


```

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
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS
LLLLLLLLLLLLLL  IIIIII               SSSSSSSS

```

SECRET
.....


```
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 Page 2 (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, ids, and
:	54	0053	1	copyright dates. Changed require files to BLISS library.
:	55	0054	1	
:	56	0055	1	--

SUBST
V04-000

Process <SUBSTITUTE flag>
Module Level Declarations

G 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 Page 3 (3)

```

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:DPLLIB'; ! 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\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1 Page 4 (3)

:	115	0243	1	KHAR;
:	116	0244	1	
:	117	0245	1	EXTERNAL LITERAL
:	118	0246	1	RNFFNA,
:	119	0247	1	RNFUDS;
:	120	0248	1	
:	121	0249	1	EXTERNAL ROUTINE
:	122	0250	1	CNV DAT,
:	123	0251	1	CNV TIM,
:	124	0252	1	END CHR,
:	125	0253	1	ERMS,
:	126	0254	1	GSLU,
:	127	0255	1	PACBAS;

!Error messages

SUBST
V04-000

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
DISK\$VMSMASTER:[RUNOFF.SRC]SUBST.BLI;1
Page 5
(4)

```
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$ALLOCATION(50)],
159 0286 2 WORK_PTR,
160 0287 2 WORK_LENGTH,
161 0288 2 HOLD_FLAG,
162 0289 2 SUBST_INDEX;
163 0290 2
164 0291 2 HOLD_FLAG=.KHAR; !Remember current <SUBSTITUTE flag>
165 0292 2
166 0293 2 KCNS (); !Get next character.
167 0294 2
168 0295 2 !Make sure user has specified doubled <SUBSTITUTE flag>s.
169 0296 2 !(Single <SUBSTITUTE flag>s are for extensions, i.e., macro facility).
170 0297 2 IF .KHAR NEQ .HOLD_FLAG THEN !Check for error case first.
171 0298 2 BEGIN !Not doubled <SUBSTITUTE flag>s
172 0299 2 ERMS ( RNFFNA
173 0300 2 .FS_START (IRA)
174 0301 2 ,CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)) );
175 0302 2
176 0303 2 RETURN;
177 0304 2 END;
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      GSLU_RESULT = GSLU (IRA, DUMMY_FS);
187      0314
188      0315      !Make sure something was found.
189      0316      IF .GSLU_RESULT NEQ GSLU_NORMAL THEN           ! If nothing was found,
190      0317      BEGIN                                           ! something's wrong.
191      0318          ERMS ( RNFFNA
192      0319              ,.FS_START (IRA)
193      0320              ,CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
194      0321      RETURN;
195      0322      END;
196      0323
197      0324      ! Attempt to identify the word. The result of this block is either a RETURN
198      0325      ! if the word was not identified, or an indicator as to which routine to
199      0326      ! call to translate the string into something displayable.
200      0327      BEGIN
201      0328      LOCAL
202      0329      PTR;
203      0330
204      0331      !Assume the word won't be identified.
205      0332      SUBST_INDEX = 0;
206      0333
207      0334      INCR I FROM 0 TO (.SUBST_NAMES[-1]-1) DO
208      0335      BEGIN
209      0336      IF CH$EQL ( .FS_LENGTH(DUMMY_FS)
210      0337          ,.FS_START(DUMMY_FS)
211      0338          ,CH$RCHAR(.SUBST_NAMES[I])
212      0339          ,CH$PLUS(.SUBST_NAMES[I],1) ) THEN
213      0340          BEGIN                                     !Keyword identified!!!!
214      0341          SUBST_INDEX = .I + 1;                     !Remember which word it was.
215      0342          EXITLOOP
216      0343          END
217      0344      END;
218      0345
219      0346      !See if the keyword was found.
220      0347      IF .SUBST_INDEX EQL 0 THEN
221      0348      BEGIN                                           !Word was not identified
222      0349      ERMS (RNFUDS, .FS_START (IRA), CH$DIFF (.FS_NEXT (IRA), .FS_START (IRA)));
223      0350      RETURN
224      0351      END;
225      0352      END;                                           !End of identification of word.
226      0353
227      0354      ! Now get the character string that the user has requested. The result of
228      0355      ! this CASE statement is that WORK_AREA will contain the text, text, and
229      0356      ! WORK_LENGTH will be the number of characters.
230      0357
231      0358      WORK_PTR = CH$PTR(WORK_AREA);
232      0359      CASE ".SUBST_INDEX FROM FIRST_INDEX TO LAST_INDEX OF
233      0360      SET
234      0361      [SUBST_DATE]:
235      0362      BEGIN
236      0363      CNVDAT(DATTIM, WORK_PTR, WORK_LENGTH);
237      0364      WORK_PTR = CH$PTR(WORK_AREA)
238      0365      END;
239      0366
240      0367      [SUBST_TIME]:
241      0368      BEGIN
242      0369      CNVTIM(DATTIM, WORK_PTR, WORK_LENGTH);

```


SUBST
V04-000

Process <SUBSTITUTE flag>
SUBST --

K 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 Page 7 (4)

```
.. 243      0370      3      WORK_PTR = CH$PTR(WORK_AREA)
.. 244      0371      3      END;
.. 245      0372      3
.. 246      0373      3      [SUBST_YEAR]:
.. 247      0374      3      BEGIN
.. 248      0375      3      WORK_LENGTH = PACBAS (.DATIM [0], WORK_PTR, 10);
.. 249      0376      3      WORK_PTR = CH$PTR(WORK_AREA);
.. 250      0377      3      END;
.. 251      0378      3
.. 252      0379      3      [SUBST_MONTH]:
.. 253      0380      3      BEGIN
.. 254      0381      3      BIND
.. 255      0382      3      MONTHS = UPLIT (
.. 256      U 0383      3      %IF german %THEN
.. 257      UU 0384      3      CH$PTR(UPLIT(%STRING('Jaenner'))),
.. 258      UU 0385      3      CH$PTR(UPLIT(%STRING('Februar'))),
.. 259      UU 0386      3      CH$PTR(UPLIT(%STRING('Maerz'))),
.. 260      UU 0387      3      CH$PTR(UPLIT(%STRING('April'))),
.. 261      UU 0388      3      CH$PTR(UPLIT(%STRING('Mai'))),
.. 262      UU 0389      3      CH$PTR(UPLIT(%STRING('Juni'))),
.. 263      UU 0390      3      CH$PTR(UPLIT(%STRING('Juli'))),
.. 264      UU 0391      3      CH$PTR(UPLIT(%STRING('August'))),
.. 265      UU 0392      3      CH$PTR(UPLIT(%STRING('September'))),
.. 266      UU 0393      3      CH$PTR(UPLIT(%STRING('Oktober'))),
.. 267      UU 0394      3      CH$PTR(UPLIT(%STRING('November'))),
.. 268      U 0395      3      CH$PTR(UPLIT(%STRING('Dezember'))),
.. 269      0396      3      %ELSE
.. 270      U 0397      3      %IF french %THEN
.. 271      UU 0398      3      CH$PTR(UPLIT(%STRING('January'))),
.. 272      UU 0399      3      CH$PTR(UPLIT(%STRING('February'))),
.. 273      UU 0400      3      CH$PTR(UPLIT(%STRING('March'))),
.. 274      UU 0401      3      CH$PTR(UPLIT(%STRING('April'))),
.. 275      UU 0402      3      CH$PTR(UPLIT(%STRING('May'))),
.. 276      UU 0403      3      CH$PTR(UPLIT(%STRING('June'))),
.. 277      UU 0404      3      CH$PTR(UPLIT(%STRING('July'))),
.. 278      UU 0405      3      CH$PTR(UPLIT(%STRING('August'))),
.. 279      UU 0406      3      CH$PTR(UPLIT(%STRING('September'))),
.. 280      UU 0407      3      CH$PTR(UPLIT(%STRING('October'))),
.. 281      UU 0408      3      CH$PTR(UPLIT(%STRING('November'))),
.. 282      U 0409      3      CH$PTR(UPLIT(%STRING('December'))),
.. 283      0410      3      %ELSE
.. 284      U 0411      3      %IF italian %THEN
.. 285      UU 0412      3      CH$PTR(UPLIT(%STRING('January'))),
.. 286      UU 0413      3      CH$PTR(UPLIT(%STRING('February'))),
.. 287      UU 0414      3      CH$PTR(UPLIT(%STRING('March'))),
.. 288      UU 0415      3      CH$PTR(UPLIT(%STRING('April'))),
.. 289      UU 0416      3      CH$PTR(UPLIT(%STRING('May'))),
.. 290      UU 0417      3      CH$PTR(UPLIT(%STRING('June'))),
.. 291      UU 0418      3      CH$PTR(UPLIT(%STRING('July'))),
.. 292      UU 0419      3      CH$PTR(UPLIT(%STRING('August'))),
.. 293      UU 0420      3      CH$PTR(UPLIT(%STRING('September'))),
.. 294      UU 0421      3      CH$PTR(UPLIT(%STRING('October'))),
.. 295      UU 0422      3      CH$PTR(UPLIT(%STRING('November'))),
.. 296      U 0423      3      CH$PTR(UPLIT(%STRING('December'))),
.. 297      0424      3      %ELSE
.. 298      0425      3      CH$PTR(UPLIT(%STRING('January'))),
.. 299      0426      3      CH$PTR(UPLIT(%STRING('February'))),
```

SUBST
V04-000

Process <SUBSTITUTE flag>
SUBST --

L 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 Page 8
(4)

TIT
V04

```

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
314      0441      MONTHL : VECTOR [12] INITIAL (
315      U 0442      %IF german %THEN
316      U 0443      7, 7, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
317      U 0444      %ELSE
318      U 0445      %IF french %THEN
319      U 0446      7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
320      U 0447      %ELSE
321      U 0448      %IF italian %THEN
322      U 0449      7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
323      0450      %ELSE
324      0451      7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
325      0452      %FI %FI %FI
326      0453      );
327      0454      WORK_PTR = .MONTHS [.DATTIM[1] - 1];
328      0455      WORK_LENGTH = .MONTHL [.DATTIM[1] - 1];
329      0456      END;
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);
```


SUBST
V04-000

Process <SUBSTITUTE flag>
SUBST --

M 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
Page 9
(4)

```

: 357      0484      3      !If the result was at least two digits, forget the leading zero.
: 358      0485      3      IF .WORK_LENGTH EQL 1 THEN
: 359      0486      3      WORK_LENGTH = .WORK_LENGTH + 1
: 360      0487      3      ELSE
: 361      0488      3      WORK_PTR = CH$PLUS(.WORK_PTR, 1);
: 362      0489      3      END;
: 363      0490      3
: 364      0491      3      [SUBST_SECONDS]:
: 365      0492      3      BEGIN
: 366      0493      3      !Assume result will be less than two digits, and prepad with a '0'.
: 367      0494      3      CH$WCHAR(%C'0', CH$PTR(WORK_AREA));
: 368      0495      3      WORK_PTR = CH$PLUS(.WORK_PTR, 1);
: 369      0496      3      WORK_LENGTH = PACBAS (.DATTIM [5], WORK_PTR, 10);
: 370      0497      3      WORK_PTR = CH$PTR(WORK_AREA);
: 371      0498      3      !If the result was at least two digits, forget the leading zero.
: 372      0499      3      IF .WORK_LENGTH EQL 1 THEN
: 373      0500      3      WORK_LENGTH = .WORK_LENGTH + 1
: 374      0501      3      ELSE
: 375      0502      3      WORK_PTR = CH$PLUS(.WORK_PTR, 1);
: 376      0503      3      END;
: 377      0504      3      TES;
: 378      0505      3
: 379      0506      3      !Now feed these characters into the document.
: 380      0507      3      INCR I FROM 1 TO .WORK_LENGTH DO
: 381      0508      3      ENDCHR ( CH$RCHAR_A (WORK_PTR) );
: 382      0509      3
: 383      0510      3      !That's all there is, so just return.
: 384      0511      3      RETURN;
: 385      0512      3      END;

```

!End of SUBST

```

                                .TITLE SUBST Process <SUBSTITUTE flag>
                                .IDENT  \V04-000\
                                .PSECT  $SPLITS,NOWRT,NOEXE,2
00 00 00 45 54 41 44 04 00000 P.AAB: .ASCII <4>\DATE\<0><0><0>
00 00 00 45 4D 49 54 04 00008 P.AAC: .ASCII <4>\TIME\<0><0><0>
00 00 00 52 41 45 59 04 00010 P.AAD: .ASCII <4>\YEAR\<0><0><0>
00 00 48 54 4E 4F 4D 05 00018 P.AAE: .ASCII <5>\MONTH\<0><0>
                                59 41 44 03 00020 P.AAF: .ASCII <3>\DAY\
00 00 53 52 55 4F 48 05 00024 P.AAG: .ASCII <5>\HOURS\<0><0>
53 45 54 55 4E 49 4D 07 0002C P.AAH: .ASCII <7>\MINUTES\
53 44 4E 4F 43 45 53 07 00034 P.AAI: .ASCII <7>\SECONDS\
                                00000008 0003C .LONG 8
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 00040 P.AAA: .ADDRESS P.AAB, P.AAC, P.AAD, P.AAE, P.AAF, -
                                00000000' 00000000' 00058 P.AAG, P.AAH, P.AAI
00 79 72 61 75 6E 61 4A 00060 P.AAK: .ASCII \January\<0>
79 72 61 75 72 62 65 46 00068 P.AAL: .ASCII \February\
00 00 00 68 63 72 61 4D 00070 P.AAM: .ASCII \March\<0><0><0>
00 00 00 6C 69 72 70 41 00078 P.AAN: .ASCII \April\<0><0><0>
                                00 79 61 4D 00080 P.AAO: .ASCII \May\<0>
                                65 6E 75 4A 00084 P.AAP: .ASCII \June\
                                79 6C 75 4A 00088 P.AAQ: .ASCII \July\
00 00 74 73 75 67 75 41 0008C P.AAR: .ASCII \August\<0><0>
00 00 00 72 65 62 6D 65 74 70 65 53 00094 P.AAS: .ASCII \September\<0><0><0>
00 72 65 62 6F 74 63 4F 000A0 P.AAT: .ASCII \October\<0>

```


SUBST
V04-000

Process <SUBSTITUTE flag>
SUBST --

N 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

Page 10
(4)

```

      72 65 62 6D 65 76 6F 4E 000A8 P.AAU: .ASCII \November\
      72 65 62 6D 65 63 65 44 000B0 P.AAV: .ASCII \December\
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000B8 P.AAJ: .ADDRESS P.AAK, P.AAL, P.AAM, P.AAN, P.AAO, -
00000000' 00000000' 00000000' 00000000' 00000000' 00000000' 000D0 P.AAP, P.AAQ, P.AAR, P.AAS, P.AAT, P.AAU, -
                                         P.AAV

```

.PSECT \$OWNS\$,NOEXE,2

```

00000004 00000003 00000005 00000005 00000008 00000007 00000 MONTHL: .LONG 7, 8, 5, 5, 3, 4, 4, 6, 9, 7, 8, 8
00000008 00000008 00000007 00000009 00000006 00000004 00018

```

SUBST_NAMES=
MONTHS=

P.AAA
P.AAJ

```

.EXTRN DATTIM, IRA, KHAR
.EXTRN RNFFNA, RNFFDS, CNVDAT
.EXTRN CNVTIM, ENDCHR, ERMS
.EXTRN GSLU, PACBAS, RINTES

```

.PSECT \$CODE\$,NOWRT,2

```

      07FC 00000
5A 00000000G EF 9E 00002
59 00000000' EF 9E 00009
58 00000000G EF 9E 00010
57 00000000G EF 9E 00017
56 00000000G EF 9E 0001E
5E 80 AE 9E 00025
50 68 D0 00029
      66 D5 0002C
      09 14 0002E
68 00G 8F 9A 00030
66 01 CE 00034
      09 11 00037
68 F8 B6 9A 00039 1$:
      F8 A6 D6 0003D
      66 D7 00040
50 68 D1 00042 2$:
      36 12 00045
      66 D5 00047
      09 14 00049
68 00G 8F 9A 0004B
66 01 CE 0004F
      09 11 00052
68 F8 B6 9A 00054 3$:
      F8 A6 D6 00058
      66 D7 0005B
44 AE 32 7D 0005D 4$:
3C AE 4C AE 9E 00061
40 AE 3C AE D0 00066
      3C AE 9F 0006B
      F4 A6 9F 0006E
      00000000G EF 02 FB 00071
      01 50 D1 00078
      11 13 0007B
7E F8 A6 F4 A6 C3 0007D 5$:
      F4 A6 DD 00083
      00000000G 8F DD 00086

```

```

.ENTRY SUBST, Save R2,R3,R4,R5,R6,R7,R8,R9,R10
MOVAB PACBAS, R10
MOVAB SUBST_NAMES, R9
MOVAB KHAR, R8
MOVAB DATTIM, R7
MOVAB IRA+12, R6
MOVAB -128(SP), SP
MOVL KHAR, HOLD_FLAG
TSTL IRA+12
BGTR 1$
MOVZBL #RINTES, KHAR
MNEGL #1, IRA+12
BRB 2$
MOVZBL @IRA+4, KHAR
INCL IRA+4
DECL IRA+12
CMPL KHAR, HOLD_FLAG
BNEQ 5$
TSTL IRA+12
BGTR 3$
MOVZBL #RINTES, KHAR
MNEGL #1, IRA+12
BRB 4$
MOVZBL @IRA+4, KHAR
INCL IRA+4
DECL IRA+12
MOVQ #50, DUMMY_FS+8
MOVAB DUMMY_FS+16, DUMMY_FS
MOVL DUMMY_FS, DUMMY_FS+4
PUSHAB DUMMY_FS
PUSHAB IRA
CALLS #2, GSLU
CMPL GSLU_RESULT, #1
BEQL 6$
SUBL3 IRA, IRA+4, -(SP)
PUSHL IRA
PUSHL #RNFFNA

```

0257

0291
0293

0297
0306

0309
0310
0313

0316
0320

51	00	3C	BE	48	01	AE	2D	0009C	BRB	10\$		
				01	A0			000A3	CLRL	SUBST_INDEX		0332
					06	12	000A5		MNEGL	#1, I		0337
					05	11	000AB		BRB	8\$		
					A9	F2	000AD	8\$:	MOVL	SUBST_NAMES[I], R0		0338
					55	D5	000B2	9\$:	MOVZBL	(R0), R1		
					17	12	000B4		CMPC5	DUMMY_FS+12, @DUMMY_FS, #0, R1, 1(R0)		0339
					A6	C3	000B6		BNEQ	8\$		
					A6	DD	000BC		MOVAB	1(R4), SUBST_INDEX		0341
					8F	DD	000BF		BRB	9\$		0340
					03	FB	000C5	10\$:	AOBLSS	SUBST_NAMES-4, I, 7\$		0335
					04	000CC			TSTL	SUBST_INDEX		0347
					08	AE	9E	000CD	BNEQ	11\$		
					55	CF	000D2	11\$:	SUBL3	IRA, IRA+4, -(SP)		0349
					0010		000D6	12\$:	PUSHL	IRA		
					004D		000DE		PUSHL	#RNFUDS		
									CALLS	#3, ERMS		0348
									RET			0358
									MOVAB	WORK_AREA, WORK_PTR		0359
									CASEL	SUBST_INDEX, #1, #7		
									.WORD	13\$-12\$,-		
										14\$-12\$,-		
										15\$-12\$,-		
										16\$-12\$,-		
										17\$-12\$,-		
										20\$-12\$,-		
										21\$-12\$,-		
										23\$-12\$		
					5E	DD	000E6	13\$:	PUSHL	SP		0363
					AE	9F	000E8		PUSHAB	WORK_PTR		
					57	DD	000EB		PUSHL	R7		
					03	FB	000ED		CALLS	#3, CNVDAT		0364
					3B	11	000F4		BRB	19\$		0369
					5E	DD	000F6	14\$:	PUSHL	SP		
					AE	9F	000F8		PUSHAB	WORK_PTR		
					57	DD	000FB		PUSHL	R7		
					03	FB	000FD		CALLS	#3, CNVTIM		
					2B	11	00104		BRB	19\$		0370
					0A	DD	00106	15\$:	PUSHL	#10		0375
					AE	9F	00108		PUSHAB	WORK_PTR		
					67	DD	0010B		PUSHL	DATTIM		
					1C	11	0010D		BRB	18\$		
					A7	D0	0010F	16\$:	MOVL	DATTIM+4, R0		0453
					74	A940	D0	00113	MOVL	MONTHS-4[R0], WORK_PTR		
					6E	00000000	EF40	D0	MOVL	MONTHL-4[R0], WORK_LENGTH		0454
							6D	11	BRB	26\$		0359
							0A	DD	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		
					AE	9E	00131	19\$:	MOVAB	WORK_AREA, WORK_PTR		0460
					58	11	00136		BRB	26\$		0359
					30	90	00138	20\$:	MOVB	#48, WORK_AREA		0466
					AE	D6	0013C		INCL	WORK_PTR		0467
					0A	DD	0013F		PUSHL	#10		0468

		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		
			03	FB	00158	22\$:	CALLS	#3, PACBAS		
	04	6A	50	D0	0015B		MOVL	R0, WORK_LENGTH		
		AE	08	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		
			03	FB	00179		CALLS	#3, PACBAS		
	04	6A	50	D0	0017C		MOVL	R0, WORK_LENGTH		
		AE	08	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	27\$:	MOVZBL	@WORK_PTR, -(SP)	0508
			08	AE	D6	00198		INCL	WORK_PTR	
				01	FB	0019B		CALLS	#1, ENDCHR	
EE	00000000G	EF	6E	F3	001A2	28\$:	AOBLEQ	WORK_LENGTH, I, 27\$		
		52	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)

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

COMMAND QUALIFIERS

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

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

0349

AH-BT13A-SE
 VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY