

RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000000000		FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNNNNN	NNN	000	000	FFF	FFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU		UUU	NNN	NNN	000	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNNNNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUU		UUU	NNN	NNN	000	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU		UUUUUUUUUUUUUUUU	NNN	NNN	000000000		FFF	FFF

```
AAAAAA  RRRRRRRR  EEEEEEEEE  CCCCCCCC
AAAAAA  RRRRRRRR  EEEEEEEEE  CCCCCCCC
AA      AA      RR      RR      CC
AA      AA      RR      RR      CC
AA      AA      RR      RR      CC
AA      AA      RRRRRRRR  EEEEEEEE  CC
AA      AA      RRRRRRRR  EEEEEEEE  CC
AAAAAAAA  RR      RR      EE      CC
AAAAAAAA  RR      RR      EE      CC
AA      AA      RR      RR      CC
AA      AA      RR      RR      CC
AA      AA      RR      RR      CC
AA      AA      RR      RR      CC
EEEEEEEE  CCCCCCCC  ....
EEEEEEEE  CCCCCCCC  ....
EEEEEEEE  CCCCCCCC  ....
EEEEEEEE  CCCCCCCC  ....
```

```
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
LLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLL  IIIIII  SSSSSSSS
```



```
0001 0 MODULE AREC (
0002 0 IDENT = 'V04-000'
0003 0 %BLISS32[
0004 0 ADDRESSING_MODE(EXTERNAL=LONG_RELATIVE, NONEXTERNAL=LONG_RELATIVE)
0005 0 ]
0006 0 ) =
0007 1 BEGIN
0008 1
0009 1
0010 1 *****
0011 1 *
0012 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0013 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0014 1 * ALL RIGHTS RESERVED.
0015 1 *
0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0021 1 * TRANSFERRED.
0022 1 *
0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0025 1 * CORPORATION.
0026 1 *
0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0029 1 *
0030 1 *
0031 1 *****
0032 1
0033 1
0034 1 ++
0035 1 FACILITY: DSR (Digital Standard RUNOFF) / DSRPLUS
0036 1
0037 1 ABSTRACT: Performs preliminary analysis on an uneaten string.
0038 1
0039 1
0040 1 ENVIRONMENT: Transportable
0041 1
0042 1 AUTHOR: R.W.Friday CREATION DATE: May, 1978
0043 1
```

AREC
V04-000

Revision History

H 7
15-Sep-1984 23:49:08
14-Sep-1984 13:05:29

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

:	45	0044	1	%SBTTL 'Revision History'		
:	46	0045	1			
:	47	0046	1	MODIFIED BY:		
:	48	0047	1			
:	49	0048	1	007	KFA00007	Ken Alden 07-Mar-1983
:	50	0049	1		Global edit of all modules. Updated module names, idents,	
:	51	0050	1		copyright dates. Changed require files to BLISS library.	
:	52	0051	1			
:	53	0052	1	--		


```
: 55      0053 1 %SBTTL 'Module Level Declarations'
: 56      0054 1
: 57      0055 1
: 58      0056 1 : TABLE OF CONTENTS:
: 59      0057 1
: 60      0058 1
: 61      0059 1 : INCLUDE FILES:
: 62      0060 1
: 63      0061 1 LIBRARY 'NXPOR:XPORT';      ! XPORT Library
: 64      0062 1 REQUIRE 'REQ:RNODEF';      ! RUNOFF variant definitions
: 65      0193 1
: 66      U 0194 1 %IF DSRPLUS %THEN
: 67      U 0195 1 LIBRARY 'REQ:DPLLIB';      ! DSRPLUS BLISS Library
: 68      0196 1 %ELSE
: 69      0197 1 LIBRARY 'REQ:DSRLIB';      ! DSR BLISS Library
: 70      0198 1 %FI
: 71      0199 1
: 72      0200 1
: 73      0201 1 : EQUATED SYMBOLS:
: 74      0202 1
: 75      0203 1
: 76      0204 1 : OWN STORAGE:
: 77      0205 1
: 78      0206 1
: 79      0207 1 : EXTERNAL REFERENCES:
: 80      0208 1
: 81      0209 1
: 82      0210 1 EXTERNAL LITERAL
: 83      0211 1 RINTES : UNSIGNED (8);
: 84      0212 1
: 85      0213 1 EXTERNAL
: 86      0214 1 FNCT : FNCT_DEFINITION,
: 87      0215 1 FLGT : FLAG_TABLE [FLAG_COUNT],
: 88      0216 1 GCA : VECTOR [GCA_SIZE],
: 89      0217 1 IRA : FIXED_STRING,
: 90      0218 1 KHAR;
```

Routine AREC

```

92 0219 1 %SBTTL 'Routine AREC'
93 0220 1 GLOBAL ROUTINE AREC =
94 0221 1
95 0222 1
96 0223 1 ++
97 0224 1 FUNCTIONAL DESCRIPTION:
98 0225 1 AREC is used to determine what the current record looks like.
99 0226 1 It also detects the '!' and ';' forms of comments, as well
100 0227 1 as .BREAK commands indicated by a single '.' followed by
101 0228 1 either white space or end of record.
102 0229 1
103 0230 1 FORMAL PARAMETERS: None
104 0231 1
105 0232 1 IMPLICIT INPUTS: None
106 0233 1
107 0234 1 IMPLICIT OUTPUTS: None
108 0235 1
109 0236 1 ROUTINE VALUE:
110 0237 1 COMPLETION CODES:
111 0238 1
112 0239 1 See ARECCC.REQ for a description of completion codes.
113 0240 1
114 0241 1 SIDE EFFECTS: None
115 0242 1
116 0243 1 --
117 0244 1
118 0245 2 BEGIN
119 0246 2 !First see if there is anything at all on the record.
120 0247 2
121 0248 2 IF .KHAR EQL RINTES
122 0249 2 THEN
123 0250 2 !The record is empty.
124 0251 2 RETURN END_OF_STRING;
125 0252 2
126 0253 2 !Remember start of possible command for error analysis later.
127 0254 2 GCA_COM_START = CH$PLUS (.FS NEXT (IRA), -1);
128 0255 2 !Check for the <ENDFOOTNOTE flag> if footnotes are being
129 0256 2 !collected, and the <ENDFOOTNOTE flag> is enabled.
130 0257 2
131 0258 2 IF .FNCT_COLLECTING
132 0259 2 AND (.KHAR EQL .FLGT [EFO_FLAG, FLAG_CHARACTER])
133 0260 2 AND (.FLGT [EFO_FLAG, FLAG_ENABLED])
134 0261 2 THEN
135 0262 2 !Yes, it was the <ENDFOOTNOTE flag>.
136 0263 2 RETURN END_FOOTNOTE;
137 0264 2
138 0265 2 !Check for the <CONTROL flag> (usually a '.')
139 0266 2
140 0267 2 IF (.KHAR EQL .FLGT [CON_FLAG, FLAG_CHARACTER])
141 0268 2 AND (.FLGT [CON_FLAG, FLAG_ENABLED])
142 0269 2 THEN
143 0270 2 BEGIN
144 0271 2 KCNS ();
145 0272 2 !See if this is a special form of .BREAK.
146 0273 2
147 0274 2 IF (.KHAR EQL RINTES)
148 0275 2 OR (.KHAR EQL %C' ')
149 0276 2 !end of record?
150 0277 2 !space?
151 0278 2
```



```
149 0276 4 OR (.KHAR EQL %'011') !tab?
150 0277 4 THEN
151 0278 4 !Yes, it was a special form of .BREAK.
152 0279 4 RETURN CF_BREAK;
153 0280 4
154 0281 4 !Two consecutive <CONTROL flag>s force a single one as a normal character.
155 0282 4
156 0283 4 IF .KHAR EQL .FLGT [CON_FLAG, FLAG_CHARACTER]
157 0284 4 THEN
158 0285 4 RETURN CF_CF; ! Two <CONTROL flag>s.
159 0286 4
160 0287 4 !See if it's a comment starting like '!' or '.;'.
161 0288 4
162 0289 4 IF (.FLGT [COM_FLAG, FLAG_ENABLED]
163 0290 4 AND (.KHAR EQL .FLGT [COM_FLAG, FLAG_CHARACTER]))
164 0291 4 OR (.KHAR EQL %C';')
165 0292 4 THEN
166 0293 4 !Yes, it's a comment.
167 0294 4 RETURN CF_COMMENT;
168 0295 4
169 0296 4 !The last possibility is that it's a command (even though that's the
170 0297 4 !usual case).
171 0298 4
172 0299 4 IF LETTER (.KHAR)
173 0300 4 THEN
174 0301 4 RETURN CF_LETTER;
175 0302 4
176 0303 4 RETURN CF_OTHER; ! Not recognized.
177 0304 4 End ! End analysis of <CONTROL flag>.
178 0305 4 ELSE
179 0306 4 BEGIN ! Something other than <CONTROL flag>.
180 0307 4
181 0308 4 IF (.KHAR EQL %C' ') !space?
182 0309 4 OR (.KHAR EQL %'011') !tab?
183 0310 4 THEN
184 0311 4 !First character is a space or tab.
185 0312 4 RETURN FIRST_SPACE;
186 0313 4
187 0314 4 IF .KHAR EQL %C';'
188 0315 4 THEN
189 0316 4 !First character is a ';'.
190 0317 4 RETURN FIRST_SEMI;
191 0318 4
192 0319 4 END;
193 0320 4
194 0321 4 RETURN NOT_SPACE; ! None of the special cases tried above.
195 0322 4 END; !End of AREC
```

```
.TITLE AREC
.IDENT \V04-000\
```

```
.EXTRN RINTES, FNCT, FLGT
.EXTRN GCA, IRA, KHAR
```

```
.PSECT $CODE$,NOWRT,2
```

			003C	00000	.ENTRY	AREC, Save R2,R3,R4,R5	0220
55		00G	8F	9A	MOVZBL	#RINTES, R5	
54	00000000G		EF	9E	MOVAB	KHAR, R4	
53	00000000G		EF	9E	MOVAB	IRA+4, R3	
52	00000000G		EF	9E	MOVAB	FLGT+80, R2	
50			64	D0	MOVL	KHAR, R0	0248
55			50	D1	CMPL	R0, R5	
			04	12	BNEQ	1\$,	
50			01	D0	MOVL	#1, R0	0251
				04	RET		
63			01	C3	SUBL3	#1, IRA+4, GCA	0254
0E	00000000G		EF	E9	BLBC	FNCT+20, 2\$	0258
FC			50	D1	CMPL	R0, FLGT+76	0259
A2			08	12	BNEQ	2\$,	
04	B4		A2	E9	BLBC	FLGT+4, 2\$	0260
50			06	D0	MOVL	#6, R0	0263
				04	RET		
62			50	D1	CMPL	R0, FLGT+80	0267
			79	12	BNEQ	14\$,	
75	B8		A2	E9	BLBC	FLGT+8, 14\$	0268
	08		A3	D5	TSTL	IRA+12	0271
			09	14	BGTR	3\$,	
08			55	9A	MOVZBL	R5, KHAR	
A3			01	CE	MNEGL	#1, IRA+12	
			09	11	BRB	4\$,	
64	00		B3	9A	MOVZBL	@IRA+4, KHAR	
			63	D6	INCL	IRA+4	
	08		A3	D7	DECL	IRA+12	
50			64	D0	MOVL	KHAR, R0	0274
55			50	D1	CMPL	R0, R5	
			0A	13	BEQL	5\$,	
20			50	D1	CMPL	R0, #32	0275
			05	13	BEQL	5\$,	
09			50	D1	CMPL	R0, #9	0276
			04	12	BNEQ	6\$,	
50			02	D0	MOVL	#2, R0	0279
				04	RET		
62			50	D1	CMPL	R0, FLGT+80	0283
			04	12	BNEQ	7\$,	
50			03	D0	MOVL	#3, R0	0285
				04	RET		
06	E8		A2	E9	BLBC	FLGT+56, 8\$	0289
30			50	D1	CMPL	R0, FLGT+128	0290
			05	13	BEQL	9\$,	
3B			50	D1	CMPL	R0, #59	0291
			04	12	BNEQ	10\$,	
50			04	D0	MOVL	#4, R0	0294
				04	RET		
00000041	8F		50	D1	CMPL	R0, #65	0299
			09	19	BLSS	11\$,	
0000005A	8F		50	D1	CMPL	R0, #90	
			12	15	BLEQ	12\$,	
00000061	8F		50	D1	CMPL	R0, #97	
			0D	19	BLSS	13\$,	
0000007A	8F		50	D1	CMPL	R0, #122	
			04	14	BGTR	13\$,	
50			05	D0	MOVL	#5, R0	0301

50	09	04 000BD	RET			
		D0 000BE 13\$:	MOVL	#9, R0		0303
20	50	04 000C1	RET			
	05	D1 000C2 14\$:	CMPL	R0, #32		0308
09	50	13 000C5	BEQL	15\$		
	50	D1 000C7	CMPL	R0, #9		0309
50	04	12 000CA	BNEQ	16\$		
	07	D0 000CC 15\$:	MOVL	#7, R0		0312
38	50	04 000CF	RET			
	50	D1 000D0 16\$:	CMPL	R0, #59		0314
50	04	12 000D3	BNEQ	17\$		
	08	D0 000D5	MOVL	#8, R0		0317
50	04	000D8	RET			
	0A	D0 000D9 17\$:	MOVL	#10, R0		0321
	04	000DC	RET			0322

; Routine Size: 221 bytes, Routine Base: \$CODE\$ + 0000

: 196

: 197

: 198

0323 1

0324 1 END

0325 0 ELUDOM

!End of module

PSECT SUMMARY		
Name	Bytes	Attributes
\$CODE\$	221	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	00:00.1
\$255\$DUA28:[RUNOFF.SRC]DSRLIB.L32;1	1248	32	2	86	00:00.3

COMMAND QUALIFIERS

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

Size: 221 code + 0 data bytes

Run Time: 00:05.1

AREC
V04-000

Routine AREC

N 7
15-Sep-1984 23:49:08

VAX-11 Bliss-32 V4.0-742

Page 8

: Elapsed Time: 00:14.5
: Lines/CPU Min: 3831
: Lexemes/CPU-Min: 14180
: Memory Used: 61 pages
: Compilation Complete

BOT
V04-

0337 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

