

[illegible]

3-

Sy

LI

LI
LILI
LI
LILI
LI

LI

LI
LILI
LILI
LI

LI

LI
LI

LI

LI
LI

LI

LI
LI

LI

11

LI

LI
LI

LI

LI
LI

21

LI

LI
LI

22

11

LI
LI

LI

LI
LI

LI

```

LL      IIIIII  BBBB BBBB  AAAAAA  SSSSSSSS  CCCCCCCC  TTTT TTTT TTTT  IIIIII  MM      MM
LL      IIIIII  BBBB BBBB  AAAAAA  SSSSSSSS  CCCCCCCC  TTTT TTTT TTTT  IIIIII  MM      MM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MMMM  MMMM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MMMM  MMMM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MM  MM  MM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MM  MM  MM
LL      II      BBBB BBBB  AA      AA  SSSSSS  CC      TT      II      MM      MM
LL      II      BBBB BBBB  AA      AA  SSSSSS  CC      TT      II      MM      MM
LL      II      BB      BB  AAAAAAAAAA  SS      CC      TT      II      MM      MM
LL      II      BB      BB  AAAAAAAAAA  SS      CC      TT      II      MM      MM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MM      MM
LL      II      BB      BB  AA      AA  SS      CC      TT      II      MM      MM
LL      IIIIII  BBBB BBBB  AA      AA  SSSSSSSS  CCCCCCCC  TT      IIIIII  MM      MM
LL      IIIIII  BBBB BBBB  AA      AA  SSSSSSSS  CCCCCCCC  TT      IIIIII  MM      MM
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
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS

```

```
1 0001 0 MODULE LIB$SYS_ASCTIM (
2 0002 0
3 0003 0 IDENT = '1-008' ! File: LIBASCTIM.B32 EDIT: RKR1008
4 0004 0
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1 *****
9 0009 1 *
10 0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
11 0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
12 0012 1 * ALL RIGHTS RESERVED.
13 0013 1 *
14 0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
15 0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
16 0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
17 0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
18 0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
19 0019 1 * TRANSFERRED.
20 0020 1 *
21 0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
22 0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
23 0023 1 * CORPORATION.
24 0024 1 *
25 0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
26 0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
27 0027 1 *
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1 ++
33 0033 1 FACILITY: General Library Support
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine calls $ASCTIM to convert binary date and time
38 0038 1 to an ASCII string, returning the resultant string using the
39 0039 1 semantics of the caller's string
40 0040 1 Parameter CONVFLG is presented to this routine by-reference
41 0041 1 and promoted to by-value for presentation to $ASCTIM.
42 0042 1
43 0043 1 ENVIRONMENT: VAX-11 User Mode
44 0044 1
45 0045 1 AUTHOR: R. Reichert, CREATION DATE: 20-SEPT-1979
46 0046 1
47 0047 1 MODIFIED BY:
48 0048 1
49 0049 1 1-001 - Original. RKR 20-SEPT-1979
50 0050 1 1-002 - Change name to LIB$SYS_ASCTIM. RKR 15-OCT-79
51 0051 1 1-003 - Change logic for status returned. Improve comments.
52 0052 1 RKR 20-OCT-79
53 0053 1 1-004 - Add code to promote by-reference arguments to by-value.
54 0054 1 RKR 1-NOV-79
55 0055 1 1-005 - Change STR$ codes to LIB$ codes. RKR 22-JAN-80
56 0056 1 1-006 - Enhance to recognize additional classes of string descriptors
57 0057 1 on output by always getting data into internal buffer and
```

```
: 58      0058 1  | using LIB$SCOPY_DXDX6 to deliver to caller's buffer.  
: 59      0059 1  | LIB$ANALYZE_SDESC_R3 is used to compute the number of bytes  
: 60      0060 1  | actually copied after the copy operation completes.  
: 61      0061 1  | Change to use LIB$SCOPY_DXDX6 instead of STR$COPY_DX. This  
: 62      0062 1  | avoids having to change STR$ statuses to LIB$ statuses and  
: 63      0063 1  | eliminates need for a handler.  
: 64      0064 1  | RKR 29-MAY-1981.  
: 65      0065 1  | 1-007 - Add special-case code to speed up performance for strings that  
: 66      0066 1  | "read" like fixed strings. RKR 7-OCT-1981.  
: 67      0067 1  | 1-008 - Redirect jsb's from LIB$ANALYZE_SDESC_R3 to  
: 68      0068 1  | LIB$ANALYZE_SDESC_R2. Redirect jsb from LIB$SCOPY_DXDX6 to  
: 69      0069 1  | LIB$SCOPY_R_DX6 to gain a little more speed.  
: 70      0070 1  | RKR 18-NOV-1981.  
: 71      0071 1  | --  
: 72      0072 1  |  
: 73      0073 1  | !<BLF/PAGE>
```

```
: 75      0074 1 |
: 76      0075 1 | SWITCHES:
: 77      0076 1 |
: 78      0077 1 |
: 79      0078 1 | SWITCHES ADDRESSING MODE
: 80      0079 1 | (EXTERNAL = GENERAL, NONEXTERNAL = WORD_RELATIVE);
: 81      0080 1 |
: 82      0081 1 |
: 83      0082 1 | LINKAGES:
: 84      0083 1 |
: 85      0084 1 | REQUIRE 'RTLIN:STRLNK';           ! Linkage for LIB$ANALYZE_SDESC_R2
: 86      0269 1 |
: 87      0270 1 | TABLE OF CONTENTS:
: 88      0271 1 |
: 89      0272 1 |
: 90      0273 1 | FORWARD ROUTINE
: 91      0274 1 |     LIB$SYS_ASCTIM;               ! Convert date and time
: 92      0275 1 |
: 93      0276 1 |
: 94      0277 1 | INCLUDE FILES:
: 95      0278 1 |
: 96      0279 1 |
: 97      0280 1 | REQUIRE 'RTLIN:RTLPSECT';         ! Macros for defining psects
: 98      0375 1 |
: 99      0376 1 | LIBRARY 'RTL$STARLE';             ! System symbols and macros
100      0377 1 |
101      0378 1 |
102      0379 1 | MACROS:
103      0380 1 |
104      0381 1 |     NONE
105      0382 1 |
106      0383 1 | EQUATED SYMBOLS:
107      0384 1 |
108      0385 1 | LITERAL
109      0386 1 |     LOCAL_BUF_SIZE = 42 ; ! The maximum number of characters that
110      0387 1 |                          ! can be returned by a call to $ASCTIM
111      0388 1 |
112      0389 1 | PSECTS:
113      0390 1 |
