

```

1  *   GENERAL AUTOMATION, INC.  ALL RIGHTS RESERVED
2  *****
3  *
4  *   PROGRAM NAME   FPH-07
5  *
6  *   MODEL NUMBER  RF007
7  *
8  *   PURPOSE       FORTRAN PHASE-07
9  *
10 *   PROGRAMMER    DICK WALLMANN, MODS-MARK FLEFIELD
11 *
12 * ***** REVISION LIST *****
13 *
14 *   PV DATE      SCO   BY   REASON FOR CHANGE
15 *   -----
16 *
17 *   01 11/16/70 NONE  INITIAL RELEASE
18 *
19 *****
20 *****
21 *STATUS-VERSION 1, MODIFICATION 0
22 *FUNCTION/OPERATION ANALYSES AND MODIFIES THE
23 * FOLLOWING STATEMENTS...
24 * *DEFINE FILE- DECODES STATEMENT, CHECKS FOR
25 * SYNTAX ERRORS AND DUPLICATE FILE NUMBERS
26 * OUTPUTS SIX WORD FILE TABLE INTO STATEMENT
27 * STRING AND MAKES ENTRY OF FILE NUMBER IN
28 * SYMBOL TABLE
29 * *CALL EXIT- DECODES AND CHECKS FOR SYNTAX ERR
30 * SETS UP NEW CODE AND INSERTS IN STRING
31 * *CALL LINK- DECODES AND CHECKS FOR SYNTAX ERR
32 * SETS UP NEW CODES AND INSERTS IN STRING
33 * ALL OTHER STATEMENTS ARE MOVED FROM THE INPUT
34 * STRING TO THE OUTPUT STRING.
35 *ENTRY POINTS-
36 * * START-PHASE 7 IS LOADED INTO CORE BY PHASE
37 * 6 VIA THE ROLRX ROUTINE AND EXECUTION
38 * BEGUN AT LOCATION START.
39 *INPUT-THE STATEMENT STRING AND SYMBOL TABLE BEING
40 * PROCESSED BY THE COMPILER
41 *OUTPUT-THE MODIFIED STATEMENT STRING AND SYMBOL
42 * TABLE AS PROCESSED BY THIS PHASE
43 *EXTERNAL REFERENCES-
44 * * SUBROUTINES-
45 * * ROLRX - PHASE ROLLING ROUTINE
46 * * OTHER PHASES-NONE
47 *EXITS-
48 * * NORMAL-IF OVERLAP OF STRING AND SYMBOL TABLE
49 * HAS NOT OCCURRED THEN PHASE 8 IS
50 * CALLED VIA ROLPX AND EXECUTED
51 * * ERRORS-A SYNTAX OR COMPILATION ERROR DOES
52 * NOT INTERRUPT PROCESSING. AN OVER-
53 * LAP CONDITION TERMINATES COMPILATION,
54 * SETS AN ERROR COND IN FCUM AND CALLS
55 * PHASE 8 VIA ROLRX. CONTROL IS THEN
56 * PASSED FROM PHASE TO PHASE UNTIL FR21
57 * * ERRORS DETECTED IN THIS PHASE ARE
58 * NUMBERS 71 AND 72.
59 *TABLES/WORK AREAS-STRING,SYMBOL TABLE,FORTRAN

```

```

60 * COMMUNICATION AREA
61 *ATTRIBUTES-NONE.
62 *NOTES
63 * 6 WORD DEFINE FILE TABLE CREATED BY DEFN FILE
64 * WORD 1 FILE NUMBER
65 * WORD 2 NUMBER OF RECORDS
66 * WORD 3 RECORD LENGTH IN WORDS
67 * WORD 4 ADDRESS OF ASSOCIATED VARIABLE
68 * WORD 5 ZEROS
69 * WORD 6 NUMBER OF RECORDS PER SECTOR
70 ABS REF CORE
71 * SYSTEM AND FORTRAN EQUATES
72 MEMRY EQU 4*320 CORE MAXIMUM CORE SIZE
73 PHSIZ EQU 4*320 MAXIMUM PHASE SIZE
74 OVERL EQU MEMRY-PHSIZ PHASES 2-29 START
75 FCOM EQU OVERL-22 FORTRAN COMM. TABLE
76 PHNTR EQU FCOM-56 PHASE TABLE
77 ROLRX EQU PHNTR-50 INTERPHASE CALL
78 * FORTRAN COMMUNICATION AREA
79 ORG FCOM BEGIN PHASE
80 SOFS BSS 1 START OF STRING
81 EOFS BSS 1 END OF STRING
82 SOFST BSS 1 START OF SYMBOL TABLE
83 SOFNS BSS 1 START OF NON-STMNT NUMBERS
84 SOFXT BSS 1 START OF SUBSC TEMPS
85 SOFGT BSS 1 START OF GENERATED TEMPS
86 EOFST BSS 1 END OF SYMBOL TABLE
87 COMMON BSS 1 NEXT AVAILABLE COMMON
88 CSIZE BSS 1 SIZE OF COMMON
89 ERROR BSS 1 OVERLAP ERROR
90 FNAME BSS 1 PROGRAM NAME
91 BSS 1 2ND WD OF NAME
92 SORF BSS 1 SURP (-) OR FUNCTION (+)
93 CCWD BSS 1 CONTROL CARD WORD
94 * BIT 15 TRANSFER TRACE
95 * BIT 14 ARITHMETIC TRACE
96 * BIT 13 EXTENDED PRECISION
97 * BIT 12 LIST SYMBOL TABLE
98 * BIT 11 LIST SUBPROGRAM NAMES
99 * BIT 10 LIST SOURCE PROGRAM
100 * BIT 9 ONE WORD INTEGERS
101 IOCS BSS 1 IOCS CONTROL CARD WORD
102 * SEE PHASE ONE FOR BIT PATTERNS
103 DFCNT BSS 1 DEFINE FILE COUNT
104 LCOMN BSS 2 SIZE OF INSKEL COMMON
105 ICCR BSS 2 IOCS CONTROL CARD ERROR
106 BSS 2 SYSTEM LOADER USE
107 * END OF FORTRAN COMMUNICATION
108 * AREA
109 ORG OVERL PHASE ENTRY
110 START LDX 17 SOFS INITIALIZE INPUT STRING PT
111 LDX 12 SOFS INITIALIZE OUTPUT STRING P
112 * CHECK FOR OVERLAP ERROR
113 LD ERROR LOAD OVERLAP ERROR WD
114 BSC L EXIT,Z BRANCH IF SO
115 * INITIALIZE FOR STATEMENT
116 CK1 LDX L3 V SET INDEX FOR CONSTANTS
117 STX 1 10SV1 STRING ADDR. I/P STRING
118 STX L2 10SV2 STRING ADDR. O/P STRING
119 LD 1 0 PICK UP ID WORD FROM STRNG

```

120		STO	3	ORIG-V	SAVE IN TEMP. STO
121		AND	3	HF803-V	MASK OUT STMT NORM
122		PST	3	OUT-V	STORE IN O/P STRING
123		LD	1	-1	PICK UP MASKED ID WORD
124		SLA		15	TEST BIT 15 FOR STMT NO.
125		RSC	L	*+2,-	BR IF NO STMT NO.
126		LD	1	0	STMT NO. FOUND
127		BSJ	3	OUT-V	MOVE STMT NO. TO STRING
128	*				TEST FOR DEFINE FILE STMT
129		LD	3	ORIG-V	GET STRING ID WD -TEMP STO
130		AND	3	HF802-V	MASK OUT STATEMENT NORM
131		S	3	DFILE-V	TEST FOR DEFINE FILE ID
132		RSC	L	DF1,+	BRANCH IF YES.
133	*				TEST FOR CALL EXIT STMT
134		S	3	CEXIT-V	TEST FOR CALL EXIT ID
135		RSC	L	CE1,+	BRANCH IF YES
136	*				TEST FOR CALL LINK STMT
137		S	3	CLINK-V	TEST FOR CALL LINK ID
138		RSC	L	CL1,+	BRANCH IF YES.
139	*				END STATEMENT
140		S	3	ENDCN-V	TEST FOR END ID
141		RSC	L	FD1,+	BRANCH IF YES
142	*				MOVE TO NEXT STATEMENT
143		LDX	L1	*-*	GET I/P STRING ADDR.
144	IDSV1	EOU		*-1	
145		LDX	I2	IDSV2	GET O/P STRING ADDR
146		LD	3	ORIG-V	GET STMT ID WORD
147		STO	1	0	STORE STMT ID BACK IN I/P
148		SLA		5	SHIFT OFF ALL BITS OUT
149		SRA		7	STATEMENT NORM
150		STO		*+1	
151		LDX	L3	*-*	GET NO WDS THIS STMT
152	CK2	LD	1	0	MOVE STATEMENT FROM I/P
153		STO	2	0	TO O/P STRING
154		MOX	1	1	INCR. I/P COUNTER
155		MOX	2	1	INCR. O/P COUNTER
156		HDX	3	-1	CONTINUE UNTIL ALL WDS IN
157		MOX		CK2	*STATEMENT MOVED
158		MOX		CK1	GET NEXT STATEMENT
159	*				PROCESS CALL EXIT STATEMENT
160	*				CHECK FOR SYNTAX ERROR
161	*				DELETE INTERPHASE COMM BIT IN ID WOR
162	*				ADD 2 FLAG WORDS(HEX 5F00) TO STRING
163	*				TO INDICATE CALL EXIT
164	*				CALL EXIT STATEMENT
165	*				REMOVE BIT 14 OF ID WORD
166	CE1	LD	I	IDSV2	GET ID WD FROM STRING
167		AND	3	HEFFD-V	MASK OUT BIT 14
168		STO	I	IDSV2	SAVE BACK IN ID WORD
169	*				SEMT-COLON
170		LD	1	0	GET 1ST DATA WD IN STMT
171		S	3	SEMI-V	TEST FOR SEMI-COLON
172		RSC	L	CE3,+	BRANCH IF SEMI-COLON
173	*				PUT ERROR NO. 3
174	ER3	LD	3	D0003-V	SYNTAX ERROR CODE
175		BSI	3	ERR-1-V	BRANCH TO OUTPUT ERROR.
176	*				SEE IF STRING IS TO BE MOVED
177	CE3	STX	L1	TEMP	SAVE XR1
178		LD	3	TEMP-V	LD A WORD FROM THE STRING
179		STX	L2	TEMP	STORE XR2 IN HERE

```

180      S      3 TEMP-V      COMPARE XR1 AND XR2
181      RSC    L      CE4,Z   BR ON NO COMPARE
182      *
183      MDX    I      1
184      *
185      LD     L      EOF5     GET ADDR OF STRING END
186      S      3 TEMP-V      SUBTRACT STRING START
187      STO    *+1         AND
188      LDX   L3 *-*        LOAD IT INTO XR3
189      MDX   3 1          INCREMENT XR3 BY 1
190      LDX   I2 EOF5     GET THE LAST WD OF STRING
191      LD    2 0          LD A STRING WD
192      STO   2 1          MOVE IT UP ONE ADDR
193      MDX   2 -1         DECREMENT XR2
194      MDX   3 -1         DECREMENT XR3
195      MDX   *-5          LOOP BACK UNTIL STRING MVD
196      MDX   L EOF5,1    INCREASE END STRING ADDR
197      LDX   L3 V        RESTORE XR3
198      LDX   I2 TEMP     RESTORE XR2
199      *
200      LD     L      EOF5     GET STRING END AND
201      S      L      EOF5T    COMPARE TO SYMBOL TABLE
202      RSC    L      CE4,++   BR ON NO OVERLAP
203      LDX   I      1        ELSE SET XR1 TO ERROR CODE
204      RSC    L      EXIT     EXIT PHASE ON ERROR
205      *
206      CE4   LD     3 H5F00-V  GET SPEC CALL EXIT WORD
207      RST   3 OUT-V        PUT IT OUT ON STRING
208      RST   3 OUT-V        TWICE IE 5F005F00
209      MDX   I      -1       DECREMENT COUNTER AND LOOP
210      MDX   CK1          BACK UNTIL COUNT IS ZERO
211      *
212      *
213      *
214      LD     I      -1       GET LAST WD PRIOR TO MOVE
215      S      3 SEMI-V      COMPARE TO SEMICOLON
216      MDX   ER3          ELSE GO TO ERROR 3 WORD
217      *
218      CL1   LD     I      0   PICK UP WD. IN I/P STRING
219      S      3 LPAR-V      TEST FOR LEFT PARENTHESES
220      RSC    L      ER3,Z   BRANCH TO ERROR IF NOT
221      *
222      MDX   I      1
223      *
224      LD     I      0
225      RSC    L      ER3,-   BRANCH IF OPERATOR
226      SLA   2
227      RSC    L      *+2,-   BRANCH IF ALPHABETIC
228      RSC    L      ER3,C   BRANCH IF NUMERIC
229      *
230      LD     I      0
231      RST   3 OUT-V        LOAD WD FROM INPUT STRING
232      *
233      LD     3 H8000-V     BRANCH TO PUT WORD O/P STR
234      RST   3 OUT-V        OUTPUT BLANK SECOND WORD
235      *
236      LD     I      -1
237      RSC    L      CL2,-   LOAD CONSTANT BLANK
238      *
239      *
240      *
241      *
242      *
243      *
244      *
245      *
246      *
247      *
248      *
249      *
250      *
251      *
252      *
253      *
254      *
255      *
256      *
257      *
258      *
259      *
260      *
261      *
262      *
263      *
264      *
265      *
266      *
267      *
268      *
269      *
270      *
271      *
272      *
273      *
274      *
275      *
276      *
277      *
278      *
279      *
280      *
281      *
282      *
283      *
284      *
285      *
286      *
287      *
288      *
289      *
290      *
291      *
292      *
293      *
294      *
295      *
296      *
297      *
298      *
299      *
300      *
301      *
302      *
303      *
304      *
305      *
306      *
307      *
308      *
309      *
310      *
311      *
312      *
313      *
314      *
315      *
316      *
317      *
318      *
319      *
320      *
321      *
322      *
323      *
324      *
325      *
326      *
327      *
328      *
329      *
330      *
331      *
332      *
333      *
334      *
335      *
336      *
337      *
338      *
339      *
340      *
341      *
342      *
343      *
344      *
345      *
346      *
347      *
348      *
349      *
350      *
351      *
352      *
353      *
354      *
355      *
356      *
357      *
358      *
359      *
360      *
361      *
362      *
363      *
364      *
365      *
366      *
367      *
368      *
369      *
370      *
371      *
372      *
373      *
374      *
375      *
376      *
377      *
378      *
379      *
380      *
381      *
382      *
383      *
384      *
385      *
386      *
387      *
388      *
389      *
390      *
391      *
392      *
393      *
394      *
395      *
396      *
397      *
398      *
399      *
400      *
401      *
402      *
403      *
404      *
405      *
406      *
407      *
408      *
409      *
410      *
411      *
412      *
413      *
414      *
415      *
416      *
417      *
418      *
419      *
420      *
421      *
422      *
423      *
424      *
425      *
426      *
427      *
428      *
429      *
430      *
431      *
432      *
433      *
434      *
435      *
436      *
437      *
438      *
439      *
440      *
441      *
442      *
443      *
444      *
445      *
446      *
447      *
448      *
449      *
450      *
451      *
452      *
453      *
454      *
455      *
456      *
457      *
458      *
459      *
460      *
461      *
462      *
463      *
464      *
465      *
466      *
467      *
468      *
469      *
470      *
471      *
472      *
473      *
474      *
475      *
476      *
477      *
478      *
479      *
480      *
481      *
482      *
483      *
484      *
485      *
486      *
487      *
488      *
489      *
490      *
491      *
492      *
493      *
494      *
495      *
496      *
497      *
498      *
499      *
500      *
501      *
502      *
503      *
504      *
505      *
506      *
507      *
508      *
509      *
510      *
511      *
512      *
513      *
514      *
515      *
516      *
517      *
518      *
519      *
520      *
521      *
522      *
523      *
524      *
525      *
526      *
527      *
528      *
529      *
530      *
531      *
532      *
533      *
534      *
535      *
536      *
537      *
538      *
539      *
540      *
541      *
542      *
543      *
544      *
545      *
546      *
547      *
548      *
549      *
550      *
551      *
552      *
553      *
554      *
555      *
556      *
557      *
558      *
559      *
560      *
561      *
562      *
563      *
564      *
565      *
566      *
567      *
568      *
569      *
570      *
571      *
572      *
573      *
574      *
575      *
576      *
577      *
578      *
579      *
580      *
581      *
582      *
583      *
584      *
585      *
586      *
587      *
588      *
589      *
590      *
591      *
592      *
593      *
594      *
595      *
596      *
597      *
598      *
599      *
600      *
601      *
602      *
603      *
604      *
605      *
606      *
607      *
608      *
609      *
610      *
611      *
612      *
613      *
614      *
615      *
616      *
617      *
618      *
619      *
620      *
621      *
622      *
623      *
624      *
625      *
626      *
627      *
628      *
629      *
630      *
631      *
632      *
633      *
634      *
635      *
636      *
637      *
638      *
639      *
640      *
641      *
642      *
643      *
644      *
645      *
646      *
647      *
648      *
649      *
650      *
651      *
652      *
653      *
654      *
655      *
656      *
657      *
658      *
659      *
660      *
661      *
662      *
663      *
664      *
665      *
666      *
667      *
668      *
669      *
670      *
671      *
672      *
673      *
674      *
675      *
676      *
677      *
678      *
679      *
680      *
681      *
682      *
683      *
684      *
685      *
686      *
687      *
688      *
689      *
690      *
691      *
692      *
693      *
694      *
695      *
696      *
697      *
698      *
699      *
700      *
701      *
702      *
703      *
704      *
705      *
706      *
707      *
708      *
709      *
710      *
711      *
712      *
713      *
714      *
715      *
716      *
717      *
718      *
719      *
720      *
721      *
722      *
723      *
724      *
725      *
726      *
727      *
728      *
729      *
730      *
731      *
732      *
733      *
734      *
735      *
736      *
737      *
738      *
739      *
740      *
741      *
742      *
743      *
744      *
745      *
746      *
747      *
748      *
749      *
750      *
751      *
752      *
753      *
754      *
755      *
756      *
757      *
758      *
759      *
760      *
761      *
762      *
763      *
764      *
765      *
766      *
767      *
768      *
769      *
770      *
771      *
772      *
773      *
774      *
775      *
776      *
777      *
778      *
779      *
780      *
781      *
782      *
783      *
784      *
785      *
786      *
787      *
788      *
789      *
790      *
791      *
792      *
793      *
794      *
795      *
796      *
797      *
798      *
799      *
800      *
801      *
802      *
803      *
804      *
805      *
806      *
807      *
808      *
809      *
810      *
811      *
812      *
813      *
814      *
815      *
816      *
817      *
818      *
819      *
820      *
821      *
822      *
823      *
824      *
825      *
826      *
827      *
828      *
829      *
830      *
831      *
832      *
833      *
834      *
835      *
836      *
837      *
838      *
839      *
840      *
841      *
842      *
843      *
844      *
845      *
846      *
847      *
848      *
849      *
850      *
851      *
852      *
853      *
854      *
855      *
856      *
857      *
858      *
859      *
860      *
861      *
862      *
863      *
864      *
865      *
866      *
867      *
868      *
869      *
870      *
871      *
872      *
873      *
874      *
875      *
876      *
877      *
878      *
879      *
880      *
881      *
882      *
883      *
884      *
885      *
886      *
887      *
888      *
889      *
890      *
891      *
892      *
893      *
894      *
895      *
896      *
897      *
898      *
899      *
900      *
901      *
902      *
903      *
904      *
905      *
906      *
907      *
908      *
909      *
910      *
911      *
912      *
913      *
914      *
915      *
916      *
917      *
918      *
919      *
920      *
921      *
922      *
923      *
924      *
925      *
926      *
927      *
928      *
929      *
930      *
931      *
932      *
933      *
934      *
935      *
936      *
937      *
938      *
939      *
940      *
941      *
942      *
943      *
944      *
945      *
946      *
947      *
948      *
949      *
950      *
951      *
952      *
953      *
954      *
955      *
956      *
957      *
958      *
959      *
960      *
961      *
962      *
963      *
964      *
965      *
966      *
967      *
968      *
969      *
970      *
971      *
972      *
973      *
974      *
975      *
976      *
977      *
978      *
979      *
980      *
981      *
982      *
983      *
984      *
985      *
986      *
987      *
988      *
989      *
990      *
991      *
992      *
993      *
994      *
995      *
996      *
997      *
998      *
999      *
1000      *

```

240	*				MOVE POINTER
241		MOX	1	1	
242	*				RIGHT PARENTHESIS
243	CL2	LD	1	-1	
244		S	3	RPAR-V	
245		BSC	L	ER3,Z	BRANCH IF NOT R PAR
246	*				RESHAPE CALL LINK
247		LD	2	-1	MOVE SECOND WORD OF NAME
248		STO	2	+1	TWO LOCS
249		LD	2	-2	MOVE FIRST WORD OF NAME
250		STO	2	-1	ONE LOCATION
251		LD	3	H5800-V	INSERT HEX CONST 5800
252		STO	2	-2	BEFORE FIRST AND 2ND NAMES
253		STO	2	0	
254		PSI	3	OUT-V	INCR XR1, XR2, AND NFORM
255		LD	2	0	
256		PSI	3	OUT-V	INCR XR1, XR2, AND NFORM
257	*				MOVE POINTER
258		MOX	1	-1	I/P STRING TOO FAR
259	*				SEMICOLON
260		BSC	L	CK1	GO PROCESS NEXT STATEMENT
261	*				DEFINE FILE STATEMENT
262	*				CHECK SYNTAX OF STATEMENT
263	*				CHECK FOR DUPLICATE FILE NOS.
264		MOX	1	-1	MOVE POINTER
265	*				CHECK FOR SEMICOLON
266	*				DEFN FILE ILLEGAL IN SUBP
267	DF1	LD	L	SORF	GET SUBPROGRAM INDICATOR
268		BSC	L	ER72,Z	BRANCH ERROR IF SUBPROGRAM
269	*				CLEAR FCT
270		STO	3	FCT-V	PUT ZERO FROM ACC IN FCT
271	*				COLLECT VALID CONSTANT
272	DF5	PSI	3	COLL-V	FROM NEXT 2 WDS (FILE NO.)
273		MOX	L	FCT,6	INCREMENT FILE COUNT
274	*				CHECK DUPLICATE OR
275	*				EXCESSIVE DEFINE FILES.
276		PSI	3	FILES-V	
277	*				OUTPUT FILE NUMBER
278		PSI	3	OUT-V	IN BINARY
279	*				LEFT PARENTHESIS
280		LD	1	0	GET NEXT WD. FROM STRING
281		S	3	LPAR-V	TEST FOR LEFT PARENTHESIS
282		BSC	L	ER71,Z	BRANCH IF NO
283	*				MOVE POINTER
284		MOX	1	1	
285	*				COLLECT VALID CONSTANT
286	*				GET NUMBER OF RECORDS, CONVERT TO BI
287		PSI	3	COLL-V	
288	*				OUTPUT NUMBER OF RECORDS
289		PSI	3	OUT-V	
290	*				COMMA
291		LD	1	0	TEST NEXT OP FOR COMMA
292		S	3	COMA-V	
293		BSC	L	ER71,Z	BRANCH IF NO
294	*				MOVE POINTER
295		MOX	1	1	
296	*				COLLECT VALID CONSTANT
297		PSI	3	COLL-V	AND CONVERT REC. LNO TO BI
298	*				OUTPUT RECORD LENGTH
299		PSI	3	OUT-V	

300	*			RECORD LENGTH LESS THAN 32
301		LD	2 -1	LOAD RECORD LENGTH
302		S	3 DC320-V	COMPARE WITH 320
303		RSC	L EF71,7-	BR IF RECORD LENGTH GT 320
304	*			COMMA
305		LD	1 0	GET NEXT WORD IN I/O STRING
306		S	3 COMA-V	TEST FOR COMMA
307		RSC	L EF71,7	BRANCH IF NO
308	*			U (UNEDITED)
309		LD	1 1	TEST FOR U, BUT DO NOT USE
310		S	3 U-V	
311		RSC	L EF71,7	BRANCH IF NO
312	*			COMMA
313		LD	1 2	TEST FOR COMMA
314		S	3 COMA-V	
315		RSC	L EF71,7	BRANCH IF NO
316	*			MOVE POINTER
317		MDX	1 3	BYPASS PREV. 3 WDS r/r STR
318	*			GATHER CONSTANT
319		RSI	3 GATH-V	GET VAR. NAME REF. FILE
320		LD	3 WD2-V	GET 2ND WD OF NAME
321		SRA	1	
322		OR	3 H8000-V	ADD FLAG
323		STO	3 WD2-V	
324	*			VALID NAME
325		LD	3 WD1-V	GET FIRST WD OF NAME
326		SLA	2	
327		RSC	L *+2,-	BRANCH IF ALPHA
328		RSC	L EF71,0	BRANCH IF NUMERIC
329	*			VARIABLE IN SYMBOL TABLE
330		LD	L SCFNS	FIND DIFF BET END OF NON-
331		S	L ECFST	STMT NOS AND SYM TBL
332		A	3 D0003-V	INCR BY 3
333		STO	3 TEMP-V	
334		STX	2 XF2R	SAVE O/P STRING ADDR
335		LDX	I2 SCFNS	GET LAST WD SYMBOL TABLE
336		MDX	DF3	BRANCH TO MODIFY
337	DF2	LD	2 1	COMPARE SYM TBL ENTRY
338		S	3 WD1-V	WITH STRING WORD
339		RSC	L DF3-1,7	BRANCH NOT CURRENT ENTRY
340		LD	2 2	COMPARE 2ND WD. IN SYM TBL
341		S	3 WD2-V	WITH 2ND STRING WORD
342		RSC	L DF4,+-	BRANCH IF FOUND
343		MDX	2 -3	DECR SYM TBL ADDRESS
344	DF3	MDX	L TEMP,-3	DECR. SYM TBL COUNTER
345		MDX	DF2	GO TO NXT TBL ENTRY UNLESS
346	*			TABLE FINISHED
347	*			PUT IN SYMBOL TABLE
348		LD	3 WD1-V	GET FIRST WD OF NAME
349		STO	2 1	SAVE IN SYMBOL TABLE+1
350		LD	3 WD2-V	GET SECOND WD. OF NAME
351		STO	2 2	SAVE IN SYMBOL TABLE+2
352		LD	3 H4000-V	GET ID FOR INTEGER VAR.
353		STO	2 0	SAVE IN SYMBOL TABLE +0
354	*			ADJUST SYM TBL POINTERS
355		MDX	L SCFGT,-3	
356		MDX	L SCFAT,-3	
357		MDX	L ECFST,-3	
358	*			TEST FOR INTEGER VARIABLE
359		LD	3 WD1-V	GET FIRST WD OF NAME

360	AND	3	H7E00-V	MASK OUT ALL BUT 1ST CHAR
361	S	3	H1200-V	SEE IF GREATER THAN *L*
362	BSC	L	ER71,Z+	BRANCH IF NO
363	S	3	H1800-V	SEE IF LESS THAN *O*
364	BSC	L	ER71,Z-	BRANCH IF NO.
365	*			CHECK LEGAL VARIABLE
366	DF4	LD	2 0	GET SYMBOL TABLE ID WD
367	AND	2	H0FDF-V	TEST IF ID WORD CONTAINS
368	FOR	3	H4000-V	FLAG FOR VARIABLE INTEGER
369	BSC	L	ER71,Z	BRANCH IF NO
370	*			INDICATE DEFINED
371	LD	2	0	GET SYMBOL TABLE ID WORD
372	OP	3	H0020-V	ADD IN DEFINED VARIABLE
373	STO	2	0	FLAG
374	*			COMPUTE S.T. POINTER
375	STX	2	TEMP	SAVE SYMBOL TABLE ADDR.
376	LD	L	SOEST	FIND DIFF BET. START OF
377	S	3	TEMP-V	SYMBOL TABLE AND CUR LOC.
378	SRT		16	
379	D	3	D0003-V	DIVIDE BY 3 AND
380	A	3	H8001-V	MAKE RLTV TO SYMBOL TABLE
381	*			OUTPUT VARIABLE
382	LDX	L2	*-*	GET STRING ADDR O/P
383	AKZR	FOU	*-]	ADD SYMBOL TABLE ENTRY
384	BST	2	OUT-V	TO O/P STRING
385	*			RIGHT PARENTHESIS
386	LD	1	0	GET NEXT WD OF STRING
387	S	3	PFAR-V	TEST FOR RIGHT PARENTHESIS
388	BSC	L	ER71,Z	BRANCH IF NO
389	*			MOVE POINTER BACK BY 1
390	MOX	1	-1	TO ALLOW FOR INCR. BY OUT
391	*			OUTPUT ZERO
392	SLA		16	
393	PSI	3	OUT-V	FIFTH WD OF FILE BLOCK
394	*			COMPUTE RECORDS/SECTOR
395	LD	3	D0320-V	NO REC/SECTOR = 320
396	SRT		16	DIVIDED BY
397	D	2	-3	NUMBER WDS/RECORD
398	*			OUTPUT RECORDS/SECTOR
399	PSI	2	OUT-V	
400	BOX	1	1	MOVE POINTER
401	*			COMMA
402	LD	1	-1	
403	S	3	COMA-V	
404	BSC	L	DF5,+-	BRANCH IF YES
405	*			SEMI-COLON
406	LD	1	-1	
407	S	3	SEMI-V	
408	BSC	L	ER71,Z	BRANCH IF YES
409	*			UPDATE FILE COUNT
410	LD	L	DFCNT	INTERPHASE FILE WD. COUNT
411	A	3	FCT-V	NO. WDS THIS FILE DFSC
412	STO	L	DFCNT	SAVE
413	BSC	L	CK1	RR TO PROCESS NEW SYMUT
414	*			PUT ERROR 71
415	ER71	LD	3 D0071-V	SYNTAX ERROR DEFINE FILE
416		MOX	ERR	GO PUT OUT ERROR
417	*			PUT ERROR 72
418	ER72	LD	3 D0072-V	DUPLICATE DEFINE FILE NO.
419		MOX	ERR	GO PUT OUT ERROR

Address	Symbol	Value	Description
420	*		CONSTANTS
421	V	EQU **128	RLTV ADDR FACTOR FOR CON
422	HFFFF	DC /FFFF	MASK FOR STRING ID WD
423	HF803	DC /F803	MASK FOR STRING ID WD
424	HF802	DC /F802	MASK FOR STRING ID WD
425	HDFDE	DC /DFDE	SYMBOL TABLE MASK
426	HA00R	DC /A00R	ERROR ID WORD
427	H8001	DC /8001	CON FOR COMP SYM TBL DT
428	H8000	DC /8000	CON USED FOR BLANK WORD
429	H7E00	DC /7E00	MASK FOR INT. WD. TEST
430	H633R	DC /633R	CALL EXIT CONSTANT
431	H5F00	DC /5F00	CALL EXIT CONSTANT
432	H5800	DC /5800	CALL LINK CONSTANT
433	H4000	DC /4000	SYM TBL ID. FOR INTEGER
434	H1800	DC /1800	TEST FOR INTEGER VARIABLE
435	H1200	DC /1200	TEST FOR INTEGER VARIABLE
436	H0020	DC /0020	SYMBOL TABLE MASK
437	H0004	DC /0004	USED TO INCR STMT NORM
438	H0001	DC /0001	USEFUL CONSTANT
439	D000R	DC 3	USEFUL CONSTANT
440	D0009	DC 9	CONSTANT TEST NUMERIC
441	D0010	DC 10	CONSTANT USED BIN. CONVERT
442	D0071	DC 71	ERROR FLAG NO. 71
443	D0072	DC 72	ERROR FLAG NO. 72
444	D0320	DC 320	MAX. REC. SIZE DEFINE FILE
445	FCT	DC *-*	NO. WDS. IN DEFN FILE O/P
446	TEMP	DC *-*	TEMPORARY STORAGE AREA
447	TEMPA	DC *-*	TEMPORARY STORAGE AREA.
448	ORIG	DC *-*	ORIGINAL ID WORD
449	FILET	DC 00DT	LOC OF FILE DESCRIPTOR TBL
450	FCNT	DC 75	MAX OF 75 DEFINED FILES
451	DFILE	DC /F000	DEFN FILE ID FLAG TEST
452	CEXIT	DC /E002-/E000	CALL EXIT ID FLAG TEST
453	CLINK	DC /E000-/E002	CALL LINK ID FLAG TEST
454	ENDCN	DC /I000-/E000	END ID TEST
455	SEMI	DC /1E	SEMI-COLON
456	LPAR	DC /00	(
457	RPAR	DC /10	)
458	COMA	DC /2B	,
459	J	DC /C800	J
460		BSS E 0	
461	WD1	DC *-*	TEMP STO USED TO CONVERT
462	WD2	DC *-*	VARIABLE CONSTANTS (PACKED)
463	*		THE OUTPUT ROUTINE STORES THE
464	*		THE CONTENTS OF THE A REGISTER
465	*		AT XR2, INCREMENT XR1 AND XR2,
466	*		AND INCREMENTS THE NORM OF THE
467	*		STATEMENT.
468	OUT	DC *-*	BSI ENTRY POINT
469	STO	2 0	SAVE ACC IN O/P STRING
470	NDX	1 1	INCREMENT I/P STRING CTR
471	NDX	2 1	INCREMENT O/P STRING CTR
472	NDX	L *-*,4	INCREMENT STMT NORM BY 1
473	IDSVP	EQU *-1	
474	RSC	I 0 JT	RETURN
475	*		THE COLLECT CONSTANT ROUTINE
476	*		CALLS GATHER CONSTANT ROUTINE
477	*		AND THE CONVERTS THE CONSTANT
478	*		TO BINARY. THE ROUTINE GOES TO
479	*		ERROR NUMBER 71 IF CONSTANT IS



```

480 *                INVALID, ZERO OR OVER 32767.
481 COLL DC          *-*          BSI ENTRY POINT
482          RST      3 GATH-V     BRANCH TO GET 2 WD CONST
483          SLA      16
484          STO      3 TEMP-V     ZERO TEMP STO WD
485 COLL1 LD         3 WD1-V      GET 1ST OF 2 WD CONSTANT
486          SLA      2
487          RSC      L *+3,C     TEST FOR POSSIBLE NUMERIC
488          RSC      L ER71,Z     ZERO CON, SYNTAX ERROR
489          MDX      COLL2      BRANCH TO RETURN
490          RSC      L ER71,-    TEST FOR ZERO, INVALID CON
491          SLA      1          SHIFT OFF ONE BIT
492          SPA      12         PUT BITS 3-9 INTO BITS9-30
493          STO      3 TEMP-V     SAVE TEMPORARILY
494          S        3 D0009-V    DETERMINE IF NO. BETWEEN
495          RSC      L ER71,-2   0 AND 9. ERROR IF NOT
496          LD        3 TEMP-V     MULT PREV. CON BY 10
497          I        3 D0010-V
498          SLT      1          TEST FOR OVERFLOW
499          RSC      L ER71,Z     NO. GT 32767, ERROR
500          SLT      15         PUT PREV CON IN ACC
501          A        3 TEMP-V     ADD CURRENT WD
502          STO      3 TEMP-V     SAVE IN PREV. CON
503          RSC      L ER71,+Z    ERROR IF NO. GT 32767
504          LDD      3 WD1-V     GET NEXT 6 BITS WD1-WD2
505          SLT      6
506          STD      3 WD1-V     RESET WD1-WD2
507          MDX      COLL1      GO PROCESS NEXT 6 BITS
508 COLL2 LD         3 TEMP-V
509          RSC      L ER71,+     GO SET UP FOR ERROR
510          RSC      I COLL      BR IF LE 0 TO ERROR
511 *                RETURN
512 *                THE GATHER CONSTANT ROUTINE
513 *                PICK TWO WORD CONSTANT AND
514 *                PUT THEM IN WD1 AND WD2. POINTS
515 *                REMAINS ON LAST WORD USED. ON
516 *                ERRORS THE ROUTINE GOES TO
517 *                ERROR 71.
518 GATH DC          *-*          BSI ENTRY
519          LD         1 0        GET WORD FROM I/P STRING
520          STO      3 WD1-V     SAVE IN WD1
521          SLA      16
522          STO      3 WD2-V     ZERO WD2.
523          LD         1 1        TEST 2ND WD I/P STRING
524          RSC      L GATH1,-    RETURN IF WD NOT FLAGGED
525          SLA      1          REMOVE FLAG.
526          STO      3 WD2-V     SAVE WORD 2.
527          MDX      1 1        INCREMENT I/P STRING COUNT
528          LD         1 1        TEST 3RD WD OF STRING FOR
529          RSC      L ER71,+Z    VAR. SYNTAX ERR IF FOUND
530 GATH1 RSC        I GATH      RETURN
531 *                THE ERROR ROUTINE REPLACES THE
532 *                STATEMENT WITH AN ERROR MESSAGE
533 *                THE ERROR NUMBER IS IN THE
534 *                ACCUMULATOR WHEN COMING HERE
535          DC          *-*          ENTRY POINT
536 ERR  LD          11 IUSV1     GET STRING I/P IN WD ADDR
537          LD          12 IUSV2   GET STRING O/P IN WD ADDR
538          STO      3 TEMP-V     STORE ERROR CONST TEMP.
539          LD         3 ORIG-V    GET ID WD FROM I/P STRING

```

```

540 AND 3 H0001-V GET STMT NO. FLAG FROM ID
541 A 3 HA000-V INCR BY ERROR ID WORD.
542 RSC E TEST IF STMT NO. PRESENT
543 A 3 H0004-V INCR. NORM IF STMT HAS NO.
544 STO 2 0 SAVE ID WORD IN O/P STRING
545 SLA 15 TEST FOR NUMBERED STMT
546 RSC L *+3,- BRANCH IF NO
547 LD 1 1 MOVE NUMBER FROM I/P TO
548 STO 2 1 O/P STRING
549 MDX 2 1 INCR O/P STRING COUNT
550 LD 3 TEMP-V GET ERROR NO.
551 STO 2 1 PUT IN O/P STRING
552 MDX 2 2 INCR O/P STRING CTR
553 * SKIP OVER REMAINDER OF THE
554 * STATEMENT.
555 SKIP LD 3 ORIG-V GET STMT ID I/P STRING
556 SLA 5
557 SPA 7 DECODE STMT NORM
558 STO *+1 INCR I/P STRING CTR BY
559 MDX L1 *-* STATEMENT NORM
560 RSC L CK1 GO PROCESS NEXT STMT
561 * THIS SUBROUTINE SCANS THE TABLE
562 * OF DEFINE FILE NUMBERS TO ENSUR
563 * THAT THERE ARE NO DUPLICATES 0
564 * THAT THERE ARE NOT MORE THAN
565 * 75 DEFINED FILES.
566 * ACC CONTAINS BIN FILE NO.
567 FILES DC *-* BSI ENTRY POINT
568 STO 3 TEMP-V SAVE ACC.
569 LD 3 FCNT-V TEST IF NO. FILES LEFT
570 S 3 H0001-V IS GT 1
571 RSC L ER72,+ IF NOT, OVERLAP ERROR
572 STO 3 FCNT-V
573 LD 3 FILET-V GET LOC OF FILE TABLE
574 STO FL1+1
575 FL1 LD L *-* GET ENTRY FROM TABLE
576 RSC L FL2,+ IF ENTRY=0, ADD FILE NO.
577 S 3 TEMP-V COMPARE ENTRY WITH FILE NO
578 RSC L ER72,+ EQUAL, DUP FILE ERROR
579 MDX L FL1+1,-1 MODIFY FILE TABLE LOC.
580 MDX FL1 GO COMPARE ENTRY NEXT FILE
581 FL2 MDX L FILET+1 INCR. FILE TABLE LOC
582 LD 3 TEMP-V GET BINARY FILE NO.
583 STO I FILET SAVE IN FILE TABLE
584 RSC I FILES RETURN
585 * RESTORE STRING POINTERS
586 EDI MDX 2 -1
587 STX L2 EOF5
588 * READ AND GO TO NEXT PHASE
589 EXIT RST L ROLRX CALL DOWN PHASE 8
590 DC 08 NEXT PHASE NUMBER
591 * DEFINE FILE NUMBER TABLE
592 ODDT DC 0
593 RSC 75 DEFINED FILE TABLE
594 DIFF DC /A000-/8000 ERROR STMT ID
595 RSC OVERL-#+320*2 PHASE-07 PATCH AREA

```