

ACCESS NAMES TABLE

SOURCE ACCESS NAME= PPC2.P359.SRC.SCAN
OBJECT ACCESS NAME= PPC2.P359.OBJ.SCANS
LISTING ACCESS NAME= PPC2.P359.LST.SCANS
ERROR ACCESS NAME=
OPTIONS= XREF
MACRO LIBRARY PATHNAME=

LINE	KEY	NAME
0002	A	VERSION =>PPC2.P359.SRC.P359

```
0031 IDT 'SCAN'
0032 *****
0033 *
0034 *          SSSS          CCCC          AA          N          N          *
0035 *          S          S          C          C          A          A          NN          N          *
0036 *          S          C          A          A          N          N          N          *
0037 *          SSSS          C          AAAAAA          N          N          N          *
0038 *          S          C          A          A          N          N          N          *
0039 *          S          S          C          C          A          A          N          NN          *
0040 *          SSSS          CCCC          A          A          N          N          *
0041 *
0042 *          P P P P P          333          55555          9999          *
0043 *          P          P          3          3          5          9          9          *
0044 *          P          P          3          5          9          9          *
0045 *          P P P P P          333          555          99999          *
0046 *          P          3          5          9          *
0047 *          P          3          3          5          5          9          9          *
0048 *          P          333          555          9999          *
0049 *
0050 *****
```

0055			DEF	SCAN
0056	0000			
0057			REF	PGMCHR, SETREG, PGMPTR, XFLAG, FORNET, RTNADD
0058			REF	PUTSTK, GETSTK, SAVREG, RESET, PRGFLG, STACK
0059			REF	LINUM, EXTRAM, TABSAV, RAMTOP, ROLB, SYMTAB
0060			REF	RESOLV, GET, GETG, GET1, C4, LNBUF, DATA
0061			REF	C3, CBH94, CBHFF
0062	0000			
0063	0081	ELSE\$	EQU	>81
0064	0082	SSEP\$	EQU	>82
0065	0083	TREM\$	EQU	>83
0066	0084	IF\$	EQU	>84
0067	0085	GO\$	EQU	>85
0068	0086	GOTO\$	EQU	>86
0069	0087	GOSUB\$	EQU	>87
0070	0088	RETUR\$	EQU	>88
0071	0089	DEF\$	EQU	>89
0072	008A	DIM\$	EQU	>8A
0073	008B	END\$	EQU	>8B
0074	008C	FOR\$	EQU	>8C
0075	0092	INPUT\$	EQU	>92
0076	0093	DATA\$	EQU	>93
0077	0096	NEXT\$	EQU	>96
0078	009A	REM\$	EQU	>9A
0079	009B	DN\$	EQU	>9B
0080	009D	CALL\$	EQU	>9D
0081	009E	OPTIO\$	EQU	>9E
0082	00A1	SUB\$	EQU	>A1
0083	00A3	IMAGE\$	EQU	>A3
0084	00A7	SUBXT\$	EQU	>A7
0085	00A8	SUBND\$	EQU	>A8
0086	00AA	LINPU\$	EQU	>AA
0087	00B0	THEN\$	EQU	>B0
0088	00C7	NUM\$	EQU	>C7
0089	00C8	STRIN\$	EQU	>C8
0090	00C9	LN\$	EQU	>C9

```

0092 *****
0093 * SCAN - is the main looping structure of the prescan
0094 * routine. Takes care of scanning each statement in
0095 * a line. Goes back to GPL to scan the special cases *
0096 * (DEF, OPTION, DIM, SUB, CALL, SUBEND, SUBEXIT) and *
0097 * also goes to GPL to enter variables into the symbol *
0098 * table. All statements which are not allowed to be *
0099 * imperative are checked directly without going to *
0100 * GPL. The NOCARE label is where a non-special *
0101 * statement is scanned, looking for variables to enter*
0102 * them into the symbol table. *
0103 *****
0104 0000
0105 0000 SCAN
0106 *-----CONDITIONAL ASSEMBLY-----*
0107 ASMIF VERS=DX10
0108 MOVB *R15+,R0 Read Scan code
0109 ASMELS
0110 0000 D01D MOVB *R13,R0 Read scan code
0111 ASMEND
0112 *-----END OF CONDITIONAL ASSEMBLY-----*
0113 0002 06A0 BL @PUTSTK Save GROM address
0114 0004 0000
0114 0006 06A0 BL @SETREG Set up R8/R9 with CHAT/SUBSTK
0115 0008 0000
0116 000A
0116 * First, decode the type of XML being executed
0117 * Types are: >00 - initial call with program
0118 * >01 - resume within a statement after
0119 * call to GPL for some reason
0120 * >02 - Initial call for imperative stmt
0121 000A
0122 000A 0980 SRL R0,8 Set condition
0123 000C 1305 JEQ SCAN05 If calling scan routine w/pgm
0124 000E 0600 DEC R0 Returning from call to GPL?
0125 0010 1357 JEQ JNCARE Yes-continue w/in line
0126 0012 C819 MOV *R9,@RTNADD
0127 0014 0000
0127 0016 104A JMP SCAN10
0128 0018
0129 0018 A660 SCAN05 A @C3,*R9 Skip following XML and select
0130 001A 0000
0130 001C C819 MOV *R9,@RTNADD Set up rtn to common GPL loc
0131 001E 0014'
0131 0020 04E0 CLR @DATA Assume out of data
0132 0022 0000
0132 0024 8820 SCAN5A C @LINUM,@EXTRAM End of program yet?
0133 0026 0000
0133 0028 0000
0133 002A 161A JNE SCAN07 No-get next line
0134 002C D020 SCAN5B MOVB @FORNET,R0 Check fornex counter
0135 002E 0000
0135 0030 1651 JNE ERRFN For/Next error
0136 0032 D020 MOVB @XFLAG,R0 Check subprogram bits
0137 0034 0000
0137 0037' CBH40 EQU $+1
0138 0036 0A40 SLA R0,4 Subprogram encountered?
0139 0038 1107 JLT SCAN6A Yes-check subend
0140 003A 0200 SCAN06 LI R0,>A000 Initialize data stack
0141 003C A000
    
```

0141	003E	D800		MOVB	RO,@STACK	
	0040	0000				
0142	0042	06A0		BL	@RESOLV	Resolve any subprogram refs
	0044	0000				
0143	0046	104F		JMP	GPL05L	Return
0144	0048					
0145	0048	0A40	SCAN6A	SLA	RO,4	Subend encountered?
0146	004A	1707		JNC	ERRMS	No-text ended w/o subend
0147	004C	0203		LI	R3,TABSAV	Get main symbol table's ptr
	004E	0000				
0148	0050	06A0		BL	@GET1	Get it
	0052	0000				
0149	0054	C801		MOV	R1,@SYMTAB	
	0056	0000				
0150	0058	10F0		JMP	SCAN06	Merge back in
0151	005A	0203	ERRMS	LI	R3,>18	* MISSING SUBEND
	005C	0018				
0152	005E	1043		JMP	GPL05L	
0153	0060					
0154	0060	6820	SCAN07	S	@C4,@EXTRAM	Go to next line in program
	0062	0000				
	0064	0028				
0155	0066	D020		MOVB	@RAMTOP,RO	ERAM program?
	0068	0000				
0156	006A	1604		JNE	SCAN08	Yes-handle ERAM
0157	006C	06A0		BL	@GET	No-get new line pointer in VDP
	006E	0000				
0158	0070	0064		DATA	EXTRAM	
0159	0072	1003		JMP	SCAN09	
0160	0074	06A0	SCAN08	BL	@GETG	Get new line pointer from GRAM
	0076	0000				
0161	0078	0070		DATA	EXTRAM	
0162	007A	C801	SCAN09	MOV	R1,@PGMPTR	Put new line pointer into perm
	007C	0000				
0163	007E	5820		SZCB	@CBH40,@XFLAG	Reset IFFLAG only on new line
	0080	0037				
	0082	0034				
0164	0084	06A0	SCAN9A	BL	@PGMCHR	Get 1st token on line
	0086	0000				
0165	0088	5820		SZCB	@CBH94,@XFLAG	Reset ENTER,STRFLG, and FNCFLG
	008A	0000				
	008C	0082				
0166	008E	D020		MOVB	@XFLAG,RO	Get flag bits
	0090	008C				
0167	0092	0A80		SLA	RO,8	Shift to check REMODE
0168	0094	170B		JNC	SCAN10	If not in REMODE
0169	0096	D208		MOVB	RB,RB	Check if token
0170	0098	1103		JLT	SCAN11	If token-look further
0171	009A	0203	ERRIBS	LI	R3,>1E	* ILLEGAL BETWEEN SUBPROGRAMS
	009C	001E				
0172	009E	1023		JMP	GPL05L	Goto error return
0173	00A0	0706	SCAN11	SETO	R6	Set up index into table
0174	00A2	0586	SCAN12	INC	R6	Increment to 1st/next element
0175	00A4	9988		CB	RB,@IBSTAB(R6)	A legal stmt between subprogs?
	00A6	01E0				
0176	00AB	1BFC		JH	SCAN12	Not able to tell-check further
0177	00AA	1AF7		JL	ERRIBS	Illegal statement here
0178	00AC					
0179	00AC	04C6	SCAN10	CLR	R6	Offset into special stmt table
0180	00AE	C0E6	SCAN15	MOV	@SCNTAB(R6),R3	Read entry from special table

00B0 01B0 /				
0181 00B2 9203	CB	R3,R8		Match this token?
0182 00B4 1306	JEQ	SCAN20		Yes - handle special case
0183 00B6 1B45	JH	NOCARE		Didn't match but passed in tau
0184 00B8 05C6	INCT	R6		Increment offset into table
0185 00BA 0286	CI	R6, TABLEN		Reach end of table?
00BC 0030				
0186 00BE 16F7	JNE	SCAN15		No- check further
0187 00C0 1040	JNCARE	JMP	NOCARE	End of table-not special case
0188 00C2				
0189 00C2 0A83	SCAN20	SLA	R3,8	Look at special case byte
0190 00C4 1169		JLT	SCNGPL	If MSBit set - goto GPL
0191 00C6 06C3		SWPB	R3	MSBit reset - offset into 9900
0192 00C8 0463		B	@OFFSET(R3)	Branch to 9900 special handler
00CA 00CC /				
0193 00CC	OFFSET			
0194 00CC D020	SCAN25	MOVB	@PRGFLG,R0	In program mode?
00CE 0000				
0195 00D0 16A9		JNE	SCAN5A	Yes-check next line
0196 00D2 10AC		JMP	SCAN5B	No- check for/next subs&rtn
0197	*			
0198 00D4	ERRFN			
0199 00D4 105B		JMP	ERRFNN	
0200	*			

```

0202          *          9900 code special case handlers
0203 00D6 F820 IF          SDCB @CBH40,@XFLAG      Indicate scan of 'IF' stmt
      00DB 0037'
      00DA 0090'

0204          *          Special handler for program-only statements
0205 00DC D020 IMPER      MOVB @PRGFLG,RO      Executing in a program?
      00DE 00CE'

0206 00E0 1648          JNE  NXTCHR          Yes - proceed in don't care
0207 00E2 0203 ERRIMP  LI   R3,>12          Illegal imperative return code
      00E4 0012

0208 00E6 105B GPL05L  JMP  GPL05          Return to GPL with error
0209 00E8

0210          *          Special handler for data-statements
0211 00E8 D020 DATA1   MOVB @DATA,RO      Data already encountered?
      00EA 0022'

0212 00EC 1606          JNE  IMAGE          Yes - don't set pointer
0213 00EE C820          MOV  @EXTRAM,@LNBUF      Save line buffer pointer
      00F0 0078'
      00F2 0000

0214 00F4 C820          MOV  @PGMPTR,@DATA      Set line pointer
      00F6 007C'
      00FB 00EA'

0215          *          Special handler for image-statements
0216 00FA D020 IMAGE    MOVB @PRGFLG,RO      In program mode?
      00FC 00DE'

0217 00FE 1692          JNE  SCAN5A          Yes-no need to scan line
0218 0100 10F0          JMP  ERRIMP          No-illegal imperative
0219 0102

0220          *          Special handler for for-statements
0221 0102 05A0 FOR     INC  @XFLAG          Increment the nesting counter
      0104 00DA'

0222 0106 D020          MOVB @XFLAG,RO      Fetch the IFFLAG
      0108 0104'

0223 010A 0240          ANDI RO,>4000          Inside an if-statement?
      010C 4000

0224 010E 1331          JEQ  NXTCHR          No-continue in don't care mode
0225 0110 0203 ERRSYN  LI   R3,>1A          * SYNTAX ERROR
      0112 001A

0226 0114 1044          JMP  GPL05
0227

0228 0116          *          Special handler for next-statements
      NEXT

0229 0116 C020          MOV  @XFLAG,RO      Get flag and for-next counter
      0118 0108'

0230 011A 0240          ANDI RO,>40FF          Get rid of flag bits except IF
      011C 40FF

0231 011E D000          MOVB RO,RO          IFFLAG set?
0232 0120 16F7          JNE  ERRSYN          Yes-syntax error
0233 0122 0600          DEC  RO            Decrement counter by one.
0234 0124 D820          MOVB @ROLB,@FORNET  Move back to the real counter.
      0126 0000
      0128 002E'

0235 012A 1323          JEQ  NXTCHR          Returning to top level - OK
0236 012C 1522          JGT  NXTCHR          Still at a secondary level-OK
0237 012E 0203          LI   R3,>14          For/next nesting return code
      0130 0014

0238 0132 1035          JMP  GPL05          Return to GPL with error
0239 0134 D020 ELSE    MOVB @XFLAG,RO      Get flag byte
      0136 0118'

0240 0138 0240          ANDI RO,>4000          Inside an if?
      013A 4000
    
```

0241 013C 13E9	JEQ ERRSYN	No => error
0242	* SMTSEP B	Special handler for statement separator
0243 013E 0460 0140 0084'	@SCAN9A	Skip the ':' and check next
0244 0142		
0245		
0246	* NDCARE CI	General don't-care scan. Simply looks for variables to enter into symbol table; stops on end of statement
0247 0142-0288 0144 8200	RB, SSEP**256	At a statement separator
0248 0146 13FB	JEQ SMTSEP	Skip and scan next statement
0249 0148-0288 014A 8300	CI RB, TREM**256	At a tail remark?
0250 014C 13BF	JEQ SCAN25	Yes - Check mode
0251 014E D208	MOV B RB, RB	At an end-of-line or symbol?
0252 0150 13BD	JEQ SCAN25	EOL - check mode
0253 0152 151F	JGT ENTER	Symbol - ENTER in symbol tab
0254 0154-0288 0156 C900	CI RB, LN**256	Special line number token?
0255 0158 130F	JEQ SKIPLN	Yes - need to skip it
0256 015A-0288 015C C700	CI RB, NUM**256	Special numeric token?
0257 015E 130F	JEQ SKPSTR	Yes - need to skip it
0258 0160-0288 0162 C800	CI RB, STRIN**256	Special string token?
0259 0164 130C	JEQ SKPSTR	Yes - need to skip it
0260 0166-0288 0168 B000	CI RB, THEN**256	Hit a then-clause?
0261 016A 13E4	JEQ ELSE	Yes-treat like a stmt-sep
0262 016C-0288 016E 8100	CI RB, ELSE**256	Hit an else-clause?
0263 0170 13E1	JEQ ELSE	Yes-treat like a stmt-sep
0264 0172 06A0 0174 0086'	NXTCHR BL @PGMCHR	Get next token
0265 0176 10E5	JMP NDCARE	And continue loop
0266 0178		
0267 0178 05E0 017A 00F6'	SKIPLN INCT @PGMPTR	Skip line number
0268 017C 10FA	JMP NXTCHR	And get next token
0269 017E		
0270 017E 06A0 0180 0174'	SKPSTR BL @PGMCHR	Get length of string/num
0271 0182 06C8	SWPB RB	Swap for add
0272 0184 A808 0186 017A'	A RB, @PGMPTR	Skip the string or num
0273 0188 04C8	CLR RB	Clear LSByte of character
0274 018A 10F3	JMP NXTCHR	And get next token


```
0276          *      Code to return to GPL to handle special case or
0277          *      an end-of-line return
0278 018C 0203  ERRFNN LI   R3,>16          FOR/NEXT NESTING
          018E 0016
0279 0190 1006          JMP   GPL05
0280 0192
0281 0192
0282 0192 0203  ENTER  LI   R3,>10          Load return code for ENTER
          0194 0010
0283 0196 1003          JMP   GPL05          Goto GPL
0284 0198
0285 0198
0286 0198 0243  SCNGPL ANDI R3,>7F00       Throw away GPL flag
          019A 7F00
0287 019C 0983          SRL   R3,B          Shift to use as index for rtn
0288 019E C660  GPL05  MOV  @RTNADD,*R9     Make sure right GROM addr
          01A0 001E
0289 01A2 A643          A     R3,*R9       Add offset to old GROM addr
0290 01A4 06A0          BL   @SAVREG       Save R8/R9 in CHAT/SUBSTK
          01A6 0000
0291 01A8 06A0          BL   @GETSTK       Restore old GROM address
          01AA 0000
0292 01AC 0460          B     @RESET        Goto GPL w/condition reset
          01AE 0000
```

```
0294 *****
0295 * Table of specially scanned statements
0296 * 2 bytes / special token
0297 * Byte 1 - token value
0298 * Byte 2 - "address" of special handler
0299 * If MSBit set then GPL and rest is offset from
0300 * the XML that got us here
0301 * If MSBit reset then 9900 code and is offset
0302 * from label OFFSET in this assembly of the
0303 * special case handler
0304 *****
0305 01B0 81 SCNTAB BYTE ELSE$, ELSE-OFFSET
      01B1 68
0306 01B2 82 BYTE SSEP$, SMTSEP-OFFSET
      01B3 72
0307 01B4 83 BYTE TREM$, SCAN25-OFFSET
      01B5 00
0308 01B6 84 BYTE IF$, IF-OFFSET
      01B7 0A
0309 01B8 85 BYTE GO$, IMPER-OFFSET
      01B9 10
0310 01BA 86 BYTE GOTO$, IMPER-OFFSET
      01BB 10
0311 01BC 87 BYTE GOSUB$, IMPER-OFFSET
      01BD 10
0312 01BE 88 BYTE RETUR$, IMPER-OFFSET
      01BF 10
0313 01C0 89 BYTE DEF$, >82
      01C1 82
0314 01C2 8A BYTE DIM$, >84
      01C3 84
0315 01C4 8C BYTE FOR$, FOR-OFFSET
      01C5 36
0316 01C6 92 BYTE INPUT$, IMPER-OFFSET
      01C7 10
0317 01C8 93 BYTE DATA$, DATA1-OFFSET
      01C9 1C
0318 01CA 96 BYTE NEXT$, NEXT-OFFSET
      01CB 4A
0319 01CC 9A BYTE REM$, SCAN25-OFFSET
      01CD 00
0320 01CE 9B BYTE ON$, IMPER-OFFSET
      01CF 10
0321 01D0 9D BYTE CALL$, >86
      01D1 86
0322 01D2 9E BYTE OPTIO$, >88
      01D3 88
0323 01D4 A1 BYTE SUB$, >8A
      01D5 8A
0324 01D6 A3 BYTE IMAGE$, IMAGE-OFFSET
      01D7 2E
0325 01D8 A7 BYTE SUBXT$, >8C
      01D9 8C
0326 01DA AB BYTE SUBND$, >8E
      01DB 8E
0327 01DC AA BYTE LINPU$, IMPER-OFFSET
      01DD 10
0328 01DE B0 BYTE THEN$, ELSE-OFFSET
      01DF 68
0329 0030 TABLEN EQU $-SCNTAB
```


SCAN LABEL	VALUE	DEFN	REFERENCES
\$	01E6'		0137 0329
C3	R 001A'	0061	0129
C4	R 0062'	0060	0154
CALL\$	009D	0080	0321
CBH40	0037'	0137	0163 0203
CBH94	R 008A'	0061	0165
CBHFF	R	0061	
DATA	R 00F8'	0060	0131 0211 0214
DATA\$	0093	0076	0317
DATA1	00E8'	0211	0317
DEF\$	0089	0071	0313
DIM\$	008A	0072	0314
DX10	0001	0003	0004 0107
ELSE	0134'	0239	0261 0263 0305 0328
ELSE\$	0081	0063	0262 0305
END\$	008B	0073	0338
ENTER	0192'	0282	0253
ERRFN	00D4'	0198	0135
ERRFNN	018C'	0278	0199
ERRIBS	009A'	0171	0177
ERRIMP	00E2'	0207	0218
ERRMS	005A'	0151	0146
ERRSYN	0110'	0225	0232 0241
EXTRAM	R 00F0'	0059	0132 0154 0158 0161 0213
FDR	0102'	0221	0315
FDR\$	008C	0074	0315
FDRNET	R 0128'	0057	0134 0234
GET	R 006E'	0060	0157
GET1	R 0052'	0060	0148
GETG	R 0076'	0060	0160
GETSTK	R 01AA'	0058	0291
GD\$	0085	0067	0309
GOSUB\$	0087	0069	0311
GOTO\$	0086	0068	0310
GPLO5	019E'	0288	0208 0226 0238 0279 0283
GPLO5L	00E6'	0208	0143 0152 0172
IBSTAB	01E0'	0336	0175
IF	00D6'	0203	0308
IF\$	0084	0066	0308
IMAGE	00FA'	0216	0212 0324
IMAGE\$	00A3	0083	0324
IMPER	00DC'	0205	0309 0310 0311 0312 0316 0320 0327
INPUT\$	0092	0075	0316
JNCARE	00C0'	0187	0125
LINPU\$	00AA	0086	0327
LINUM	R 0026'	0059	0132
LN\$	00C9	0090	0254
LNBUF	R 00F2'	0060	0213
NEXT	0116'	0228	0318
NEXT\$	0096	0077	0318
NOCARE	0142'	0247	0183 0187 0265
NUM\$	00C7	0088	0256
NXTCHR	0172'	0264	0206 0224 0235 0236 0268 0274
OFFSET	00CC'	0193	0192 0305 0306 0307 0308 0309 0310 0311 0312 0313 0314 0315 0316 0317 0318 0319 0320 0324 0327 0328 0329
ON\$	009B	0079	0320
OPTID\$	009E	0081	0322
P359	0000	0003	0003
PGMCHR	R 0180'	0057	0164 0264 0270
PGMPTR	R 0186'	0057	0162 0214 0267 0272

SCAN LABEL	VALUE	DEFN	REFERENCES									PAGE 0013
PRGFLG R	00FC'	0058	0194	0205	0216							
PUTSTK R	0004'	0058	0113									
RO	0000		0110	0122	0124	0134	0136	0138	0140	0141	0145	
			0155	0166	0167	0194	0205	0211	0216	0222	0223	
			0229	0230	0231	0231	0233	0239	0240			
ROLB R	0126'	0059	0234									
R1	0001		0149	0162								
R13	000D		0110									
R3	0003		0147	0151	0171	0180	0181	0189	0191	0192	0207	
			0225	0237	0278	0282	0286	0287	0289			
R6	0006		0173	0174	0175	0179	0180	0184	0185			
R8	0008		0169	0169	0175	0181	0247	0249	0251	0251	0254	
			0256	0258	0260	0262	0271	0272	0273			
R9	0009		0126	0129	0130	0288	0289					
RAMTOP R	0068'	0059	0155									
REM\$	009A	0078	0319	0339								
RESET R	01AE'	0058	0292									
RESOLV R	0044'	0060	0142									
RETUR\$	0088	0070	0312									
RTNADD R	01A0'	0057	0126	0130	0288							
SAVREG R	01A6'	0058	0290									
SCAN D	0000'	0105	0055									
SCAN05	0018'	0129	0123									
SCAN06	003A'	0140	0150									
SCAN07	0060'	0154	0133									
SCAN08	0074'	0160	0156									
SCAN09	007A'	0162	0159									
SCAN10	00AC'	0179	0127	0168								
SCAN11	00A0'	0173	0170									
SCAN12	00A2'	0174	0176									
SCAN15	00AE'	0180	0186									
SCAN20	00C2'	0189	0182									
SCAN25	00CC'	0194	0250	0252	0307	0319						
SCAN5A	0024'	0132	0195	0217								
SCAN5B	002C'	0134	0196									
SCAN6A	0048'	0145	0139									
SCAN9A	0084'	0164	0243									
SCNGPL	0198'	0286	0190									
SCNTAB	01B0'	0305	0180	0329								
SETREG R	0008'	0057	0114									
SKIPLN	0178'	0267	0255									
SKPSTR	017E'	0270	0257	0259								
SMTSEP	013E'	0243	0248	0306								
SSEP\$	0082	0064	0247	0306	0336							
STACK R	0040'	0058	0141									
STRIN\$	00C8	0089	0258									
SUB\$	00A1	0082	0323	0340								
SUBND\$	00A8	0085	0326									
SUBXT\$	00A7	0084	0325									
SYMTAB R	0056'	0059	0149									
TABLEN	0030	0329	0185									
TABSAV R	004E'	0059	0147									
THEN\$	00B0	0087	0260	0328								
TREM\$	0083	0065	0249	0307	0337							
VERMAC M		A0001	0003									
VERS	0000	0003	0004	0107								
XFLAG R	0136'	0057	0136	0163	0165	0166	0203	0221	0222	0229	0239	