114      0391 1 | DECLARE_PSECTS (LIB);             ! Declare psects for LIB$ facility
115      0392 1 |
116      0393 1 | OWN STORAGE:
117      0394 1 |
118      0395 1 |     NONE
119      0396 1 |
120      0397 1 | EXTERNAL REFERENCES:
121      0398 1 |
122      0399 1 |
123      0400 1 | EXTERNAL ROUTINE
124      0401 1 |     LIB$ANALYZE_SDESC_R2 : LIB$ANALYZE_SDESC_JSB_LINK, ! Extract length
125      0402 1 |                                                              ! and address of
126      0403 1 |                                                              ! 1st data byte
127      0404 1 |                                                              ! from descriptor
: 128      0405 1 |     LIB$COPY_R_DX6 : STRING_JSB;   ! Copy string
```

```

: 130      0406 1 GLOBAL ROUTINE LIBSYS_ASCTIM (      ! Convert date and time
: 131      0407 1
: 132      0408 1      LENGTH,      ! Length returned
: 133      0409 1      RESULT_ADDR, ! Descriptor of user's string
: 134      0410 1      TIMEADR,     ! Time value to convert
: 135      0411 1      CONVFLG      ! Flags
: 136      0412 1
: 137      0413 1      ) : =
: 138      0414 1
: 139      0415 1 ++
: 140      0416 1      FUNCTIONAL DESCRIPTION:
: 141      0417 1          This routine calls $ASCTIM to convert binary date and time
: 142      0418 1          to an ASCII string, returning the resultant string using the
: 143      0419 1          semantics of the caller's string
: 144      0420 1          Parameter CONVFLG is presented to this routine by-reference
: 145      0421 1          and promoted to by-value for presentation to $ASCTIM.
: 146      0422 1
: 147      0423 1
: 148      0424 1      FORMAL PARAMETERS:
: 149      0425 1
: 150      0426 1          LENGTH.ww.r      Address of a word to receive the length
: 151      0427 1          of the output string returned.
: 152      0428 1          (Optional parameter)
: 153      0429 1
: 154      0430 1          RESULT_ADDR.wt.dx Address of the descriptor into which
: 155      0431 1          the resulting date and time is written.
: 156      0432 1
: 157      0433 1          TIMEADR.rqu.r     Address of the 64-bit time value to be
: 158      0434 1          converted. If no address is specified,
: 159      0435 1          or is specified as zero (the default),
: 160      0436 1          the current system date and time are
: 161      0437 1          returned. A positive time value
: 162      0438 1          represents an absolute time. A Negative
: 163      0439 1          time value represents a delta time. If
: 164      0440 1          a delta time is specified, it must be
: 165      0441 1          less than 10,000 days.
: 166      0442 1          (Optional parameter)
: 167      0443 1
: 168      0444 1          CONVFLG.rlu.r     Address of a
: 169      0445 1          Conversion indicator. A value of 1
: 170      0446 1          causes only the hour, minute, second,
: 171      0447 1          and hundredths of a second to be
: 172      0448 1          returned, depending on the length of the
: 173      0449 1          buffer. A value of 0 (the default)
: 174      0450 1          causes the full date and time to be
: 175      0451 1          returned, depending on the length of the
: 176      0452 1          buffer.
: 177      0453 1          (Optional parameter)
: 178      0454 1
: 179      0455 1      IMPLICIT INPUTS:
: 180      0456 1
: 181      0457 1          NONE
: 182      0458 1
: 183      0459 1      IMPLICIT OUTPUTS:
: 184      0460 1
: 185      0461 1          NONE
: 186      0462 1
```

```
187 0463 1 | COMPLETION CODES:
188 0464 1 |
189 0465 1 |     SSS_NORMAL      Procedure successfully completed
190 0466 1 | From LIB$COPY_R DX6
191 0467 1 |     LIB$STRTRD      Success, but source string truncated
192 0468 1 |     LIB$INSVIRMEM   Insufficient virtual memory
193 0469 1 |     LIB$INVSTRDES   Invalid string descriptor
194 0470 1 | From $ASCTIM
195 0471 1 |     SSS_IVTIME      The specified delta time is greater than or
196 0472 1 |                     equal to 10,000 days.
197 0473 1 |
198 0474 1 |
199 0475 1 | SIDE EFFECTS:
200 0476 1 |
201 0477 1 |     May request the current date and time from VAX/VMS.
202 0478 1 |
203 0479 1 | --
204 0480 1 |
205 0481 2 | BEGIN
206 0482 2 | BUILTIN
207 0483 2 |     NULLPARAMETER,
208 0484 2 |     ACTUALPARAMETER ;
209 0485 2 |
210 0486 2 | LOCAL
211 0487 2 |                                     ! output buffer
212 0488 2 |     COPY_STATUS,                ! Status from LIB$COPY_R DX6 call
213 0489 2 |     STR_STATUS,                 ! Status from $ASCTIM call
214 0490 2 |     DATE_TIME_BUF : VECTOR [LOCAL_BUF_SIZE, BYTE], ! Local buffer
215 0491 2 |     BUFFER_DESC : BLOCK [8, BYTE]; ! Descr for local
216 0492 2 |                                     ! buffer
217 0493 2 |
218 0494 2 | MAP
219 0495 2 |     RESULT_ADDR : REF BLOCK [BYTE] ,
220 0496 2 |     LENGTH: REF VECTOR [,WORD] ;
221 0497 2 |
222 0498 2 | +
223 0499 2 | Construct a static descriptor to receive the date and time from
224 0500 2 | VAX/VMS
225 0501 2 | -
226 0502 2 |     BUFFER_DESC [DSC$W_LENGTH] = LOCAL_BUF_SIZE;
227 0503 2 |     BUFFER_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
228 0504 2 |     BUFFER_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
229 0505 2 |     BUFFER_DESC [DSC$A_POINTER] = DATE_TIME_BUF;
230 0506 2 |
231 0507 2 | +
232 0508 2 | Call $ASCTIM using the appropriate combination of caller's
233 0509 2 | parameters, default parameters and internal variables.
234 0510 2 | -
235 0511 2 |
236 P 0512 2 |     STR_STATUS = $ASCTIM (
237 P 0513 2 |         TIMLEN = BUFFER_DESC, ! NOTE: length will get rewritten with
238 P 0514 2 |         TIMBUF = BUFFER_DESC, ! actual length returned
239 P 0515 2 |         TIMADR = (IF NULLPARAMETER(3)
240 P 0516 2 |             THEN 0
241 P 0517 2 |             ELSE ACTUALPARAMETER(3) ),
242 P 0518 2 |         CVTFLG = ( IF NULLPARAMETER(4)
243 P 0519 2 |             THEN 0
```

```

: 244      P 0520      2      ELSE .ACTUALPARAMETER(4) )
: 245      0521      2      ) ;
: 246      0522      2
: 247      0523      2
: 248      0524      2      + Copy internal buffer to caller's buffer, letting LIB$SCOPY_R_DX6
: 249      0525      2      worry about its semantics.
: 250      0526      2
: 251      0527      2      COPY_STATUS = LIB$SCOPY_R_DX6 ( .BUFFER_DESC [DSC$W_LENGTH],
: 252      0528      2      .BUFFER_DESC [DSC$A_POINTER],
: 253      0529      2      .RESULT_ADDR ) ;
: 254      0530      2
: 255      0531      2      +
: 256      0532      2      If caller supplied optional length parameter, update it.
: 257      0533      2      To do this accurately, we need to inspect the resulting descriptor
: 258      0534      2      as it appears after the copy. Get LIB$ANALYZE_SDESC_R2 to do it.
: 259      0535      2      No need to check LIB$ANALYZE_SDESC_R2 status. If RESULT_ADDR descr
: 260      0536      2      was bad, LIB$SCOPY_R_DX6 would have told us about it.
: 261      0537      2
: 262      0538      2      IF (NOT NULLPARAMETER(1))
: 263      0539      2      THEN
: 264      0540      2      BEGIN
: 265      0541      2      LOCAL
: 266      0542      2      RESULT_LEN,      ! No. of bytes delivered to caller
: 267      0543      2      RESULT_ADDRESS;      ! Address of 1st data byte of callers
: 268      0544      2      ! buffer
: 269      0545      2      IF .RESULT_ADDR [ DSC$B_CLASS] GTRU DSC$K_CLASS_D
: 270      0546      2      THEN      ! Must use general extraction
: 271      0547      2      BEGIN
: 272      0548      2      LIB$ANALYZE_SDESC_R2 ( .RESULT_ADDR ;
: 273      0549      2      RESULT_LEN, RESULT_ADDRESS );
: 274      0550      2      END
: 275      0551      2
: 276      0552      2      ELSE      ! Can fetch length directly
: 277      0553      2
: 278      0554      2      RESULT_LEN = .RESULT_ADDR [DSC$W_LENGTH] ;
: 279      0555      2
: 280      0556      2      LENGTH [0] = MIN ( .BUFFER_DESC [DSC$W_LENGTH], .RESULT_LEN) ;
: 281      0557      2      END;
: 282      0558      2      +
: 283      0559      2      At this point we have 2 statuses that we could return to caller.
: 284      0560      2      Determine which one is the "most meaningful".
: 285      0561      2
: 286      0562      2      +
: 287      0563      2      If original $ASCTIM call failed, tell him about that
: 288      0564      2      else give him the copy status.
: 289      0565      2
: 290      0566      2      IF NOT .STR_STATUS THEN RETURN (.STR_STATUS)
: 291      0567      2      ELSE RETURN (.COPY_STATUS) ;
: 292      0568      2
: 293      0569      1      END;      ! end of LIB$SYS_ASCTIM
```

```

      .TITLE  LIB$SYS_ASCTIM
      .IDENT  \1-008\

      .EXTRN  LIB$ANALYZE_SDESC_R2
      .EXTRN  LIB$SCOPY_R_DX6
```

