


```

1 0001 0 %TITLE 'Process <SUBSTITUTE flag>'
2 0002 0 MODULE SUBST (
3 0003 0 IDENT = 'V04-000'
4 P 0004 0 %BLISS32[
5 P 0005 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
6 0006 0 ]
7 0007 0 ) =
8 0008 1 BEGIN
9 0009 1
10 0010 1 *****
11 0011 1 *
12 0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
13 0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *
31 0031 1 *****
32 0032 1
33 0033 1 ++
34 0034 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
35 0035 1
36 0036 1 ABSTRACT: Processes the <SUBSTITUTE flag>
37 0037 1
38 0038 1
39 0039 1 ENVIRONMENT: Transportable
40 0040 1
41 0041 1 AUTHOR: R.W.Friday CREATION DATE: May, 1978
42 0042 1

```

SUBST
V04-000

Process <SUBSTITUTE flag>
Revision History

F 13
16-Sep-1984 01:52:20
14-Sep-1984 13:08:17

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 (2) Page 2

```

: 44 0043 1 %SBTTL 'Revision History'
: 45 0044 1
: 46 0045 1   MODIFIED BY:
: 47 0046 1
: 48 0047 1   004   REM00004   Ray Marshall   30-Mar-1984
: 49 0048 1   Added conditionals for foreign language support. Provided
: 50 0049 1   German translations for the names of the months.
: 51 0050 1
: 52 0051 1   003   REM00003   Ray Marshall   07-Mar-1983
: 53 0052 1   Global edit of all modules. Updated module names, idents,
: 54 0053 1   copyright dates. Changed require files to BLISS library.
: 55 0054 1
: 56 0055 1   --

```

```

58 0056 1 %SBTTL 'Module Level Declarations'
59 0057 1
60 0058 1  TABLE OF CONTENTS:
61 0059 1
62 0060 1
63 0061 1  INCLUDE FILES:
64 0062 1
65 0063 1
66 0064 1  LIBRARY 'NXPORT:XPORT';           ! XPORT Library
67 0065 1  REQUIRE 'REQ:RNODEF';           ! RUNOFF variant definitions
68 0196 1
69 U 0197 1 %IF DSRPLUS %THEN
70 U 0198 1  LIBRARY 'REQ:DPDLLIB';       ! DSRPLUS BLISS Library
71 0199 1  %ELSE
72 0200 1  LIBRARY 'REQ:DSRLIB';       ! DSR BLISS Library
73 0201 1  %FI
74 0202 1
75 0203 1
76 0204 1  MACROS:
77 0205 1
78 0206 1  MACRO
79 M 0207 1      NAME_ENTRY(X) =
80 0208 1      CHSPTR( (UPLIT ( %STRING( %CHAR(%CHARCOUNT(X)), X) ) ) ) %;
81 0209 1
82 0210 1  EQUATED SYMBOLS:
83 0211 1
84 0212 1  LITERAL
85 0213 1      FIRST_INDEX = 1,           !Don't change this!!!!
86 0214 1      SUBST_DATE = 1,
87 0215 1      SUBST_TIME = 2,
88 0216 1      SUBST_YEAR = 3,
89 0217 1      SUBST_MONTH = 4,
90 0218 1      SUBST_DAY = 5,
91 0219 1      SUBST_HOURS = 6,
92 0220 1      SUBST_MINUTES = 7,
93 0221 1      SUBST_SECONDS = 8,
94 0222 1      LAST_INDEX = 8;           !Keep updated for additional symbols!!!!
95 0223 1
96 0224 1  OWN STORAGE:
97 0225 1
98 0226 1  BIND
99 0227 1      SUBST_NAMES = PLIT (NAME_ENTRY('DATE'),
100 0228 1      NAME_ENTRY('TIME'),
101 0229 1      NAME_ENTRY('YEAR'),
102 0230 1      NAME_ENTRY('MONTH'),
103 0231 1      NAME_ENTRY('DAY'),
104 0232 1      NAME_ENTRY('HOURS'),
105 0233 1      NAME_ENTRY('MINUTES'),
106 0234 1      NAME_ENTRY('SECONDS')
107 0235 1      ) : VECTOR;
108 0236 1
109 0237 1  EXTERNAL REFERENCES:
110 0238 1
111 0239 1
112 0240 1  EXTERNAL
113 0241 1      DATTIM : VECTOR,           !Date and time pieces.
114 0242 1      IRA : FIXED_STRING,

```

SUBST
V04-000

Process <SUBSTITUTE flag>
Module Level Declarations

H 13
16-Sep-1984 01:52:20
14-Sep-1984 13:08:17

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[RUNOFF.SRC]SUBST.BLI;1 (3)

TIT
V04

```
.. 115 0243 1 KHAR;  
.. 116 0244 1  
.. 117 0245 1 EXTERNAL LITERAL  
.. 118 0246 1 RNFNA,  
.. 119 0247 1 RNFUDS;  
.. 120 0248 1  
.. 121 0249 1 EXTERNAL ROUTINE  
.. 122 0250 1 CNVDAT,  
.. 123 0251 1 CNVTIM,  
.. 124 0252 1 ENDCHR,  
.. 125 0253 1 ERMS,  
.. 126 0254 1 GSLU,  
.. 127 0255 1 PACBAS;
```

!Error messages

SUBST
VV -000

Process <SUBSTITUTE flag>
SUBST --

I 13
16-Sep-1984 01:52:20
14-Sep-1984 13:08:17

VAX-11 Bliss-32 V4.0-742 Page 5
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 (4)

TIT
V04

```
: 129 0256 1 %SBTTL 'SUBST --'
: 130 0257 1 GLOBAL ROUTINE subst : NOVALUE =
: 131 0258 1
: 132 0259 1 |++
: 133 0260 1 |FUNCTIONAL DESCRIPTION:
: 134 0261 1 |
: 135 0262 1 |     See ABSTRACT, above.
: 136 0263 1 |
: 137 0264 1 | FORMAL PARAMETERS:     None
: 138 0265 1 |
: 139 0266 1 | IMPLICIT INPUTS:
: 140 0267 1 |
: 141 0268 1 |     KHAR must be the current <SUBSTITUTE flag>, the first of
: 142 0269 1 |     the (presumed) pair.
: 143 0270 1 |
: 144 0271 1 | IMPLICIT OUTPUTS:     None
: 145 0272 1 |
: 146 0273 1 | ROUTINE VALUE:
: 147 0274 1 | COMPLETION CODES:     None
: 148 0275 1 |
: 149 0276 1 | SIDE EFFECTS:         None
: 150 0277 1 |
: 151 0278 1 | --
: 152 0279 1 |
: 153 0280 2 | BEGIN
: 154 0281 2 |
: 155 0282 2 | LOCAL
: 156 0283 2 |     DUMMY_FS : FIXED_STRING[50],
: 157 0284 2 |     GSLU_RESULT,
: 158 0285 2 |     WORK_AREA : VECTOR [CH$ALLOCATION(50)],
: 159 0286 2 |     WORK_PTR,
: 160 0287 2 |     WORK_LENGTH,
: 161 0288 2 |     HOLD_FLAG,
: 162 0289 2 |     SUBST_INDEX;
: 163 0290 2 |
: 164 0291 2 | HOLD_FLAG=.KHAR;           !Remember current <SUBSTITUTE flag>
: 165 0292 2 |
: 166 0293 2 | KCNS ();                   !Get next character.
: 167 0294 2 |
: 168 0295 2 | !Make sure user has specified doubled <SUBSTITUTE flag>s.
: 169 0296 2 | !\Single <SUBSTITUTE flag>s are for extensions, i.e., macro facility).
: 170 0297 2 | IF .KHAR NEQ .HOLD_FLAG THEN !Check for error case first.
: 171 0298 2 |     BEGIN                   !Not doubled <SUBSTITUTE flag>s
: 172 0299 3 |     ERMS ( RNFFNA
: 173 0300 3 |         .FS_START (IRA)
: 174 0301 3 |         ,CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)) );
: 175 0302 3 |     RETURN;
: 176 0303 3 |     END;
: 177 0304 2 |
: 178 0305 2 | !Position to character immediately after the second <SUBSTITUTE flag>
: 179 0306 2 | KCNS ();
: 180 0307 2 |
: 181 0308 2 | !Fudge up DUMMY_FS to look like a FIXED_STRING;
: 182 0309 2 | FS_MAXSIZE (DUMMY_FS) = 50;
: 183 0310 2 | FS_INIT (DUMMY_FS);
: 184 0311 2 |
: 185 0312 2 | !Attempt to pick up the word DATE or TIME.
```

```
186 0313 2    GSLU_RESULT = GSLU (IRA, DUMMY_FS);
187 0314 2
188 0315 2    !Make sure something was found.
189 0316 2    IF .GSLU_RESULT NEQ GSLU_NORMAL THEN      ! If nothing was found,
190 0317 2    BEGIN                                ! something's wrong.
191 0318 2    ERMS ( RNFFNA
192 0319 2    ..FS_START (IRA)
193 0320 2    ,CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
194 0321 2    RETURN;
195 0322 2    END;
196 0323 2
197 0324 2    ! Attempt to identify the word. The result of this block is either a RETURN
198 0325 2    ! if the word was not identified, or an indicator as to which routine to
199 0326 2    ! call to translate the string into something displayable.
200 0327 2    BEGIN
201 0328 2    LOCAL
202 0329 2    PTR;
203 0330 2
204 0331 2    !Assume the word won't be identified.
205 0332 2    SUBST_INDEX = 0;
206 0333 2
207 0334 2    INCR I FROM 0 TO (.SUBST_NAMES[-1]-1) DO
208 0335 2    BEGIN
209 0336 2    IF CH$EQL ( .FS_LENGTH(DUMMY_FS)
210 0337 2    ..FS_START(DUMMY_FS)
211 0338 2    ,CH$RCHAR(.SUBST_NAMES[.I])
212 0339 2    ,CH$PLUS(.SUBST_NAMES[.I],1) ) THEN
213 0340 2    BEGIN                                !Keyword identified!!!!
214 0341 2    SUBST_INDEX = .I + 1;                !Remember which word it was.
215 0342 2    EXITLOOP
216 0343 2    END
217 0344 2    END;
218 0345 2
219 0346 2    !See if the keyword was found.
220 0347 2    IF .SUBST_INDEX EQL 0 THEN
221 0348 2    BEGIN                                !Word was not identified
222 0349 2    ERMS (RNFUDS, .FS_START (IRA), CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
223 0350 2    RETURN
224 0351 2    END;
225 0352 2    END;                                !End of identification of word.
226 0353 2
227 0354 2    ! Now get the character string that the user has requested. The result of
228 0355 2    ! this CASE statement is that WORK_AREA will contain the text, text, and
229 0356 2    ! WORK_LENGTH will be the number of characters.
230 0357 2
231 0358 2    WORK_PTR = CH$PTR(WORK_AREA);
232 0359 2    CASE .SUBST_INDEX FROM FIRST_INDEX TO LAST_INDEX OF
233 0360 2    SET
234 0361 2    [SUBST_DATE]:
235 0362 2    BEGIN
236 0363 2    CNVDAT(DATTIM, WORK_PTR, WORK_LENGTH);
237 0364 2    WORK_PTR = CH$PTR(WORK_AREA)
238 0365 2    END;
239 0366 2
240 0367 2    [SUBST_TIME]:
241 0368 2    BEGIN
242 0369 2    CNVTIM(DATTIM, WORK_PTR, WORK_LENGTH);
```

```

243      0370      WORK_PTR = CH$PTR(WORK_AREA)
244      0371      END;
245      0372
246      0373      [SUBST_YEAR]:
247      0374      BEGIN
248      0375      WORK_LENGTH = PACBAS (.DATTIM [0], WORK_PTR, 10);
249      0376      WORK_PTR = CH$PTR(WORK_AREA);
250      0377      END;
251      0378
252      0379      [SUBST_MONTH]:
253      0380      BEGIN
254      0381      BIND
255      0382      MONTHS = UPLIT (
256      U 0383      %IF german %THEN
257      U U 0384      CH$PTR(UPLIT(%STRING('Jaenner'))),
258      U U 0385      CH$PTR(UPLIT(%STRING('Februar'))),
259      U U 0386      CH$PTR(UPLIT(%STRING('Maerz'))),
260      U U 0387      CH$PTR(UPLIT(%STRING('April'))),
261      U U 0388      CH$PTR(UPLIT(%STRING('Mai'))),
262      U U 0389      CH$PTR(UPLIT(%STRING('Juni'))),
263      U U 0390      CH$PTR(UPLIT(%STRING('Juli'))),
264      U U 0391      CH$PTR(UPLIT(%STRING('August'))),
265      U U 0392      CH$PTR(UPLIT(%STRING('September'))),
266      U U 0393      CH$PTR(UPLIT(%STRING('Oktober'))),
267      U U 0394      CH$PTR(UPLIT(%STRING('November'))),
268      U 0395      CH$PTR(UPLIT(%STRING('Dezember'))),
269      U 0396      %ELSE
270      U 0397      %IF french %THEN
271      U U 0398      CH$PTR(UPLIT(%STRING('January'))),
272      U U 0399      CH$PTR(UPLIT(%STRING('February'))),
273      U U 0400      CH$PTR(UPLIT(%STRING('March'))),
274      U U 0401      CH$PTR(UPLIT(%STRING('April'))),
275      U U 0402      CH$PTR(UPLIT(%STRING('May'))),
276      U U 0403      CH$PTR(UPLIT(%STRING('June'))),
277      U U 0404      CH$PTR(UPLIT(%STRING('July'))),
278      U U 0405      CH$PTR(UPLIT(%STRING('August'))),
279      U U 0406      CH$PTR(UPLIT(%STRING('September'))),
280      U U 0407      CH$PTR(UPLIT(%STRING('October'))),
281      U U 0408      CH$PTR(UPLIT(%STRING('November'))),
282      U 0409      CH$PTR(UPLIT(%STRING('December'))),
283      U 0410      %ELSE
284      U 0411      %IF italian %THEN
285      U U 0412      CH$PTR(UPLIT(%STRING('January'))),
286      U U 0413      CH$PTR(UPLIT(%STRING('February'))),
287      U U 0414      CH$PTR(UPLIT(%STRING('March'))),
288      U U 0415      CH$PTR(UPLIT(%STRING('April'))),
289      U U 0416      CH$PTR(UPLIT(%STRING('May'))),
290      U U 0417      CH$PTR(UPLIT(%STRING('June'))),
291      U U 0418      CH$PTR(UPLIT(%STRING('July'))),
292      U U 0419      CH$PTR(UPLIT(%STRING('August'))),
293      U U 0420      CH$PTR(UPLIT(%STRING('September'))),
294      U U 0421      CH$PTR(UPLIT(%STRING('October'))),
295      U U 0422      CH$PTR(UPLIT(%STRING('November'))),
296      U 0423      CH$PTR(UPLIT(%STRING('December'))),
297      U 0424      %ELSE
298      U 0425      CH$PTR(UPLIT(%STRING('January'))),
299      U 0426      CH$PTR(UPLIT(%STRING('February'))),

```

```

300 0427 CH$PTR(UPLIT(%STRING('March'))),
301 0428 CH$PTR(UPLIT(%STRING('April'))),
302 0429 CH$PTR(UPLIT(%STRING('May'))),
303 0430 CH$PTR(UPLIT(%STRING('June'))),
304 0431 CH$PTR(UPLIT(%STRING('July'))),
305 0432 CH$PTR(UPLIT(%STRING('August'))),
306 0433 CH$PTR(UPLIT(%STRING('September'))),
307 0434 CH$PTR(UPLIT(%STRING('October'))),
308 0435 CH$PTR(UPLIT(%STRING('November'))),
309 0436 CH$PTR(UPLIT(%STRING('December'))),
310 0437 %FI %FI %FI
311 0438 ) : VECTOR;
312 0439
313 0440 OWN MONTHL : VECTOR [12] INITIAL (
314 U 0441 %IF german %THEN 7, 7, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
315 U 0442
316 U 0443 %ELSE
317 U 0444 %IF french %THEN 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
318 U 0445
319 U 0446 %ELSE
320 U 0447 %IF italian %THEN 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
321 U 0448
322 0449 %ELSE 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
323 0450
324 0451 %FI %FI %FI
325 0452 );
326 0453 WORK_PTR = .MONTHS [.DATTIM[1] - 1];
327 0454 WORK_LENGTH = .MONTHL [.DATTIM[1] - 1];
328 0455 END;
329 0456
330 0457 [SUBST_DAY]:
331 0458 BEGIN
332 0459 WORK_LENGTH = PACBAS (.DATTIM [2], WORK_PTR, 10);
333 0460 WORK_PTR = CH$PTR(WORK_AREA);
334 0461 END;
335 0462
336 0463 [SUBST_HOURS]:
337 0464 BEGIN
338 0465 !Assume result will be less than two digits, and prepad with a '0'.
339 0466 CH$WCHAR(%'0', CH$PTR(WORK_AREA));
340 0467 WORK_PTR = CH$PLUS(.WORK_PTR, 1);
341 0468 WORK_LENGTH = PACBAS (.DATTIM [3], WORK_PTR, 10);
342 0469 WORK_PTR = CH$PTR(WORK_AREA);
343 0470 !If the result was at least two digits, forget the leading zero.
344 0471 IF .WORK_LENGTH EQL 1 THEN
345 0472 WORK_LENGTH = .WORK_LENGTH + 1
346 0473 ELSE
347 0474 WORK_PTR = CH$PLUS(.WORK_PTR, 1);
348 0475 END;
349 0476
350 0477 [SUBST_MINUTES]:
351 0478 BEGIN
352 0479 !Assume result will be less than two digits, and prepad with a '0'.
353 0480 CH$WCHAR(%'0', CH$PTR(WORK_AREA));
354 0481 WORK_PTR = CH$PLUS(.WORK_PTR, 1);
355 0482 WORK_LENGTH = PACBAS (.DATTIM [4], WORK_PTR, 10);
356 0483 WORK_PTR = CH$PTR(WORK_AREA);

```


				37	11	0008C		BRB	10\$			
				55	D4	0008E	6\$:	CLRL	SUBST_INDEX		0332	
			54	01	CE	00090		MNEGL	#1, I		0337	
				18	11	00093		BRB	8\$			
			50	6944	D0	00095	7\$:	MOVL	SUBST_NAMES[I], R0		0338	
			51	60	9A	00099		MOVZBL	(R0), R1			
51	00	3C	BE	48	AE	2D	0009C	CMPC5	DUMMY_FS+12, @DUMMY_FS, #0, R1, 1(R0)		0339	
				01	A0							
				06	12	000A5		BNEQ	8\$			
			55	01	A3	9E	000A7	MOVAB	1(R4), SUBST_INDEX		0341	
				05	11	000AB		BRB	9\$		0340	
	E3		54	FC	A9	F2	000AD	8\$:	AOBLSS	SUBST_NAMES-4, I, 7\$	0335	
				55	D5	000B2	9\$:	TSTL	SUBST_INDEX		0347	
				17	12	000B4		BNEQ	11\$			
	7E	F8	A6	F4	A6	C3	000B6	SUBL3	IRA, IRA+4, -(SP)		0349	
				F4	A6	DD	000BC	PUSHL	IRA			
				00000000G	EF	00000000G	8F	DD	000BF			
				03	FB	000C5	10\$:	CALLS	#3, ERMS			
				04	000CC			RET			0348	
		04	AE	08	AE	9E	000CD	11\$:	MOVAB	WORK_AREA, WORK_PTR	0358	
	07		01	55	CF	000D2		CASEL	SUBST_INDEX, #1, #7		0359	
0039	0030	0020		0010		000D6	12\$:	.WORD				
0094	0073	0062		004D		000DE						
									13\$-12\$,-			
									14\$-12\$,-			
									15\$-12\$,-			
									16\$-12\$,-			
									17\$-12\$,-			
									20\$-12\$,-			
									21\$-12\$,-			
									23\$-12\$			
				08	5E	DD	000E6	13\$:	PUSHL	SP	0363	
				08	AE	9F	000E8		PUSHAB	WORK_PTR		
					57	DD	000EB		PUSHL	R7		
			00000000G	EF	03	FB	000ED		CALLS	#3, CNVDAT		
					3B	11	000F4		BRB	19\$	0364	
					5E	DD	000F6	14\$:	PUSHL	SP	0369	
				08	AE	9F	000F8		PUSHAB	WORK_PTR		
					57	DD	000FB		PUSHL	R7		
			00000000G	EF	03	FB	000FD		CALLS	#3, CNVTIM		
					2B	11	00104		BRB	19\$	0370	
					0A	DD	00106	15\$:	PUSHL	#10	0375	
				08	AE	9F	00108		PUSHAB	WORK_PTR		
					67	DD	0010B		PUSHL	DATTIM		
					1C	11	0010D		BRB	18\$		
			04	50	04	A7	D0	0010F	16\$:	MOVL	DATTIM+4, R0	0453
				AE	74	A940	D0	00113		MOVL	MONTHS-4[R0], WORK_PTR	
			6E	00000000	EF40	D0	00119		MOVL	MONTHL-4[R0], WORK_LENGTH	0454	
					6D	11	00121		BRB	26\$	0359	
					0A	DD	00123	17\$:	PUSHL	#10	0459	
				08	AE	9F	00125		PUSHAB	WORK_PTR		
				08	A7	DD	00128		PUSHL	DATTIM+8		
					03	FB	0012B	18\$:	CALLS	#3, PACBAS		
					50	D0	0012E		MOVL	R0, WORK_LENGTH		
			04	6A					MOVAB	WORK_AREA, WORK_PTR	0460	
				6E					BRB	26\$	0359	
			04	AE	08	AE	9E	00131	19\$:	MOVAB	WORK_AREA, WORK_PTR	0466
									BRB	26\$	0467	
			08	AE	04	30	90	00138	20\$:	MOVAB	#48, WORK_AREA	0468
									INCL	WORK_PTR		
					0A	DD	0013F		PUSHL	#10		

		08	AE	9F	00141		PUSHAB	WORK_PTR		
		0C	A7	DD	00144		PUSHL	DATTIM+12		
			0F	11	00147		BRB	22\$		
	08	AE	30	90	00149	21\$:	MOVB	#48, WORK_AREA		0480
			04	AE	D6	0014D	INCL	WORK_PTR		0481
			0A	DD	00150		PUSHL	#10		0482
		08	AE	9F	00152		PUSHAB	WORK_PTR		
		10	A7	DD	00155		PUSHL	DATTIM+16		
	6A		03	FB	00158	22\$:	CALLS	#3, PACBAS		
	6E		50	D0	0015B		MOVL	R0, WORK_LENGTH		
	04	AE	08	AE	9E	0015E	MOVAB	WORK_AREA, WORK_PTR		0483
		01	6E	D1	00163		CMPL	WORK_LENGTH, #1		0485
			21	13	00166		BEQL	24\$		
			23	11	00168		BRB	25\$		0488
	08	AE	30	90	0016A	23\$:	MOVB	#48, WORK_AREA		0494
			04	AE	D6	0016E	INCL	WORK_PTR		0495
			0A	DD	00171		PUSHL	#10		0496
		08	AE	9F	00173		PUSHAB	WORK_PTR		
		14	A7	DD	00176		PUSHL	DATTIM+20		
	6A		03	FB	00179		CALLS	#3, PACBAS		
	6E		50	D0	0017C		MOVL	R0, WORK_LENGTH		
	04	AE	08	AE	9E	0017F	MOVAB	WORK_AREA, WORK_PTR		0497
		01	6E	D1	00184		CMPL	WORK_LENGTH, #1		0499
			04	12	00187		BNEQ	25\$		
			6E	D6	00189	24\$:	INCL	WORK_LENGTH		0500
			03	11	0018B		BRB	26\$		
		04	AE	D6	0018D	25\$:	INCL	WORK_PTR		0502
			52	D4	00190	26\$:	CLRL	I		0507
			0E	11	00192		BRB	28\$		
		7E	04	BE	9A	00194	MOVZBL	@WORK_PTR, -(SP)		0508
		08	AE	D6	00198		INCL	WORK_PTR		
	00000000G	EF	01	FB	0019B		CALLS	#1, ENDCHR		
EE		52	6E	F3	001A2	28\$:	AOBLEQ	WORK_LENGTH, I, 27\$		
			04	001A6			RET			0512

: Routine Size: 423 bytes, Routine Base: \$CODE\$ + 0000

: 386 0513 1
: 387 0514 1 END
: 388 0515 0 ELUDOM

!End of module

PSECT SUMMARY

Name	Bytes	Attributes
\$PLITS	232	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$OWNS	48	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$CODE\$	423	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

SUBST
V04-000

Process <SUBSTITUTE flag>
SUBST --

D 14
16-Sep-1984 01:52:20
14-Sep-1984 13:08:17

VAX-11 Bliss-32 V4.0-742
DISK\$VMMASTER:[RUNOFF.SRC]SUBST.BLI;1 (4)

TIT
V04

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]XPORT.L32;1	590	0	0	252	00:00.2
_\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	13	1	86	00:00.2

000

000

COMMAND QUALIFIERS

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

000

000

: Size: 423 code + 280 data bytes
: Run Time: 00:10.8
: Elapsed Time: 00.33.1
: Lines/CPU Min: 2850
: Lexemes/CPU-Min: 21154
: Memory Used: 131 pages
: Compilation Complete

000

000

