	RET			

; Routine Size: 98 bytes, Routine Base: _PASSCODE + 0000

: 207
: 208 0268 1
0269 1 !<BLF/PAGE>

PASSOCT
1-001

Convert value in base 8 to string
PASSOCT - Convert value in base 8 to string

B 10
16-Sep-1984 01:45:34
14-Sep-1984 12:51:40

VAX-11 Bliss-32 V4.0-742
[PASRTL.SRC]PASOCT.B32;1

: 210 0270 1 END
: 211 0271 1
: 212 0272 0 ELUDOM

! End of module PASSOCT

.EXTRN LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
_PASS\$CODE	98	NOVEC,NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	6	0	581	00:01.0
_\$255\$DUA28:[PASRTL.OBJ]PASLIB.L32;1	427	3	0	33	00:00.4

COMMAND QUALIFIERS

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

: Size: 98 code + 0 data bytes
: Run Time: 00:04.6
: Elapsed Time: 00:18.0
: Lines/CPU Min: 3586
: Lexemes/CPU-Min: 8070
: Memory Used: 54 pages
: Compilation Complete