				.EXTRN	SYSSASCTIM	
				.PSECT	_LIB\$CODE,NOWRT, SHR, PIC,2	
				.ENTRY	LIBSYS_ASCTIM, Save R2,R3,R4,R5,R6,R7,R8	: 0406
	5E		01FC 00000	SUBL2	#48, SP	: 0502
04	AE	010E002A	8F DD 00005	PUSHL	#17694762	: 0505
	04	08	AE 9E 0000B	MOVAB	DATE_TIME_BUF, BUFFER_DESC+4	: 0521
			6C 91 00010	CMPB	(AP), #4	
		10	05 1F 00013	BLSSU	1\$	
			AC D5 00015	TSTL	16(AP)	
			04 12 00018	BNEQ	2\$	
			7E D4 0001A 1\$:	CLRL	-(SP)	
			03 11 0001C	BRB	3\$	
		10	BC DD 0001E 2\$:	PUSHL	@16(AP)	
03			6C 91 00021 3\$:	CMPB	(AP), #3	
			05 1F 00024	BLSSU	4\$	
		0C	AC D5 00026	TSTL	12(AP)	
			04 12 00029	BNEQ	5\$	
			7E D4 0002B 4\$:	CLRL	-(SP)	
			03 11 0002D	BRB	6\$	
		0C	AC DD 0002F 5\$:	PUSHL	12(AP)	
		08	AE 9F 00032 6\$:	PUSHAB	BUFFER_DESC	
		0C	AE 9F 00035	PUSHAB	BUFFER_DESC	
00000000G	00		04 FB 00038	CALLS	#4, SYSSASCTIM	
	58		50 D0 0003F	MOVL	R0, STR_STATUS	: 0529
	57	08	AC D0 00042	MOVL	RESULT_ADDR, R7	: 0527
	52		57 D0 00046	MOVL	R7, R2	
	51	04	AE D0 00049	MOVL	BUFFER_DESC+4, R1	
	50		6E 3C 0004D	MOVZWL	BUFFER_DESC, R0	
		00000000G	00 16 00050	JSB	LIB\$COPY_R DX6	
	53		50 D0 00056	MOVL	R0, COPY_STATUS	: 0538
			6C 95 00059	TSTB	(AP)	
			28 13 0005B	BEQL	10\$	
		04	AC D5 0005D	TSTL	4(AP)	
			23 13 00060	BEQL	10\$	
	02	03	A7 91 00062	CMPB	3(R7), #2	: 0545
			0B 1B 00066	BLEQU	7\$	
	50		57 D0 00068	MOVL	R7, R0	: 0548
		00000000G	00 16 0006B	JSB	LIB\$ANALYZE_SDESC_R2	
			03 11 00071	BRB	8\$	: 0545
	51		67 3C 00073 7\$:	MOVZWL	(R7), RESULT_LEN	: 0554
	50		6E 3C 00076 8\$:	MOVZWL	BUFFER_DESC, R0	: 0556
	51		50 D1 00079	CMPL	R0, RESULT_LEN	
			03 15 0007C	BLEQ	9\$	
	50		51 D0 0007E	MOVL	RESULT_LEN, R0	
04	BC		50 B0 00081 9\$:	MOVW	R0, @LENGTH	
	04		58 E8 00085 10\$:	BLBS	STR_STATUS, 11\$	: 0566
	50		58 D0 00088	MOVL	STR_STATUS, R0	: 0567
			04 0008B	RET		
	50		53 D0 0008C 11\$:	MOVL	COPY_STATUS, R0	
			04 0008F	RET		: 0569

; Routine Size: 144 bytes, Routine Base: _LIB\$CODE + 0000

; 294 0570 1

LIB\$SYS_ASCTIM
1-008

D 10
16-Sep-1984 00:35:55 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:38:23 [LIBRTL.SRC]LIBASCTIM.B32;1

Page 8
(3)

: 295 0571 1 END
: 296 0572 1
: 297 0573 0 ELUDOM

! end of module LIB\$SYS_ASCTIM

PSECT SUMMARY

:
: Name Bytes Attributes
: _LIB\$CODE 144 NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(2)

Library Statistics

:
: File ----- Symbols ----- Pages Processing
: Total Loaded Percent Mapped Time
: _\$255\$DUA28:[SYSLIB]STARLET.L32;1 9776 10 0 581 00:00.7

COMMAND QUALIFIERS

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

: Size: 144 code + 0 data bytes
: Run Time: 00:04.6
: Elapsed Time: 00:21.2
: Lines/CPU Min: 7539
: Lexemes/CPU-Min: 31184
: Memory Used: 80 pages
: Compilation Complete

0203

AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY