


```

IIIIII  FFFFFFFF  IIIIII  FFFFFFFF  NN  NN  EEEEEEEEE
IIIIII  FFFFFFFF  IIIIII  FFFFFFFF  NN  NN  EEEEEEEEE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NNNN  NN  EE
II      FF      II      FF      NNNN  NN  EE
II      FFFFFFFF  II      FFFFFFFF  NN  NN  EEEEEEEEE
II      FFFFFFFF  II      FFFFFFFF  NN  NN  EEEEEEEEE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NN  NN  EE
II      FF      II      FF      NN  NN  EE
IIIIII  IIIIII  IIIIII  IIIIII  NN  NN  EEEEEEEEE
IIIIII  IIIIII  IIIIII  IIIIII  NN  NN  EEEEEEEEE

```

```

LL      IIIIII  SSSSSSS
LL      IIIIII  SSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSS
LL      II      SSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLL  IIIIII  SSSSSSS
LLLLLLLL  IIIIII  SSSSSSS

```



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

```

P 0001 0 MODULE ififne ( IDENT = 'V04-000'
0002 0 %BLISS32[, ADDRESSING_MODE (EXTERNAL = LONG_RELATIVE,
0003 0 NONEXTERNAL = LONG_RELATIVE})
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 **
0031 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0032 1
0033 1 ABSTRACT:
0034 1
0035 1 Processes the .IF, .IFNOT, .ELSE, .ENDIF, and .VARIABLE commands.
0036 1
0037 1 ENVIRONMENT: Transportable
0038 1
0039 1 AUTHOR: R.W.Friday CREATION DATE: June, 1978
0040 1
0041 1

```

IFIFNE
V04-000

Revision History

K 11
16-Sep-1984 00:44:45
14-Sep-1984 13:06:43

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]IFIFNE.BLI;1

Page 2
(2)

IFI
V04

:	43	0042	1	XSBTTL 'Revision History'
:	44	0043	1	
:	45	0044	1	MODIFIED BY:
:	46	0045	1	
:	47	0046	1	005 RER00005 Ron Randall 07-Mar-1983
:	48	0047	1	Global edit of all modules. Updated module names, idents,
:	49	0048	1	copyright dates. Changed require files to BLISS library.
:	50	0049	1	
:	51	0050	1	--
:	52	0051	1	

: R
:

Module Level Declarations

```

54 0052 1 %SBTTL 'Module Level Declarations'
55 0053 1
56 0054 1
57 0055 1 : TABLE OF CONTENTS:
58 0056 1
59 0057 1 FORWARD ROUTINE
60 0058 1     IFIFNE : NOVALUE,
61 0059 1     VR : NOVALUE,
62 0060 1     VRENTN : NOVALUE,
63 0061 1     VRFIND;
64 0062 1
65 0063 1
66 0064 1 : INCLUDE FILES:
67 0065 1
68 0066 1 LIBRARY 'NXPORT:XPORT';           ! XPORT Library
69 0067 1 REQUIRE 'REQ:RNODEF';         ! RUNOFF variant definitions
70 0198 1
71 U 0199 1 %IF DSRPLUS %THEN
72 U 0200 1 LIBRARY 'REQ:DPLLIB';         ! DSRPLUS BLISS Library
73 0201 1 %ELSE
74 0202 1 LIBRARY 'REQ:DSRLIB';         ! DSR BLISS Library
75 0203 1 %FI
76 C204 1
77 0205 1
78 0206 1 : EXTERNAL REFERENCES:
79 0207 1
80 0208 1 EXTERNAL LITERAL
81 0209 1     RINTES : UNSIGNED (8);
82 0210 1
83 0211 1 EXTERNAL
84 0212 1     FS01 : FIXED STRING,
85 0213 1     GCA : GCA_DEFINITION,
86 0214 1     IFSTK : IFSTACK,
87 0215 1     IRA : FIXED_STRING,
88 0216 1     IRAC : IRAC_DEFINITION,
89 0217 1     KHAR;
90 0218 1
91 0219 1 EXTERNAL                               !See VR.REQ for definition of items.
92 0220 1     VRCNT,                               !Number of variables.
93 0221 1     VRNAME : VRNAME_DEF,                 !The variable names.
94 0222 1     VRLNG : VRLNG_DEF,                   !The length of the names.
95 0223 1     VRBOOL : VRBOOL_DEF,                 !TRUE/FALSE value
96 0224 1     VRFFLG : VRFFLG_DEF,                 !FALSE flag
97 0225 1     VRTFLG : VRTFLG_DEF,                 !TRUE flag
98 0226 1     VRSRC : VRSRC_DEF;                   !Source of variable definition.
99 0227 1
100 0228 1 EXTERNAL LITERAL                       !Error messages
101 0229 1     RNFBVN,
102 0230 1     RNFDVN,
103 0231 1     RNFEVL,
104 0232 1     RNFINI,
105 0233 1     RNFITD,
106 0234 1     RNFSKC;
107 0235 1
108 0236 1 EXTERNAL ROUTINE
109 0237 1     ERMA,
110 0238 1     GNAME,

```

IFIFNE
V04-000

Module Level Declarations

:	111	0239	1	RSKIPS,
:	112	0240	1	SKPSEP;
:	113	0241	1	

M 11
16-Sep-1984 00:44:45
14-Sep-1984 13:06:43

VAX-11 Bliss-32 V4.0-742
[RUNOFF.SRC]IFIFNE.BLI;1

Page 4
(3)

IF1
V04

: R

```

115 0242 1 GLOBAL ROUTINE IFIFNE (HANDLER_CODE) : NOVALUE =
116 0243 1
117 0244 1 ++
118 0245 1 FUNCTIONAL DESCRIPTION:
119 0246 1
120 0247 1     See ABSTRACT, above.
121 0248 1
122 0249 1 FORMAL PARAMETERS:
123 0250 1
124 0251 1     HANDLER_CODE indicates which command is to be processed.
125 0252 1
126 0253 1 IMPLICIT INPUTS:      None
127 0254 1
128 0255 1 IMPLICIT OUTPUTS:     None
129 0256 1
130 0257 1 ROUTINE VALUE:
131 0258 1 COMPLETION CODES:     None
132 0259 1
133 0260 1 SIDE EFFECTS:
134 0261 1
135 0262 1     -- Swallows a portion of the uneaten string.
136 0263 1
137 0264 1
138 0265 2     BEGIN
139 0266 2     LOCAL
140 0267 2     GNAME_RESULT,
141 0268 2     VR_INDEX,           !Will be used to point into the variables tables
142 0269 2     X;                   !Just a copy of the IFSTACK depth
143 0270 2
144 0271 2     RSKIPS (IRA);       !Position to variable name.
145 0272 2     GNAME_RESULT = GNAME (IRA, FS01);
146 0273 2
147 0274 2     IF .GNAME_RESULT NEQ GNAME_NORMAL
148 0275 2     THEN
149 0276 3     BEGIN
150 0277 3     ERMA (RNFBVN, TRUE); !Skip to end of command and output message
151 0278 3     RETURN;
152 0279 2     END;
153 0280 2
154 0281 2     SELECT .HANDLER_CODE OF
155 0282 2     SET
156 0283 2
157 0284 2     [H_IF, H_IFNOT] :
158 0285 3     BEGIN
159 0286 3     !See if this variable exists already.
160 0287 3     VR_INDEX = VRFIND (.FS_START (FS01), .FS_LENGTH (FS01));
161 0288 3
162 0289 3     IF .VR_INDEX EQL -1
163 0290 3     THEN
164 0291 4     BEGIN           !Unrecognized variable.
165 0292 4
166 0293 4     IF .VRCNT EQL MAX_VR_NAMES
167 0294 4     THEN
168 0295 5     BEGIN           !Table would overflow if this were added!!!!
169 0296 5     ERMA (RNFEVL, FALSE);
170 0297 5     RETURN;
171 0298 4     END;

```



```

229 0356 3      IFSTK [.X, IFSTK_ELSE_FLG] = .VRTFLG [.VR_INDEX];
230 0357 3      END;
231 0358 3
232 0359 3      [H_ELSE, H_ENDIF] :
233 0360 3      BEGIN                                !Variable must match one on stack.
234 0361 3      X = .IFSTK [0, IFSTK_DEPTH];
235 0362 3
236 0363 3      IF CH$NEQ (.FS_LENGTH (FS01),
237 0364 3          .FS_START (FS01),
238 0365 3          .VR[NG [.IFSTK [.X, IFSTK_VR]],
239 0366 3          CH$PTR (VRNAME [.IFSTK [.X, IFSTK_VR], 0, 0, 0, 0]))
240 0367 3      THEN
241 0368 4          BEGIN                                !Improperly nested, etc
242 0369 4          ERMA (RNFINI, FALSE);
243 0370 4          RETURN;
244 0371 4          END;                                !.ELSE and .ENDIF must appear in same file as .IF/.IFNOT
245 0372 3
246 0373 3      IF .IFSTK [.X, IFSTK_REQ_D] NEQ .GCA_REQ_DEPTH
247 0374 3      THEN
248 0375 3          ERMA (RNFSKC, FALSE);
249 0376 3
250 0377 3      END;
251 0378 3
252 0379 3      [H_ELSE] :
253 0380 3      BEGIN
254 0381 3          !'X' gets set in the error checking block, above.
255 0382 3          IFSTK [.X, IFSTK_BOOL] = ( NOT .IFSTK [.X, IFSTK_BOOL]) AND .IFSTK [.X - 1, IFSTK_BOOL];
256 0383 3
257 0384 3      IF .GCA_DEBUG_CND
258 0385 3      THEN
259 0386 3          !If a /DRAFT document, don't skip.
260 0387 3          GCA_SKIPPING = FALSE
261 0388 3      ELSE
262 0389 3          GCA_SKIPPING = NOT (.IFSTK [.X, IFSTK_BOOL] AND .IFSTK [.X - 1, IFSTK_BOOL]);
263 0390 3
264 0391 3          !Pick up appropriate /DRAFT flag
265 0392 3          IRAC_DRAFT_FLG = .IFSTK [.X, IFSTK_ELSE_FLG];
266 0393 3      END;
267 0394 3
268 0395 3      [H_ENDIF] :
269 0396 3      BEGIN
270 0397 3          X = .IFSTK [0, IFSTK_DEPTH] - 1;
271 0398 3          IFSTK [0, IFSTK_DEPTH] = .X;
272 0399 3
273 0400 3      IF .GCA_DEBUG_CND
274 0401 3      THEN
275 0402 4          BEGIN
276 0403 4          GCA_SKIPPING = FALSE;
277 0404 4          END
278 0405 3      ELSE
279 0406 4          BEGIN
280 0407 4
281 0408 4          IF .X EQL 0
282 0409 4          THEN
283 0410 4              GCA_SKIPPING = FALSE                ! No more active .IFs
284 0411 4          ELSE
285 0412 4              GCA_SKIPPING = NOT (.IFSTK [.X, IFSTK_BOOL] AND .IFSTK [.X - 1, IFSTK_BOOL]);

```

```

: 286 0413 4
: 287 0414
: 288 0415
: 289 0416
: 290 0417
: 291 0418
: 292 0419
: 293 0420
: 294 0421
: 295 0422
: 296 0423

```

```

                END;
                !Restore interrupted /DRAFT flag
                !Note that we have already popped the stack, and so need to
                !look back in the stack. (The data's still valid though).
                IRAC_DRAFT_FLG = .IFSTK [.X+1, IFSTK_FLAG];
                END;
TES;
END;

```

!End of IFIFNE

```

                .TITLE IFIFNE
                .IDENT \V04-000\

                .EXTRN RINTES, FS01, GCA
                .EXTRN IFSTK, IRA, IRAC
                .EXTRN KHAR, VRCNT, VRNAME
                .EXTRN VRLNG, VRBOOL, VRFPLG
                .EXTRN VRTFLG, VRSRC, RNFBVN
                .EXTRN RNFDVN, RNFEVL, RNFINI
                .EXTRN RNFITD, RNFSKC, ERMA
                .EXTRN GNAME, RSKIPS, SKPSEP

                .PSECT $CODE$,NOWRT,2

                OFFC 00000
                .ENTRY IFIFNE, Save R2,R3,R4,R5,R6,R7,R8,R9,R10,-
                R11
                MOVAB VRCNT, R11
                MOVAB IRAC+24, R10
                MOVAB FS01, R9
                MOVAB GCA+156, R8
                MOVAB IFSTK, R7
                PUSHAB IRA
                CALLS #1, RSKIPS
                PUSHL R9
                PUSHAB IRA
                CALLS #2, GNAME
                CMPL GNAME_RESULT, #1
                BEQL 1$
                PUSHL #1
                PUSHL #RNFBVN
                BRB 6$
                MOVL HANDLER_CODE, R6
                CMPL R6, #10T
                BGEQ 3$
                BRW 10$
                CMPL R6, #102
                BGTR 2$
                PUSHL FS01+12
                PUSHL FS01
                CALLS #2, VRFIND
                MOVL R0, VR_INDEX
                CMPL VR_INDEX, #-1
                BNEQ 5$
                CMPL VRCNT, #20

```

```

                5B 00000000G EF 9E 00002
                5A 00000000G EF 9E 00009
                59 00000000G EF 9E 00010
                58 00000000G EF 9E 00017
                57 00000000G EF 9E 0001E
                00000000G EF 9F 00025
                00000000G EF 01 FB 0002B
                00000000G EF 59 DD 00032
                00000000G EF 9F 00034
                01 EF 02 FB 0003A
                01 50 D1 00041
                00000000G 0A 13 00044
                00000000G 01 DD 00046
                00000000G 8F DD 00048
                56 04 AC D0 00050 1$:
                8F 56 D1 00054
                03 18 0005B
                00000065 00CC 31 0005D 2$:
                8F 56 D1 00060 3$:
                0C F4 14 00067
                0C A9 DD 00069
                00000000V EF 69 DD 0006C
                52 EF 02 FB 0006E
                FFFFFFFF 8F 50 D0 00075
                52 D1 00078
                25 12 0007F
                14 6B D1 00081

```

```

                : 0242
                :
                : 0271
                : 0272
                : 0274
                : 0277
                : 0281
                : 0284
                :
                : 0287
                :
                : 0289
                : 0293

```

		0A	12	00084	BNEQ	4\$		
		7E	D4	00086	CLRL	-(SP)		0296
		00000000G	8F	DD 00088	PUSHL	#RNFVFL		
			27	11 0008E	BRB	6\$		
		7E	01	7D 00090	4\$: MOVQ	#1, -(SP)		0301
			20	DD 00093	PUSHL	#32		
			20	DD 00095	PUSHL	#32		
			0C	A9 DD 00097	PUSHL	FS01+12		
			69	DD 0009A	PUSHL	FS01		
		00000000V	EF	06 FB 0009C	CALLS	#6, VRENT		
			52	6B D0 000A3	MOVL	VRCNT, VR_INDEX		0302
54			67	01 C1 000A6	5\$: ADDL3	#1, IFSTK, X		0305
			0A	54 D1 000AA	CMPL	X, #10		0307
				0B 15 000AD	BLEQ	7\$		
			7E	D4 000AF	CLRL	-(SP)		0310
			00000000G	8F DD 000B1	PUSHL	#RNFITD		
			00EE	31 000B7	6\$: BRW	14\$		
		67	54	D0 000BA	7\$: MOVL	X, IFSTK		0315
50		54	05	78 000BD	ASHL	#5, X, R0		0316
			0C	A740 9F 000C1	PUSHAB	IFSTK+12[R0]		
		9E	56	D0 000C5	MOVL	R6, @(SP)+		
			04	A740 9F 000C8	PUSHAB	IFSTK+4[R0]		0317
		9E	52	D0 000CC	MOVL	VR_INDEX, @(SP)+		
			10	A740 9F 000CF	PUSHAB	IFSTK+16[R0]		0318
		9E	20	A8 D0 000D3	MOVL	GCA+188, @(SP)+		
			14	A740 9F 000D7	PUSHAB	IFSTK+20[R0]		0319
		9E	F4	AA D0 000DB	MOVL	IRAC+12, @(SP)+		
			18	A740 9F 000DF	PUSHAB	IFSTK+24[R0]		0320
		9E	F0	AA D0 000E3	MOVL	IRAC+8, @(SP)+		
			1C	A740 9F 000E7	PUSHAB	IFSTK+28[R0]		0321
		9E	6A	D0 000EB	MOVL	IRAC+24, @(SP)+		
		50	08	A740 9E 000EE	MOVAB	IFSTK+8[R0], R0		0322
		60	00000000GEF	42 D0 000F3	MOVL	VRBOOL[VR_INDEX], (R0)		
02	D8	A8	02	E1 000FB	BBC	#2, GCA+116, 8\$		0325
			60	D4 00100	CLRL	(R0)		0327
		000000066	8F	56 D1 00102	8\$: CMPL	R6, #102		0329
			03	12 00109	BNEQ	9\$		
		60	60	D2 0010B	MCOML	(R0), (R0)		0331
55		54	05	78 0010E	9\$: ASHL	#5, X, R5		0335
			E8	A745 9F 00112	PUSHAB	IFSTK-24[R5]		
		51	9E	D2 00116	MCOML	@(SP)+, R1		
		60	51	CA 00119	BICL2	R1, (R0)		
		51	60	D2 0011C	MCOML	(R0), R1		0339
68		01	00	51 F0 0011F	INSV	R1, #0, #1, GCA+156		
		03	D8	A8 02 E1 00124	BBC	#2, GCA+116, 10\$		0341
			68	01 8A 00129	BICB2	#1, GCA+156		0343
		000000065	8F	56 D1 0012C	10\$: CMPL	R6, #101		0347
			18	12 00133	BNEQ	11\$		
		55	54	05 78 00135	ASHL	#5, X, R5		0349
			20	A745 9F 00139	PUSHAB	IFSTK+32[R5]		
			9E	00000000GEF	42 D0 0013D	MOVL	VRFFLG[VR_INDEX], @(SP)+	
			6A	00000000GEF	42 D0 00145	MOVL	VRTFLG[VR_INDEX], IRAC+24	
		000000066	8F	56 D1 0014D	11\$: CMPL	R6, #102		0350
			18	12 00154	BNEQ	12\$		0353
			6A	00000000GEF	42 D0 00156	MOVL	VRFFLG[VR_INDEX], IRAC+24	
55		54	05	78 0015E	ASHL	#5, X, R5		0355
			20	A745 9F 00162	PUSHAB	IFSTK+32[R5]		0356

		9E	00000000GEF42	D0	00166	MOVL	VRTFLG[VR_INDEX], @(SP)+		
		2E	56	D1	0016E	12\$:	CMPL	R6, #46	0359
			05	13	00171		BEQL	13\$	
		37	56	D1	00173		CMPL	R6, #55	
			51	12	00176		BNEQ	16\$	
		54	67	D0	00178	13\$:	MOVL	IFSTK, X	0361
55		54	05	78	0017B		ASHL	#5, X, R5	0365
			04 A745	9F	0017F		PUSHAB	IFSTK+4[R5]	
		51	9E	D0	00183		MOVL	@(SP)+, R1	
50		51	04	78	00186		ASHL	#4, R1, R0	0366
		00	00000000GEF41	DF	0018A		PUSHAL	VRLNG[R1]	
9E		00	0C A9	2D	00191		CMPCS	FS01+12, @FS01, #0, @(SP)+, VRNAME[R0]	
			00000000GEF40		00198				
			10	13	0019E		BEQL	15\$	
			7E	D4	001A0		CLRL	-(SP)	0369
			00000000G	8F	DD 001A2		PUSHL	#RNFINI	
		00000000G	EF	02	FB 001A8	14\$:	CALLS	#2, ERMA	
				04	001AF		RET		0368
			10 A745	9F	001B0	15\$:	PUSHAB	IFSTK+16[R5]	0373
		20	A8	9E	D1 001B4		CMPL	@(SP)+, GCA+188	
				0F	13 001B8		BEQL	16\$	
				7E	D4 001BA		CLRL	-(SP)	0375
			00000000G	8F	DD 001BC		PUSHL	#RNF5KC	
		00000000G	EF	02	FB 001C2		CALLS	#2, ERMA	
			2E	56	D1 001C9	16\$:	CMPL	R6, #46	0379
				3D	12 001CC		BNEQ	19\$	
51		54	05	78	001CE		ASHL	#5, X, R1	0382
		52	08 A741	9E	001D2		MOVAB	IFSTK+8[R1], R2	
50		54	05	78	001D7		ASHL	#5, X, R0	
			E8 A740	9F	001DB		PUSHAB	IFSTK-24[R0]	
62		9E	62	CB	001DF		BICL3	(R2), @(SP)+, (R2)	
05	D8	A8	02	E1	001E3		BBC	#2, GCA+116, 17\$	0384
		68	01	8A	001E8		BICB2	#1, GCA+156	0387
			17	11	001EB		BRB	18\$	
50		54	05	78	001ED	17\$:	ASHL	#5, X, R0	0389
			E8 A740	9F	001F1		PUSHAB	IFSTK-24[R0]	
		53	9E	D2	001F5		MCOML	@(SP)+, R3	
52		62	53	CB	001F8		BICL3	R3, (R2), R2	
		50	52	D2	001FC		MCOML	R2, R0	
68		01	00	F0	001FF		INSV	R0, #0, #1, GCA+156	
			20 A741	9F	00204	18\$:	PUSHAB	IFSTK+32[R1]	0392
		6A	9E	D0	00208		MOVL	@(SP)+, IRAC+24	
		37	56	D1	0020B	19\$:	CMPL	R6, #55	0395
			3C	12	0020E		BNEQ	23\$	
54		67	01	C3	00210		SUBL3	#1, IFSTK, X	0397
		67	54	D0	00214		MOVL	X, IFSTK	0398
02	D8	A8	02	E0	00217		BBS	#2, GCA+116, 20\$	0400
			05	12	0021C		BNEQ	21\$	0408
		68	01	8A	0021E	20\$:	BICB2	#1, GCA+156	0410
			1F	11	00221		BRB	22\$	
55		54	05	78	00223	21\$:	ASHL	#5, X, R5	0412
50		54	05	78	00227		ASHL	#5, X, R0	
			E8 A740	9F	0022B		PUSHAB	IFSTK-24[R0]	
		50	9E	D2	0022F		MCOML	@(SP)+, R0	
			08 A745	9F	00232		PUSHAB	IFSTK+8[R5]	
50		9E	50	CB	00236		BICL3	R0, @(SP)+, R0	
		51	50	D2	0023A		MCOML	R0, R1	


```

299 0425 1 GLOBAL ROUTINE VR (HANDLER_CODE) : NOVALUE =
300 0426 1
301 0427 1 +-
302 0428 1 FUNCTIONAL DESCRIPTION:
303 0429 1
304 0430 1 FORMAL PARAMETERS.
305 0431 1
306 0432 1     HANDLER_CODE is never checked by this routine.
307 0433 1
308 0434 1 IMPLICIT INPUTS:      None
309 0435 1
310 0436 1 IMPLICIT OUTPUTS:     None
311 0437 1
312 0438 1 ROUTINE VALUE:
313 0439 1 COMPLETION CODES:     None
314 0440 1
315 0441 1 SIDE EFFECTS:
316 0442 1
317 0443 1     -- Swallows a part of the uneaten string.
318 0444 1
319 0445 1
320 0446 2 BEGIN
321 0447 2 LOCAL
322 0448 2     VR_INDEX,           !Index in VRxxxx where variable goes.
323 0449 2     HOLD_T_FLAG,       !Temporary location of TRUE flag
324 0450 2     HOLD_F_FLAG,       !Temporary location of FALSE flag
325 0451 2     GNAME_RESULT;
326 0452 2
327 0453 2 RSKIPS (IRA);           !Position to name
328 0454 2 GNAME_RESULT = GNAME (IRA, FS01); !Get name
329 0455 2
330 0456 2 IF .GNAME_RESULT NEQ GNAME_NORMAL
331 0457 2 THEN
332 0458 2 BEGIN
333 0459 2 ERMA (RNFBVN, TRUE);    !Skip to end of command and issue message
334 0460 2 RETURN;
335 0461 2 END;
336 0462 2
337 0463 2 HOLD_T_FLAG = %C' ';    !Default TRUE/FALSE flags
338 0464 2 HOLD_F_FLAG = %C' ';
339 0465 2 SKPSEP~(IRA);          !Skip spaces,tabs, and a comma before the flags.
340 0466 2
341 0467 2 !The two flags are given as [[x] [,] [y]]
342 0468 2 !Note that a single character is always taken as the TRUE flag.
343 0469 2 !The input ",y" can be used to specify the FALSE flag alone.
344 0470 2
345 0471 2 IF .KHAR NEQ RINTES
346 0472 2 THEN
347 0473 2 BEGIN                    !There might be TRUE/FALSE flags supplied.
348 0474 2
349 0475 2     IF .KHAR NEQ %C', '
350 0476 2     THEN
351 0477 2     BEGIN                !pick up the TRUE flag
352 0478 2     HOLD_T_FLAG = .KHAR;
353 0479 2     KCNS~(T);            !Get character after the flag
354 0480 2     RSKIPS (IRA);        !Skip spaces and tabs
355 0481 2     END;

```

```

356 0482 3
357 0483 3
358 0484 3
359 0485 4
360 0486 4 'Get character after the comma
361 0487 4
362 0488 3
363 0489 3
364 0490 3
365 0491 3
366 0492 4 !Get FALSE flag
367 0493 4
368 0494 4 !Position past KHAR.
369 0495 3
370 0496 3
371 0497 2
372 0498 2
373 0499 2 VR_INDEX = VRFIND (.FS_START (FS01), .FS_LENGTH (FS01));
374 0500 2
375 0501 2
376 0502 2 IF .VR_INDEX EQL -1
377 0503 2 THEN
378 0504 2 BEGIN !It's a new entry
379 0505 3
380 0506 3 IF .VRCNT EQL MAX_VR_NAMES
381 0507 4 THEN
382 0508 4 BEGIN !Table would overflow if this were added!!!!
383 0509 4 ERMA (RNFEVL, FALSE);
384 0510 4 RETURN;
385 0511 3 END;
386 0512 3 !Allocate a new spot and put this stuff in.
387 0513 3 VRENTR (.FS_START (FS01), .FS_LENGTH (FS01), .HOLD_T_FLAG, .HOLD_F_FLAG, 1, FALSE);
388 0514 3 END
389 0515 2 ELSE
390 0516 2 BEGIN !It's probably an entry made via /VARIANT.
391 0517 2
392 0518 2 IF .VRSRC [.VR_INDEX] EQL 1
393 0519 3 THEN
394 0520 4 BEGIN !This is duplicate declaration!!!
395 0521 4 ERMA (RNFDVN, FALSE);
396 0522 4 RETURN;
397 0523 3 END;
398 0524 3
399 0525 3 VRSRC [.VR_INDEX] = 1; !'1' means from a .IF or .VARIABLE command.
400 0526 3 VRTFLG [.VR_INDEX] = .HOLD_T_FLAG; !Save TRUE flag
401 0527 3 VRFFLG [.VR_INDEX] = .HOLD_F_FLAG; !Save FALSE flag
402 0528 2 END;
403 0529 2
404 0530 1 END; !End of VR

```

```

07FC 0000
5A 0000000G EF 9E 00002
59 00G 8F 9A 00009

```

```

.ENTRY VR, Save R2,R3,R4,R5,R6,R7,R8,R9,R10
MOVAB VRSRC, R10
MOVZBL #RINTÉS, R9

```

```

: 0425
:
:

```

	58	00000000G	EF	9E	0000D	MOVAB	RSKIPS, R8	
	57	00000000G	EF	9E	00014	MOVAB	FS01, R7	
	56	00000000G	EF	9E	0001B	MOVAB	KHAR, R6	
	55	00000000G	EF	9E	00022	MOVAB	IRA+12, R5	
		F4	A5	9F	00029	PUSHAB	IRA	0453
	68		01	FB	0002C	CALLS	#1, RSKIPS	
			57	DD	0002F	PUSHL	R7	0454
		F4	A5	9F	00031	PUSHAB	IRA	
00000000G	EF		02	FB	00034	CALLS	#2, GNAME	
	01		50	D1	0003B	CMPL	GNAME_RESULT, #1	0456
			0B	13	0003E	BEQL	1\$	
			01	DD	00040	PUSHL	#1	0459
		00000000G	8F	DD	00042	PUSHL	#RNFBN	
			00C5	31	00048	BRW	12\$	
	54		20	D0	0004B	MOVL	#32, HOLD_T_FLAG	0463
	53		20	D0	0004E	MOVL	#32, HOLD_F_FLAG	0464
		F4	A5	9F	00051	PUSHAB	IRA	0465
00000000G	EF		01	FB	00054	CALLS	#1, SKPSEP	
	50		66	D0	0005B	MOVL	KHAR, R0	0471
	59		50	D1	0005E	CMPL	R0, R9	
			60	13	00061	BEQL	9\$	
	2C		50	D1	00063	CMPL	R0, #44	0475
			1E	13	00066	BEQL	4\$	
	54		50	D0	00068	MOVL	R0, HOLD_T_FLAG	0478
			65	D5	0006B	TSTL	IRA+12	0479
			08	14	0006D	BGTR	2\$	
	66		59	9A	0006F	MOVZBL	R9, KHAR	
	65		01	CE	00072	MNEGL	#1, IRA+12	
			09	11	00075	BRB	3\$	
	66	F8	B5	9A	00077	MOVZBL	@IRA+4, KHAR	2\$:
		F8	A5	D6	0007B	INCL	IRA+4	
			65	D7	0007E	DECL	IRA+12	
		F4	A5	9F	00080	PUSHAB	IRA	3\$:
	68		01	FB	00083	CALLS	#1, RSKIPS	
	2C		66	D1	00086	CMPL	KHAR, #44	4\$:
			1B	12	00089	BNEQ	7\$	
			65	D5	0008B	TSTL	IRA+12	0486
			08	14	0008D	BGTR	5\$	
	66		59	9A	0008F	MOVZBL	R9, KHAR	
	65		01	CE	00092	MNEGL	#1, IRA+12	
			09	11	00095	BRB	6\$	
	66	F8	B5	9A	00097	MOVZBL	@IRA+4, KHAR	5\$:
		F8	A5	D6	0009B	INCL	IRA+4	
			65	D7	0009E	DECL	IRA+12	
		F4	A5	9F	000A0	PUSHAB	IRA	6\$:
	68		01	FB	000A3	CALLS	#1, RSKIPS	
	59		66	D1	000A6	CMPL	KHAR, R9	7\$:
			18	13	000A9	BEQL	9\$	
	53		66	D0	000AB	MOVL	KHAR, HOLD_F_FLAG	0493
			65	D5	000AE	TSTL	IRA+12	0494
			08	14	000B0	BGTR	8\$	
	66		59	9A	000B2	MOVZBL	R9, KHAR	
	65		01	CE	000B5	MNEGL	#1, IRA+12	
			09	11	000B8	BRB	9\$	
	66	F8	B5	9A	000BA	MOVZBL	@IRA+4, KHAR	8\$:
		F8	A5	D6	000BE	INCL	IRA+4	
			65	D7	000C1	DECL	IRA+12	

Module Level Declarations

		OC	A7	DD	000C3	9\$:	PUSHL	FS01+12		0499
			67	DD	000C6		PUSHL	FS01		
00000000V	EF		02	FB	000C8		CALLS	#2, VRFIND		
	52		50	D0	000CF		MOVL	R0, VR_INDEX		
FFFFFFFF	8F		52	D1	000D2		CMPL	VR_INDEX, #-1		0501
			27	12	000D9		BNEQ	11\$		
		14	00000000G	EF	D1	000DB	CMPL	VRCNT, #20		0505
				0A	12	000E2	BNEQ	10\$		
				7E	D4	000E4	CLRL	-(SP)		0508
				8F	DD	000E6	PUSHL	#RNFVFL		
				22	11	000EC	BRB	12\$		
				7E	01	7D	000EE	10\$:		0513
				53	DD	000F1	PUSHL	HOLD_F_FLAG		
				54	DD	000F3	PUSHL	HOLD_T_FLAG		
				OC	A7	DD	000F5	PUSHL	FS01+12	
				67	DD	000F8	PUSHL	FS01		
00000000V	EF		06	FB	000FA		CALLS	#6, VRENTN		0501
				04	00101		RET			
				01	6A42	D1	00102	11\$:		0518
				10	12	00106	BNEQ	13\$		
				7E	D4	00108	CLRL	-(SP)		0521
				8F	DD	0010A	PUSHL	#RNFVFN		
00000000G	EF		02	FB	00110	12\$:	CALLS	#2, ERMA		
				04	00117		RET			0520
				01	D0	00118	13\$:	MOVL	#1, VRSRC[VR_INDEX]	0525
				54	D0	0011C	MOVL	HOLD_T_FLAG, VRTFLG[VR_INDEX]		0526
00000000GEF42				53	D0	00124	MOVL	HOLD_F_FLAG, VRFFLG[VR_INDEX]		0527
00000000GEF42				04	0012C		RET			0530

: Routine Size: 301 bytes, Routine Base: \$CODE\$ + 024D

: 405 0531 1

```

: 407 0532 1 GLOBAL ROUTINE VRFIND (PTR, LENGH) =
: 408 0533 1
: 409 0534 1 +-
: 410 0535 1 FUNCTIONAL DESCRIPTION:
: 411 0536 1
: 412 0537 1 Returns the index to the name, if found. otherwise, returns -1.
: 413 0538 1
: 414 0539 1 FORMAL PARAMETERS:
: 415 0540 1
: 416 0541 1 PTR is a ch$ptr to a name to be located; LENGH is its length
: 417 0542 1
: 418 0543 1 IMPLICIT INPUTS: None
: 419 0544 1
: 420 0545 1 IMPLICIT OUTPUTS: None
: 421 0546 1
: 422 0547 1 ROUTINE VALUE:
: 423 0548 1 COMPLETION CODES: None
: 424 0549 1
: 425 0550 1 SIDE EFFECTS: None
: 426 0551 1 --
: 427 0552 1
: 428 0553 2 BEGIN
: 429 0554 2 !In the loop, I starts at 1 even though there is a 0th element.
: 430 0555 2 !This is because the 0th one is a dummy.
: 431 0556 2
: 432 0557 2 INCR I FROM 1 TO .VRCNT DO
: 433 0558 2 BEGIN
: 434 0559 2
: 435 0560 2 IF .VRLNG [.I] EQL .LENGH
: 436 0561 2 THEN
: 437 0562 2
: 438 0563 2 IF CH$EQL (.LENGH, .PTR, .LENGH, CH$PTR (VRNAME [.I, 0, 0, 0, 0]))
: 439 0564 2 THEN
: 440 0565 2 RETURN .I; !Variable found
: 441 0566 2
: 442 0567 2 END;
: 443 0568 2
: 444 0569 2 RETURN -1; !Variable not found
: 445 0570 1 END; !End of VRFIND

```

				001C 00000	.ENTRY VRFIND, Save R2,R3,R4	: 0532
			54	D4 00002	CLRL I	: 0560
			20	11 00004	BRB 2\$	
	08	AC 00000000GEF44	44	D1 00006 1\$:	CMPL VRLNG[I], LENGH	
			15	12 0000F	BNEQ 2\$	
	50	54	04	78 00011	ASHL #4, I, R0	: 0563
00000000GEF40	04	BC 08	AC	29 00015	CMPC3 LENGH, @PTR, VRNAME[R0]	
			04	12 00020	BNEQ 2\$	
		50	54	D0 00022	MOVL I, R0	: 0565
				04 00025	RET	
	D8	54 00000000G	EF	F3 00026 2\$:	AOBLEQ VRCNT, I, 1\$: 0557
		50	01	CE 0002E	MNEGL #1, R0	: 0569
			04	00031	RET	: 0570


```

: 448 0572 1 GLOBAL ROUTINE VRENTR (VARIABLE, VARIABLE_LNG, TRUE_FLAG, FALSE_FLAG, SOURCE_FLAG, LOGICAL_VALUE) : NOVALUE
: 449 0573 1
: 450 0574 1 ++
: 451 0575 1 FUNCTIONAL DESCRIPTION:
: 452 0576 1
: 453 0577 1     VRENTR saves a variable in the variable tables, along with
: 454 0578 1     associated information.
: 455 0579 1
: 456 0580 1 FORMAL PARAMETERS:
: 457 0581 1
: 458 0582 1     VARIABLE is a CHSPTR to the variable name. VARIABLE_LNG is its length.
: 459 0583 1     TRUE_FLAG and FALSE_FLAG are the two draft flags.
: 460 0584 1     SOURCE_FLAG indicates whether the definition came from a command
: 461 0585 1     or a /VARIANT switch.
: 462 0586 1     LOGICAL_VALUE is the TRUE/FALSE setting of the variable.
: 463 0587 1
: 464 0588 1 IMPLICIT INPUTS:      None
: 465 0589 1
: 466 0590 1 IMPLICIT OUTPUTS:    None
: 467 0591 1
: 468 0592 1 ROUTINE VALUE:
: 469 0593 1 COMPLETION CODES:    None
: 470 0594 1
: 471 0595 1 SIDE EFFECTS:        None
: 472 0596 1 --
: 473 0597 1
: 474 0598 2 BEGIN
: 475 0599 2 VRCNT = .VRCNT + 1;           !Allocate a new slot.
: 476 0600 2 CH$MOVE (.VARIABLE_LNG, !Move variable name into the slot.
: 477 0601 2 .VARIABLE,
: 478 0602 2 CH$PTR (VRNAME [.VRCNT, 0, 0, 0, 0]));
: 479 0603 2 VRLNG [.VRCNT] = .VARIABLE_LNG; !Save length of name.
: 480 0604 2 VRBOOL [.VRCNT] = .LOGICAL_VALUE; !Save TRUE/FALSE value.
: 481 0605 2 VRSRC [.VRCNT] = .SOURCE_FLAG; !Remember how variable got defined.
: 482 0606 2 VRTFLG [.VRCNT] = .TRUE_FLAG; !Save TRUE flag
: 483 0607 2 VRFFLG [.VRCNT] = .FALSE_FLAG; !Save FALSE flag
: 484 0608 1 END; !End of VRENTR

```

			00FC 0000	.ENTRY	VRENTR, Save R2,R3,R4,R5,R6,R	: 0572
	57	00000000G	EF 9E 00002	MOVAB	VRCNT, R7	: 0599
		56	67 D6 00009	INCL	VRCNT	: 0602
		56	67 D0 0000B	MOVL	VRCNT, R6	
		56	04 78 0000E	ASHL	#4, R6, R0	
00000000GEF40	50	04 BC 08	AC 28 00012	MOV3	VARIABLE_LNG, @VARIABLE, VRNAME[R0]	
		00000000GEF46	08 AC D0 0001D	MOVL	VARIABLE_LNG, VRLNG[R6]	: 0603
		00000000GEF46	18 AC D0 00026	MOVL	LOGICAL VALUE, VRBOOL[R6]	: 0604
		00000000GEF46	14 AC D0 0002F	MOVL	SOURCE_FLAG, VRSRC[R6]	: 0605
		00000000GEF46	0C AC D0 00038	MOVL	TRUE_FLAG, VRTFLG[R6]	: 0606
		00000000GEF46	10 AC D0 00041	MOVL	FALSE_FLAG, VRFFLG[R6]	: 0607
			04 0004A	RET		: 0608

: Routine Size: 75 bytes. Routine Base: \$CODE\$ + 03AC

```

: 485      0609 1
: 486      0610 1 END
: 487      0611 0 ELUDOM
                                !End of module

```

PSECT SUMMARY

```

: Name          Bytes          Attributes
: $CODE$       1015 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	45	3	86	00:00.3

COMMAND QUALIFIERS

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

```

: Size:          1015 code + 0 data bytes
: Run Time:      00:17.3
: Elapsed Time: 00:35.3
: Lines/CPU Min: 2122
: Lexemes/CPU-Min: 18132
: Memory Used:  144 pages
: Compilation Complete

```


