

OPTIONS NODECK,LIST,XREF,NOREL,OBJ(P)

THE LIST OF OPTIONS USED DURING THIS ASSEMBLY IS-- NODECK,LIST,XREF,NOREL,OBJ

0000	1	#KRVLA	START	0
	2		PRINT	ON,NODATA
	3	*	@SYS	EXP-N
	214+		PRINT	ON
	215	*	@FXD	EXP-N
	620+		PRINT	ON
	621	*	@CAN	EXP-N
	724+		PRINT	ON
	725	*	@ERM	EXP-N
	1347+		PRINT	ON
	1348	*	@SPF	EXP-N
	1811+		PRINT	ON
	1812	*	@B@E	EXP-N
	2712+		PRINT	ON

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE	3
	2714				*****				
	2715	*	5703-XM1		COPYRIGHT IBM CORP. 1970				*
	2716	*			REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083				*
	2717	*							*
	2718				*****				
	2719	*			*STATUS				*
	2720	*			VERSION 1 MODIFICATION 0				*
	2721	*							*
	2722	*			*FUNCTION				*
	2723	*			THE FUNCTION OF KRVLAY IS TO PROCESS THE LABEL TABLE CREATED BY				*
	2724	*			KRLABL BY RELABELING THE SPECIFIED VARIABLES AND WRITING THE				*
	2725	*			MODIFIED LINES BACK TO THE WORKAREA.				*
	2726	*							*
	2727	*			*ENTRY POINTS				*
	2728	*			THE ENTRY POINT TO KRVLAY IS TO THE FIRST BYTE FOLLOWING THE				*
	2729	*			PROGRAM HEADER. (LABEL IS #KRVLA)				*
	2730	*							*
	2731	*			*INPUT				*
	2732	*			INPUT TO KRVLAY IS THE LABEL TABLE CREATED IN KRLABL.				*
	2733	*							*
	2734	*			*OUTPUT				*
	2735	*			OUTPUT FROM KRVLAY IS THE MODIFIED PROGRAM IN THE WORKAREA.				*
	2736	*							*
	2737	*			*EXTERNAL REFERENCES				*
	2738	*			* \$DISKN - ENTRY TO PHYSICAL DISK ROUTINE				*
	2739	*			* \$CIMSK - NUCLEUS BYTE SET TO MASK AGAINST INTERRUPTS				*
	2740	*			* \$INDR1 - NUCLEUS BYTE CONTAINING 'FIT IN CORE' INDICATOR				*
	2741	*			* \$CAERR - ERROR CODE SAVE AREA				*
	2742	*			* \$CAERK - EXIT TO LOAD #ERRPG, THE ERROR PROGRAM				*
	2743	*			* \$CARPL - EXIT TO LOAD #GUFUD, THE FILE UPDATE PROGRAM				*
	2744	*			* \$ERRCT - NUCLEUS BYTE SET TO INDICATE TWO ERROR MESSAGES				*
	2745	*			* \$\$ERSK - NUCLEUS BYTES SET WITH TWO ERROR CODES				*
	2746	*			* SVARAB - ENTRY TO MODULE TO FIND VARIABLES IN A FILE LINE				*
	2747	*			* SVAVTC - VARIABLE TYPE CODE SET BY SVARAB				*
	2748	*			* GPUTIT - ENTRY TO MODULE TO WRITE FILE LINES BACK TO DISK				*
	2749	*			* SCANIT - ENTRY TO DELIMITER SCAN MODULE				*
	2750	*			* GRABIT - ENTRY TO MODULE TO RETRIEVE FILE LINES				*
	2751	*			* GRWHAT - GRABIT REQUEST CODE				*
	2752	*			* GRSRDA - GRABIT AREA FOR INITIAL DISK ADDRESS				*
	2753	*			* GRTEXT - AREA WHERE GRABIT RETURNS FILE LINE TEXT				*
	2754	*			* GRTEND - TWO-BYTE FIELD CONTAINING ADDR OF EOS IN A FILE LINE				*
	2755	*							*
	2756	*			*EXITS, NORMAL				*
	2757	*			NORMAL EXIT FROM KRVLAY IS TO \$CARPL TO LOAD #GUFUD.				*
	2758	*							*
	2759	*			*EXITS, ERROR				*
	2760	*			ERROR EXIT FROM KRVLAY IS TO \$CAERK TO LOAD #ERRPG.				*
	2761	*							*
	2762	*			*TABLES/WORKAREAS				*
	2763	*			* TABLE OF LABEL PAIRS CREATED IN #KRLAB				*
	2764	*			* TWO-SECTOR BUFFER FOR GPUTIT OUTPUT.				*
	2765	*			* ONE-SECTOR BUFFER USED FOR GPUTIT INPUT, GRABIT OUTPUT, AND				*
	2766	*			GCPACK.				*
	2767	*			* TWO-SECTOR BUFFER FOR GRABIT INPUT.				*
	2768	*			* THE FILE INDEX TABLE IS BUILT BY GPUTIT AT X'1D00-X'1FFF'.				*
	2769	*							*

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE	4
		2770	*	ATTRIBUTES				*
		2771	*	RELOCATABLE				*
		2772	*					*
		2773	*	CHARACTER CODE DEPENDENCY				*
		2774	*	NONE				*
		2775	*					*
		2776	*	NOTES				*
		2777	*	ERROR PROCEDURES				*
		2778	*	AN APPROPRIATE ERROR MSG IS GIVEN IF ONE OR MORE FILE LINES				*
		2779	*	HAVE BEEN TRUNCATED, OR IF THE SIZE OF THE WORKFILE IS				*
		2780	*	EXCEEDED, OR BOTH (PRODUCES TWO ERROR MSG). EXIT IS MADE				*
		2781	*	TO \$CAERK.				*
		2782	*					*
		2783	*	REGISTER USAGE				*
		2784	*	* REGISTER 1 (@BR) IS USED AS A POINTER ACROSS THE LABEL TABLE.				*
		2785	*	* REGISTER 2 (@XR) IS USED AS A POINTER ACROSS THE TEXT OF A				*
		2786	*	FILE LINE.				*
		2787	*					*
		2788	*	SAVED/RESTORED AREAS				*
		2789	*	NONE				*
		2790	*					*
		2791	*	MODIFICATION CONSIDERATIONS				*
		2792	*	NONE				*
		2793	*					*
		2794	*	REQUIRED MODULES				*
		2795	*	* @SYSEQ COMMON SYSTEM EQUATES				*
		2796	*	* @FXDEQ NUCLEUS FIXED ADDRESS EQUATES				*
		2797	*	* @ERMEQ ERROR MESSAGE EQUATES				*
		2798	*	* @CANEQ FIXED ADDRESSES OUTSIDE NUCLEUS EQUATES				*
		2799	*	* \$B@EQU BASIC COMPILER PARAMETER AND SYSTEM EQUATES				*
		2800	*	* SVARAB - BASIC SYNTAX SCAN MODULE				*
		2801	*	* GRABIT MODULE TO RETRIEVE BASIC FILE LINES				*
		2802	*	* GCPACK - MODULE TO PACK BASIC FILE LINES				*
		2803	*	* GPUTIT MODULE TO WRITE FILE LINES TO DISK				*
		2804	*	* SCANIT DELIMITER SCAN MODULE				*
		2805	*	* DL4ICS 4-TRACK LOGICAL DISK IOCS MODULE				*
		2806	*					*
		2807	*	OTHER				*
		2808	*	NONE				*
		2809	*	*****				*

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 5

```

0800          2811      ORG    $ENDNU+X'0200'
              2812      *
              2813      *          HDR    #KRVLA,1          GENERATE PROGRAM HEADER
              2814      *****
              2815      * PROGRAM HEADER FOR DISK LOAD
              2816      *****
              2817      *#$KRVL EQU    X'0710'          DISK ADDR OF IKRVLA
              2818      *#$KRV EQU    X'0800'          CORE LOAD ADDRESS OF #KRVLA
              2819      *#$@KRV EQU    013          SECTOR CNT OF IKRVLA
0800          2820      ORG    $$$KRV          CORE LOAD ADDRESS
              2821      $$$$ EQU    *          FIRST LOCATION IN PROGRAM
0800 7BD2D9E5D3C1 0805 2822      DC    CL6'#KRVLA'          PROGRAM NAME
0806 2C          0806 2823      DC    IL1'044'          PROGRAM NUMBER OF #KRVLA
              0807 2824      $KRVLA EQU    *          ENTRY POINT TO PROGRAM
              2825      *** END OF EXPANSION ***
              2826      *
              2827      *          INITIALIZE AND CALL GRABIT
              2828      *
0807 C0 87 0025 0807 2829      KRVLAY EQU    *          ENTRY POINT
080B 097A          2830      B    $DISKN          PRIME GRABIT BUFFERS, USING
              080C 2831      DC    AL(@CADDR)(KRVDPG)          * KRVDPG AS PPL
              2832      *
080D 3C 00 1028 2833      MVI    GRWHAT,GRAEFI          SET INITIAL INDICATOR FOR GRABIT
0811 0C 01 101E 0981 2834      MVC    GRSRDA(@DADDR),KRVFVM          INITIALIZE GRABIT DISK ADDR
              2835      *
0817 C0 87 0EA3 2836      B    GRABIT          INITIAL CALL TO GRABIT
              2837      *
081B 3C 80 0476 2838      MVI    $CIMSK,@NOP          MASK AGAINST INTERRUPTS
081F 3C 01 1028 2839      MVI    GRWHAT,GRAEFR          SET RETURN TEXT IND IN GRABIT
0823 3A 10 03D4 2840      SBN    $INDR1,$FITIN          SET FIT IN CORE IND ON
              2841      *
              2842      *          SEARCH FILE LINE FOR LABELS
              2843      *
0827 C0 87 0EA3 2844      KRV380 B    GRABIT          GET A LINE OF FILE
              2845      *
082B 3D 1C 1A07 2846      CLI    GRTEXT,@EOF          IS EOF INDICATED ?
082F C0 81 094F 2847      BE    KRV600          YES, EXIT LOOP
              2848      *
0833 C2 02 1A07 2849      LA    GRTEXT,@XR          POINT XR TO TEXT
0837 C0 87 0990 2850      KRV390 B    SVARAB          FIND A LABEL IN THE FILE LINE
083B 0C 00 0985 0DB2 2851      MVC    KRVTMP(1),SVALNG          EXPAND VAR LNG TO 2 BYTE FIELD
              2852      *
0841 BD 1E 00 2853      CLI    0(,@XR),@EOS          IF XR DOES NOT POINT TO EOS,
0844 F2 01 08 2854      JNE    KRV395          * START TABLE SEARCH
              2855      *
0847 C0 87 10B2 2856      KRV392 B    GPUTIT          WRITE LINE BACK
084B C0 87 0827 2857      B    KRV380          GET NEXT LINE

```

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	6
				2859	*						
				2860	*		SEARCH TABLE FOR A MATCH				
				2861	*						
084F	C2	01	0707	2862	KRV395	LA	KRVTBL,@BR			POINT BR TO 1ST LBL IN LBL TABLE	
				2863	*						
0853	6D	00	00 00	2864	KRV400	CLC	0(,@BR),0(1,@XR)			DO FIRST CHARACTERS MATCH ?	
0857	F2	81	12	2865		JE	KRV410			YES, CHECK SECOND CHARACTERS	
				2866	*						
085A	D2	01	04	2867	KRV405	LA	KRVFOR(,@BR),@BR			ELSE, POINT BR TO NEXT LBL PAIR	
085D	7D	4E	00	2868		CLI	0(,@BR),@CPLUS			IS THIS PAST END OF TBL	
0860	C0	01	0853	2869		BNE	KRV400			NO, CHECK NEXT TBL ENTRY	
				2870	*						
				2871	*		LABEL DOES NOT NEED SWITCHING OR HAS BEEN SWITCHED				
				2872	*						
0864	36	02	0985	2873	KRV407	A	KRVTMP,@XR			POINT XR PAST LABEL FOUND	
0868	C0	87	0837	2874		B	KRV390			CHECK NEXT LABEL	
				2875	*						
				2876	*		FIRST CHARACTERS MATCH CHECK SECOND CHARACTERS				
				2877	*						
086C	38	10	0DB1	2878	KRV410	TBN	SVAVTC,SVALDC			IS THIS LBL A LTR-DGT VARIABLE ?	
0870	F2	10	0B	2879		JT	KRV420			YES, JUMP TO TEST FOR A MATCH	
				2880	*						
0873	4D	00	01 0DB1	2881		CLC	KRVD02(,@BR),SVAVTC(1)			ELSE, ARE LABELS A MATCH ?	
0878	F2	01	16	2882		JNE	KRV440			NO, CHECK NEXT ENTRY	
087B	F2	87	17	2883		J	KRV450			YES, GO TO PROCESS A MATCH	
				2884	*						
087E	34	02	0890	2885	KRV420	ST	KRV430+@OP1,@XR			SAVE XR	
				2886	*						
0882	E2	02	01	2887		LA	1(,@XR),@XR			INCR XR PAST ALPHA CHAR	
0885	C0	87	12F4	2888		B	SCANIT			POINT XR TO DGT	
				2889	*						
0889	9D	00	00 01	2890		CLC	0(,@XR),KRVD02(1,@BR)			DO DIGITS MATCH ?	
088D	C2	02	0000	2891	KRV430	LA	*-*,@XR			RESTORE XR	
				2892	*						
0891	C0	01	085A	2893	KRV440	BNE	KRV405			IF NOT A MATCH, CHECK NEXT ENTRY	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	7
					2895	*					
					2896	*	LABELS MATCH -- SWITCH NAMES				
					2897	*					
0895	34	02	0910		2898	KRV450 ST	KRV530+@OP1,@XR			SAVE XR	
					2899	*					
0899	D2	01	02		2900	LA	KRVTB2(,@BR),@BR			POINT BR TO REPLACEMENT LET	
089C	3C	01	0982		2901	MVI	KRVLNG,KRVANY			SET REPLACEMENT LENGTH TO ONE	
08A0	78	10	01		2902	TBN	KRVD02(,@BR),SVALDC			IS IT A LTR-DGT LABEL 7	
08A3	F2	90	04		2903	JF	KRV460			NO, JUMP TO COMPARE LENGTHS	
					2904	*					
08A6	3C	02	0982		2905	MVI	KRVLNG,KRVLLT			YES, REPLACEMENT LENGTH = 2	
08AA	0D	00	0985 0982		2906	KRV460 CLC	KRVTMP(1),KRVLNG			REPLACEMENT LNG > AVAIL LNG ?	
08B0	F2	02	30		2907	JNL	KRV500			NO, GO CLEAR OUT OLD LABEL	
					2908	*					
					2909	*	MOVE USERS LINE OVER ONE BYTE				
					2910	*					
08B3	0C	01	08DC 0FD9		2911	MVC	KRV470+@OP2(@CADDR),GRTEND			SET MOVE FROM ADDRESS	
					2912	*					
08B9	0F	01	0FD9 0910		2913	SLC	GRTEND,KRV530+@OP1(@CADDR)			COMPUTE LENGTH OF MOVE	
08BF	0C	00	08D8 0FD9		2914	MVC	KRV470+@Q,GRTEND(1)			SET MOVE LENGTH COUNT	
08C5	0C	01	08DA 08DC		2915	MVC	KRV470+@OP1(@CADDR),KRV470+@OP2			CALCULATE MOVE TO	
08CB	0E	01	08DA 0987		2916	ALC	KRV470+@OP1(@CADDR),KRVCC1			* ADDRESS	
					2917	*					
08D1	0E	00	0985 0987		2918	ALC	KRVTMP(1),KRVCC1			INCR VAR LENGTH SPACE BY ONE	
08D7	0C	00	0000 0000		2919	KRV470 MVC	*-*(@VQ),*-*			MOVE USER'S LINE	
					2920	*					
08DD	0C	01	0FD9 08DA		2921	MVC	GRTEND,KRV470+@OP1(@CADDR)			SET NEW LAST CHAR LOCATION	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 8
				2923	*			
				2924	*		CLEAR OUT OLD LABEL	
				2925	*			
08E3	3D	01	0985	2926	KRV500	CLI	KRVTMP,KRVONE	IS OLD FIELD ONE CHAR LONG ?
08E7	F2	81	20	2927		JE	KRV520	YES, JUMP TO MOVE IN ONE BLANK
				2928	*			
08EA	0F	00	0985 0988	2929		SLC	KRVTMP,KRVTWO(1)	SUBT '2' FROM OLD FLD LENGTH
08F0	36	02	0985	2930		A	KRVTMP,@XR	PT XR 1 BYTE FROM OLD FLD
08F4	0C	00	08FE 0985	2931		MVC	KRV510+@Q(1),KRVTMP	SET LENGTH FOR MOVE INSTRUCTION
08FA	BC	40	01	2932		MVI	1(,@XR),@BLANK	CLEAR RIGHT MOST BYTE TO BLANKS
08FD	AC	00	00 01	2933	KRV510	MVC	0(,@XR),1(@VQ,@XR)	PROPOGATE BLANKS THROUGH FIELD
				2934	*			
0901	0E	00	0985 0988	2935		ALC	KRVTMP(1),KRVTWO	RESTORE SVALNG
0907	F2	87	03	2936		J	KRV530	JUMP TO MOVE IN NEW LABEL
				2937	*			
090A	BC	40	00	2938	KRV520	MVI	0(,@XR),@BLANK	CLEAR FIELD
				2939	*			
				2940	*		MOVE IN NEW LABEL	
				2941	*			
090D	C2	02	0000	2942	KRV530	LA	*-*,@XR	RESTORE XR
				2943	*			
0911	9C	00	00 00	2944		MVC	0(1,@XR),0(,@BR)	MOVE NEW LBL TO BUFFER
0915	78	10	01	2945		TBN	KRVD02(,@BR),SVALDC	IS IT A LTR-DGT VAR ?
0918	F2	90	07	2946		JF	KRV550	NO, TEST FOR CHAR ARRAY
				2947	*			
091B	9C	00	01 01	2948		MVC	1(1,@XR),1(,@BR)	ELSE, MOVE DGT TO LINE
091F	F2	87	0F	2949		J	KRV560	GO TEST FOR TRUNCATED LINE
0922	78	04	01	2950	KRV550	TBN	KRVD02(,@BR),SVACVC	IS THIS A CHARACTER VAR ?
0925	F2	10	06	2951		JT	KRV555	YES, JUMP TO MOVE 'S' TO LINE
				2952	*			
0928	78	02	01	2953		TBN	KRVD02(,@BR),SVACAC	IS THIS A CHARACTER ARRAY ?
092B	F2	90	03	2954		JF	KRV560	NO, GO TEST FOR TRUNCATED LINE
092E	BC	5B	01	2955	KRV555	MVI	1(,@XR),@DOLAR	MOVE '\$' TO LINE
				2956	*			
0931	3D	FA	08DA	2957	KRV560	CLI	KRV470+@OP1,KRVMAX	GO GET NEXT LABEL IF NEW LINE
0935	C0	04	0864	2958		BNH	KRV407	* DID NOT OVERFLOW BUFFER
				2959	*			
0939	3C	78	1A06	2960		MVI	GRTYPE,B@TDUM	ELSE,SET TRUNCATED LH TYPE CODE
093D	0C	01	0946 08DC	2961		MVC	KRV570+@OP1(@CADDR),KRV470+@OP2	RESTORE AN EOS OVER THE
0943	3C	1E	0000	2962	KRV570	MVI	*-*,@EOS	* LAST CHARACTER
0947	3C	01	0983	2963		MVI	KRVERS,KRVONE	SET IND FOR TRUNCATED LINE
094B	C0	87	0847	2964		B	KRV392	GO WRITE THIS LINE BACK

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 9
			2966	*		
			2967	*	END OF FILE ENCOUNTERED	
			2968	*		
094F	C0 87 10B2		2969	KRV600 B	GPUTIT CALL GPUTIT LAST TIME	
			2970	*		
0953	3C 89 03CD		2971	MVI	\$CAERR,@E500 SET 'TRUNCATED LINE' ERR CODE	
0957	3D 00 0983		2972	CLI	KRVERS,@ZERO WAS AT LEAST ONE LN TRUNCATED?	
095B	C0 01 0469		2973	BNE	\$CAERK YES, CALL ERROR PROGRAM	
			2974	*		
095F	C0 87 04A1		2975	B	\$CARPL ELSE, GOOD EXIT	
		0963	2976	GPUERR EQU	* GPUTIT ERROR EXIT	
0963	3D 00 0983		2977	CLI	KRVERS,@ZERO WERE ANY LINES TRUNCATED	
0967	F2 81 0C		2978	JE	KRV650 NO, CALL ERR PROG-FILE OVERFLOW	
			2979	*		
096A	0C 01 03CF 098A		2980	MVC	\$ERRCT,KRVECT(KRVSE2-KRVSE1) SET ERR CODE FOR STACK	
0970	0C 04 1C04 098F		2981	MVC	\$ERSK+KRVSE2,KRVSE2,KRVSE2+1) SET STACK	
0976	C0 87 0469		2982	KRV650 B	\$CAERK CALL ERROR PROGRAM	
			2983	*		
			2984	*	EQUATES FOR BUFFERS	
			2985	*		
		1800	2986	GPUBF1 EQU	X'1800' GPUTIT OUTPUT BUFFER	
			2987	*		
		1A00	2988	GPUSMT EQU	GPUBF1+X'0200' GPUTIT INPUT BUFFER	
			2989	*		
		1B00	2990	GRBFR1 EQU	GPUSMT+X'0100' GRABIT INPUT BUFFER	
			2991	*		
		1A05	2992	GRLINE EQU	GPUSMT+@SBLN GRABIT LINE NO. SAVE AREA ADDR	
		1A06	2993	GRTYPE EQU	GPUSMT+@STYPE GRABIT TYPE CODE SAVE AREA ADDR	
		1A07	2994	GRTEXT EQU	GPUSMT+@STEXT ADDR OF 1ST BYTE OF TEXT LINE	
			2995	*		
		1A00	2996	GCPBFR EQU	GPUSMT GCPACK BUFFER	
			2997	*		
			2998	*	MISCELLANEOUS EQUATES	
			2999	*		
		008A	3000	GPUECD EQU	@E501 GPUTIT ERROR CODE	
		00FA	3001	KRVMAX EQU	@SDFLN+243 LENGTH OF MAXIMUM FILE LINE	
		0004	3002	KRVFOR EQU	4 LENGTH OF LABEL-PAIR TBL ENTRY	
		0707	3003	KRVTLB EQU	\$SKLD2+@HDLN LABEL TABLE ADDRESS	
		0001	3004	KRVONE EQU	1 LENGTH OR ONE EBCDIC CHARACTER	
		0001	3005	KRVD02 EQU	1 DISP TO 2ND BYTE IN LBL ENTRY	
		0002	3006	KRVTLB EQU	2 DISP TO REPLACEMENT LABEL	
		0002	3007	KRV002 EQU	2 LENGTH OF LABEL TABLE ENTRY	
		0001	3008	KRVANY EQU	1 LENGTH OF ALPHA PART OF A LABEL	
		0002	3009	KRVLLT EQU	2 LENGTH OF LTR-DGT CHAR	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 10

3011 *
3012 *
3013 *
CONSTANTS AND SAVE AREAS

097A 01 097A 3014 KRVDPG DC AL1(@DGET) DPL FOR READING TO CORE THE
097B 070C 097C 3015 DC XL(@DADDR)'070C' * FIRST TWO SECTORS OF THE
097D 02 097D 3016 DC IL1'2' * WORKFILE -- USED TO PRIME
097E 1B00 097F 3017 DC AL(@CADDR)(GRBFR1) * GRABIT'S BUFFERS
3018 *
0980 0703 0981 3019 KRVFVM DC XL(@DADDR)'0703' FIRST LOGICAL SECTOR OF VM
3020 *
0982 0982 3021 KRVLNG DS XL1 SAVE AREA FOR LENGTH OF LABEL
3022 * A REPLACEMENT
0983 0983 3023 KRVERS DS XL1 SAVE AREA FOR INDICATING A
0983 3024 ORG KRVERS * LINE WAS TRUNCATED --
0983 00 0983 3025 DC XL1'00' * INITIALIZED TO NONE
3026 *
0984 0984 3027 KRVTM1 EQU * SAVE AREA FOR VARIABLE LENGTH.
0984 0985 3028 KRVTMP DS XL(@REGL) * COMPUTED IN SVARAB, EXPANDED
0984 0000 0985 3029 ORG KRVTM1 * HERE TO TWO BYTES (LEFT BYTE
0984 0000 0985 3030 DC XL(@REGL)'0000' * ZERO) FOR ADDING TO REGISTER
3031 *
0986 0001 0987 3032 KRVCC1 DC XL(@CADDR)'01' CONSTANT ONE
3033 *
0988 02 0988 3034 KRVTWO DC XL1'02' CONSTANT TWO
3035 *
3036 *
3037 *
CONSTANTS USED FOR ERROR CODE STACKING

0989 30 0989 3038 KRVSE1 DC AL1(\$ERSTK) ERROR STACK INDICATOR
098A 02 098A 3039 KRVECT DC IL1'2' ERROR COUNT
098B 89 098B 3040 KRVSE2 DC AL1(@@E500) LINE TRUNCATION ERROR CCDE
098C A0 098C 3041 DC AL1(\$\$\$NLN) NO LINE-NUMEZR REF
098D 098D 3042 KRVDUM DS CL1 DUMMY BYTE
098E 8A 098E 3043 DC AL1(@@E501) FILE TRUNCATION ERROR CODE
098F A0 098F 3044 KRVSER DC AL1(\$\$\$NLN) NO LINE-NUMBER REF
3045 *

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE 11
		3047		*****			
		3048	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		3049	*	REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083			*
		3050	*				*
		3051		*****			*
		3052	*				*
		3053	*	STATUS -			*
		3054	*	VERSION 1 MODIFICATION 0			*
		3055	*				*
		3056	*	FUNCTION -			*
		3057	*	* SVARAB SCANS THE BASIC LINE, BEGINNING AT A POINT PASSED BY			*
		3058	*	THE CALLING ROUTINE			*
		3059	*	* A POINTER IS SET TO THE FIRST VARIABLE OR ARRAY SYMBOL			*
		3060	*	ENCOUNTERED			*
		3061	*	* SVAVTC IS SET TO A CODE THE WILL INDICATE THE VARIABLE TYPE			*
		3062	*	* SVALNG IS SET TO THE VARIABLE LENGTH			*
		3063	*				*
		3064	*	ENTRY POINTS -			*
		3065	*	* SVARAB HAS ONLY 1 ENTRY POINT			*
		3066	*	* THE CALLING SEQUENCE IS			*
		3067	*	B SVARAB			*
		3068	*				*
		3069	*	INPUT -			*
		3070	*	* REGISTER @XR - CONTAINS THE CORE ADDRESS OF THE INITIAL			*
		3071	*	CHARACTER TO BE EXAMINED			*
		3072	*	* GRTYPE - CONTAINS THE STATEMENT TYPE CODE FOR THE BASIC			*
		3073	*	STATEMENT LINE BEING PROCESSED			*
		3074	*				*
		3075	*	OUTPUT -			*
		3076	*	* REGISTER @XR - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		3077	*	CHARACTER OF THE FIRST VARIABLE ENCOUNTERED			*
		3078	*	* IF NO VARIABLE EXISTS, @XR CONTAINS CORE ADDRESS OF THE			*
		3079	*	CARRIAGE RETURN BYTE			*
		3080	*	* SVAVTC - 1 BYTE, CONTAINS THE VARIABLE TYPE CODE OF THE			*
		3081	*	VARIABLE			*
		3082	*	* SVALNG - 1 BYTE, CONTAINS THE LENGTH OF THE VARIABLE			*
		3083	*				*
		3084	*	EXTERNAL REFERENCES			*
		3085	*	GRTYPE - BASIC STATEMENT TYPE CODE			*
		3086	*				*
		3087	*	EXITS, NORMAL -			*
		3088	*	NORMAL EXIT IS TO THE FIRST INSTRUCTION FOLLOWING THE CALLING			*
		3089	*	SEQUENCE. THE BASE REGISTER IS RESTORED. THE RETURN ADDRESS IS			*
		3090	*	IN THE ADDRESS RECALL REGISTER (@ARR).			*
		3091	*				*
		3092	*	OUTS, ERROR -			*
		3093	*	N/A			*
		3094	*				*
		3095	*	TABLES/WORK AREAS -			*
		3096	*	* THE STATEMENT BRANCH TABLE A ONE ENTRIES FOR EACH BASIC			*
		3097	*	STATEMENT TYPE, EACH ENTRY IS 3 BYTES AND CONTAINS			*
		3098	*	A LENGTH OF STATEMENT KEYWORD - 1 BYTE			*
		3099	*	* CORE ADDRESS OF STATEMENT PROCESSING ROUTINE - 2 BYTES			*
		3100	*	* THE CONSTANTS AND WORK AREAS RESIDE AT THE END OF THE			*
		3101	*	EXECUTABLE CODE AND ARE REFERENCED BY @BR			*
		3102	*				*

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE 12
		3103	*	ATTRIBUTES -			*
		3104	*	N/A			*
		3105	*				*
		3106	*	CHARACTER CODE DEPENDENCY			*
		3107	*	THE OPERATION OF THIS MODULE DEPENDS UPON THE FOLLOWING			*
		3108	*	PROPERTIES OF THE INTERNAL REPRESENTATION OF THE EXTERNAL			*
		3109	*	CHARACTER SET			*
		3110	*	* MOST CODING HAS BEEN ARRANGED SO THAT REDEFINITION OF			*
		3111	*	CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT IN A CORRECT			*
		3112	*	MODULE FOR THE NEW DEFINITION			*
		3113	*	* ALPHABETIC LETTERS A THROUGH I ARE PRESUMED TO BE CODED IN			*
		3114	*	INCREASING COLLATING SEQUENCE, AND THE RANGE OF CHARACTER			*
		3115	*	CONSTANTS FOR THIS SERIES IS EXPECTED TO EXCLUDE ALL NUMERIC			*
		3116	*	CHARACTER CONSTANTS			*
		3117	*	* NUMERIC CHARACTERS 0 - 9 ARE PRESUMED TO BE CODED IN			*
		3118	*	INCREASING COLLATING SEQUENCE			*
		3119	*	* EXTENDED ALPHABETIC LETTERS (\$, #, @) ARE PRESUMEMED TO BE			*
		3120	*	IN INCREASING COLLATING SEQUENCE, AND ARE ALL EXPECTED TO			*
		3121	*	COLLATE LOWER THAN LETTER (A)			*
		3122	*	* DECIMAL NUMBERS MUST BE CODED SO THAT THE LOW ORDER FOUR			*
		3123	*	BITS, WHEN CONSIDERED AS A BINARY INTEGER, IDENTIFY THE			*
		3124	*	VALUE OF THE DIGIT			*
		3125	*	THE SPECIFIC INSTRUCTIONS (INSTRUCTION SEQUENCES) WHICH REQUIRE			*
		3126	*	MODIFICATION IF THESE PROPERTIES OF THE CHARACTER SET ARE CHANGED			*
		3127	*	MAY BE IDENTIFIED BY -			*
		3128	*	* INSTRUCTION SEQUENCES AT LABELS SVA075 AND SVA080			*
		3129	*	* INSTRUCTION SEQUENCES AT LABELS SVA460 AND SVA465			*
		3130	*	* INSTRUCTION SEQUENCES AT LABEL SVA930			*
		3131	*				*
		3132	*	NOTES -			*
		3133	*	ERROR PROCEDURES			*
		3134	*	N/A			*
		3135	*				*
		3136	*	REGISTER USAGE			*
		3137	*	* REGISTER @XR IS BOTH AN INPUT AND OUTPUT PARAMETER			*
		3138	*	* REGISTER @BR IS SAVED ON ENTRY, USED DURING EXECUTION,			*
		3139	*	AND RESTORED ON EXIT			*
		3140	*				*
		3141	*	SAVED/RESTORED			*
		3142	*	N/A			*
		3143	*				*
		3144	*	MODIFICATION CONSIDERATIONS			*
		3145	*	N/A			*
		3146	*				*
		3147	*	REQUIRED MODULES			*
		3148	*	@SYSEQ - COMMON SYSTEM EQUATES			*
		3149	*	\$B@EQU - COMPILER SYSTEM EQUATES			*
		3150	*				*
		3151	*	OTHER			*
		3152	*	N/A			*
		3153	*				*
		3154	*	*****			*

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 13
					3156	*****				
					3157	*****				
					3158	*				*
					3159	*****				
					3160	*				*
					3161	*				*
					3162	*	SVARAB BASIC SYNTAX SCAN ROUTINE			*
					3163	*				*
					3164	*				*
					3165	*****				
					3166	*				*
					3167	*****				
					3168	*****				
					3170	*				
					3171	*	SVARAB ENTRY, SET RETURN LINKAGE AND ADDRESSABILITY			
					3172	*				
				0990	3173	SVARAB EQU *	SVARAB ENTRY POINT			
0990	34	08	0B64		3174	ST SVA320+@OP1,@ARR	SAVE RETURN ADDR			
0994	34	01	0B60		3175	ST SVA315+@OP1,@BR	SAVE PT			
				0D8F	3176	USING SVA960,@BR	SET BASE ADDR			
0998	C2	01	0D8F		3177	LA SVA960,@BR	LOAD BASE			
099C	7C	78	24		3178	MVI SVAMAG(,@BR),B@TDUM	SET MAGIC TYPE			1-4
099F	7C	00	25		3179	MVI SVAZRO(,@BR),@ZERO	CLEAR INDEX			1-4
					3180	*				
					3181	*	DETERMINE THE BRACH TABLE INDEX			
					3182	*				
09A2	0C	00	0DAB 1A06		3183	MVC SVASTC(1),GRTYPE	SAVE STATEMENT TYPE CODE			
09A8	3B	80	0DAB		3184	SVA020 SBF SVASTC,SVADIS	SET DISABLE SW OFF			
					3185	* CODE FOR STRING FUNCTION				1-4
09AC	5F	00	24 1C		3186	SLC SVAMAG(,@BR),SVASTC(,@BR)	CHECK TYPE CODE			1-4
09B0	F2	02	0C		3187	JNL SVA030	NOT SPECIAL---SKIP			1-4
09B3	5F	00	25 24		3188	SLC SVAZRO(,@BR),SVAMAG(,@BR)	COMPLEMENT TYPE			1-4
09B7	5E	00	25 25		3189	ALC SVAZRO(,@BR),SVAZRO(,@BR)	DOUBLE IT			1-4
09BB	5E	00	1C 25		3190	ALC SVASTC(1,@BR),SVAZRO(,@BR)	ACCUMULATE			1-4
09BF	C2	01	0E19		3191	SVA030 LA SVABRT-3,@BR	DISTRIBUTOR TABLE			1-4
09C3	36	01	0DAB		3192	A SVASTC,@BR	INDEX TABLE ENTRY			
09C7	1C	00	0DA9 00		3193	MVC SVAKWL(1),SVA0TD(,@BR)	SAVE KEYWORD LENGTH			
09CC	75	01	02		3194	L SVA2TD(,@BR),@BR	LOAD BRANCH ADDR			
09CF	D0	87	00		3195	B SVAPD0(,@BR)	BRANCH TO PROC RTN			

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 14
		3197		*****	
		3198		*	*
		3199		*****	
		3200		*	*
		3201	*	ROUTINE TO PROCESS STATEMENT LINES THAT CONTAIN NO VARIABLES	*
		3202		*	*
		3203		*****	
		3204		*	*
		3205		*****	
		3206		*	
09D2	BD 1E 00	3207	SVA050	CLI SVAPD0(,@XR),B@EOST	AT END OF STATEMENT
09D5	C0 81 0B5D	3208		BE SVA315	YES, RETURN TO CALLING RTN
09D9	36 02 0DB6	3209		A SVAI01,@XR	INCR PT TO NEXT BYTE
09DD	C0 87 09D2	3210		B SVA050	CYCLE LOOP UNTIL EOS

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 15
				3212		*****				
				3213		*				*
				3214		*****				
				3215		*				*
				3216	*	ROUTINE TO PROCESS ARITHMETIC AND CHARACTER ASSIGNMENT STATEMENTS				*
				3217		*				*
				3218		*****				
				3219		*				*
				3220		*****				
				3221		*				
09E1	C2	01	0D8F	3222	SVA070	LA	SVA960,@BR		SET	BASE ADDR
09E5	7D	00	18	3223		CLI	SVABSW(,@BR),SVAOFF		IS	BRANCH SW OFF
09E8	F2	01	25	3224		JNE	SVA085		NO,	SCAN FOR VARIABLE
09EB	BD	C1	00	3225	SVA075	CLI	SVAPD0(,@XR),@CHARA		IF	BYTE IS IN STANDARD
09EE	F2	82	06	3226		JL	SVA080		*	ALPHABET, EXIT LOOP
09F1	BD	E9	00	3227		CLI	SVAPD0(,@XR),@CHARZ		*	AND GO SCAN FOR VARIABLES
09F4	F2	04	19	3228		JNH	SVA085		*	IN LINE
09F7	BD	7B	00	3229	SVA080	CLI	SVAPD0(,@XR),@NUMBR		TEST	FOR SPECIAL ALPHABETIC
09FA	F2	81	13	3230		JE	SVA085		*	CHARACTERS. IF EQUAL TO
09FD	BD	7C	00	3231		CLI	SVAPD0(,@XR),@ASIGN		*	ANY, GO SCAN FOR ANY
0A00	F2	81	0D	3232		JE	SVA085		*	VARIABLES IN THE LINE
0A03	BD	5B	00	3233		CLI	SVAPD0(,@XR),@DOLAR		*	\$ INCLUDED FOR WTC
0A06	F2	81	07	3234		JE	SVA085		*	CONSIDERATIONS
0A09	E2	02	01	3235		LA	1(,@XR),@XR		INCR	PT
0A0C	C0	87	09EB	3236		B	SVA075		LOOP	UNTIL ALPHA CHAR IS FOUND
0A10	C0	87	0B6C	3237	SVA085	B	SVA395		SCAN	FOR VARIABLE TYPE
0A14	7C	01	1D	3238		MVI	SVADSW(,@BR),SVAONN		SET	BR SW ON
0A17	C0	87	0B5D	3239		B	SVA315		RETURN	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	16
				3241		*****					
				3242	*						*
				3243		*****					
				3244	*						*
				3245	*	ROUTINE TO PROCESS THE COMPUTED GOTO STATEMENT					*
				3246	*						*
				3247		*****					
				3248	*						*
				3249		*****					
				3250	*						
0A1B	C2	01	0D8F	3251	SVA090	LA	SVA960,@BR			SET	BASE ADDR
				3252	*						
				3253	*	TEST BRANCH SWITCH					
				3254	*						
0A1F	7D	00	18	3255	SVA095	CLI	SVABSW(,@BR),SVAOFF			IS	BR SW OFF
0A22	F2	01	12	3256		JNE	SVA110			NO,	SKIP INCR PAST KEYWORD
				3257	*						
				3258	*	INCREMENT LINE POINTER PAST KEYWORD 'ON'					
				3259	*						
0A25	7C	01	18	3260	SVA100	MVI	SVABSW(,@BR),SVAONN			BR	SW ON
0A28	C0	87	0CDC	3261		B	SVA700			INCR	PAST KEYWORD
0A2C	C0	87	0CF3	3262		B	SVA900			INCR	TO 1ST ALPHA BYTE
0A30	7C	02	1A	3263		MVI	SVAKWL(,@BR),B@LKON			SET	KEYWORD LNG
0A33	C0	87	0CDC	3264		B	SVA700			INCR	PAST KEYWORD
				3265	*						
				3266	*	DETERMINE VARIABLE TYPE AND RETURN					
				3267	*						
0A37	C0	87	0B6C	3268	SVA110	B	SVA395			DETERMINE	VAR TYPE
0A3B	C0	87	0B5D	3269		B	SVA315			RETURN	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 17
					3271	*****				
					3272	*				
					3273	*****				
					3274	*				
					3275	*	ROUTINE TO PROCESS THE MAT ASSIGNMENT STATEMENTS			
					3276	*				
					3277	*****				
					3278	*				
					3279	*****				
					3280	*				
	0A3F	C2	01	0D8F	3281	SVA120 LA	SVA960,@BR SET BASE ADDR			
					3282	*				
					3283	*	TEST BRANCH SWITCH			
					3284	*				
	0A43	7D	00	18	3285	SVA125 CLI	SVABSW(,@BR),SVAOFF IS BRAN SW OFF ?			
	0A46	F2	01	23	3286	JNE	SVA150 YES, CHECK FOR VAR			
					3287	*				
					3288	*	INCREMENT LINE POINTER PAST KEYWORD			
					3289	*				
	0A49	C0	87	0CDC	3290	SVA130 B	SVA700 INCR PAST KEYWORD			
	0A4D	7C	01	18	3291	MVI	SVABSW(,@BR),SVAONN SET BR SW ON			
	0A50	7C	00	1E	3292	MVI	SVAPCT(,@BR),SVAOFF SET PAREN SW			
					3293	*				
					3294	*	INCREMENT TO THE ARRAY VARIABLE AND SET TYPE CODE			
					3295	*				
	0A53	C0	87	0CF3	3296	SVA140 B	SVA900 TO 1ST ALPHA BYTE			
	0A57	34	01	0A67	3297	ST	SVA148+@OP1,@BR SAVE VAR ADDR			
	0A5B	7C	01	23	3298	MVI	SVALNG(,@BR),SVAVL1 SET VAR LNG TO			
	0A5E	D0	87	00	3299	SVA144 B	SVA960(,@BR) DETM VAR LNG			
	0A61	7C	08	22	3300	MVI	SVAVTC(,@BR),SVANAC SET TYPE CODE TO ARITH ARRAY			
	0A64	C2	02	0000	3301	SVA148 LA	*-*,@XR RESTORE VAR ADDR			
	0A68	C0	87	0B5D	3302	B	SVA315 RETURN			
					3303	*				
					3304	*	STACK NEXT THREE NON-BLANK BYTES			
					3305	*				
	0A6C	BD	7E	00	3306	SVA150 CLI	SVAPD0(,@XR),B@EQL AT EQ SIGN			
	0A6F	F2	81	06	3307	JE	SVA151 YES, INCR PAST IT			
	0A72	BD	40	00	3308	CLI	SVAPD0(,@XR),B@BLNK AT BLANK			
	0A75	F2	01	03	3309	JNE	SVA152 NO, SAVE PRESENT PT			
	0A78	D0	87	00	3310	SVA151 B	SVA960(,@BR) YES, SKIP TO NON-BLANK BYTE			
	0A7B	34	02	0AD8	3311	SVA152 ST	SVA165+@OP1,@XR SAVE VAR ADDR			
	0A7F	6C	00	1F 00	3312	MVC	SVALS1(,@BR),SVAPD0(1,@XR) STACK CHAR			
	0A83	D0	87	00	3313	B	SVA960(,@BR) TO NEXT NON BLANK BYTE			
	0A86	6C	00	20 00	3314	MVC	SVALS2(,@BR),SVAPD0(1,@XR) STACK CHAR			
	0A8A	D0	87	00	3315	B	SVA960(,@BR) TO NEXT NON BLANK BYTE			
	0A8D	6C	00	21 00	3316	MVC	SVALS3(,@BR),SVAPD0(1,@XR) STACK CHAR			
					3317	*				
					3318	*	TEST FOR FUNCTIONS INV AND TRN			
					3319	*				
	0A91	5D	02	21 80	3320	SVA154 CLC	SVALSA(B@LIFN,@BR),SVAINV(,@BR) FUNC 'INV'			
	0A95	F2	81	07	3321	JE	SVA156 YES, INCR TO VAR			
	0A98	5D	02	21 83	3322	SVA155 CLC	SVALSA(B@LIFN,@BR),SVATRN(,@BR) FUNC 'TRN'			
	0A9C	F2	01	1B	3323	JNE	SVA160 NO, SCAN FOR VAR			
					3324	*				
					3325	*	INCREMENT LINE PT TO ARGUMENT VAR AND SET TYPE CODE			
					3326	*				

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 18
	0A9F	76	02	27	3327	SVA156	A SVAI01(,@BR),@XR			INCR PAST CHAR
	0AA2	C0	87	0CF3	3328		B SVA900			TO 1ST ALPHA BYTE
	0AA6	34	02	0AB6	3329		ST SVA158+@OP1,@XR			SAVE VAR ADOR
	0AAA	7C	01	23	3330		MVI SVALNG(,@BR),SVAVL1			SET VAR LNG TO 1
	0AAD	D0	87	00	3331		B SVA960(,@BR)			TO NEXT NON BLANK BYTE
	0AB0	7C	08	22	3332		MVI SVAVTC(,@BR),SVANAC			SET TYPE CODE TO ARITH ARRAY
	0AB3	C2	02	0000	3333	SVA158	LA *-*,@XR			RESTORE VAR ADDR
	0AB7	F2	87	A3	3334		J SVA315			RETURN
					3335	*				
					3336	*	TEST FOR FUNCTION CON, IDN, AND ZER			
					3337	*				
	0ABA	5D	02	21 86	3338	SVA160	CLC SVALSA(B@LIFN,@BR),SVACON(,@BR) FUNC 'CON'			
	0ABE	F2	81	0E	3339		JE SVA163			YES, PROC FUNC
	0AC1	5D	02	21 89	3340		CLC SVALSA(B@LIFN,@BR),SVAIDN(,@BR) FUNC 'IDN'			
	0AC5	F2	81	07	3341		JE SVA163			YES, PROC FUNC
	0AC8	5D	02	21 8C	3342		CLC SVALSA(B@LIFN,@BR),SVAZER(,@BR) FUNC 'ZER'			
	0ACC	F2	01	06	3343		JNE SVA165			NO, SCAN FOR VAR
	0ACF	E2	02	01	3344	SVA163	LA 1(,@XR),@XR			INCR PT
	0AD2	F2	87	04	3345		J SVA168			SCAN FOR VARS
					3346	*				
					3347	*	BRANCH TO SCAN RTN TO FIND A VARIABLE AND RETURN			
					3348	*				
	0AD5	C2	02	0000	3349	SVA165	LA *-*,@XR			RESTORE VAR ADDR
	0AD9	C0	87	0B3B	3350	SVA168	B SVA270			DETM VAR TYPE

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 19
		3352		*****	
		3353	*		*
		3354		*****	
		3355	*		*
		3356	*	ROUTINE TO PROCESS GET AND PUT STATEMENTS	*
		3357	*		*
		3358		*****	
		3359	*		*
		3360		*****	
		3361	*		
0ADD C2 01 0D8F		3362	SVA170 LA	SVA960,@BR	SET BASE ADDR
		3363	*		
		3364	*	TEST BRANCH SWITCH	
		3365	*		
0AE1 7D 00 18		3366	SVA175 CLI	SVABSW(,@BR),SVAOFF	IS BR SW OFF
0AE4 F2 01 07		3367		JNE SVA190	NO, CHECK FOR VAR
0AE7 7C 01 18		3368		MVI SVABSW(,@BR),SVAONN	SET BR SW ON
0AEA C0 87 0CDC		3369		B SVA700	SKIP KEYWORD
		3370	*		
		3371	*	SCAN FOR VARIABLE AND RETURN	
		3372	*		
0AEE C0 87 0B6C		3373	SVA190 B	SVA395	SCAN FOR VAR
0AF2 F2 87 68		3374		J SVA315	RETURN

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	20
				3376		*****					
				3377		*					*
				3378		* ROUTINE TO PROCESS THE REMAINING MATRIX STATEMENTS					*
				3379		*					*
				3380		*****					*
				3381		*					*
				3382		*****					*
				3383		*					
0AF5	C2	01	0D8F	3384	SVA192	LA	SVA960,@BR			SET BASE ADOR	
				3385		*					
				3386		* TEST BRANCH SWITCH					
				3387		*					
0AF9	7D	00	18	3388	SVA194	CLI	SVABSW(,@BR),SVAOFF			IS BR SW OFF ?	
0AFC	F2	01	3C	3389		JNE	SVA270			NO, SCAN FOR VARS	
				3390		*					
				3391		* INCREMENT PAST THE KEYWORDS					
				3392		*					
0AFF	7C	01	18	3393	SVA196	MVI	SVABSW(,@BR),SVAONN			SET BR SW ON	
0B02	7C	00	1E	3394		MVI	SVAPCT(,@BR),SVAOFF			SET PAREN COUNT TO 0	
0B05	C0	87	0CDC	3395		B	SVA700			INCR PAST KEYWORDS	
0B09	F2	87	2F	3396		J	SVA270			SCAN FOR VARS	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 21
		3398		*****	
		3399		*	*
		3400		* ROUTINE TO PROCESS REMAINING NON-MATRIX STATEMENTS	*
		3401		*	*
		3402		*****	
		3403		*	*
		3404		*****	
		3405		*	
0B0C C2 01 0D8F		3406	SVA200 LA	SVA960,@BR	SET BASE ADDR
		3407		*	
		3408		* TEST BRANCH SWITCH	
		3409		*	
0B10 7D 00 18		3410	SVA205 CLI	SVABSW(,@BR),SVAOFF	IS BR SW OFF ?
0B13 F2 01 07		3411		JNE SVA220	NO, SCAN FOR VAR
		3412		*	
		3413		* INCREMENT PAST KEYWORD	
		3414		*	
0B16 C0 87 0CDC		3415	SVA210 B	SVA700	INCR PAST KEYWORD
0B1A 7C 01 18		3416		MVI SVABSW(,@BR),SVAONN	SET BR SW ON
		3417		*	
		3418		* SCAN FOR VARIABLE AND RETURN	
		3419		*	
0B1D C0 87 0B6C		3420	SVA220 B	SVA395	SCAN FOR VAR
0B21 F2 87 39		3421		J SVA315	RETURN

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23	PAGE 22
					3423	*****			
					3424	*			
					3425	* ROUTINE TO PROCESS MATRIX GET AND PUT STATEMENTS			
					3426	*			
					3427	*****			
					3428	*			
					3429	*****			
					3430	*			
	0B24	C2	01	0D8F	3431	SVA250 LA	SVA960,@BR	SET BASE ADDR	
					3432	*			
					3433	* TEST BRANCH SWITCH			
					3434	*			
	0B28	7D	00	18	3435	SVA255 CLI	SVABSW(,@BR),SVAOFF	IS BR SW OFF	
	0B2B	F2	01	0D	3436	JNE	SVA270	NO, SCAN FOR VAR	
					3437	*			
					3438	* INCREMENT PAST THE KEYWORDS AND FILE NAME			
					3439	*			
	0B2E	7C	01	18	3440	SVA260 MVI	SVABSW(,@BR),SVAONN	SET BR SW ON	
	0B31	7C	00	1E	3441	MVI	SVAPCT(,@BR),SVAOFF	SET PAREN CT TO 0	
	0B34	C0	87	0CDC	3442	B	SVA700	SKIP KEYWORD	
	0B38	F2	87	0A	3443	J	SVA305	PROCESS AS VARIABLE	1-4
					3444	*			
					3445	* DETERMINE IF VARIABLE IS IN AN ARITHMETIC EXPRESION			
					3446	*			
	0B3B	C0	87	0CF3	3447	SVA270 B	SVA900	TO FIRST ALPHA BYTE	
	0B3F	7D	00	1E	3448	SVA300 CLI	SVAPCT(,@BR),@ZERO	IS PAREN COUNT ZERO	
	0B42	F2	81	07	3449	JE	SVA310	YES, SET TYPE CODE	
	0B45	C0	87	0B6F	3450	SVA305 B	SVA400	NO, SCAN FOR VAR TYPE	
	0B49	F2	87	11	3451	J	SVA315	RETURN	
					3452	*			
					3453	* SET VARIABLE TYPE CODE			
					3454	*			
	0B4C	34	02	0B59	3455	SVA310 ST	SVA312+@OP1,@XR	SAVE VAR ADDR	
	0B50	7C	01	23	3456	MVI	SVALNG(,@BR),SVAVL1	SET LNG TO 1	
	0B53	D0	87	00	3457	B	SVA960(,@BR)	COUNT BLANKS	
	0B56	C2	02	0000	3458	SVA312 LA	*-*,@XR	RESTORE VAR ADDR	
	0B5A	7C	08	22	3459	MVI	SVAVTC(,@BR),SVANAC	TYPE CODE TO ARITH ARRAY	
	0B5D	C2	01	0000	3460	SVA315 LA	*-*,@BR	RESTORE PT	
	0B61	C0	87	0000	3461	SVA320 B	*-*	RETURN TO CALLING RTN	
					3463	*****			
					3464	*			
					3465	* END OF STATEMENT ROUTINE			
					3466	*			
					3467	*****			
					3468	*			
					3469	*****			
					3470	*			
					3471	* TURN BRANCH SWITCH OFF AND RETURN TO CALLING ROUTINE			
					3472	*			
	0B65	7C	00	18	3473	SVA330 MVI	SVABSW(,@BR),SVAOFF	SET BR SW OFF	
	0B68	C0	87	0B5D	3474	B	SVA315	RETURN	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 23
					3476	*****				
					3477	*				*
					3478	* VARIABLE SCAN ROUTINE				*
					3479	*				*
					3480	*****				*
					3481	*				*
					3482	*****				*
					3483	*				*
	0B6C	7C	0F	1E	3484	SVA395 MVI	SVAPCT(, @BR), SVAL15		SET PAREN CT TO A MAX	
					3485	*				*
					3486	* ENTER AND SAVE RETURN ADDRESS				*
					3487	*				*
	0B6F	34	08	0CDB	3488	SVA400 ST	SVA670+@OP1, @ARR		SAVE RETURN ADDR	
					3489	*				*
					3490	* INCREMENT TO THE NEXT ALPHA BYTE ENCOUNTERED IN THE LINE AND STACK IT				*
					3491	*				*
	0B73	C0	87	0CF3	3492	SVA410 B	SVA900		TO ALPHA BYTE	
	0B77	6C	00	1F 00	3493	MVC	SVALS1(1, @BR), SVAPD0(, @XR)		STACK BYTE	
	0B7B	34	02	0CD7	3494	ST	SVA660+@OP1, @XR		SAVE VAR ADDRESS	
					3495	*				*
					3496	* SET VARIABLE LENGTH INITIALLY TO ONE				*
					3497	*				*
	0B7F	7C	01	23	3498	SVA415 MVI	SVALNG(, @BR), SVAVL1		VAR LNG EQ 1	
					3499	*				*
					3500	* INCREMENT THE LINE POINTER TO 1ST NON-BLANK BYTE AND STACK BYTE				*
					3501	*				*
	0B82	D0	87	00	3502	SVA420 B	SVA960(, @BR)		TO 1ST NON-BLANK BYTE	
	0B85	6C	00	20 00	3503	MVC	SVALS2(1, @BR), SVAPD0(, @XR)		STACK BYTE	
					3504	*				*
					3505	* TEST FOR A LETTER-DIGIT VARIABLE REFERENCE				*
					3506	*				*
	0B89	BD	F0	00	3507	SVA430 CLI	SVAPD0(, @XR), B@DEC0		BYTE A DIGIT	
	0B8C	F2	02	36	3508	JNL	SVA480		YES, SET TYPE CODE	
					3509	*				*
					3510	* TEST FOR AN ARITHMETIC ARRAY REFERENCE				*
					3511	*				*
	0B8F	BD	4D	00	3512	SVA440 CLI	SVAPD0(, @XR), B@LPAR		BYTE A LEFT PAREN	
	0B92	F2	81	3D	3513	JE	SVA490		YES, SET TYPE CODE	
					3514	*				*
					3515	* TEST CHARACTER VARIABLE OR ARRAY REFERENCE				*
					3516	*				*
	0B95	BD	5B	00	3517	SVA450 CLI	SVAPD0(, @XR), B@CVAR		BYTE A \$	
	0B98	F2	81	3D	3518	JE	SVA500		YES, TEST FOR ARRAY REF	
					3519	*				*
					3520	* TEST FOR A KEYWORD OR FUNCTION REFERENCE				*
					3521	*				*
	0B9B	BD	C1	00	3522	SVA460 CLI	SVAPD0(, @XR), @CHARA		IF BYTE IS IN STANDARD ALPHABET	
	0B9E	F2	82	06	3523	JL	SVA465		* TEST FOR KEYWORD OR	
	0BA1	BD	E9	00	3524	CLI	SVAPD0(, @XR), @CHARZ		* FUNCTION	
	0BA4	F2	04	4B	3525	JNH	SVA530		* REFERENCE	
	0BA7	BD	7B	00	3526	SVA465 CLI	SVAPD0(, @XR), @NUMBR		TEST FOR SPECIAL ALPHABETIC	
	0BAA	F2	81	45	3527	JE	SVA530		* CHARACTERS, IF EQUAL TO	
	0BAD	BD	7C	00	3528	CLI	SVAPD0(, @XR), @ASIGN		* ANY, TEST FOR FUNCTION	
	0BB0	F2	81	3F	3529	JE	SVA530		* REFERENCE OR KEYWORD	
	0BB3	BD	5B	00	3530	CLI	SVAPD0(, @XR), @DOLAR		* \$ INCLUDED FOR WTC	
	0BB6	F2	81	39	3531	JE	SVA530		* CONSIDERATION	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 24
				3533		*****				
				3534	*	ASSUME A LETTER VARIABLE REFERENCE AND SET TYPE CODE PARAMETER	*			
				3535		*****				
				3536	*					
0BB9	7D	00	1E	3537	SVA470	CLI	SVAPCT(,@BR),@ZERO		IS PAREN CT 0	
0BBC	F2	81	13	3538		JE	SVA490		YES, IN A MAT STATEMENT	
0BBF	7C	01	22	3539		MVI	SVAVTC(,@BR),SVALVC		SET VAR TYPE CODE	
0BC2	F2	87	2A	3540		J	SVA525		RETURN	
				3541	*					
				3542		*****				
				3543	*	SET TYPE CODE AND VARIABLE LENGTH PARAMETERS FOR LETTER-DIGIT VAR	*			
				3544		*****				
				3545	*					
0BC5	5E	00	23 27	3546	SVA480	ALC	SVALNG(,@BR),SVAI01(1,@BR)		INCR LNG COUNT	
0BC9	D0	87	00	3547		B	SVA960(,@BR)		COUNT BLANKS TO NON-BLANK BYTE	
0BCC	7C	10	22	3548		MVI	SVAVTC(,@BR),SVALDC		SET VAR TYPE CODE	
0BCF	F2	87	1D	3549		J	SVA525		RETURN	
				3551		*****				
				3552	*	SET TYPE CODE FOR AN ARITHMETIC ARRAY VARIABLE	*			
				3553		*****				
0BD2	7C	08	22	3554	SVA490	MVI	SVAVTC(,@BR),SVANAC		SET VAR TYPE CODE	
0BD5	F2	87	17	3555		J	SVA525		RETURN	
				3557		*****				
				3558	*					
				3559	*	CHARACTER REFERENCE PROCESSING ROUTINE	*			
				3560	*					
				3561		*****				
				3562	*					
				3563	*	INCREMENT LINE POINTER TO NEXT NON-BLANK BYTE				
				3564	*					
0BD8	5E	00	23 27	3565	SVA500	ALC	SVALNG(,@BR),SVAI01(1,@BR)		INCR LNG COUNT	
0BDC	D0	87	00	3566		B	SVA960(,@BR)		TO NEXT NON-BLANK BYTE	
				3567	*					
				3568	*	TEST FOR CHARACTER ARRAY REFERENCE				
				3569	*					
0BDF	BD	4D	00	3570	SVA505	CLI	SVAPD0(,@XR),B@LPAR		IS BYTE A LEFT PAREN	
0BE2	C0	81	0BEC	3571		BE	SVA520		YES, SET CHAR ARRAY TYPE CODE	
				3572	*					
				3573		*****				
				3574	*					
				3575	*	SET TYPE CODE PARAMETER FOR CHARACTER VARIABLE	*			
				3576	*					
				3577		*****				
0BE6	7C	04	22	3578	SVA510	MVI	SVAVTC(,@BR),SVACVC		SET VAR TYPE CODE	
0BE9	F2	87	03	3579		J	SVA525		RETURN	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 25
					3581	*****				
					3582	*				
					3583	* SET TYPE CODE PARAMETER FOR A CHARACTER ARRAY VARIABLE				
					3584	*				
					3585	*****				
	0BEC	7C	02	22	3586	SVA520 MVI	SVAVTC(, @BR), SVACAC		SET VS TYPE CONE	
	0BEF	F2	87	E2	3587	SVA525 J	SVA660		RETUNN	
					3589	*****				
					3590	*				
					3591	* KEYWORD OR FUNCTION REFERENCE DISCRIMINATION ROUTINE				
					3592	*				
					3593	*****				
					3594	*				
					3595	* TEST FOR PRESENCE OF AN EMBEDDED KEYWORD - IT IS ASSUMED THAT NO				
					3596	* INTRINSIC FUNCTION NAME BEGINS WITH A KEYWORD IDENTIFIER				
					3597	*				
	0BF2	5D	01	2A 20	3598	SVA530 CLC	SVAKTO(SVAKLN, @BR), SVALS2(, @BR)		KEYWORD 'TO'	
	0BF6	F2	81	B7	3599		JE SVA635		YES, PROCESS KEYWORD	
	0BF9	5D	01	2C 20	3600		CLC SVAKST(SVAKLN, @BR), SVALS2(, @BR)		KEYWORD 'STEP'	
	0BFD	F2	01	07	3601		JNE SVA535		NO, GO CHECK 'THEN'	
	0C00	3C	01	0DA8	3602		MVI SVASSS, @B1		SET IND FOR POSSIBLE 'STEP'	
	0C04	F2	87	91	3603		J SVA630		GO PROCESS KEYWORD, MAYBE	
	0C07	5D	01	2E 20	3604	SVA535 CLC	SVAKTH(SVAKLN, @BR), SVALS2(, @BR)		KEYWORD 'THEN'	
	0C0B	F2	81	8A	3605		JE SVA630		YES, PROCESS KEYWORD	
	0C0E	5D	01	30 20	3606		CLC SVAKGO(SVAKLN, @BR), SVALS2(, @BR)		KEYWORD 'GOTO'	
	0C12	F2	81	83	3607		JE SVA630		YES, PROCESS KEYWORD	
					3608	*				
					3609	* STACK NEXT NON-BLANK BYTE				
					3610	*				
	0C15	D0	87	00	3611	SVA540 B	SVA960(, @BR)		INCR LINE PT TO NON-BLANK BYTE	
	0C18	5C	00	21 00	3612	SVA545 MVC	SVALS3(, @BR), SVAPD0(1, @BR)		STACK BYTE	
					3613	*				
					3614	* TEST FOR USER DEFINED FUNCTION - IT IS ASSUMED THAT NO				
					3615	* INTRINSIC FUNCTION NAME BEGINS WITH A USER FUNCTION IDENTIFIER				
					3616	*				
	0C1C	5D	01	38 20	3617	SVA550 CLC	SVAFNC(B@LUFN, @BR), SVALS2(, @BR)		USER FUNCTION ?	
	0C20	F2	81	94	3618		JE SVA640		YES, PROCESS USER FUNC	
	0C23	5D	02	3B 21	3619		CLC SVASTR(, @BR), SVALS3(, @BR)		STR FUNCTION ?	1-4
	0C27	F2	81	8D	3620		JE SVA640		TREAT AS USER FUNCTION	1-4
					3621	*				
					3622	* TEST FOR PRESENCE OF AN INTRINSIC FUNCTION NAME IT IS ASSUMED THAT				
					3623	* NO INTRINSIC FUNCTION NAME CONTAINS A KEYWORD IDENTIFIER				
					3624	*				
	0C2A	34	02	0C4F	3625	SVA560 ST	SVA580+@OP1, @XR		SAVE PRESENT LINE PT CADDR	
	0C2E	34	02	0CCF	3626		ST SVA650+@OP1, @XR		SAVE PRESENT LINE PT CADDR	
	0C32	3C	41	0C3C	3627		MVI SVA570+@OP1, SVAFTD		DISP TO LAST SYM ENTRY	
	0C36	D2	02	3C	3628	SVA565 LA	SVAIFT(, @BR), @XR		CADDR INTRINSIC FUNC	
	0C39	E2	02	00	3629	SVA570 LA	*-(, @XR), @XR		ACCESS SYM ENTRY	
	0C3C	6D	02	21 00	3630		CLC SVALS3(B@LIFN, @BR), SVAPD0(, @XR)		STACKED LETTERS A FUNC	
	0C40	F2	81	89	3631		JE SVA650		YES, PROCESS THE FUNC	
					3632	*				
					3633	* DECREMENT TABLE DISP TO NEXT FUNCTION SYMBOL ENTRY				
					3634	*				
	0C43	1F	00	0C3B 28	3635		SLC SVA570+@D1, SVAFIL(1, @BR)		DECR TO NEXT TABLE ENTRY	
	0C48	C0	84	0C36	3636		BH SVA565		LOOP UNTIL DISP = 0	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 26

```

3637 *
3638 * TEST FOR THE RANDOM NUMBER FUNCTION SYMBOL
3639 *
0C4C C2 02 0000 3640 SVA580 LA *-*,@XR RESTORE LINE PT
0C50 5D 02 21 36 3641 CLC SVALS3(B@LIFN,@BR),SVARND(,@BR) THE RANDOM FUNC SYM
0C54 F2 01 0D 3642 JNE SVA600 NO, TEST FOR DET FUNC
3643 *
3644 * PROCESS RANDOM NUMBER FUNCTION
3645 *
0C57 D0 87 00 3646 SVA590 B SVA960(,@BR) TO NEXT NON-BLANK BYTE
0C5A BD 4D 00 3647 CLI SVAPD0(,@XR),B@LPAR A LEFT PAREN
0C5D F2 81 5A 3648 JE SVA645 YES, SCAN FOR A VAR
0C60 C0 87 0B73 3649 B SVA410 CONTINUE SCAN
3650 *
3651 * TEST FOR THE DETERMINANT FUNCTION SYMBOL
3652 *
0C64 5D 02 21 33 3653 SVA600 CLC SVALS3(B@LIFN,@BR),SVADET(,@BR) THE DET FUNC SYM
0C68 F2 01 17 3654 JNE SVA620 NO, PROCESS VAR
3655 *
3656 * PROCESS THE DETERMINANT FUNCTION
3657 *
0C6B 76 02 27 3658 SVA610 A SVAI01(,@BR),@XR INCR TO NEXT BYTE
0C6E C0 87 0CF3 3659 B SVA900 TO NEXT ALPHA BYTE
0C72 34 02 0CD7 3660 ST SVA660+@OP1,@XR SAVE VAR ADDR
0C76 7C 01 23 3661 MVI SVALNG(,@BR),SVAVL1 RESET LNG CT TO 1
0C79 D0 87 00 3662 B SVA960(,@BR) DETERMINE LNG OF VAR
0C7C 7C 08 22 3663 MVI SVAVTC(,@BR),SVANAC SET VAR TYPE CODE
0C7F F2 87 52 3664 J SVA660 RETURN
3665 *
3666 * ASSUME THAT WE HAVE A SIMPLE LETTER VARIABLE FOLLOWED WITH AN
3667 * EMBEDDED STATEMENT KEYWORD
3668 *
0C82 0C 01 0C8E 0CD7 3669 SVA620 MVC SVA625+@OP1,SVA660+@OP1(@CADDR) R ESTORE PT TO VAR BYTE
0C88 7C 01 22 3670 MVI SVAVTC(,@BR),SVALVC SET VAR TYPE CODE
0C8B C2 02 0000 3671 SVA625 LA *-*,@XR RESTORE VAR ADDR TO PT
0C8F 7C 01 23 3672 MVI SVALNG(,@BR),SVAVL1 RESET LNG CT TO 1
0C92 D0 87 00 3673 B SVA960(,@BR) DETERMINE LNG OF VAR
0C95 F2 87 3C 3674 J SVA660 RETURN
3676 *****
3677 *
3678 * PROCESS EMBEDDED KEYWORD
3679 *
3680 *****
3681 *
0C98 D0 87 00 3682 SVA630 B SVA960(,@BR) INCR TO 3RD LETTER
0C9B 3D 01 0DA8 3683 CLI SVASSS,@B1 IS 'STEP' POSSIBLE KEYWORD ?
0C9F F2 01 0B 3684 JNE SVA632 NO, GO INCR TO 4TH LETTER
0CA2 3C 00 0DA8 3685 MVI SVASSS,@ZERO SET 'STEP' INDICATOR OFF
0CA6 BD C5 00 3686 CLI 0(,@XR),B@EXPC DOES CHAR 'E' FOLLOW 'ST' ?
0CA9 C0 01 0C18 3687 BNE SVA545 IF NOT, RETURN TO SIMPLE VAR
0CAD D0 87 00 3688 SVA632 B SVA960(,@BR) INCR TO 4TH LETTER
0CB0 D0 87 00 3689 SVA635 B SVA960(,@BR) TO 1ST NON-BLANK BYTE AFTER
3690 * * KEYWORD
0CB3 C0 87 0B73 3691 B SVA410 CONTINUE SCAN

```

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	27
				3693		*****					
				3694		*					*
				3695		* PROCESS USER DEFINED FUNCTION REFERENCE					*
				3696		*					*
				3697		*****					*
		0CB7	D0 87 00	3698	SVA640	B	SVA960(,@BR) TO LEFT PAREN				
		0CBA	5E 00 1E 27	3699	SVA645	ALC	SVAPCT(1,@BR),SVAI01(,@BR) INCR PAREN COUNT				
		0CBE	D0 87 00	3700		B	SVA960(,@BR) INCR TO NEXT NON-BLANK BYTE				
		0CC1	BD 4D 00	3701		CLI	0(,@XR),B@LPAR AT LEFT PAREN				
		0CC4	C0 01 0B73	3702		BNE	SVA410 NO, SCAN FOR VARS				
		0CC8	C0 87 0CBA	3703		B	SVA645 YES, INCR PAREN COUNT				
				3704		*					
				3705		*****					*
				3706		*					*
				3707		* PROCESS INTRINSIC FUNCTION REFERENCE					*
				3708		*					*
				3709		*****					*
				3710		*					*
		0CCC	C2 02 0000	3711	SVA650	LA	*-*,@XR RESTORE LINE PT				
		0CD0	C0 87 0CB7	3712		B	SVA640 SCAN TO VAR IN PARENS				
				3713		*****					*
				3714		*					*
				3715		* VARIABLE SCAN EXIT ROUTINE					*
				3716		*					*
				3717		*****					*
				3718		*					*
		0CD4	C2 02 0000	3719	SVA660	LA	*-*,@XR				
		0CD8	C0 87 0000	3720	SVA670	B	*-*				

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	28
				3722		*****					
				3723		*					*
				3724		*****					*
				3725		*					*
				3726	*	ROUTINE TO INCREMENT PAST THE STATEMENT KEYWORD					*
				3727		*					*
				3728		*****					*
				3729		*					*
				3730		*****					*
				3731		*					*
0CDC	34	08	0CF2	3732	SVA700	ST	SVA750+@OP1,@ARR			SAVE	RETURN ADDR
				3733		*					*
				3734	*	INCREMENT TO NEXT ALPHA BYTE					*
				3735		*					*
0CE0	C0	87	0CF3	3736	SVA710	B	SVA900			GO TO	NEXT ALPHA BYTE
0CE4	E2	02	01	3737		LA	1(,@XR),@XR			INCR	PT
				3738		*					*
				3739	*	DECREMENT KEYWORD LETTER COUNT AND TEST FOR ZERO					*
				3740		*					*
0CE7	5F	00	1A 27	3741	SVA720	SLC	SVAKWL(,@BR),SVAI01(,@BR)			DECR	LETTER CT
0CEB	C0	84	0CE0	3742		BH	SVA710			LOOP	UNTIL CT = 0
				3743		*					*
				3744	*	RETURN					*
				3745		*					*
0CEF	C0	87	0000	3746	SVA750	B	*-*			RETURN	

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 29
					3748	*****		
					3749	*		
					3750	* SCAN STATEMENT LINE FOR FIRST ALPHABETIC BYTE		
					3751	*		
					3752	*****		
					3753	*		
					3754	*****		
					3755	*		
0CF3	34	08	0D83		3756	SVA900 ST	SVA940+@OP1,@ARR	SAVE RETURN ADDR
0CF7	7C	40	1D		3757	MVI	SVADSW(,@BR),B@BLNK	INIT DIGIT SW
					3758	*		
					3759	* TEST FOR INTERNAL CONSTANTS		
					3760	*		
0CFA	BD	50	00		3761	SVA902 CLI	0(,@XR),B@ICON	AN INTERNAL CON
0CFD	F2	01	18		3762	JNE	SVA910	NO, TEST FOR LITERAL
					3763	*		
					3764	* INCREMENT PAST THE INTERNAL CONSTANT		
					3765	*		
0D00	D0	87	00		3766	SVA904 B	SVA960(,@BR)	TO NEXT LETTER
0D03	BD	C5	00		3767	CLI	0(,@XR),B@CIEX	&E
0D06	F2	81	0C		3768	JE	SVA908	YES, INCR PAST IT
0D09	BD	D7	00		3769	CLI	0(,@XR),B@CIPI	&PI
0D0C	F2	81	03		3770	JE	SVA906	YES, INCR PAST IT
0D0F	D0	87	00		3771	B	SVA960(,@BR)	ASSUME &SQR2
0D12	D0	87	00		3772	SVA906 B	SVA960(,@BR)	INCR TO NEXT LETTER
0D15	D0	87	00		3773	SVA908 B	SVA960(,@BR)	INCR TO NEXT LETTER
					3774	*		
					3775	* TEST FOR LITERAL		
					3776	*		
0D18	BD	7D	00		3777	SVA910 CLI	SVAPD0(,@XR),B@SQUO	IS BYTE A QUOTE ?
0D1B	F2	01	11		3778	JNE	SVA920	NO, CHECK EOS
0D1E	76	02	27		3779	SVA915 A	SVAI01(,@BR),@XR	INCR TO NEXT BYTE
0D21	BD	7D	00		3780	CLI	SVAPD0(,@XR),B@SQUO	IS BYTE A QUOTE ?
0D24	C0	01	0D1E		3781	BNE	SVA915	NO, GET NEXT BYTE
0D28	76	02	27		3782	A	SVAI01(,@BR),@XR	INCR TO NEXT BYTE
0D2B	C0	87	0CFA		3783	B	SVA902	CHECK FOR QUOTE
					3784	*		
					3785	* TEST FOR EOS		
					3786	*		
0D2F	BD	1E	00		3787	SVA920 CLI	SVAPD0(,@XR),@EOS	AT EOS ?
0D32	C0	81	0B65		3788	BE	SVA330	YES, RETURN
					3789	*		
					3790	* TEST AND PROCESS PARENS		
					3791	*		
0D36	BD	4D	00		3792	SVA925 CLI	0(,@XR),B@LPAR	A LEFT PAREN
0D39	F2	01	07		3793	JNE	SVA928	NO, TEST FOR RT PAREN
0D3C	5E	00	1E 27		3794	ALC	SVAPCT(1,@BR),SVAI01(,@BR)	INCR PAREN COUNT
0D40	F2	87	41		3795	J	SVA950	INCR TO NEXT BYTE
0D43	BD	5D	00		3796	SVA928 CLI	0(,@XR),B@RPAR	A RT PAREN
0D46	F2	01	07		3797	JNE	SVA930	NO. TEST FOR A LETTER
0D49	5F	00	1E 27		3798	SLC	SVAPCT(1,@BR),SVAI01(,@BR)	DECR PAREN COUNT
0D4D	F2	87	34		3799	J	SVA950	TO NEXT BYTE
					3801	*		
					3802	* TEST FOR ALPHABETIC BYTE		
					3803	*		

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	30
	0D50	7D	5B	00	3804	SVA930	CLI SVAPD0(,@BR),@DOLAR				IS BYTE A \$?
	0D53	F2	81	2A	3805		JE SVA940				YES, RETURN
	0D56	BD	7B	00	3806		CLI SVAPD0(,@XR),@NUMBR				IS BYTE A # ?
	0D59	F2	81	24	3807		JE SVA940				YES, RETURN
	0D5C	BD	7C	00	3808		CLI SVAPD0(,@XR),@ASIGN				IS BYTE A @ ?
	0D5F	F2	81	1E	3809		JE SVA940				YES, RETURN
	0D62	BD	C1	00	3810		CLI SVAPD0(,@XR),@CHARA				LT LETTER A ?
	0D65	F2	82	1C	3811		JL SVA950				YES, INCR TO NEXT BYTE
	0D68	BD	E9	00	3812		CLI SVAPD0(,@XR),@CHARZ				GT LETTER Z ?
	0D6B	F2	84	16	3813		JH SVA950				YES, INCR TO NEXT BYTE
	0D6E	BD	C5	00	3814		CLI SVAPD0(,@XR),B@EXPC				AN EXP ?
	0D71	F2	01	0C	3815		JNE SVA940				NO, RETURN
	0D74	7D	F0	1D	3816		CLI SVADSW(,@BR),B@DEC0				PREVIOUS CHAR A DIGIT ?
	0D77	F2	02	0A	3817		JNL SVA950				YES, INCR PAST EXP
	0D7A	7D	4B	1D	3818		CLI SVADSW(,@BR),B@DPNT				A DEC POINT ?
	0D7D	F2	81	04	3819		JE SVA950				YES, SKIP EXP
	0D80	C0	87	0000	3820	SVA940	B *-*				RETURN
					3821	*					
					3822	*	INCREMENT LINE POINTER AND RECYCLE LOOP				
					3823	*					
	0D84	6C	00	1D	00	3824	SVA950 MVC SVADSW(,@BR),SVAPD0(,@XR)				SAVE PRESENT CHAR
	0D88	D0	87	00	3825		B SVA960(,@BR)				INCR PT
	0D8B	C0	87	0CFA	3826		B SVA902				TEST FOR LITERAL

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 31
		3828		*****	
		3829		*	*
		3830		* SCAN PAST BLANKS AND KEEP A COUNT	*
		3831		*	*
		3832		*****	
		3833		*	*
		3834		*****	
		3835		*	
		3836		* SAVE RETURN ADDRESS AND ZERO BLANK COUNTER	
		3837		*	
0D8F 74 08 0C		3838	SVA960 ST	SVA970+@OP1(,@BR),@ARR	SAVE RETURN ADDR
0D92 76 02 27		3839	A	SVAI01(,@BR),@XR	INCR TO NEXT BYTE
		3840		*	
		3841		* TEST FOR BLANK	
		3842		*	
0D95 BD 40 00		3843	SVA966 CLI	SVAPD0(,@XR),@BLANK	AT A BLANK ?
0D98 C0 01 0000		3844	SVA970 BNE	*-*	NO, RETURN
		3845		*	
		3846		* INCREMENT LINE POINTER AND BLANK COUNT	
		3847		*	
0D9C 76 02 27		3848	SVA975 A	SVAI01(,@BR),@XR	INCR LINE PT
0D9F 5E 00 23 27		3849	ALC	SVALNG(,@BR),SVAI01(1,@BR)	INCR BLANK CT
0DA3 C0 87 0D95		3850	B	SVA966	CHECK FOR A BLANK

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 32
			3852	*****		
			3853	*****		
			3854	*		*
			3855	* SVARAB WORK AREAS, CONSTANTS AND EQUATES		*
			3856	*		*
			3857	*****		
			3858	*****		
			3859	*		
			3860	* SVRAB EQUATES REFERENCING CONSTANTS		
			3861	*		
		0000	3862	SVAPD0 EQU	0	PT DISP OF 0
		0000	3863	SVA0TD EQU	0	TBL DISP OF 0
		0000	3864	SVAOFF EQU	0	TEST FOR BR SW OFF
		0001	3865	SVAONN EQU	1	TO TURN BR SW ON
		0001	3866	SVA1TD EQU	1	TBL DISP TO BR ADDR
		0001	3867	SVAVL1 EQU	1	VAR LNG OF 1
		0002	3868	SVA2TD EQU	2	TBL DISP TO KEYWORD LNG
		0002	3869	SVAKLN EQU	2	LNG OF EMBEDDED KEYWORD INDR
		0002	3870	SVAVL2 EQU	2	VAR LNG OF 2
		0003	3871	SVAVL3 EQU	3	VAR LNG OF 3
		000F	3872	SVAL15 EQU	15	PAREN COUNT SET SO IT WILL
			3873	*		* NOT BECOME 0 IN A SCALAR INST
		0041	3874	SVAFTD EQU	65	DISP TO LAST BYTE FUNC TABLE
		0080	3875	SVADIS EQU	X'80'	MASK TO TEST FOR DISABLE INST
			3876	*		
			3877	* VARIABLE REFERENCE TYPE CODES		
			3878	*		
		0001	3879	SVALVC EQU	X'01'	CODE FOR A LETTER VAR REF
		0010	3880	SVALDC EQU	X'10'	CODE FOR A LETTER DIGIT VAR REF
		0004	3881	SVACVC EQU	X'04'	CODE FOR A CHAR VAR REF
		0008	3882	SVANAC EQU	X'08'	CODE FOR A ARITH ARRAY REF
		0002	3883	SVACAC EQU	X'02'	CODE FOR A CHAR ARRAY REF
			3884	*		
			3885	* SVARAB WORK AREAS		
			3886	*		
0DA7		0DA7	3887	SVABSW DS	CL1	BRANCH SWITCH USED TO DETERMINE
0DA7			3888	ORG	*-1	* IF THE KEYWORD NEEDS TO BE
0DA7 00		0DA7	3889	DC	XL1'00'	* BY-PASSED, INITIALLY ZERO
0DA8		0DA8	3890	SVASSS DS	XL1	INDICATOR FOR 'STEP' FUNCTION -
0DA8			3891	ORG	SVASSS	* '01' -> POSSIBLE 'STEP', 'ST'
0DA8 00		0DA8	3892	DC	XL1'00'	* HAS BEEN FOUND. '00' -> OFF
0DA9		0DA9	3893	SVAKWL DS	CL1	KEYWORD LENGTH SAVE AREA,
0DA9			3894	ORG	SVAKWL	* INITIALLY SET TO ZERO. 1ST
0DA9 00		0DA9	3895	DC	XL1'00'	* BYTE ALWAYS ZERO
0DAA		0DAB	3896	SVASTC DS	CL2	STATEMENT TYPE CODE SAVE AREA,
0DAA			3897	ORG	*-2	* INITIALLY SET TO ZERO. 1ST
0DAA 0000		0DAB	3898	DC	XL2'00'	* BYTE ALWAYS ZERO
0DAC		0DAC	3899	SVADSW DS	CL1	DIGIT SW
0DAD		0DAD	3900	SVAPCT DS	CL1	PAREN COUNTER
0DAE		0DB0	3901	SVALSA DS	CL3	LETTER SAVE AREA
0DB1		0DB1	3902	SVAVTC DS	CL1	VARIABLE TYPE CODE SAVE AREA
0DB2		0DB2	3903	SVALNG DS	CL1	VAR LNG SAVE AREA
0DB3		0DB3	3904	SVAMAG DS	CL1	STR TYPE HOLDER 1-4
0DB4		0DB4	3905	SVAZRO DS	CL1	STR ZERO 1-4
			3906	*		
			3907	* SVARAB CONSTANTS		

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 33
					3908	*		
	0DB5	0001		0DB6	3909	SVAI01 DC	IL2'01'	INTEGER OF 1
	0DB7	03		0DB7	3910	SVAFIL DC	IL1'03'	VALUE TO DECR FUNC TABLE PT
					3911	*		
					3912	* EMBEDDED STATEMENT KEYWORD IDENTIFIERS		
					3913	*		
	0DB8	E3D6		0DB9	3914	SVAKTO DC	CL2'TO'	IDENTIFIER FOR KEYWORD 'TO'
	0DBA	E2E3		0DBB	3915	SVAKST DC	CL2'ST'	IDENTIFIER FOR KEYWORD 'STEP'
	0DBC	E3C8		0DBD	3916	SVAKTH DC	CL2'TH'	IDENTIFIER FOR KEYWORD 'THEN'
	0DBE	C7D6		0DBF	3917	SVAKGO DC	CL2'GO'	IDENTIFIER FOR KEYWORD 'GOTO'
					3918	*		
					3919	* DETERMINANT AND RANDOM NUMBER FUNCTION IDENTIFIERS		
					3920	*		
	0DC0	C4C5E3		0DC2	3921	SVADET DC	CL(B@LIFN)'DET'	DETERMINANT FUNCTION IDENTIFIER
	0DC3	D9D5C4		0DC5	3922	SVARNDC	CL(B@LIFN)'RND'	RND NUMBER FUNC IDENTIFIER
					3923	*		
					3924	* USER DEFINED FUNCTION IDENTIFIER		
					3925	*		
	0DC6	E3D5		0DC7	3926	SVAFNC DC	CL2'TN'	USER FUNCTION IDENTIFIER
	0DC8	E2E3D9		0DCA	3927	SVASTR DC	CL(B@LIFN)'STR'	1-4
					3928	*		
					3929	* INTRINSIC FUNCTION TABLE		
					3930	*		
				0DCB	3931	SVAIFT EQU	*	ADDR INTRINSIC FUNCTION TABLE
	0DCB	C1C2E2		0DCD	3932	DC	CL(B@LIFN)'ABS'	FUNCTION SYMBOL FOR ABSOLUTE
	0DCE	C9D5E3		0DD0	3933	DC	CL(B@LIFN)'INT'	FUNCTION SYMBOL FOR INTEGER
	0DD1	E2C7D5		0DD3	3934	DC	CL(B@LIFN)'SGN'	FUNCTION SYMBOL FOR SIGN
	0DD4	E2D6D9		0DD6	3935	DC	CL(B@LIFN)'SOR'	FUNCTION SYMBOL FOR SQ ROOT
	0DD7	D3D6C7		0DD9	3936	DC	CL(B@LIFN)'LOG'	FUNCTION SYMBOL FOR LOG E
	0DDA	D3C7E3		0DDC	3937	DC	CL(B@LIFN)'LGT'	FUNCTION SYMBOL FOR LOG 10
	0DDD	D3E3E6		0DDF	3938	DC	CL(B@LIFN)'LTW'	FUNCTION SYMBOL FOR LOG 2
	0DE0	C5E7D7		0DE2	3939	DC	CL(B@LIFN)'EXP'	FUNCTION SYMBOL FOR EXPONENTIAL
	0DE3	E3C1D5		0DE5	3940	DC	CL(B@LIFN)'TAN'	FUNCTION SYMBOL FOR TANGENT
	0DE6	C3D6E3		0DE8	3941	DC	CL(B@LIFN)'COT'	FUNCTION SYMBOL FOR COTANGENT
	0DE9	E2C9D5		0DEB	3942	DC	CL(B@LIFN)'SIN'	FUNCTION SYMBOL FOR SINE
	0DEC	E3D6E2		0DEE	3943	DC	CL(B@LIFN)'TOS'	FUNCTION SYMBOL FOR COSINE
	0DEF	E2C5C3		0DF1	3944	DC	CL(B@LIFN)'SEC'	FUNCTION SYMBOL FOR SECANT
	0DF2	C3E2C3		0DF4	3945	DC	CL(B@LIFN)'CSC'	FUNCTION SYMBOL FOR COSECANT
	0DF5	C1E3D5		0DF7	3946	DC	CL(B@LIFN)'ATN'	FUNCTION SYMBOL FOR AROTANGEN
	0DF8	C1E2D5		0DFA	3947	DC	CL(B@LIFN)'ASN'	FUNCTION SYMBOL FOR ARCSINE
	0DFB	C1C3E2		0DFD	3948	DC	CL(B@LIFN)'ACS'	FUNCTION SYMBOL FOR ARCCOSINE
	0DFE	C8E3D5		0E00	3949	DC	CL(B@LIFN)'HTN'	HYPERBOLIC TANGENT FUNC SYM
	0E01	C8E2D5		0E03	3950	DC	CL(B@LIFN)'HSN'	HYPERBOLIC SINE FUNC SYM
	0E04	C8C3E2		0E06	3951	DC	CL(B@LIFN)'HCS'	HYPERBOLIC COSINE FUNC SYM
	0E07	E3C5C7		0E09	3952	DC	CL(B@LIFN)'TEG'	CONVERT RAD TO DEG FUNC SYM
	0E0A	D9C1C4		0E0C	3953	DC	CL(B@LIFN)'RAD'	CONVERT DEG TO RAD FUNC SYM
					3954	*		
					3955	* INVERSE AND TRANSPOSE IDENTIFIERS		
					3956	*		
	0E0D	C9D5E5		0E0F	3957	SVAINV DC	CL(B@LIFN)'INV'	FUNCTION SYMBOL FOR INVERSE
	0E10	E3D9D5		0E12	3958	SVATRN DC	CL(B@LIFN)'TRN'	FUNCTION SYMBOL FOR TRANSPOSE
					3959	*		
					3960	* IDENTIFIERS FOR CONSTANT, IDENTITY AND ZERO.		
					3961	*		
	0E13	C3D6D5		0E15	3962	SVACON DC	CL(B@LIFN)'CON'	FUNCTION SYMBOL FOR CONSTANT
	0E16	C9C4D5		0E18	3963	SVAIDN DC	CL(B@LIFN)'IDN'	FUNCTION SYMBOL FOR IDENTITY

[illegible]

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 35
				3972	*****	
				3973	*****	
				3974	*	*
				3975	*****	
				3976	*	*
				3977	* SVARAB DISTRIBUTOR TABLE	*
				3978	*	*
				3979	*****	
				3980	*	*
				3981	*****	
				3982	*****	
				3983	*	
				3984	*	
		0E1C	3985	SVABRT EQU *	CADDR 1ST BYTE DISTRIBUTOR TBL	
			3986	*		
0E1C	03	0E1C	3987	DC	AL1(B@LREM)	LNG OF KEYWORD REM
0E1D	09D2	0E1E	3988	DC	AL(@CADDR)(SVA050)	RTN FOR LINES WITH NO VARS
			3989	*		
0E1F	04	0E1F	3990	DC	AL1(B@LDAT)	LNG OF KEYWORD DATA
0E20	09D2	0E21	3991	DC	AL(@CADDR)(SVA050)	RTN FOR LINES WITH NO VARS
			3992	*		
0E22	03	0E22	3993	DC	AL1(B@LDEF)	LNG OF KEYWORD DEF
0E23	0B0C	0E24	3994	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			3995	*		
0E25	03	0E25	3996	DC	AL1(B@LDIM)	LNG OF KEYWORD DIM
0E26	0B0C	0E27	3997	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			3998	*		
0E28	03	0E28	3999	DC	AL1(B@LLET)	LNG OF KEYWORD LET
0E29	0B0C	0E2A	4000	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4001	*		
0E2B	00	0E2B	4002	DC	XL1'00'	LNG ASSIGN SIMPLE
0E2C	09E1	0E2D	4003	DC	AL(@CADDR)(SVA070)	RTN FOR ASSIGNMENT SIMPLE
			4004	*		
0E2E	03	0E2E	4005	DC	AL1(B@LLET)	LNG OF KEYWORD LET
0E2F	0B0C	0E30	4006	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4007	*		
0E31	00	0E31	4008	DC	XL1'00'	LNG ASSIGN SIMPLE
0E32	09E1	0E33	4009	DC	AL(@CADDR)(SVA070)	RTN FOR ASSIGNMENT SIMPLE
			4010	*		
0E34	03	0E34	4011	DC	AL1(B@LLET)	LNG OF KEYWORD LET
0E35	0B0C	0E36	4012	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4013	*		
0E37	00	0E37	4014	DC	XL1'00'	LNG ASSIGN SIMPLE
0E38	09E1	0E39	4015	DC	AL(@CADDR)(SVA070)	RTN FOR ASSIGNMENT SIMPLE
			4016	*		
0E3A	03	0E3A	4017	DC	AL1(B@LKFR)	LNG OF KEYWORD FOR
0E3B	0B0C	0E3C	4018	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4019	*		
0E3D	04	0E3D	4020	DC	AL1(B@LNEX)	LNG OF KEYWORD NEXT
0E3E	0B0C	0E3F	4021	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4022	*		
0E40	02	0E40	4023	DC	AL1(B@LKIF)	LNG OF KEYWORD IF
0E41	0B0C	0E42	4024	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS
			4025	*		
0E43	02	0E43	4026	DC	AL1(B@LKIF)	LNG OF KEYWORD IF
0E44	0B0C	0E45	4027	DC	AL(@CADDR)(SVA200)	RTN FOR REMAINING NON-MATS

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 36
					4028	*				
	0E46	04		0E46	4029	DC	AL1(B@LGTO)			LNG OF KEYWORD GOTO SIMPLE
	0E47	0B0C		0E48	4030	DC	AL(@CADDR)(SVA200)			RTN FOR REMAINING NON-MATS
					4031	*				
	0E49	04		0E49	4032	DC	AL1(B@LGTO)			LNG OF KEYWORD GOTO
	0E4A	0A1B		0E4B	4033	DC	AL(@CADDR)(SVA090)			RTN FOR COMPUTED GOTO
					4034	*				
	0E4C	05		0E4C	4035	DC	AL1(B@LGSB)			LNG OF KEYWORD GO SUB
	0E4D	09D2		0E4E	4036	DC	AL(@CADDR)(SVA050)			RTN FOR LINES WITH NO VARS
					4037	*				
	0E4F	06		0E4F	4038	DC	AL1(B@LRTN)			LNG OF KEYWORD RETURN
	0E50	09D2		0E51	4039	DC	AL(@CADDR)(SVA050)			RTN FOR LINES WITH NO VARS
					4040	*				
	0E52	03		0E52	4041	DC	AL1(B@LKGT)			LNG OF KEYWORD GET
	0E53	0ADD		0E54	4042	DC	AL(@CADDR)(SVA170)			RTN FOR GET AND PUT
					4043	*				
	0E55	03		0E55	4044	DC	AL1(B@LKPT)			LNG OF KEYWORD PUT
	0E56	0ADD		0E57	4045	DC	AL(@CADDR)(SVA170)			RTN FOR GET AND PUT
					4046	*				
	0E58	05		0E58	4047	DC	AL1(B@LKRT)			LNG OF KEYWORD RESET
	0E59	0ADD		0E5A	4048	DC	AL2(SVA170)			RTN TO PROCESS VAR FILE REF
					4049	*				
	0E5B	05		0E5B	4050	DC	AL1(B@LKCL)			LNG OF KEYWORD CLOSE
	0E5C	0ADD		0E5D	4051	DC	AL2(SVA170)			RTN TO PROCESS VAR FILE REF
					4052	*				
	0E5E	05		0E5E	4053	DC	AL1(B@LINP)			LNG OF KEYWORD INPUT
	0E5F	0B0C		0E60	4054	DC	AL(@CADDR)(SVA200)			RTN FOR REMAINING NON-MATS
					4055	*				
	0E61	04		0E61	4056	DC	AL1(B@LREA)			LNG OF KEYWORD READ
	0E62	0B0C		0E63	4057	DC	AL(@CADDR)(SVA200)			RTN FOR REMAINING NON-MATS
					4058	*				
	0E64	07		0E64	4059	DC	AL1(B@LKRR)			LNG OF KEYWORD RESTORE
	0E65	09D2		0E66	4060	DC	AL(@CADDR)(SVA050)			RTN FOR LINES WITH NO VARS
					4061	*				
	0E67	05		0E67	4062	DC	AL1(B@LPRT)			LNG OF KEYWORD PRINT
	0E68	0B0C		0E69	4063	DC	AL(@CADDR)(SVA200)			RTN FOR REMAINING NON-MATS
					4064	*				
	0E6A	0A		0E6A	4065	DC	AL1(B@LKPU)			LNG OF KEYWORD PRINT USING
	0E6B	0B0C		0E6C	4066	DC	AL(@CADDR)(SVA200)			RTN FOR REMAINING NON-MATS
					4067	*				
	0E6D	01		0E6D	4068	DC	AL1(B@LIMG)			LNG OF KEYWORD IMAGE
	0E6E	09D2		0E6F	4069	DC	AL(@CADDR)(SVA050)			RTN FOR LINES WITH NO VMS
					4070	*				
	0E70	03		0E70	4071	DC	AL1(B@LMAT)			LNG OF KEYWORD MAT
	0E71	0A3F		0E72	4072	DC	AL(@CADDR)(SVA120)			RTN FOR MAT ASSIGNMENT
					4073	*				
	0E73	06		0E73	4074	DC	AL1(B@LMGT)			LNG OF KEYWORD MAT GET
	0E74	0B24		0E75	4075	DC	AL(@CADDR)(SVA250)			RTN FOR REMAINING MAT STMTS
					4076	*				
	0E76	08		0E76	4077	DC	AL1(B@LMIN)			LNG OF KEYWORD MAT INPUT
	0E77	0AF5		0E78	4078	DC	AL(@CADDR)(SVA192)			RTN FOR REMAINING MAT STMTS
					4079	*				
	0E79	07		0E79	4080	DC	AL1(B@LMRD)			LNG OF KEYWORD MAT READ
	0E7A	0AF5		0E7B	4081	DC	AL(@CADDR)(SVA192)			RTN FOR REMAINING MAT STMTS
					4082	*				
	0E7C	06		0E7C	4083	DC	AL1(B@LMPT)			LNG OF KEYWORD MAT PUT

#KRVLA - SET KEYWORD COMMAND ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	37
	0E7D	0B24		0E7E	4084	DC	AL(@CADDR)(SVA250)				RTN FOR REMAINING MAT STMTS
					4085	*					
	0E7F	08		0E7F	4086	DC	AL1(B@LMPR)				LNG OF KEYWORD MAT PRINT
	0E80	0AF5		0E81	4087	DC	AL(@CADDR)(SVA192)				RTN FOR REMAINING MAT STMTS
					4088	*					
	0E82	0D		0E82	4089	DC	AL1(B@LMPU)				LNG OF KEYWORD MAT PRINT USING
	0E83	0AF5		0E84	4090	DC	AL(@CADDR)(SVA192)				RTN FOR REMAINING MAT STMTS
					4091	*					
	0E85	05		0E85	4092	DC	AL1(B@LPSE)				LNG OF KEYWORD PAUSE
	0E86	09D2		0E87	4093	DC	AL(@CADDR)(SVA050)				RTN FOR LINES WITH NO VARS
					4094	*					
	0E88	04		0E88	4095	DC	AL1(B@LSTP)				LNG OF KEYWORD STOP
	0E89	09D2		0E8A	4096	DC	AL(@CADDR)(SVA050)				RTN FOR LINES WITH NO VARS
					4097	*					
	0E8B	03		0E8B	4098	DC	AL1(B@LEND)				LNG OF KEYWORD END
	0E8C	09D2		0E8D	4099	DC	AL(@CADDR)(SVA050)				RTN FOR LINES WITH NO VARS
					4100	*					
	0E8E			0E8E	4101	DS	CL1				DUMMY BYTE
	0E8F	09D2		0E90	4102	DC	AL(@CADDR)(SVA050)				RTN FOR LINES WITH NO VARS
					4103	*					
	0E91			0E91	4104	DS	CL1				DUMMY BYTE
	0E92	09D2		0E93	4105	DC	AL(@CADDR)(SVA050)				RTN FOR LINES WITH NO VARS
					4106	*					1-4
	0E94	03		0E94	4107	DC	AL1(B@LLET)				LNG OF LET(STR)
	0E95	0B0C		0E96	4108	DC	AL(@CADDR)(SVA200)				RTN, (STR)--SIMPLE
					4109	*					1-4
	0E97	03		0E97	4110	DC	AL1(B@LLET)				LNG OF LET(STR)
	0E98	0B0C		0E99	4111	DC	AL(@CADDR)(SVA200)				RTN, (STR)--MULTIPLE
					4112	*					1-4
	0E9A	00		0E9A	4113	DC	XL1'00'				LNG OF ASSIGN(STR)
	0E9B	09E1		0E9C	4114	DC	AL(@CADDR)(SVA070)				RTN, (STR)--SIMPLE
					4115	*					1-4
	0E9D	00		0E9D	4116	DC	XL1'00'				LNG OF ASSIGN(STR)
	0E9E	09E1		0E9F	4117	DC	AL(@CADDR)(SVA070)				RTN, (STR)--MULTIPLE
					4118	*					1-4
	0EA0	02		0EA0	4119	DC	AL1(B@LKIF)				LNG OF IF(STR)
	0EA1	0B0C		0EA2	4120	DC	AL(@CADDR)(SVA200)				RTN.(STR) IF STATEMENT
					4121	*					*
					4122	*	*****				
					4123	*					

GRABIT - RETRIEVE FILE STATEMENTS

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 38

```

4125 *****
4126 * 5703-XM1      COPYRIGHT IBM CORP. 1970
4127 *              REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083
4128 *
4129 *****
4130 *STATUS
4131 *  VERSION 1 MODIFICATION 0
4132 *
4133 *FUNCTION
4134 *  GRABIT LOCATES SEQUENTIAL STATEMENTS IN THE FILE SPECIFIED BY THE
4135 *  USER, AND, DEPENDING UPON THE OPTION CHOSEN, PASSES BACK THE
4136 *  STATEMENT OR SKIPS TO THE NEXT.
4137 *  AFTER BEING PRIMED BY THE CALLING PROGRAM, GRABIT READS LOGICALLY
4138 *  CONSECUTIVE BLOCKS OF SEGMENTED STATEMENTS, FROM THE FILE
4139 *  SPECIFIED BY THE USER, INTO CORE.  GRABIT RETURNS WITH @XR
4140 *  POINTING TO THE BINARY LINE NUMBER OF THE NEXT STATEMENT.
4141 *  IN ADDITION TO @XR, GRABIT PARAMETERS CAN BE SET TO CAUSE THE
4142 *  BINARY LINE NR, THE TYPE CODE AND THE UNPACKED, NON-SEGMENTED
4143 *  TEXT OF THE NEXT STMT TO BE PLACED IN AREAS DEFINED BY THE USER.
4144 *  IF GRABIT IS USED TO SKIP THROUGH THE STMTS WITHOUT UNPACKING
4145 *  THEM OR CHANGING THEIR LENGTH OR SEGMENTED CONDITION, GRABIT CAN
4146 *  BE INSTRUCTED TO RETURN THE BLOCKS TO THEIR ORIGINAL DISK ADDRESS
4147 *  IF THE SPECIFIED FILE IS ACCESSED BY DL4ICS.
4148 *
4149 *NOTES
4150 *  THIS VERSION OF GRABIT USES ONLY DL4ICS TO ACCESS THE NEXT DATA
4151 *  BLOCK.  GRABIT IN THE SUBROUTINE LIBRARY USES DL4ICS AND DL2ICS.
4152 *****
0F81 4153      USING GRABSE,@BR
0EA3 4154 GRABIT EQU *          ENTRY POINT TO ROUTINE
0EA3 4155      ST      GRASBR,@BR      SAVE CALLING PROG'S BASE REG.
0EA7 C2 01 0F81 4156      LA      GRABSE,@BR      LOAD LOCAL BASE TO BASE REG.
0EAB 34 08 0F24 4157      ST      GRASAR,@ARR      SAVE RETURN ADDR.
0EAF 7D 00 A7 4158      CLI    GRWHAT(,@BR),GRAEFI  IS FUNC REQ'D INITIALIZATION ?
0EB2 F2 81 13 4159      JE      GRA100          YES, GO TO INITIALIZATION RTN
4160 * THE ADDRESS OF THE NEXT SEGMENT IN THE CURRENT BUFFER IS INITLZ'D
4161 * AND MAINTAINED IN THE NEXT INST, WHICH LOADS IT TO THE @XR.
0EB5 C2 02 0000 4162 GRA020 LA      *-*,@XR          LOAD NEXT STMT CADDR TO @XR
0EB9 7D 01 A7 4163      CLI    GRWHAT(,@BR),GRAEFR  IS FUNC REQ'D RETURN TEXT ?
0EBC F2 81 87 4164      JE      GRA300          YES, GO RETURN STMT ROUTINE
0EBF 7D 02 A7 4165      CLI    GRWHAT(,@BR),GRAEFS  IS FUNC REQ'D SKIP STATEMENT
0EC2 F2 81 35 4166      JE      GRA200          YES, GO TO SKIP STMT ROUTINE
0EC5 F2 87 38 4167      J      GRA210          GO TO SKIP SEGMENT RTN
4168 *
4169 *          INITIALIZATION ROUTINE
4170 *
0EC8 75 02 A0 4171 GRA100 L      GRBFRA(,@BR),@XR      LOAD 1ST BFR ADDR TO DB
0ECB 74 02 A6 4172      ST      GRANCA(,@BR),@XR      PROPAGATE IT TO NEXT BFR DPL
0ECE 5C 01 A3 9D 4173      MVC     GRANDA(@DADDR,@BR),GRSRDA(,@BR)  INITLZ NEXT BR'F DADDR
0ED2 7C FF AC 4174      MVI     GRASIZ(,@BR),GRAEBS      INITLZ BUFFER SIZE COUNTER
0ED5 5C 00 9E A4 4175      MVC     GRACSC(1,@BR),GRSCTR(,@BR)  INITLZ SCTR COUNT IN DPL
0ED9 C0 87 0025 4176      B      $DISKN          WAIT FOR FIRST DATA BLOCKS TO
0EDD 057F 0EDE 4177      DC      AL2($WAITF)          * GET INTO CORE
0EDF 7C 97 B5 4178      MVI     GRAERR+@Q(,@BR),@E550      SET ERR CODE TO SPECIFY WRKFILE
0EE2 5E 01 A6 A9 4179      ALC     GRANCA(@CADDR,@BR),GRASSZ(,@BR)  SET CADDR OF NEXT BFR
0EE6 BD 00 00 4180 GRA140 CLI    GRAELK(,@XR),GRAELN      IS 1ST DB LINK CODE = 0 ?

```

GRABIT - RETRIEVE FILE STATEMENTS

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 39
	0EE9	F2	81 07		4181	JE	GRA150	YES, GO INCR TO NEXT LOGICAL DB
	0EEC	7C	02 A3		4182	MVI	GRANDA(, @BR), GRAEDB	SET DADDR OF NEXT DB
	0EEF	6E	00 A3 00		4183	ALC	GRANDA(1, @BR), GRAELK(, @XR) *	
	0EF3	5E	00 A3 AB		4184	GRA150 ALC	GRANDA(1, @BR), GRANPB(, @BR)	INCR TO NEXT BFR DADDR
	0EF7	F2	87 2E		4185	J	GRA260	GO ACCESS FIRST STATEMENT
					4186	*		
					4187	*	ACCESS NEXT STATEMENT OR NEXT SEGMENT ROUTINE	
					4188	*		
	0EFA	BD	75 07		4189	GRA200 CLI	GRAEDT(, @XR), GRAEET	END-OF-FILE RECORD ?
	0EFD	F2	81 16		4190	JE	GRA230	YES, RESET OR TO THIS RECORD
	0F00	6F	00 AC 02		4191	GRA210 SLC	GRASIZ(1, @BR), GRAES1(, @XR)	DECR BFR CT BY SEGMENT LENGTH
	0F04	B6	02 02		4192	A	GRAES1(, @XR), @XR	INCR OR BY SEGMENT LENGTH
	0F07	7D	00 AC		4193	GRA220 CLI	GRASIZ(, @BR), @ZERO	IS BUFFER EMPTY ?
	0F0A	D0	82 B4		4194	BL	GRAERR(, @BR)	GONE NEG, GO TO BAD ERR
	0F0D	F2	81 15		4195	JE	GRA250	YES, GO TO GET NEXT BFR
	0F10	BD	80 01		4196	CLI	GRAES0(, @XR), @SNUL	IS SEGMENT NULL ?
	0F13	F2	81 0F		4197	JE	GRA250	YES, GO TO GET NEXT BFR
	0F16	34	02 0EB8		4198	GRA230 ST	GRA020+@OP1, @XR	SAVE CADDR OF NEXT SEG. IN INST.
	0F1A	E2	02 06		4199	LA	GRAEDL(, @XR), @XR	POINT @XR TO LINE NUMBER
	0F1D	C2	01 0000		4200	GRA240 LA	*-*, @BR	RESTORE THE BASE REGISTER
				0F20	4201	GRASBR EQU	GRA240+@OP1	* STORED IN INST AT GRA240
	0F21	C0	87 0000		4202	GRA245 B	*-*	RETURN TO USER
				0F24	4203	GRASAR EQU	GRA245+@OP1	* TO CADDR SAVED IN GRA245
	0F25	D0	87 67		4204	GRA250 B	GRA500(, @BR)	ACCESS NEXT BUFFER
	0F28	BD	80 01		4205	GRA260 CLI	GRAES0(, @XR), @SNUL	IS 1ST SEG. NULL ?
	0F2B	D0	81 B4		4206	BE	GRAERR(, @BR)	YES, GO TO BAD ERR
	0F2E	B9	02 03		4207	TBF	GRAES2(, @XR), GRAETP	PRIMARY SEGMENT
	0F31	C0	10 0F16		4208	BT	GRA230	YES, SAVE LOCATION
	0F35	7D	01 A7		4209	CLI	GRWHAT(, @BR), GRAEFR	ACTION REQ'D = RETURN TEXT ?
	0F38	D0	81 B4		4210	BE	GRAERR(, @BR)	YES, GO TO BAD ERR
	0F3B	7D	04 A7		4211	CLI	GRWHAT(, @BR), GRAEFG	ACTION REQ'D = SKIP SEGMENT ?
	0F3E	C0	81 0F16		4212	BE	GRA230	YES, GO SAVE LOCATION
	0F42	C0	87 0F00		4213	B	GRA210	NO, GO SKIP THIS SEGMENT
					4214	*		
					4215	*	RETURN TEXT ROUTINE	
					4216	*		
	0F46	2C	01 1A05 06		4217	GRA300 MVC	GRLINE, GRAEDL(GRAELL, @XR)	SET BINARY LINE NO. IN O/P FIELD
	0F4B	2C	00 1A06 07		4218	MVC	GRTYPE, GRAEDT(1, @XR)	SET TYPE CODE IN OUTPUT FIELD
	0F50	4C	01 58 102F		4219	MVC	GRTEND(@CADDR, @BR), GRATXT	INITLZ TEXT O/P CADDR IN INST.
	0F55	BD	75 07		4220	CLI	GRAEDT(, @XR), GRAEET	END OF FILE STATEMENT ?
	0F58	F2	01 08		4221	JNE	GRA303	NO - GO RESET SEGMENT SWITCH
	0F5B	3C	1C 1A07		4222	MVI	GRTEXT, @EOF	MOVE EOF CODE TO GRTEXT
	0F5F	C0	87 0F16		4223	B	GRA230	GO GET OUT
					4225	GRA303 MVI	GRA310+@Q(, @BR), @UCB	INITLZ BRANCH FOR ONLY SEGMENT
	0F66	BD	00 03		4226	CLI	GRAES2(, @XR), @SONLY	IS IT AN ONLY SEGMENT ?
	0F69	F2	81 03		4227	JE	GRA305	YES, BYPASS BRANCH RESET
	0F6C	7C	80 01		4228	MVI	GRA310+@Q(, @BR), @NOP	SET FOR MORE SEGMENTS
	0F6F	6F	00 AC 02		4229	GRA305 SLC	GRASIZ(1, @BR), GRAES1(, @XR)	DECR BFR CT BY SEG LENGTH
	0F73	9F	00 02 B0		4230	SLC	GRAES1(1, @XR), GRAPSG(, @BR)	DECR SEG CT BY SDF-HDR LENGTH
	0F77	6C	00 B3 02		4231	MVC	GRASEG(1, @BR), GRAES1(, @XR)	MOVE TEXT LENGTH TO TEXT CTR
	0F7B	E2	02 07		4232	LA	GRAELP(, @XR), @XR	INCR TO TYPE CODE
	0F7E	F2	87 2A		4233	J	GRA317	GO TEST FILE TYPE
	0F81	C0	87 0F07		4234	GRA310 B	GRA220	GO ACCESS NEXT STATEMENT
	0F81				4235	ORG	GRA310	* UNLESS CURRENT STATEMENT
	0F81	C0	87 0F07		4236	BC	GRA220, @UCB	* HAS MORE SEGMENTS

GRABIT - RETRIEVE FILE STATEMENTS

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 40
	0F85	6C	00 24 00		4237	MVC	GRASVC(, @BR), @ZERO(1, @XR)	SAVE CURR CHAR IN RESTORE INST
	0F89	D0	87 67		4238	B	GRA500(, @BR)	ACCESS NEXT BUFFER
	0F8C	BD	02 03		4239	CLI	GRAES2(, @XR), @SLAST	LAST SEGMENT ?
	0F8F	F2	01 03		4240	JNE	GRA313	NO, GO RESET SEG COUNTER
	0F92	7C	87 01		4241	MVI	GRA310+@Q(, @BR), @UCB	RESET BRANCH OUT
	0F95	6F	00 AC 02		4242	GRA313 SLC	GRASIZ(1, @BR), GRAES1(, @XR)	DECR BUFFER COUNTER
	0F99	9F	00 02 B2		4243	SLC	GRAES1(1, @XR), GRASSG(, @BR)	DECR SEG COUNT BY SDF LENGTH
	0F9D	6C	00 B3 02		4244	MVC	GRASEG(1, @BR), GRAES1(, @XR)	MOVE TEXT LNG TO SEG COUNTER
	0FA1	E2	02 04		4245	LA	GRAELS(, @XR), @XR	INCR @XR PAST SECONDARY SDF
	0FA4	BC	00 00		4246	GRA315 MVI	@ZERO(, @XR), *-*	RESTORE CHAR SAVED IN Q-CODE
				0FA5	4247	GRASVC EQU	GRA315+@Q	SAVED CHAR HOLD AREA
	0FA7	5E	01 58 AB		4248	GRA316 ALC	GRTEND(@CADDR, @BR), GRABOA(, @BR)	INCR RECEIVING CADDR
				0FAB	4249	GRA317 EQU	*	MOVE TEXT TO GRTEXT
	0FAB	38	80 03D4		4250	TBN	\$INDR1, \$BASIC	IS FILE TYPE = BASIC ?
	0FAF	F2	90 24		4251	JF	GRA350	NO, BYPASS REPITION CODE CHECK
	0FB2	BD	1B 01		4252	CLI	GRAENC(, @XR), GRAEMR	IS CHAR REF A REPITION CODE ?
	0FB5	F2	84 1E		4253	JH	GRA350	NO, GO RETURN REF'D CHAR
	0FB8	5C	01 3E 58		4254	MVC	GRATND(@CADDR, @BR), GRTEND(, @BR)	SET RCV'G CADDR IN INSTR
	0FBC	2C	00 0000 00		4255	GRA320 MVC	*-*, @ZERO(1, @XR)	RETURN REPEATED CHAR TO OUTPUT
				0FBF	4256	GRATND EQU	GRA320+@OP1	* ADDR SUPPLIED
	0FC1	9F	00 01 AB		4257	SLC	GRAENC(1, @XR), GRAONE(, @BR)	DECR. REPITION COUNTER
	0FC5	F2	01 07		4258	JNZ	GRA330	IF <> 0, GO INCR O/P CADDR
	0FC8	5C	01 58 3E		4259	MVC	GRTEND(@CADDR, @BR), GRATND(, @BR)	RESTORE NEW O/P CADDR
	0FCC	F2	87 0C		4260	J	GRA360	GO INCR @XR
	0FCF	5E	01 3E AB		4261	GRA330 ALC	GRATND(@CADDR, @BR), GRABOA(, @BR)	INCR O/P CADDR IN INSTR
	0FD3	D0	87 3B		4262	B	GRA320(, @BR)	GO MOVE CHAR TO OUTPUT
	0FD6	2C	00 0000 01		4263	GRA350 MVC	*-*, GRAENC(1, @XR)	MOVE NON-REPEAT CHAR TO OUTPUT
				0FD9	4264	GRTEND EQU	GRA350+@OP1	* ADDR SUPPLIED
	0FDB	E2	02 01		4265	GRA360 LA	GRAENC(, @XR), @XR	INCR @XR TO NEXT CHAR.
	0FDE	5F	00 B3 AB		4266	SLC	GRASEG(1, @BR), GRABOA(, @BR)	DECR BFR SPACE CTR
	0FE2	D0	81 00		4267	BZ	GRA310(, @BR)	NO MORE TEXT IN SEG, CHK MORE
	0FE5	D0	87 26		4268	B	GRA316(, @BR)	MORE TEXT, GO INCR RECV CADDR
					4269	*		
					4270	*	ACCESS NEXT BUFFER ROUTINE	
					4271	*		
	0FE8	74	08 9A		4272	GRA500 ST	GRA5SA(, @BR), @ARR	
	0FEB	C0	87 0025		4273	B	\$DISKN	WAIT FOR PRIOR READ TO COMPLETE
	0FEF	057F		0FF0	4274	DC	AL2(\$WAITF)	*
				0FF1	4275	GRA600 EQU	*	
					4276	*		
					4277	*	DL4ICS BEING USED - ACCESS NEXT DATA BLOCK	
					4278	*		
	0FF1	75	02 A0		4279	L	GRBFRA(, @BR), @XR	SAVE CURR BFR STARTING CADDR
	0FF4	5C	04 A0 A6		4280	MVC	GRBFRA(GRAEDS, @BR), GRANCA(, @BR)	MOVE NEXT DPL TO CURR DPI
	0FF8	74	02 A6		4281	ST	GRANCA(, @BR), @XR	RESTORE NEXT BFR STARTING CADDR
	0FFB	75	02 A0		4282	L	GRBFRA(, @BR), @XR	POINT EN TO CURR BFR CADDR
	0FFE	BD	00 00		4283	CLI	GRAELK(, @XR), GRAELN	NEXT LOGICAL DB = NEXT PHYS DB ?
	1001	F2	81 07		4284	JE	GRA620	YES, GO INCR SCTR DISP.
	1004	7C	02 A3		4285	MVI	GRANDA(, @BR), GRAEDB	SET DADDR OF NEXT DB
	1007	6E	00 A3 00		4286	ALC	GRANDA(1, @BR), GRAELK(, @XR)	*
	100B	5E	00 A3 AB		4287	GRA620 ALC	GRANDA(1, @BR), GRANPB(, @BR)	INCR SCTR DISP FOR NEXT PHYS D
	100F	C0	87 1335		4288	GRA640 B	DL4ICS	GO READ NEXT DB
	1013	1022		1014	4289	DC	AL2(GRANPL)	* CADDR OF DPL
	1015	7C	FF AC		4290	GRA660 MVI	GRASIZ(, @BR), GRAEBS	RE-INITLZ BFR SPACE COUNT
	1018	C0	87 0000		4291	GRA680 B	*-*	RETURN TO
				101B	4292	GRA5SA EQU	GRA680+@OP1	* CADDR SUPPLIED

GRABIT - RETRIEVE FILE STATEMENTS

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 41
			101C	4293	GRACPL	EQU *	DPL FOR CURRENT BUFFER
101C	02		101C	4294	GRACFN	DC AL1(@DPUT)	WRITE FUNCTION CODE
101D			101E	4295	GRSRDA	DS CL2	RELATIVE DADDR OF CURR. BFR
			101D	4296	GRACCA	EQU GRSRDA-@B1	CYLINDER BYTE OF DISK ADDR.
101D				4297		ORG *-2	* INITIALIZED TO THE
101D	0503		101E	4298		DC AL2(@WSTBL)	* 1ST DB OF THE WORK FILE
101F			101F	4299	GRACSC	DS CL1	SECTOR COUNT
1020	1B00		1021	4300	GRBFRA	DC AL2(GRBFRA1)	CADDR OF CURRENT BUFFER
			1022	4301	GRANPL	EQU *	DPL FOR NEXT BUFFER
1022	01		1022	4302		DC AL1(@DGET)	READ FUNCTION CODE
1023			1024	4303	GRANDA	DS CL2	RELATIVE DADDR OF NEXT BFR.
1025			1025	4304	GRSCTR	DS CL1	SECTOR COUNT
1025				4305		ORG *-1	* INITIALIZE TO 1
1025	01		1025	4306		DC XL1'01'	
1026			1027	4307	GRANCA	DS CL2	CADDR OF NEXT BUFFER
1028			1028	4308	GRWHAT	DS CL1	USER SPEC'D FUNCTION CODE
1028				4309		ORG *-1	SET TO ZERO FOR
1028	00		1028	4310		DC XL1'00'	* INITIALIZATION CALL
1029	0100		102A	4311	GRASSZ	DC XL2'0100'	SECTOR SIZE
102B	0001		102C	4312	GRANPB	DC XL2'01'	DISP TO NEXT PHYS BFR DADDR
			0002	4313	GRAEDB	EQU 2	DB DADDR ADJUSTMENT FACTOR
102D			102D	4314	GRASIZ	DS CL1	BUFFER SPACE COUNTER
102E	1A07		102F	4315	GRATXT	DC AL2(GRTEXT)	ADDRESS OF TEXT OUTPUT AREA
1030	0007		1031	4316	GRAPSG	DC XL2'07'	SIZE OF PRIMARY SEG. HEADER
1032	0004		1033	4317	GRASSG	DC XL2'04'	SIZE OF 2NDARY SEG. HEADER
			102C	4318	GRAONE	EQU GRANPB	DECR FACTOR FOR REPITITION CTR
			102C	4319	GRABOA	EQU GRANPB	INCR FACTOR FOR NEXT TEXT CHAR
			102C	4320	GRANXC	EQU GRANPB	CYL ADJ FACTOR
1034			1034	4321	GRASEG	DS CL1	SEGMENT TEXT COUNTER
			0000	4322	GRAEFI	EQU X'00'	INITIALIZATION FUNC. CODE
			0003	4323	GRAEFW	EQU X'03'	WRITE BACK ONLY FUNC. CODE
			0001	4324	GRAEFR	EQU X'01'	RETURN TEXT FUNC. CODE
			0002	4325	GRAEFS	EQU X'02'	SKIP STATEMENT FUNC. CODE
			0004	4326	GRAEFG	EQU X'04'	SKIP SEGMENT FUNC. CODE
			00FF	4327	GRAEBS	EQU X'FF'	BUFFER TEXT AREA SIZE
			0001	4328	GRAESC	EQU X'01'	SCTR COUNT IF DL4ICS USED
			0000	4329	GRAELK	EQU X'00'	DISP TO LINK CODE WITHIN DB
			0000	4330	GRAELN	EQU X'00'	LINK CODE TO NEXT PHYS DB
			0001	4331	GRAEXA	EQU X'01'	ADJ TO '@' EQU'S FOR @XR ADDR
			0006	4332	GRAEDL	EQU @SBLN+GRAEXA	DISP TO STMT BINARY LINE NO.
			0007	4333	GRAEDT	EQU @STYPE+GRAEXA	DISP TO STMT TYPE CODE
			0002	4334	GRAELL	EQU X'02'	LENGTH OF BINARY LINE NUMBER
			0075	4335	GRAEET	EQU @EOFTC	TYPE CODE OF END-OF-FILE STMT
			0001	4336	GRAES0	EQU @SDF0+GRAEXA	DISP TO SDF0 - NULL INDR
			0002	4337	GRAES1	EQU @SDF1+GRAEXA	DISP TO SDF1 - LENGTH
			0003	4338	GRAES2	EQU @SDF2+GRAEXA	DISP TO SDF2 - SEGMENTATION CDE
			0002	4339	GRAETP	EQU X'02'	MASK FOR A PRIMARY SEGMENT
			0007	4340	GRAELP	EQU X'07'	LENGTH OF PRIMARY SEG.
			0004	4341	GRAELS	EQU X'04'	LENGTH OF SECONDARY SEG.
			001B	4342	GRAEMR	EQU 27	MAX. REPITITION CODE
			0001	4343	GRAENC	EQU X'01'	DISP TO NEXT TEXT CHARACTER
			0001	4344	GRAEDC	EQU X'01'	DISP TO CYL IN DADDR
			0F81	4345	GRABSE	EQU GRA310	BASE ADDRESS OF GRABIT
			0005	4346	GRAEDS	EQU X'05'	LNG OF DPL DADDR, SCTR-CT.
			0006	4347	GRAEW2	EQU 6	SECOND CYL OF WORK FILE
			4348	*			

GRABIT - RETRIEVE FILE STATEMENTS

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 42		
				4349	*	ERROR ROUTINE			
				4350	*				
1035	3C	98 03CD		4351	GRAERR MVI	\$CAERR,@E551 SET BAD FILE ERROR CODE			
				4352	*	THE ABOVE ERROR CODE IS INITIALLY SET FOR A SAVED FILE,			
				4353	*	BUT IS MODIFIED TO THE WORK FILE IF DL4ICS IS USED			
1039	3A	04 03D6		4354	SBN	\$INDR3,\$ERHRD SET INDR FOR HARD ERROR			
103D	C0	87 0469		4355	B	\$CAERK GO TO ERRPGM INTERFACE			
				4356	*				
				4357	*	GCPACK			

GCPACK - BASIC STATEMENT CHARACTER PACKING ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 43

```

4359 *
4360 *      GCPBFR MUST BE EQUATED TO THE FIRST BYTE OF THE SDF
4361 *      PRECEEDING THE BASIC STATEMENT IN THE USED DEFINED AREA
4362 *
0001 4363      DROP 1      NO BASE REGISTER USED IN RTN
1041 4364 GCPACK EQU *      ENTRY TO GCPACK ROUTINE
4365 *
4366 ***      SAVE REGISTERS AND SET UP POINTERS
4367 *
1041 34 08 10AE      4368      ST      GCP140+@OP1,@ARR      SAVE RET ADDR IN RESTORE INSTR
1045 34 02 10AA      4369      ST      GCP130+@OP1,@XR      SAVE @XR IN RESTORE INSTR
1049 34 01 10A6      4370      ST      GCP120+@OP1,@BR      SAVE @RB IN RESTORE INSTR
104D C2 01 1A08      4371      LA      GCPBFR+@STEXT+@B1,@BR      SET POINTER FOR PACKED PORTION
1051 C2 02 1A07      4372      LA      GCPBFR+@STEXT,@XR      SET POINTER FOR UNPACKED PART
4373 *
4374 ***      TEST FOR EOS AND REPEAT CHARACTERS
4375 *
1055 BD 1E 00      4376 GCP020 CLI      @ZERO(,@XR),@EOS      TEST FOR CARR RETURN CHAR
1058 F2 81 3E      4377      JE      GCP110      YES, GO CALC STMT LENGTH
105B AD 00 00 01    4378      CLC      @ZERO(1,@XR),@B1(,@XR)      COMPARE FIRST TWO CHAR'S
105F F2 01 29      4379      JNE      GCP090      NOT EQUAL, GO MOVE 1ST TO PACKD
1062 AD 00 01 02    4380      CLC      @B1(1,@XR),GCPTWO(,@XR)      COMPARE 2ND 3RD CHAR'S
1066 F2 01 22      4381      JNE      GCP090      NOT EQUAL, GO MOVE 1ST TO PACKD
4382 *
4383 ***      DETERMINE LENGTH OF REPEAT COUNT
4384 *
1069 7C 02 00      4385      MVI      @ZERO(,@BR),GCPTWO      SET UP INITIAL REPEAT COUNT
106C E2 02 01      4386 GCP050 LA      @B1(,@XR),@XR      SET UNPACKED POINTER UP 1 CHAR
106F AD 00 01 02    4387      CLC      @B1(1,@XR),GCPTWO(,@XR)      TEST FOR ADDITIONAL REPEATS
1073 F2 01 19      4388      JNE      GCP100      NO, GO INCR POINTERS
4389 *
4390 ***      TEST FOR MAX REPEAT COUNT AND RETURN TO PACKING MORE CHARACTERS
4391 *
1076 7D 1B 00      4392      CLI      @ZERO(,@BR),GCPMAX      IS REPEAT COUNT AT MAX ?
1079 F2 81 09      4393      JE      GCP080      YES, GO INCR POINTERS
107C 4E 00 00 10AF  4394      ALC      @ZERO(1,@BR),GCPONE      NO, ADD ONE TO REPEAT COUNTER
1081 C0 87 106C      4395      B      GCP050      GO TEST FOR MORE REPEAT CHAR'S
1085 D2 01 01      4396 GCP080 LA      @B1(,@BR),@BR      SET POINTER OF PACKED AREA UP 1
1088 E2 02 01      4397      LA      @B1(,@XR),@XR      SET POINTER OF INPUT AREA UP 1
108B 6C 00 00 01    4398 GCP090 MVC      @ZERO(1,@BR),@B1(,@XR)      MOVE CHAR TO PACKED STMT AREA
108F D2 01 01      4399 GCP100 LA      @B1(,@BR),@BR      INCREMENT PACKED AREA POINTER
1092 E2 02 01      4400      LA      @B1(,@XR),@XR      INCREMENT INPUT AREA POINTER
1095 C0 87 1055      4401      B      GCP020      GO BACK TO CHECK NEXT CHARACTER
4402 *
4403 ***      CALCULATE STATEMENT LENGTH AND RETURN TO CALLING PROGRAM
4404 *
1099 34 01 1A01      4405 GCP110 ST      GCPBFR+@SDF1,@BR      SAVE PTR TO CALCULATE LENGTH
109D 0F 01 1A01 10B1 4406      SLC      GCPBFR+@SDF1,GCPSTL(@CADDR)      SUBTRACT STARTING LOCATION
10A3 C2 01 0000      4407 GCP120 LA      *-*,@BR      RELOAD BASE REGISTER
10A7 C2 02 0000      4408 GCP130 LA      *-*,@XR      RELOAD INDEX REGISTER
10AB C0 87 0000      4409 GCP140 B      *-*      RETURN
4411 *
4412 ***      DEFINE CONSTANTS AREA
4413 *
10AF 01      10AF 4414 GCPONE DC      XL1'01'      INCR REPEAT COUNTER FACTOR

```

GCPACK - BASIC STATEMENT CHARACTER PACKING ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	44
	10B0	1A00		10B1	4415	GCPSTL	DC AL2(GCPBFR)				
					4416	*					
					4417	***	EQUATES				
				0002	4418	GCPTWO	EQU 2				
				001B	4419	GCPMAX	EQU 27				
					4420	*	END OF GCPACK				
					4421	*					
					4422	*	GPUTIT				

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 45
		4424		*****	
		4425	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		4426	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		4427	*		*
		4428		*****	
		4429	*	*STATUS	*
		4430	*	VERSION 1 MODIFICATION 0	*
		4431	*		*
		4432	*	*FUNCTION	*
		4433	*	THE FUNCTION OF GPUTIT IS TO PUT STATEMENTS INTO THE WORK FILE.	*
		4434	*	WHEN FIRST CALLED, THE ROUTINE WILL PLACE THE STATEMENT PASSED TO	*
		4435	*	IT IN A CORE BUFFER INTO THE POSITION OF THE FIRST STATEMENT OF	*
		4436	*	THE WORK FILE, AND IF REQUESTED, WILL INITIALIZE THE FILE INDEX	*
		4437	*	TABLE. (THE CALL ROUTINE SETS THE INDR GPUFIT IN GPUIDR OFF IF A	*
		4438	*	FIT IS REQUESTED).	*
		4439	*	EACH STATEMENT PASSED VIA A SUBSEQUENT CALL TO GPUTIT WILL BE	*
		4440	*	PLACED IN THE CORE BUFFERS, FOLLOWING THE PREVIOUS STATEMENT.	*
		4441	*	AS A STATEMENT IS PLACED IN A CORE BUFFER, THE FIT IS ADJUSTED	*
		4442	*	IF IT WAS REQUESTED.	*
		4443	*	WHEN A CORE BUFFER IS FILLED IT IS WRITTEN TO DISK VIA DL4ICS.	*
		4444	*	AND FILE BUILDING WILL CONTINUE IN AN ALTERNATE CORE BUFFER.	*
		4445	*	WHEN A EOF CODE IS FINALLY PASSED TO GPUTIT, IT WILL BE REPLACED	*
		4446	*	BY AN END OF FILE RECORD AND THE LAST BLOCK WILL BE WRITTEN TO	*
		4447	*	DISK	*
		4448	*		*
		4449	*	*ENTRY POINTS	*
		4450	*	GPUTIT - THE FIRST LOCATION IN THE PROGRAM. THE CALL IS:	*
		4451	*	B GPUTIT	*
		4452	*		*
		4453	*	*INPUT	*
		4454	*	INPUT TO GPUTIT IS THE STATEMENT TO BE PROCESSED AND PUT TO THE	*
		4455	*	WORK FILE. IT IS PASSED IN A COMMON ANEA, GPUSMT. THE FORMAT OF	*
		4456	*	GPUSMT IS AS FOLLOWS:	*
		4457	*	4 BYTE SDF - FILLED IN BY GPUTIT	*
		4458	*	2 BYTE BINARY LINE NUMBER - SUPPLIED BY USER	*
		4459	*	1 BYTE TYPE CODE - SUPPLIED BY USER	*
		4460	*	244 BYTE TEXT ARE - SUPPLIED BY USER	*
		4461	*	PRIOR TO INITIAL ENTRY, THE FOLLOWING FIELDS MUST BE SET FOR	*
		4462	*	GPUTIT:	*
		4463	*	GPUCYL - STARTING CYLINDER OF THE FILE. (1 BYTE)	*
		4464	*	GRUBFR - CADDR (2 BYTES) OF THE LEFT-MOST BYTE OF THE 2 SECTOR	*
		4465	*	BUFFER AREA ASSIGNED BY USER.	*
		4466	*	GPUFIT - '0' FIT WILL BE BUILT IN CORE	*
		4467	*	'1' FIT WILL NOT BE BUILT	*
		4468	*		*
		4469	*	*OUTPUT	*
		4470	*	OUTPUT FROM GPUTIT WILL BE THE WORK FILE DISK BLOCKS WRITTEN TO	*
		4471	*	DISK AND A FIT BUILT IN CORE IF REQUESTED.	*
		4472	*		*
		4473	*	*EXTERNAL REFERENCES	*
		4474	*	DL4ICS - FOUR TRACK LOGICAL DISK IOCS	*
		4475	*	GCPACK - STATEMENT PACK ROUTINE	*
		4476	*	GPUSMT - BUFFER MONK AREA SUPPLIED BY USER	*
		4477	*	GPUERR - ERROR EXIT ROUTINE ADM	*
		4478	*	GRTEND - ADDR IN GRABIT - EOS ADDR	*
		4479	*	\$\$FITS - CORE ADDR FILE INDEX TABLE	*

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 46
		4480	*	\$INDR1 - ADDR IN SYSTEM NUCLEUS-SYSTEM STATUS INDR	*
		4481	*	\$KEYDT - MASK IN SINDR1 - KEYBOARD OR CARD FILE INDR	*
		4482	*	\$CAERR - ADDR IN SYSTEM NUCLEUS-ERROR CODE SAVE AREA	*
		4483	*		*
		4484	*	*EXITS, NORMAL	*
		4485	*	NEXT SEQUENTIAL INSTRUCTION IN CALL ROUTINE. REGISTERS FOR	*
		4486	*	CALL ROUTINE ARE RESTORED AND POINTERS ARE SAVED INTERNALLY.	*
		4487	*		*
		4488	*	*EXITS, ERROR	*
		4489	*	GPUERR - ERROR EXIT ROUTINE IN CALL PROGRAM. THE ONLY ERROR	*
		4490	*	DETECTED BY GPUTIT IS A FULL WORK FILE AND A REQUEST HAS	*
		4491	*	BEEN MADE TO ENTER ANOTHER LINE. AN ERROR CODE WHICH IS	*
		4492	*	EQUATED TO GPUECD BY THE CALL ROUTINE WILL BE PLACED IN	*
		4493	*	\$CAERR.	*
		4494	*		*
		4495	*	*TABLES/WORK AREAS	*
		4496	*	DPL'S, WORK AREAS AND CONSTANTS ARE PLACED BETWEEN THE 2 MAJOR	*
		4497	*	BLOCKS OF CODE IN ORDER TO FACILITATE BASE ADDRESSIBILITY.	*
		4498	*		*
		4499	*	*ATTRIBUTES	*
		4500	*	GPUTIT IS REUSABLE	*
		4501	*		*
		4502	*	*CHARACTER CODE DEPENDENCY	*
		4503	*	CHARACTER CODE DEPENDENCY CLASS - C	*
		4504	*	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA-	*
		4505	*	TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE	*
		4506	*	USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT RE-	*
		4507	*	DEFINITION OF CHARACTER CONSTANTS. BY REASSEMBLY, WILL RESULT IN	*
		4508	*	A CORRECT MODULE FOR THE NEW DEFINITIONS. THE FOLLOWING ARE THE	*
		4509	*	SPECIAL CONSIDERATIONS FOR THIS MODULE:	*
		4510	*	* @EOS - PART OF @SYSEQ	*
		4511	*	* @EOF - PART OF @SYSEQ - DC AS A CONSTANT	*
		4512	*	* @EOFTC - PART OF @SYSEQ - DC AS A CONSTANT	*
		4513	*		*
		4514	*	*NOTES	*
		4515	*	ERROR PROCEDURES	*
		4516	*	UPON DETECTION OF AN ERROR. THE ERROR CODE EQUATED TO GPUECD	*
		4517	*	BY THE CALL ROUTINE IS MOVED TO \$CAERR AND AN EXIT IS MADE TO	*
		4518	*	GPUERR.	*
		4519	*		*
		4520	*	REGISTER USAGE	*
		4521	*	INDEX REGISTER 1 (@BR) IS SAVED AND RESTORED AND USED AS A	*
		4522	*	BASE REGISTER TO ADDRESS CONSTANTS, WORK AREAS ETC, AND CORE.	*
		4523	*	INDEX REGISTER 2 (@XR) IS SAVED AND RESTORED AND USED AS A	*
		4524	*	POINTER TO THE FIRST UNUSED SPACE IN THE CURRENT BUFFER, AND	*
		4525	*	AS AN INDEX IN CREATING THE FIT.	*
		4526	*		*
		4527	*	SAVED/RESTORED AREAS	*
		4528	*	N/A	*
		4529	*		*
		4530	*	MODIFICATION CONSIDERATIONS	*
		4531	*	N/A	*
		4532	*		*
		4533	*	REQUIRED MODULES	*
		4534	*	@SYSEQ - COMMON SYSTEM SOFTWARE EQUATES	*
		4535	*	@FXDEQ - COMMON CORE LOCATIONS WITHIN THE SYSTEM NUCLEUS	*

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	47
		4536	*	@ERMEQ	- ERROR MESSAGE EQUATES				*
		4537	*	@CANEQ	- FIXED ADDRESSES OUTSIDE SYSTEM NUCLEUS				*
		4538	*	GRABIT	- FILE LINE RETRIEVER				*
		4539	*	GCPACK	- PACK CHARACTER ROUTINE				*
		4540	*	DL4ICS	- FOUR TRACK LOGICAL DISK IOCS				*
		4541	*						*
		4542	*	OTHER					*
		4543	*	N/A					*
		4544	*	*****					*

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	48
				4546		*****					
				4547		*					*
				4548		*	GPUTIT MODULE EQUATES				*
				4549		*					*
				4550		*****					
				4551		*					
	0001			4552	GPULN1	EQU	1			LENGTH CODE OF 1	
	0002			4553	GPULN2	EQU	2			LENGTH CODE OF 2	
	0003			4554	GPULN3	EQU	3			LENGTH CODE OF 3	
	0004			4555	GPULN4	EQU	4			LENGTH CODE OF 4	
	000C			4556	GPUL12	EQU	12			LENGTH OF FIRST FIT ENTRY	
				4557		*					
				4558		*					
	0000			4559	GPUDS0	EQU	0			DISPLACEMENT OF 0	
	0001			4560	GPUDS1	EQU	1			DISPLACEMENT OF 1	
	0002			4561	GPUDS2	EQU	2			DISPLACEMENT OF 2	
	0003			4562	GPUDS3	EQU	3			DISPLACEMENT OF 3	
	0004			4563	GPUDS4	EQU	4			DISPLACEMENT OF 4	
	000B			4564	GPUD11	EQU	11			DISPLACEMENT OF 11	
				4565		*					
	00FF			4566	GPUXFF	EQU	X'FF'			CORE BLOCK LENGTH	
				4567		*					
	00BC			4568	GPUXBC	EQU	X'BC'			NUMBER OF FIT ENTRIES TO BE	
				4569		*				* CREATED INTERNALLY	
	00BC			4570	GPU188	EQU	188			MAXIMUM DB COUNT	
				4571		*					
	0008			4572	GPU008	EQU	X'08'			LENGTH OF EOF RECORD	
				4573		*					
	0001			4574	GPUON1	EQU	X'01'			TEST MICH BUFFER TO FILL	
				4575		*					
	0008			4576	GPUX08	EQU	8			MINIMUM CB BYTES	
				4577		*					
	1D00			4578	GPUADR	EQU	X'1D00'			ADDR FIT IN CORE	
	1D0B			4579	GPUFTS	EQU	GPUADR+GPUD11			DISP OF 11 FROM FIT BEGIN	
				4580		*					*
				4581		*****					

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 49
				4583		*****	
				4584	*		*
				4585	*	INITIALIZATION OF MODULE	*
				4586	*		*
				4587		*****	
				4588	*		
				115E 4589		USING GPUDPL,@BR	
				10B2 4590	GPUNIT	EQU *	
10B2	34	01	1155	4591		ST GPU270+@OP1,@BR	SAVE BASE REGISTER
10B6	C2	01	115E	4592		LA GPUDPL,@BR	LOAD BASE REGISTER
10BA	34	02	1151	4593	GPU050	ST GPU260+@OP1,@XR	SAVE INDEX REGISTER
10BE	34	08	115D	4594		ST GPU280+@OP1,@ARR	SAVE RETURN ADDRESS
				4596		*****	
				4597	*		*
				4598	*	THE FIRST TIME IN THE ROUTINE THE BRANCH AROUND THE FIRST	*
				4599	*	PROCESSING IS NO-OP'ED. AFTER THE INITIAL PASS THROUGH,	*
				4600	*	THE BRANCH IS ALTERED TO BYPASS THE INITIALIZATION ROUTINE.	*
				4601	*		*
				4602		*****	
				4603	*		
10C2	F2	80	12	4604	GPU100 JC	GPU200,@NOP	
				4606		*****	
				4607	*		*
				4608	*	PROCESSING OF INITIAL ENTRY TO ROUTINE	*
				4609	*		*
				4610		*****	
				4611	*		
10C5	5C	01	23 05	4612	GPU150 MVC	GPUCLA(@CADDR,@BR),GPUBFR(,@BR)	MOVE DATA BUFFER ADDRESS
				4613	*		* TO CURRENT LINE ADDRESS
10C9	75	02	05	4614		L GPUBFR(,@BR),@XR	LOAD BUFFER ADDRESS
				4615	*		*
10CC	BC	00	00	4616		MVI @ZERO(,@XR),@ZERO	MOVE A ZERO TO FIRST BYTE OF
				4617	*		* FIRST BUFFER
10CF	3C	87	10C3	4618		MVI GPU100+@Q,@UCB	MODIFY BRANCH AROUND INITIAL-
				4619	*		* IZATION ROUTINE
10D3	3C	00	1A03	4620		MVI GPUSMT+@SDF3,@ZERO	INIT FOURTH BYTE OF GPUSMT
				4621	*		*
				4622		*****	

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 50

```

4624 *****
4625 *
4626 *          CREATE FILE RECORD SEGMENTS
4627 *
4628 *****
4629 *
10D7 75 02 23      4630 GPU200 L      GPUCLA(,@BR),@XR      LOAD CURRENT LINE ADDRESS
10DA 3C 00 1A02    4631          MVI      GPUSMT+@SDF2,@ZERO      INIT TYPE CODE
10DE 3D 1C 1A07    4632          CLI      GPUSMT+@STEXT,@EOF      IS THIS THE EOF RECORD ?
10E2 D0 81 40      4633          BE      GPU340(,@BR)      IF EOF, MODIFY STATUS
10E5 38 40 03D4    4634          TBN      $INDR1,$KEYDT      IS THIS A DATA FILE ?
10E9 F2 90 0E      4635          JF      GPU210      NO, PACK DATA
4636 *
4637 *          PROCESS DATA FILE LINE LENGTH
4638 *
10EC 0C 01 1A01 0FD9 4639          MVC      GPUSMT+@SDF1(@CADDR),GRTEND      GET ADDR OF EOS
10F2 1F 01 1A01 26  4640          SLC      GPUSMT+@SDF1(@CADDR),GPUMOV(,@BR)      COMPUTE LENGTH OF STMT
4641 *
10F7 F2 87 04      4642          J      GPU215      BRANCH AROUND PACK
10FA C0 87 1041    4643 GPU210 B      GCPACK      PACK TEXT DATA; COMPUTE LENGTH
4644 *
10FE 7D BC 16      4645 GPU215 CLI      GPUDBS(,@BR),GPU188      IS DATA BLOCK COUNT 188 ?
1101 3C 8A 03CD    4646          MVI      $CAERR,GPUECD      MAX FILE SIZE EXCEEDED
1105 D0 81 2D      4647          BE      GPU300(,@BR)      YES, CHECK SEGMENT LENGTH
1108 5E 01 18 28   4648 GPU220 ALC      GPUCNT(GPULN2,@BR),GPU001(,@BR)      ADD TO LINE COUNT
110C 7D 08 1D      4649          CLI      GPULN1,@BR),GPULN1      MIN 8 BYTES LEFT ?
4650 *
110F D0 82 BE      4651          BL      GPU400(,@BR)      NO, WRITE BLOCK
1112 78 80 19      4652 GPU230 TBN      GPULN1,@BR),GPULN1      IS BREAK INDR ON ?
1115 D0 90 4C      4653          BF      GPU360(,@BR)      NO, PROCESS FIT
1118 7B 80 19      4654          SBF      GPULN1,@BR),GPULN1      TURN OFF BREAK INDR
111B D0 87 A7      4655          B      GPU396(,@BR)      GO MOVE SECOND SEGMENT
111E 3C 80 12E2    4656 GPU240 MVI      GPU502+@Q,@NOP      RESET RE-ENTRY SWITCH
1122 36 02 1A01    4657          A      GPUSMT+@SDF1,@XR      ADD LENGTH OF SEGMENT TO XR
1126 4F 00 1D 1A01 4658          SLC      GPULN1,@BR),GPULN1      SUB LENGTH OF SEG-
4659 *
112B 3C FF 1141    4660          MVI      GPU245+@VQ,GPUXFF      SET Q CODE TO -1
112F 0E 00 1141 1A01 4661          ALC      GPU245+@VQ(1),GPUSMT+@SDF1      ADD SEGMENT LENGTH
1135 1C 01 1144 26  4662          MVC      GPU245+@DOP2(@CADDR),GPULN1      MOVE BASE ADDR
113A 0E 01 1144 1A01 4663          ALC      GPU245+@DOP2(@CADDR),GPUSMT+@SDF1      ADD SEGMENT LENGTH
1140 8C 00 00 0000 4664 GPU245 MVC      @ZERO(@VQ,@XR),*-*      MOVE LINE SEGMENT TO CORE BUFF
1145 78 40 19      4665 GPU247 TBN      GPULN1,@BR),GPULN1      IS EOF INDR ON ?
1148 D0 10 D6      4666          BT      GPU405(,@BR)      YES, CONTINUE PROCESSING

```

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 51
				4668		*****				
				4669		*				*
				4670		*	END OF MODULE PROCESSING			*
				4671		*				*
				4672		*****				
				4673		*				
114B	74	02	23	4674	GPU250	ST	GPUCLA(,@BR) ,@XR			
114E	C2	02	0000	4675	GPU260	LA	*-*,@XR			RESTORE REGS
1152	C2	01	0000	4676	GPU270	LA	*-*,@BR			*
1156	C0	80	0963	4677	GPU275	BC	GPUERR,@NOP			CONDITIONAL ERROR EXIT
115A	C0	87	0000	4678	GPU280	B	*-*			
				4679		*				*
				4680		*****				

GPUTIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 52
			4682	*****		
			4683	*		*
			4684	*	DATA CONSTANTS, BUFFERS, AND WORK AREAS	*
			4685	*		*
			4686	*****		
			4687	*		
			4688	*	DPL FOR WRITING FILE DATA BLOCKS TO DISK	
			4689	*		
115E 02		115E	4690	GPUDPL DC	AL1(@DPUT)	
115F 0503		1160	4691	DC	XL2'0503'	
1161 01		1161	4692	DC	AL1(@B1)	
1162 1800		1163	4693	DC	AL2(GPUBF1)	
		115F	4695	GPUCYL EQU	GPUDPL+@DCYL	CYLINDER
		1160	4696	GPUSCT EQU	GPUDPL+@DSAD	SECTOR
		1163	4697	GPUBFR EQU	GPUDPL+@DBFR2	CORE ADDR
		0005	4698	GPU005 EQU	5	CYLINDER
			4699	*		
		1164	4700	GPUSDF EQU	*	TEMPORARY SDF
1164		1167	4701	DS	CL4	
1164			4702	ORG	GPUSDF	RESET FOR INITIALIZATION
1164 00000000		1167	4703	DC	XL4'00000000'	INITIAL VALUE OF ZERO
			4704	*		
		1168	4705	GPUNUL EQU	*	NULL SDF
1168 00000000		116B	4706	DC	XL4'00000000'	INITIAL VALUE OF ZERO
			4707	*		
116C 000800002710		1171	4708	DC	XL6'000800002710'	
1172 75		1172	4709	DC	AL1(@EOFTC)	
1173 1C		1173	4710	GPURCD DC	AL1(@EOF)	
			4711	*		
1174		1174	4712	GPUDBS DS	CL1	DATA BLOCK COUNT
1174			4713	ORG	GPUDBS	RESET FOR INITIALIZATION
1174 00		1174	4714	DC	XL1'00'	INITIAL VALUE OF ZERO
1175		1176	4715	GPUCNT DS	CL2	LINE COUNTER
1175			4716	ORG	GPUCNT-1	RESET LOCATION COUNTER
1175 0000		1176	4717	DC	XL2'0000'	INITIALIZED TO ZERO
			4718	*		
1177		1177	4719	GPUIDR DS	CL1	BYTE OF INDICATORS
1177			4720	ORG	GPUIDR	RESET LOCATION COUNTER
1177 00		1177	4721	DC	XL1'00'	INITLZ INDICATORS
		0080	4722	GPUBRK EQU	X'80'	BREAK INDICATOR
			4723	*		* 0 - SEGMENT NOT BROKEN
			4724	*		* 1 - SEGMENT WAS BROKEN
		0040	4725	GPUEOF EQU	X'40'	EOF INDICATOR
			4726	*		* 0 - NOT EOF
			4727	*		* 1 - END OF FILE DETECTED
		0020	4728	GPUERD EQU	X'20'	ERROR INDICATOR
			4729	*		* 0 - NO ERROR
			4730	*		* 1 - ERROR WAS DETECTED
		0001	4731	GPUFIT EQU	X'01'	BUILD FIT INDICATOR
			4732	*		* 0 - BUILD FIT IN CORE
			4733	*		* 1 - DO NOT BUILD FIT
			4734	*	TEMPORARY FIT ENTRY	
			4735	*		
1178		1178	4736	GPUDSP DS	CL1	SECTOR DISPLACEMENT
1178			4737	ORG	GPUDSP	RESET FOR INITIALIZATION

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 53
1178	00		1178	4738	DC	XL1'00'			INITIAL VALUE OF ZERO
1179			117A	4739	GPULIN DS	CL2			BINARY LINE NUMBER
1179				4740	ORG	GPULIN-1			RESET LOCATION COUNTER
1179	0000		117A	4741	DC	XL2'0000'			INITIAL VALUE OF ZERO
117B			117B	4742	GPULSTR DS	CL1			UNUSED DB SPACE
117B				4743	ORG	GPULSTR			RESET FOR INITIALIZATION
117B	FF		117B	4744	DC	XL1'FF'			INITIAL VALUE OF 255
				4745	*				
117C	1D0B		117D	4746	GPULUD DC	XL2'1D0B'			VALUE IN FIT FOR FILE UPDATE
117E			117F	4747	GPULUE DS	CL2			FIT 'LAST USED ENTRY'
117E				4748	ORG	GPULUE-1			RESET LOCATION COUNTER
117E	1D0B		117F	4749	DC	XL2'1D0B'			LAST USED ENTRY ADDR OF FIT
				4750	*				
1180			1181	4751	GPUCLA DS	CL2			CURRENT LINE ADDRESS
1180				4752	ORG	GPUCLA-1			RESET LOCATION COUNTER
1180	0000		1181	4753	DC	XL2'0000'			INITIALIZED TO ZERO
				4754	*				
1182			1182	4755	GPUCBL DS	CL1			LENGTH FIELD WORK AREA
				4756	*				
1183	19FF		1184	4757	GPUSTT DC	AL2(GPUSMT-1)			ADDR FOR MODIFYING MOVE
			1184	4758	GPUMOV EQU	GPUSTT			ADDR FOR MOVE OF DATA LINES
				4759	*				
1185	0001		1186	4760	GPU001 DC	XL2'0001'			INCREMENT
1187	0003		1188	4761	GPU003 DC	XL2'0003'			DECREMENT LUE
1189	0004		118A	4762	GPU004 DC	XL2'0004'			INCREMENT FOR SDF
				4763	*				*
			4764	*****					

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 54
				4766		*****		
				4767	*			*
				4768	*	TEST REMAINING CB SIZE AND SET STATUS INDICATORS		*
				4769	*			*
				4770		*****		
				4771	*			
118B	7C	08	24	4772	GPU300	MVI	GPUCBL(,@BR),GPU008	WILL THE LINE SEGMENT AND EOF
118E	4E	00	24 1A01	4773		ALC	GPUCBL(1,@BR),GPUSMT+@SDF1	* BOTH FIT IN CORE BLOCK ?
1193	5D	00	24 1D	4774		CLC	GPUCBL(1,@BR),GPUPSTR(,@BR)	
1197	C0	04	1108	4775		BNH	GPU220	CONTINUE PROCESS
119B	7A	20	19	4776	GPU320	SBN	GPUIR(,@BR),GPUERD	TURN ON ERROR INDICATOR
119E	7A	40	19	4777	GPU340	SBN	GPUIR(,@BR),GPUEOF	TURN ON EOF INDICATOR
11A1	1C	07	1A07 15	4778		MVC	GPUSMT+@STEXT(GPU008),GPURCD(,@BR)	MOVE EOF RECORD
11A6	C0	87	1108	4779		B	GPU220	RETURN TO PROCESSING
11AA	4C	01	1C 1A05	4780	GPU360	MVC	GPULIN(GPULN2,@BR),GPUSMT+@SBLN	MOVE LINE NUMBER FROM
				4781	*			* GPUSMT TO TEMPORARY FIT NTRY
11AF	1D	00	1A01 1D	4782		CLC	GPUSMT+@SDF1,GPUPSTR(1,@BR)	WILL LINE SEGMENT FIT IN
				4783	*			* CURRENT CB ?
11B4	C0	04	111E	4784		BNH	GPU240	YES, ADD TO PRESENT SEGMENT
				4786		*****		
				4787	*			*
				4788	*	COMPLETE OLD SEGMENT AND INITIALIZE NEW SEGMENT		*
				4789	*			*
				4790		*****		
				4791	*			
11B8	4C	00	07 1A01	4792	GPU380	MVC	GPUSDF+@SDF1(1,@BR),GPUSMT+@SDF1	MOVE LINE LENGTH TO THE
				4793	*			* TEMPORARY SDF
11BD	1C	00	1A01 1D	4794		MVC	GPUSMT+@SDF1,GPUPSTR(1,@BR)	MOVE REMAINING SEGMENT LENGTH
				4795	*			* TO LINE LENGTH IN GPUSMT
11C2	5F	00	07 1D	4796		SLC	GPUSDF+@SDF1(1,@BR),GPUPSTR(,@BR)	SUBTRACT CB LENGTH LEFT
				4797	*			* FROM SEGMENT LENGTH TO DET-
				4798	*			* ERMIN LENGTH FOR 2ND SEG.
11C6	7C	00	1D	4799		MVI	GPUPSTR(,@BR),@ZERO	ZERO UNUSED DB SPACE
11C9	36	02	1A01	4800		A	GPUSMT+@SDF1,@XR	ADD SEGMENT LENGTH TO XR
11CD	3C	01	1A02	4801		MVI	GPUSMT+@SDF2,@SIST	SET SEGMENT TYPE INDICATOR
11D1	7C	02	08	4802		MVI	GPUSDF+@SDF2(,@BR),@SLAST	SET SEGMENT TYPE INDICATOR
				4803	*			*
				4804		*****		

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 55
				4806		*****	
				4807	*		*
				4808	*	MODIFY MOVE INSTR FOR MOVING CURRENT SEGMENT TO CB	*
				4809	*		*
				4810		*****	
				4811	*		
11D4	7C	FF	88	4812	MVI	GPU390+@VQ(,@BR),GPUXFF SET 0 CODE TO MINUS 1	
11D7	4E	00	88 1A01	4813	ALC	GPU390+@VQ(1,@BR),GPUSMT+@SDF1 ADD LENGTH TO Q CODE	
11DC	5C	01	8B 26	4814	MVC	GPU390+@DOP2(@CADDR,@BR),GPUSTT(,@BR) MOVE ADDR OF LEFT	
				4815	*	* BYTE -1 OF GPUSTT TO MOVE	
11E0	4E	01	8B 1A01	4816	ALC	GPU390+@DOP2(@CADDR,@BR),GPUSMT+@SDF1 ADD DISP FROM	
				4817	*	* 'GPUSMT' TO MOVE	
11E5	8C	00	00 0000	4818	GPU390 MVC	@ZERO(@VQ,@XR),*-* MOVE LINE SEGMENT TO CB	
				4820		*****	
				4821	*		*
				4822	*	MODIFY MOVE FOR MOVING SEGMENT TO FRONT OF BUFFER	*
				4823	*		*
				4824		*****	
				4825	*		
11EA	5C	01	B9 8B	4826	MVC	GPU398+@DOP2(@CADDR,@BR),GPU390+@DOP2(,@BR) MODIFY MOVE	
11EE	5E	01	B9 07	4827	ALC	GPU398+@DOP2(@CADDR,@BR),GPUSDF+@SDF1(,@BR) OF SECOND	
				4828	*	* SEGMENT TO BUFFER	
11F2	5E	00	07 2C	4829	ALC	GPUSDF+@SDF1(1,@BR),GPU004(,@BR) ADD SDF LENGTH TO SEG	
11F6	5C	01	9F 8B	4830	MVC	GPU395+@OP1(@CADDR,@BR),GPU390+@DOP2(,@BR) MODIFY ADDR	
				4831	*	* WHERE TO MOVE SDF	
11FA	1C	03	0000 09	4832	GPU395 MVC	*-*(GPULN4),GPUSDF+3(,@BR) MOVE SDF TO FRONT OF THE	
				4833	*	* SECONDARY SEGMENT	
11FF	7A	80	19	4834	SBN	GPUIDR(,@BR),GPUBRK TURN ON BREAK INDR	
1202	F2	87	53	4835	J	GPU450	
				4836	*		*
				4837		*****	
1205	76	02	07	4839	GPU396 A	GPUSDF+@SDF1(,@BR),@XR MODIFY FOR MOVE OF SEGMENT	
1208	5F	00	1D 07	4840	SLC	GPUSTR(1,@BR),GPUSDF+@SDF1(,@BR)	
				4841	*		
120C	7C	FF	B6	4842	MVI	GPU398+@Q(,@BR),GPUXFF MODIFY Q CODE FOR MOVE OF	
120F	5E	00	B6 07	4843	ALC	GPU398+@Q(1,@BR),GPUSDF+@SDF1(,@BR) * OF 2ND SEGMENT	
1213	8C	00	00 0000	4844	GPU398 MVC	@ZERO(@Q,@XR),*-* MOVE SECONDARY SEGMENT TO BUFF	
1218	C0	87	1145	4845	B	GPU247 RETURN TO PROCESSING	

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 56
				4847		*****		
				4848	*			*
				4849	*	CREATE NULL ENTRY		*
				4850	*			*
				4851		*****		
				4852	*			
121C	78	40	19	4853	GPU400 TBN	GPUIR(, @BR), GPUEOF	IS EOF INDR ON ?	
121F	F2	90	12	4854	JF	GPU405	NO, CONTINUE	
1222	7D	08	1D	4855	CLI	GPUIR(, @BR), GPUX08	WILL EOF RECORD FIT ?	
1225	F2	02	0C	4856	JNL	GPU405	YES, CONTINUE	
1228	3C	87	12B5	4857	MVI	GPU480+@Q, @UCB	SET RE-ENTRY SWITCH	
122C	3C	80	1273	4858	MVI	GPU457+@Q, @NOP	SET RE-ENTRY SWITCH	
1230	3C	87	12E2	4859	MVI	GPU502+@Q, @UCB	SET RE-ENTRY SWITCH	
1234	7D	04	1D	4860	GPU405 CLI	GPUIR(, @BR), GPULN4	ARE THERE 4 BYTES IN CB ?	
1237	D0	82	E0	4861	BL	GPU410(, @BR)	NO, LESS THAN 4	
				4863		*****		
				4864	*			*
				4865	*	FILL CB WITH ENTIRE NULL SDF RECORD		*
				4866	*			*
				4867		*****		
				4868	*			
123A	9C	03	04 0D	4869	MVC	GPUDS4(GPULN4, @XR), GPUNUL+@SDF3(, @BR)	MOVE IN NULL SEG	
123E	7D	00	1D	4870	GPU410 CLI	GPUIR(, @BR), @ZERO	IS THERE ANY BYTES IN CB ?	
1241	F2	81	14	4871	JE	GPU450	NO, NO NULL SDF TO MOVE; WRITE	
1244	7D	02	1D	4872	CLI	GPUIR(, @BR), GPULN2	ARE THERE 2 BYTES IN CB ?	
1247	D0	82	F7	4873	BL	GPU430(, @BR)	1 BYTE SDF	
124A	D0	81	F3	4874	BE	GPU420(, @BR)	2 BYTE SDF	
124D	9C	02	03 0D	4875	MVC	GPUDS3(GPULN3, @XR), GPUNUL+@SDF3(, @BR)	MOVE 3 BYTE SDF	
1251	9C	01	02 0C	4876	GPU420 MVC	GPUDS2(GPULN2, @XR), GPUNUL+@SDF2(, @BR)	MOVE 2 BYTE SDF	
1255	BC	80	01	4877	GPU430 MVI	GPUDS1(, @XR), @SNUL	MOVE 1 BYTE SDF	
				4878	*			
				4879	*	WRITE COMPLETED CB TO DISK		
				4880	*			
1258	C0	87	1335	4881	GPU450 B	DL4ICS		
125C	115E			125D 4882	DC	AL2(GPUDPL)		
				4883	*			
125E	78	01	19	4884	TBN	GPUIR(, @BR), GPUIR	FIT BEING BUILT IN CORE ?	
1261	F2	10	16	4885	JT	GPU460	IF NOT, CONTINUE	
				4886	*			*
				4887		*****		

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 57
				4889		*****				
				4890	*					*
				4891	*		FIT IS BUILT IN CORE			*
				4892	*					*
				4893		*****				
				4894	*					
1264	5E	01	21	2C	4895	ALC	GPULUE(@CADDR,@BR),GPU004(,@BR) ADD 4 TO FIT 'LUE'			
1268	1C	01	1270	21	4896	MVC	GPU455+@OP1(GPULN2),GPULUE(,@BR) MODIFY MOVE			
				4897	*		* WITH ADDR FIT 'LUE'			
126D	1C	03	0000	1D	4898	GPU455 MVC	*-(GPULN4),GPUSTR(,@BR) MOVE TEMP FIT ENTRY TO ADDR			
				4899	*		* REFERENCED BY GPULUE			
1272	F2	87	05		4900	GPU457 JC	GPU460,@UCB JUMP WHEN NO RE-ENTRY			
1275	4C	01	1C	1A05	4901	MVC	GPULIN(GPULN2,@BR),GPUSMT+@SBLN SET UP LINE NUMBER			
				4902	*					
127A	5E	00	16	28	4903	GPU460 ALC	GPUDBS(1,@BR),GPU001(,@BR) INCREMENT DB COUNT BY 1			
127E	7C	FF	1D		4904	MVI	GPUSTR(,@BR),GPUXFF INIT GPUSTR TO 255			
1281	5E	00	1A	28	4905	ALC	GPUDSP(1,@BR),GPU001(,@BR) INCREMENT DISPLACEMENT			
1285	5E	00	02	28	4906	ALC	GPUDPL+@DSAD(1,@BR),GPU001(,@BR) INCREMENT DPL SECTOR DIS			
1289	79	01	1A		4907	TBF	GPUDSP(,@BR),GPUON1 IS GPUDSP EVEN ?			
128C	F2	90	07		4908	JF	GPU470 NO, IT IS ODD			
128F	5F	00	04	28	4909	SLC	GPUBFR-1(1,@BR),GPU001(,@BR) DECREMENT GPUBFR BY 256			
1293	F2	87	04		4910	J	GPU475			
1296	5E	00	04	28	4911	GPU470 ALC	GPUBFR-1(1,@BR),GPU001(,@BR) INCREMENT GPUBFR BY 256			
129A	75	02	05		4912	GPU475 L	GPUBFR(,@BR),@XR LOAD XR WITH BUFFER ADDR			
129D	BC	00	00		4913	MVI	@ZERO(,@XR),@ZERO MOVE ZERO TO FIRST BUFFER BYTE			
				4914	*					*
				4915		*****				

GPUNIT - PUT STATEMENTS INTO THE WORK FILE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE	58
				4917			*****				
				4918	*						*
				4919	*		TEXT STATUS INDICATORS				*
				4920	*						*
				4921			*****				
				4922	*						
12A0	78	40	19	4923		TBN	GPUIDR(,@BR),GPUEOF			EOF INDR ON ?	
12A3	C0	90	1112	4924		BF	GPU230			NO, TEST BREAK INDR	
				4925	*						
12A7	78	80	19	4926		TBN	GPUIDR(,@BR),GPUBRK			IS BREAK INDR ON ?	
12AA	C0	10	1112	4927		BT	GPU230			YES, PROCESS SEGMENT	
				4928	*						
12AE	78	01	19	4929		TBN	GPUIDR(,@BR),GPUFIT			IS FIT TO BE BUILT IN CORE ?	
12B1	F2	10	29	4930		JT	GPU500			NO, CHECK ERROR INDR	
12B4	F2	80	26	4931	GPU480	JC	GPU500,@NOP			JUMP FOR RE-ENTRY	
				4933			*****				
				4934	*						*
				4935	*		BUILD FIT IN CORE				*
				4936	*						*
				4937			*****				
				4938	*						
12B7	1C	0B	1D0B 21	4939		MVC	GPUFTS(GPUL12),GPULUE(,@BR)			INIT BYTES OF FIT	
				4940	*						
				4941	*		SET UP DO DISPS FOR REST OP FIT ENTRIES				
				4942	*						
12BC	5F	01	21 2A	4943		SLC	GPULUE(GPULN2,@BR),GPU003(,@BR)			MODIFY LUE FOR MOVE	
12C0	75	02	21	4944		L	GPULUE(,@BR),@XR			LOAD CONTENTS OF LUE	
12C3	7C	BB	24	4945		MVI	GPUCBL(,@BR),GPUBC-1			INITIALIZE COUNTER	
12C6	6F	00	24 00	4946		SLC	GPUCBL(1,@BR),@ZERO(,@XR)			SUBTRACT ENTRY COUNT	
12CA	AC	00	04 00	4947	GPU490	MVC	GPUDS4(1,@XR),@ZERO(,@XR)			MOVE OLD DISP TO NEW	
12CE	9E	00	04 28	4948		ALC	GPUDS4(1,@XR),GPU001(,@BR)			ADD ONE TO NEW DISP	
12D2	E2	02	04	4949		LA	GPUDS4(,@XR),@XR			ADD 4 TO XR	
12D5	5F	00	24 28	4950		SLC	GPUCBL(1,@BR),GPU001(,@BR)			ALL ENTRIES COMPLETED?	
12D9	C0	02	12CA	4951		BNL	GPU490			NO, CREATE NEXT ENTRY	
12DD	3C	80	12B5	4952	GPU500	MVI	GPU480+@Q,@NOP			RESET RE-ENTRY BYPASS	
12E1	C0	80	111E	4953	GPU502	BC	GPU240,@NOP			BRANCH TO PREPARE FOR RE-ENTRY	
12E5	78	20	19	4954		TBN	GPUIDR(,@BR),GPUERD			IS ERROR IND ON	
12E8	C0	90	114E	4955		BF	GPU260			NO, RETURN	
12EC	3C	87	1157	4956		MVI	GPU275+@Q,@UCB			RESET	
12F0	C0	87	114E	4957		B	GPU260			RETURN TO CALLER	
				4958	*						*
				4959			*****	END OF ROUTINE	*****		
				4960	*						
				4961	*		\$CANI				

SCANIT - DELIMETER SCAN MODULE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE 59
4963+				*****			
4964+	*	5703-XM1		COPYRIGHT IBM CORP. 1970			*
4965+	*			REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083			*
4966+	*						*
4967+	*			*****			*
4968+	*			STATUS			*
4969+	*			VERSION 1 MODIFICATION 0			*
4970+	*						*
4971+	*			FUNCTION			*
4972+	*			THE FUNCTION OF SCANIT IS TO SCAN PAST VALID DELIMITERS AND			*
4973+	*			RETURN A POINTER TO THE FIRST CHARACTER THAT'S NOT A DELIMITER.			*
4974+	*						*
4975+	*			ENTRY POINTS			*
4976+	*			* THE ENTRY POINT IS SCANIT.			*
4977+	*			* THE CALLING SEQUENCE IS AS FOLLOWS:			*
4978+	*			B SCANIT			*
4979+	*			WITH REGISTER 2 (@XR) POINTING TO THE FIRST CHARACTER TO BE			*
4980+	*			EXAMINED.			*
4981+	*						*
4982+	*			INPUT			*
4983+	*			NONE			*
4984+	*						*
4985+	*			OUTPUT			*
4986+	*			NONE			*
4987+	*						*
4988+	*			EXTERNAL REFERENCES			*
4989+	*			\$CAERR - ERROR CODE SAVE AREA			*
4990+	*						*
4991+	*			EXITS, NORMAL			*
4992+	*			NORMAL EXIT FROM SCANIT IS TO THE BYTE FOLLOWING THE BRANCH TO			*
4993+	*			SCANIT IN THE CALLING ROUTINE. THE PSR (REGISTER 4) WILL CONTAIN			*
4994+	*			A ZERO IF NO DELIMITERS WERE FOUND OR A HIGH CONDITION IF ONE OR			*
4995+	*			MORE DELIMITERS WERE SCANNED.			*
4996+	*						*
4997+	*			EXITS, ERROR			*
4998+	*			ERROR EXIT FROM SCANIT IS TO THE BYTE FOLLOWING THE BRANCH TO			*
4999+	*			SCANIT IN THE CALLING ROUTINE. THE PSR WILL CONTAIN A LOW			*
5000+	*			CONDITION.			*
5001+	*						*
5002+	*			TABLES/WORKAREAS			*
5003+	*			* SCACNT - AREA CONTAINING NUMBERS OF DELIMITERS SCANNED			*
5004+	*			* SCAMMA - LOC WHERE SCACOM MAY BE MOVED IF ONE COMMA IS ALSO			*
5005+	*			TO BE CONSIDERED A DELIMITER. MOVING SCACOF BACK INTO SCAMMA			*
5006+	*			INDICATES THAT ONLY BLANKS SHOULD BE CONSIDERED DELIMITERS.			*
5007+	*						*
5008+	*			ATTRIBUTES			*
5009+	*			RELOCATABLE AND RE-USABLE			*
5010+	*						*
5011+	*			CHARACTER CODE DEPENDENCY			*
5012+	*			THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
5013+	*			INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
5014+	*						*
5015+	*			NOTES			*
5016+	*			ERROR PROCEDURES			*
5017+	*			THE ONLY ERROR CONDITION DETECTED BY SCANIT IS THE CASE WHERE			*
5018+	*			A CARRIAGE-RETURN CODE FOLLOWS A COMMA. UPON RETURN TO THE			*

SCANIT - DELIMETER SCAN MODULE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 30/10/23 PAGE 60

```

5019+*      CALLING ROUTINE, @PSR WILL BE SET TO A LOW CONDITION, THE      *
5020+*      ERROR CODE IS SET IN $CAERR, AND MG WILU BE POINTING TO THE      *
5021+*      CARRIAGE-RETURN CHARACTER.                                       *
5022+*                                           *
5023+*      REGISTER USAGE                                           *
5024+*      REGISTER 2 (@XR) IS USED AS A POINTER ACROSS THE AREA BEING      *
5025+*      SCANNED FOR DELIMITERS.                                           *
5026+*                                           *
5027+*      SAVED/RESTORED AREAS                                           *
5028+*      UPON ENTRY TO SCANIT, REGISTER 8 (@ARR) IS SAVED AND USED AS      *
5029+*      THE RETURN ADDRESS.                                               *
5030+*                                           *
5031+*      MODIFICATION CONSIDERATIONS                                     *
5032+*      NONE                                                             *
5033+*                                           *
5034+*      REQUIRED MODULES                                           *
5035+*      * @SYSEQ - COMMON SYSTEM EQUATES                               *
5036+*      * @FXDEQ - FIXED NUCLEUS ADDRESSES EQUATES                     *
5037+*                                           *
5038+*      OTHER                                                         *
5039+*      SCANIT IS INITIALIZED TO BYPASS BLANKS ONLY. IF SCACOM IS          *
5040+*      MOVED TO SCAMMA, ONE COMMA WILL BE SCANNED ALONG WITH BLANKS.      *
5041+*      THE INSTRUCTION TO DO THIS IS AS FOLLOWS:                       *
5042+*      MVI    SCAMMA,SCACOM                                              *
5043+*                                           *
5044+*      TO DROP THE COMMA FROM ITS DELIMITER STATUS, SCACOF SHOULD BE      *
5045+*      MOVED TO SCAMMA, USING THE FOLLOWING INSTRUCTION:                 *
5046+*      MVI    SCAMMA,SCACOF                                              *
5047+*                                           *
5048+*****

5050+*
5051+*      EQUATES USED IN THIS SUBROUTINE
5052+*
0001 5053+SCAINC EQU    1          TO INCREMENT POINTER
0001 5054+SCACOM EQU    @BNE       SWITCH TO ALLOW SCANNING COMMA
0087 5055+SCACOF EQU    @UCB       SWITCH TO SET OFF THE INDICATON
5056+*      * FOR SCANNING A COMMA
12F4 5057+SCANIT EQU    *          ENTRY POINT TO THIS SUBROUTINE
12F4 34 08 1330 5058+      ST      SCA500+@OP1,@ARR      SAVE RETURN ADDRESS
12F8 34 02 1332 5059+      ST      SCASVE,@XR           SAVE POINTER VALUE
12FC 3C 04 03CD 5060+      MVI     $CAERR,@@E110         SET ERROR CODE
1300 F2 87 03   5061+      J       SCA200               GO TO PROCESS
1303 E2 02 01   5062+SCA100 LA     SCAINC(,@XR),@XR      INCREMENT POINTER TO NEXT CHAR
1306 BD 40 00   5063+SCA200 CLI    0(,@XR),@BLANK       IS THIS CHAR BLANK ?
1309 C0 81 1303 5064+      BE      SCA100               YES, FETCH NEXT ONE
130D BD 6B 00   5065+      CLI     0(,@XR),@COMMA        IS IT A COMMA ?
1310 F2 87 10   5066+SCA250 JC     SCA400,@UCB          UCS TO RETURN -- OR NOP IF
5067+*      * SCAMMA IS ACTIVE AND CHAR
1313 E2 02 01   5068+SCA300 LA     SCAINC(,@XR),@XR      INCREMENT POINTER TO NEXT CHAR
1316 BD 40 00   5069+      CLI     0(,@XR),@BLANK       IS THIS CHAR A BLANK ?
1319 C0 81 1313 5070+      BE      SCA300               YES, FETCH NEXT ONE
131D BD 1F 00   5071+      CLI     0(,@XR),@EOS+1        IS THIS EOS ?
1320 F2 82 0A   5072+      JL      SCA500               IF NOT, SKIP ERROR ROUTINE
1323 34 02 1334 5073+SCA400 ST     SCACNT,@XR           SAVE NEW POINTER VALUE

```

SCANIT - DELIMETER SCAN MODULE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 61
	1327	0F 01 1334	1332		5074+	SLC	SCACNT(2),SCASVE			SET PSR TO EQUAL IF POINTER
					5075+*					* NOT ADVANCED
	132D	C0 87 0000			5076+SCA500	B	*-*			YES, RETURN
				1311	5077+SCAMMA	EQU	SCA250+@Q			TO SET SCAN COMMA INDICATOR
					5078+*					
					5079+*		SAVE AREA			
					5080+*					
				1331	5081+SCASV1	EQU	*			FIRST BYTE OF SCASVE
1331				1332	5082+SCASVE	DS	CL2			ORIGINAL POINTER VALUE SAVE
1333				1334	5083+SCACNT	DS	CL2			SAVE AREA FOR TOTAL CHAR SCAN
					5084+***			END OF SCANIT		***
					5085 *					
					5086 *		\$DL4P			

DL4ICS - FOUR TRACK LOGICAL IOCR

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 62
		5088+		*****	*
		5089+	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		5090+	*	REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083	*
		5091+	*		*
		5092+		*****	*
		5093+	*	STATUS	*
		5094+	*	VERSION 1 MODIFICATION 0	*
		5095+	*		*
		5096+	*	FUNCTION	*
		5097+	*	DL4ICS WILL CONVERT A RELATIVE DISK ADDRESS TO A PHYSICAL	*
		5098+	*	DISK ADDRESS AND CALL \$DISKN TO PERFORM THE SPECIFIED FUNCTION	*
		5099+	*	THE DISK ADDRESS IS A ONE BYTE CYLINDER ADDRESS AND A ONE BYTE	*
		5100+	*	SECTOR DISPLACEMENT RELATIVE TO SECTOR 0 ON A CYLINDER	*
		5101+	*	BOUNDARY	*
		5102+	*	WHEN MORE THAN 1 SECTOR IS PROCESSED, DL4ICS WILL MAKE MULTIPLE	*
		5103+	*	CALLS TO \$DISKN TO CROSS CYLINDER BOUNDARIES IF REQUIRED.	*
		5104+	*	IF 1 SECTOR ONLY IS TO BE PROCESSED, THE USER MAY OVERLAY THE	*
		5105+	*	UNUSED CODE BY ORGING HIS NEXT MODULE AT DL4SPT	*
		5106+	*		*
		5107+	*	ENTRY POINTS	*
		5108+	*	DL4ICS - ENTRY TO PROCESS A 4 SURFACE FILE. THE CALLING	*
		5109+	*	SEQUENCE IS AS FOLLOWS	*
		5110+	*	DSKL4 DPL	*
		5111+	*	WHERE DPL IS THE LABEL OF A SIX BYTE DISK PARAMETER	*
		5112+	*	LIST AS DESCRIBED FOR \$DISKN EXCEPT FOR THE SECTOR	*
		5113+	*	ADDRESS BYTE.	*
		5114+	*		*
		5115+	*	INPUT	*
		5116+	*	INPUT TO DL4ICS IS THE ADDRESS OF THE DPL TO BE PROCESSED.	*
		5117+	*		*
		5118+	*	OUTPUT	*
		5119+	*	N/A	*
		5120+	*		*
		5121+	*	EXTERNAL REFENECES	*
		5122+	*	\$DISKN - ENTRY TO SYSTEM DISK ROUTINE	*
		5123+	*		*
		5124+	*	EXITS, NORMAL	*
		5125+	*	NORMAL RETURN IS TO THE 1ST INSTRUCTION FOLLOWING THE TWO BYTE	*
		5126+	*	ADDRESS POINTING TO THE DPL.	*
		5127+	*		*
		5128+	*	EXITS, ERROR	*
		5129+	*	N/A	*
		5130+	*		*
		5131+	*	TABLES/WORK AREAS	*
		5132+	*	N/A	*
		5133+	*		*
		5134+	*	ATTRIBUTES	*
		5135+	*	RELOCATABLE	*
		5136+	*	REUSABLE	*
		5137+	*		*
		5138+	*	CHARACTER CODE DEPENDENCY	*
		5139+	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		5140+	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		5141+	*		*
		5142+	*	NOTES	*
		5143+	*	ERROR PROCEDURES	*

DL4ICS - FOUR TRACK LOGICAL IOCR

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	30/10/23	PAGE 63	
		5144+	*	N/A				*
		5145+	*					*
		5146+	*	REGISTER USAGE				*
		5147+	*	@BR IS SAVED AND RESTORED ON EXIT, @XR IS NOT USED. @ARR IS				*
		5148+	*	USED TO PROVIDE THE ADDRESS OF THE PARAMETER. THE @ARR IS				*
		5149+	*	INCREMENTED BT TWO AND SAVED AS THE RETURN ADDRESS.				*
		5150+	*					*
		5151+	*	SAVED/RESTORED AREAS				*
		5152+	*	N/A				*
		5153+	*					*
		5154+	*	MODIFICATION CONSIDERATIONS				*
		5155+	*	N/A				*
		5156+	*					*
		5157+	*	REQUIRED MODULES				*
		5158+	*	@SYSEQ - SYSTEM SOFTWARE EQUATES				*
		5159+	*	@FXDEQ - SYSTEM NUCLEUS EQUATES				*
		5160+	*					*
		5161+	*	OTHER				*
		5162+	*	NONE				*
		5163+	*	*****				*

DL4ICS - FOUR TRACK LOGICAL IOCR

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 30/10/23 PAGE 64
				1335	5165+	DL4ICS	EQU *	ENTRY TO DL4ICS
				1339	5166+		USING DL4010,@BR	ESTABLISH BASE REGISTER USAGE
1335	34	01	13A5		5167+		ST DL4900+@OP1,@BR	SAVE BASE REGISTER FOR EXIT
				1339	5168+	DL4010	EQU *	BASE ADDRESSABILITY
1339	C2	01	1339		5169+		LA DL4010,@BR	ESTABLISH BASE
133D	76	08	78		5170+		A DL4C01(,@BR),@ARR	BUMP TO HIGH END OF ADDR
1340	74	08	14		5171+		ST DL4020+@DOP2(,@BR),@ARR	SET UP MOVE INSTRUCTION
1343	76	08	78		5172+		A DL4C01(,@BR),@ARR	BUMP TO RETURN ADDR
1346	74	08	70		5173+		ST DL4920+@OP1(,@BR),@ARR	SAVE RETURN ADDR
					5174+*			
1349	4C	01	1D 0000		5175+	DL4020	MVC DL4030+@DOP2(@DADDR,@BR),*-*	MOVE DPL ADDR INTO MOVE
134E	5E	01	1D 7A		5176+		ALC DL4030+@DOP2(@CADDR,@BR),DL4C05(,@BR)	BUMP TO RIGHT END
1352	4C	05	76 0000		5177+	DL4030	MVC DL4DPL(@DPLNG,@BR),*-*	MOVE USER DPL TO WORK AREA
					5178+*			
1357	7C	00	5E		5179+	DL4035	MVI DL4100+@Q(,@BR),@ZERO	CLEAR TRACK, DISK SET INST
135A	7C	80	67		5180+		MVI DL4200+@Q(,@BR),@NOP	TURN OFF TWICE INDICATOR
					5181+*			
135D	7D	60	73		5182+	DL4040	CLI DL4SCD(,@BR),DL4E96	TEST IF DISPLACEMENT OVER 95 ?
1360	F2	82	0B		5183+		JL DL4050	JUMP IF NOT OVER 95
1363	5E	00	72 78		5184+		ALC DL4CYL(1,@BR),DL4C01(,@BR)	INCREMENT CYLINDER COUNT
1367	5F	00	73 25		5185+		SLC DL4SCD(1,@BR),DL4C96(,@BR)	DECREMENT DISP BY 96
136B	D0	87	24		5186+		B DL4040(,@BR)	GO BACK CHECK FOR NEXT CYLINDER
					5187+*			
136E	7D	30	73		5188+	DL4050	CLI DL4SCD(,@BR),DL4E48	TEST IF DISP ON NEXT DISK ?
1371	F2	82	07		5189+		JL DL4060	JUMP IF NOT OVER 48
1374	7A	01	5E		5190+		SBN DL4100+@Q(,@BR),DL4EFD	TURN ON BIT FOR FIXED DISK
1377	5F	00	73 36		5191+		SLC DL4SCD(1,@BR),DL4C48(,@BR)	DECREMENT DISP 1 DISK
137B	7D	01	74		5192+	DL4060	CLI DL4SCT(,@BR),DL4E01	IS SECTOR COUNT GREATER THEN 1 ?
137E	F2	84	33		5193+		JH DL4SPT	GO TO SPLIT CALL
1381	7D	18	73		5194+	DL4070	CLI DL4SCD(,@BR),DL4E24	DISPLACEMENT OVER 23 ?
1384	F2	82	07		5195+		JL DL4080	JUMP NOT OVER 24
1387	7A	80	5E		5196+		SBN DL4100+@Q(,@BR),DL4ETB	SET TRACK BIT ON
138A	5F	00	73 49		5197+		SLC DL4SCD(1,@BR),DL4C24(,@BR)	DECR DISP TO NEXT TRACK
138E	5E	00	73 73		5198+	DL4080	ALC DL4SCD(1,@BR),DL4SCD(,@BR)	SHIFT LEFT 1 PLACE
1392	5E	00	73 73		5199+		ALC DL4SCD(1,@BR),DL4SCD(,@BR)	SHIFT LEFT 1 PLACE
1396	7A	00	73		5200+	DL4100	SBN DL4SCD(,@BR),*-*	SET TRACK, DISK BIT
					5201+*			
1399	C0	87	0025		5202+		B \$DISKN	GO PERFORM DISK I/O
139D	13AA			139E	5203+		DC AL2(DL4LST)	ADDR OF DISK PARAM LIST
					5204+*			
139F	F2	00	3C		5205+	DL4200	JC DL4600,*-*	BRANCH OR NOP IF TWICE SET
					5206+*			
13A2	C2	01	0000		5207+	DL4900	LA *-*,@BR	RESTORE OLD BASE TO RETURN
13A6	C0	87	0000		5208+	DL4920	B *-*	RETURN TO CALLER
				13AA	5210+	DL4LST	EQU *	LEFT END OF DPL
13AA				13AF	5211+	DL4DPL	DS CL(@DPLNG)	DPL SAVE AREA
				13AB	5212+	DL4CYL	EQU DL4LST+@DCYL	CYLINDER COUNT BYTE
				13AC	5213+	DL4SCD	EQU DL4LST+@DSAD	DISPLACEMENT SECTOR COUNT
				0060	5214+	DL4E96	EQU 96	TWO DISK SECTOR COUNT PER CYL
				0030	5215+	DL4E48	EQU 48	ONE DISK SECTOR COUNT PER CYL
				0018	5216+	DL4E24	EQU 24	TRACK SECTOR COUNT
				0001	5217+	DL4E01	EQU 01	VALUE TO TEST SECTOR COUNT
				0001	5218+	DL4EFD	EQU 01	VALUE TO SET FIXED DISK BIT
				0080	5219+	DL4ETB	EQU X'80'	VALUE TO SET TRACK BIT
13B0	0001			13B1	5220+	DL4C01	DC IL2'1'	VALUE TO INCR TO CYLINDER

DL4ICS - FOUR TRACK LOGICAL IOCR

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	30/10/23	PAGE 65
13B2	0005			13B3	5221+DL4C05	DC	IL2'5'			
				135E	5222+DL4C96	EQU	DL4040+@Q			DISP TO RIGHT END OF DPL
				1382	5223+DL4C24	EQU	DL4070+@Q			VALUE TO DECR DISPLACEMENT
				13AD	5224+DL4SCT	EQU	DL4LST+@DCNT			VALUE OF 1 TRACK
				136F	5225+DL4C48	EQU	DL4050+@Q			POINTER TO DPL SECTOR COUNT
										VALUE TO DECR DISP BY 1 DISK
13B4	5C	00	14	74	5227+DL4500	MVC	DL4WRK(1,@BR),DL4SCT(,@BR)			PICKUP SECTOR COUNT
				13B4	5228+DL4SPT	EQU	DL4500			POSSIBLE OVERLAY REFERENCE
13B8	5E	00	14	73	5229+	ALC	DL4WRK(1,@BR),DL4SCD(,@BR)			BUMP BY DISPLACEMENT
13BC	7D	30	14		5230+	CLI	DL4WRK(,@BR),DL4E48			TEST FOR CYLINDER OVERLAP
13BF	D0	04	48		5231+	BNH	DL4070(,@BR)			BRANCH BACK IF NO OVERLAY
13C2	5F	00	14	36	5232+	SLC	DL4WRK(1,@BR),DL4C48(,@BR)			DECREMENT WORK BY 48
13C6	5F	00	74	14	5233+	SLC	DL4SCT(1,@BR),DL4WRK(,@BR)			SUBTRACT WORK FROM COUNT
13CA	7C	87	67		5234+	MVI	DL4200+@Q(,@BR),@UCB			SET TWICE SWITCH
13CD	5C	00	13	73	5235+	MVC	DL4SAV(1,@BR),DL4SCD(,@BR)			SAVE SECTOR DISP IN WORK AREA
13D1	78	01	5E		5236+	TBN	DL4100+@Q(,@BR),DL4EFD			DISK BIT ON IN Q CODE ?
13D4	D0	90	48		5237+	BF	DL4070(,@BR)			BRANCH NOT ON
13D7	5E	00	13	36	5238+	ALC	DL4SAV(1,@BR),DL4C48(,@BR)			BUMP TO NEXT DISK
13DB	D0	87	48		5239+	B	DL4070(,@BR)			RETURN TO CALL I/O
					5240+*					
13DE	5C	00	73	13	5241+DL4600	MVC	DL4SCD(1,@BR),DL4SAV(,@BR)			PICKUP NEXT HALF OF I/O
13E2	5E	00	75	74	5242+	ALC	DL4LST+@DBFR1(1,@BR),DL4SCT(,@BR)			BUMP CORE ADDRESS
13E6	5E	00	73	74	5243+	ALC	DL4SCD(1,@BR),DL4SCT(,@BR)			
13EA	5C	00	74	14	5244+	MVC	DL4SCT(1,@BR),DL4WRK(,@BR)			MOVE IN NEW SECTOR COUNT
13EE	D0	87	1E		5245+	B	DL4035(,@BR)			RETURN FOR SECOND PASS
					5246+*					
				134D	5247+DL4WRK	EQU	DL4020+@DOP2			1 BYTE WORK AREA FOR SPLIT CALL
				134C	5248+DL4SAV	EQU	DL4020+@DOP2-1			1 BYTE WORK AREA FOR SPLIT CALL
				13F1	5249+DL4END	EQU	*			DEFINE END OF CODE
					5250+***					***
							END OF DL4ICS			

DL4ICS - FOUR TRACK LOGICAL IOCR

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 30/10/23 PAGE 66
		5252		*****	
		5253	*	PATCH AREA 1	*
		5254		*****	
		5255	*		
		5256	*	CALCULATE AREA LEFT IN THIS SECTOR	
		5257	*		
1400		13F1 5258	\$\$\$\$L1 EQU	*	START OF PATCH AREA 1
		5259	ORG	*,256,0	SET LOC CNTR TO NEXT SECTOR
		1400 5260	\$\$\$\$T1 EQU	*	DEFINE ADDR OF SCTR WRY
13F1		5261	ORG	\$\$\$\$L1	SET LOC CNTR TO START OF
		5262	*		* PATCH AREA
13F1		13FF 5263	\$\$\$\$\$1 DS	CL(\$\$\$\$T1-\$\$\$\$L1)	PATCH AREA
		5264		*****	
		FFFF 5265		END	

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 67

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$\$\$	001	0800	2821	
\$\$\$\$\$1	015	13FF	5263	
\$\$\$\$L1	001	13F1	5258	5261 5263
\$\$\$\$T1	001	1400	5260	5263
\$\$\$CMD	001	0020	0659	
\$\$\$DAT	001	0040	0658	
\$\$\$EPL	001	0091	0655	
\$\$\$ERN	001	0080	0709	
\$\$\$FUN	001	0010	0660	
\$\$\$NLN	001	00A0	0705	3041 3044
\$\$\$STD	001	0081	0654	
\$\$BNLN	001	0605	0635	0637
\$\$CDBS	001	08C0	0685	
\$\$CDND	001	0666	0644	
\$\$CDRD	001	0890	0683	0685
\$\$CKEY	001	0603	0633	
\$\$CKFF	001	0B3D	0665	
\$\$COFF	001	0B44	0664	
\$\$CSNS	001	209C	0694	
\$\$DATB	001	0BBF	0666	
\$\$EOSA	001	0AFE	0663	
\$\$ERSK	001	1C00	0704	2981*
\$\$FITS	001	1D00	0712	
\$\$FLIB	001	06FF	0711	
\$\$ILEN	001	0601	0629	0631 0635
\$\$ILHD	001	0600	0627	0629
\$\$INLN	001	0607	0642	0644 0646
\$\$INND	001	06FA	0646	
\$\$KBDT	001	09E1	0653	0657
\$\$KBSN	001	09E2	0657	0662
\$\$KLD1	001	0600	0717	
\$\$KLD2	001	0700	0719	3003
\$\$KLD3	001	0C00	0721	
\$\$LPOS	001	09EB	0662	
\$\$PCNT	001	07E9	0678	
\$\$PLYN	001	2004	0692	
\$\$PRES	001	0890	0651	0653 0663 0664 0665 0666 0683
\$\$PRFL	001	2143	0696	
\$\$PRNT	001	0707	0672	0673 0677 0678
\$\$PRTN	001	0782	0673	
\$\$PSIO	001	07CE	0677	
\$\$PYCD	001	2200	0698	
\$\$PYMP	001	2000	0690	0692 0694 0696 0698
\$\$SLIB	001	1C00	0707	
\$\$TPCD	001	0606	0637	0642
\$\$UPAR	001	0602	0631	0633
\$\$WSPB	001	1E00	0710	
\$\$XIND	001	06FF	0708	0711
\$\$ZERO	001	0000	0223	0224 0226 0227 0228 0232 0690
\$ABORT	001	0010	0336	
\$BASIC	001	0080	0394	4250
\$BIGCD	001	0080	0470	
\$BLDPL	001	0579	0603	0605
\$BLNOE	001	0569	0593	
\$BLOAD	001	0522	0584	0586 0589 0602 0603
\$BLRTN	001	0550	0592	0593

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 68

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$BRSAB	001	03C5	0281	0282
\$BSADR	001	0587	0608	0610
\$BUFPT	001	03E3	0489	0490
\$CABLD	001	04B4	0562	0563
\$CAERK	001	0469	0539	0542 2973 2982 4355
\$CAERR	001	03CD	0287	0289 2971* 4351* 4646* 5060*
\$CAIPL	001	049D	0558	0560
\$CALLI	001	0008	0479	
\$CARDI	001	0001	0250	
\$CARPL	001	04A1	0560	0562 2975
\$CIENT	001	0483	0549	0550
\$CIEXT	001	0480	0548	0549
\$CIMSK	001	0476	0545	0548 2838*
\$CISUS	001	0496	0553	0558
\$CLBFR	001	0010	0437	
\$CMDKY	001	0008	0349	
\$CMODE	001	0002	0399	
\$CONFG	001	03DD	0462	0472
\$CRPOS	001	03E2	0488	0489
\$CRTAD	001	044D	0527	0528
\$CRTAV	001	0002	0343	
\$CRTDN	001	0002	0367	
\$CRTIN	001	03D3	0364	0371
\$CRTNO	001	0004	0346	
\$CRTPU	001	0004	0368	
\$CRTSP	001	0008	0369	
\$CRTUP	001	0001	0366	
\$CRUSH	001	0080	0475	
\$CSDPL	001	050E	0574	0575
\$C0001	001	0464	0531	0537
\$DATE	001	043A	0512	0513
\$DBGUF	001	03E0	0474	0483
\$DBLOK	001	0001	0424	
\$DFDET	001	03E8	0495	0496
\$DISKN	001	0025	0226	2830 4176 4273 5202
\$DKERR	001	0008	0405	
\$DKSIZ	001	03D7	0449	0457 0498
\$DK100	001	0001	0451	
\$DK200	001	0002	0452	
\$DK400	001	0004	0453	
\$DK600	001	0008	0454	
\$DK800	001	0010	0455	
\$DPLSV	001	0449	0523	0525
\$DTNMB	001	0040	0270	
\$DTRDR	001	0040	0358	
\$ENDNU	001	0600	0617	0627 0651 0672 0708 0717 0719 0721 1819 2811
\$ERDPL	001	046F	0542	0544
\$ERFIL	001	0040	0297	
\$ERHRD	001	0004	0429	4354
\$ERKEY	001	0080	0301	
\$ERLOG	001	0345	0231	
\$ERMAD	001	0472	0544	0545
\$ERPND	001	0004	0402	
\$ERRCT	001	03CF	0303	2980*
\$ERRPG	001	03CE	0291	
\$ERSFL	001	0035	0296	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 69

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$ERSTK	001	0030	0294	3038
\$ER050	001	0363	0232	
\$ER1N2	001	0050	0299	
\$EXADR	001	0517	0577	0579
\$EXCMD	001	0001	0331	
\$EXFTR	001	043B	0513	0518
\$FCIND	001	0010	0409	
\$FDIND	001	0040	0416	
\$FEARR	001	0004	0224	
\$FEMAP	001	0588	0610	0611
\$FILIB	001	03DA	0460	0461
\$FITIN	001	0010	0385	2840
\$FUIND	001	0020	0414	
\$GUFIO	001	0583	0607	0608
\$GUFIR	001	0008	0259	
\$HISTE	001	042E	0510	0511
\$HIST1	001	0435	0511	0512
\$HRDER	001	0020	0355	
\$INDR1	001	03D4	0371	0397 2840* 4250 4634
\$INDR2	001	03D5	0397	0422
\$INDR3	001	03D6	0422	0449 4354*
\$INLNO	001	03CF	0289	0291 0303 0310
\$INRPT	001	0020	0267	
\$IOIND	001	03D2	0338	0364
\$IOPGS	001	0010	0478	
\$IOYES	001	0002	0253	
\$IPLDV	001	05FF	0614	0617
\$IRKEY	001	0020	0477	
\$KEYBD	001	03E1	0483	0488
\$KEYCD	001	03C3	0247	0281
\$KEYDT	001	0040	0391	4634
\$KE090	001	00DE	0227	
\$KE130	001	01D5	0228	
\$KRVLA	001	0807	2824	
\$KYBSY	001	0010	0264	
\$LDRTN	001	0571	0602	
\$LEVEL	001	03DF	0472	0474
\$LIST	001	0002	0426	
\$LMRGN	001	03C1	0242	0244
\$LNPTR	001	0080	0361	
\$LOADB	001	054A	0586	
\$LOADR	001	051A	0579	0582
\$LPRIO	001	03EA	0496	
\$LPROS	001	03E5	0491	0493
\$LPRP3	001	03E4	0490	0491
\$MOUNT	001	0020	0440	
\$MPDWN	001	0001	0340	
\$NEXTB	001	03E6	0493	0494
\$NEXTL	001	03E7	0494	0495
\$NOENB	001	0008	0432	
\$NOLST	001	0004	0256	
\$NUCBS	001	03C0	0239	0240
\$NWRKF	001	0080	0445	
\$NWRKR	001	0040	0442	
\$PASWD	001	042D	0509	0510
\$PAUSD	001	04BA	0563	0565

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 70

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$PAUSE	001	0002	0333	
\$PGMDT	001	0020	0388	
\$PGMST	001	0010	0352	
\$PKERT	001	0419	0507	0509
\$PLST1	001	0454	0528	0529
\$PLST2	001	045B	0529	0530
\$PLST3	001	0462	0530	0531
\$PRDEV	001	044B	0525	0527
\$PRESN	001	0002	0376	
\$PROCI	001	0001	0373	
\$PRPOS	001	03C2	0244	0247
\$PSDBR	001	04FA	0568	
\$PSDXR	001	04F2	0567	0568
\$PSTEP	001	0004	0334	
\$PSTMT	001	0008	0335	
\$PTCH1	001	03F5	0498	0502
\$READY	001	0080	0418	
\$REORD	001	0040	0476	
\$RLOAD	001	051E	0582	0584
\$RMGRN	001	03C0	0240	0242
\$RSTR	001	04D6	0565	0567 0569 0574
\$RUNIT	001	0001	0312	
\$SFAID	001	050D	0570	
\$SPRNT	001	0465	0537	0539
\$SRTRN	001	04FE	0569	0570
\$STEPT	001	0002	0313	
\$SWPCR	001	0511	0575	0577
\$TABLN	001	03CB	0284	0287
\$TFLOW	001	0008	0319	
\$TRACE	001	0004	0314	
\$TRALL	001	0010	0320	
\$TROVR	001	054E	0589	0592
\$TRUNK	001	0080	0272	
\$TRVAR	001	0020	0321	
\$UNMSK	001	048D	0550	0553
\$USRDR	001	03DC	0461	0462
\$VMDEF	001	0080	0325	
\$VOLF1	001	03FE	0504	0505
\$VOLF2	001	040E	0506	
\$VOLID	001	03F6	0502	0503 0507
\$VOLR1	001	03F6	0503	0504
\$VOLR2	001	0406	0505	0506
\$WAITF	001	057F	0605	0607 4177 4274
\$WFDEF	001	0040	0519	
\$WFLOK	001	0008	0382	
\$WFNME	001	0443	0518	0523
\$WSIND	001	0004	0379	
\$XIND1	001	03D0	0310	0329
\$XIND2	001	03D1	0329	0338
\$XIND3	001	03D8	0457	0460
\$XPREC	001	0040	0322	
\$XRSAB	001	03C7	0282	0284
\$ZTRAD	001	05A2	0611	
\$12K	001	0004	0466	
\$16CKY	001	0008	0468	
\$16K	001	0002	0465	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 71

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$22IMP	001	0001	0463	
###BL	001	0000	1672	
###CK	001	0000	1800	
###CN	001	0000	1768	
###CO	001	0000	1560	
###CS	001	0000	1620	
###DR	001	0000	1364	
###ER	001	0000	1564	
###FS	001	0000	1660	
###IN	001	0000	1804	
###PW	001	0000	1808	
###RS	001	0000	1640	
###SA	001	0000	1628	
###SS	001	0000	1624	
###VU	001	0600	1584	
###0T	001	0700	1356	
###1T	001	0000	1360	
###BCO	001	0600	1372	
###BOV	001	0800	1644	
###DPR	001	0700	1380	
###DRE	001	0889	1396	
###DSP	001	2800	1416	
###ECM	001	0C00	1676	
###EFK	001	0C00	1696	
###ERR	001	0C00	1668	
###EXM	001	0C00	1556	
###FIL	001	0E00	1636	
###FIS	001	0E00	1632	
###FML	001	0200	1764	
###FMS	001	0200	1604	
###GRA	001	0889	1528	
###GUF	001	0C00	1664	
###INL	001	0600	1744	
###INS	001	0600	1368	
###KAL	001	0C00	1532	
###KCA	001	0C00	1748	
###KCH	001	0C00	1500	
###KCN	001	0C00	1616	
###KCT	001	0C00	1468	
###KDE	001	0C00	1464	
###KDI	001	0D00	1544	
###KDN	001	0C00	1452	
###KDO	001	0E00	1548	
###KED	001	0C00	1388	
###KEN	001	0C00	1392	
###KEX	001	0C00	1412	
###KGO	001	0C00	1384	
###KHE	001	0C00	1568	
###KKE	001	0C00	1796	
###KLI	001	0C00	1472	
###KLL	001	0920	1772	
###KLO	001	0C00	1476	
###KME	001	0D00	1456	
###KMO	001	0C00	1400	
###KNA	001	0C00	1512	
###KOV	001	0E00	1432	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 72

###KPA 001 0C00 1408
###KPO 001 0C00 1496
###KPR 001 0C00 1520
###KRE 001 0C00 1440
###KRL 001 0700 1536
###KRM 001 0C00 1404
###KRN 001 0700 1424
###KRO 001 0D00 1428
###KRS 001 0C00 1752
###KRU 001 0C00 1448
###KRV 001 0800 1540
###KSA 001 0C00 1484
###KSE 001 0E00 1524
###KSO 001 0C20 1576
###KSS 001 0C00 1508
###KSV 001 0980 1504
###KSY 001 0C00 1516
###KWI 001 0C00 1444
###KWR 001 0C00 1436
###LOA 001 0600 1376
###MIP 001 0C00 1572
###SDS 001 0C00 1684
###SFF 001 0E00 1688
###SFL 001 0F00 1680
###SFO 001 1500 1652
###SFS 001 0C00 1648
###SPA 001 0C00 1488
###SPO 001 0806 1492
###SPS 001 0C00 1480
###STR 001 1600 1656
###TDC 001 1000 1460
###TSY 001 1000 1420
###TVK 001 0FC0 1596
###UAL 001 0C00 1612
###UAT 001 0900 1708
###UCD 001 0900 1716
###UCN 001 0C00 1700
###UCP 001 0700 1704
###UDE 001 0C00 1720
###UDI 001 0C00 1724
###UEX 001 0C00 1608
###UIN 001 0C00 1712
###UPA 001 0C00 1692
###UPO 001 0C00 1760
###UPT 001 0C00 1756
###VCR 001 2000 1552
###VLO 001 0600 1588
###VOD 001 0600 1592
###VVM 001 0000 1600
###VXI 001 0600 1580
###ZDU 001 1100 1732
###ZLB 001 1100 1776
###ZLO 001 1100 1736
###ZLV 001 0F00 1792
###ZL1 001 0F00 1780
###ZL2 001 0F00 1784

2820

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 73

\$\$\$ZL3	001	0C00	1788	
\$\$\$ZTR	001	1000	1728	
\$\$\$ZUT	001	0C00	1740	
\$\$#BLN	001	18D4	1671	
\$\$#CKT	001	2118	1799	
\$\$#CNF	001	2000	1767	
\$\$#COR	001	0800	1559	
\$\$#CSA	001	1000	1619	
\$\$#DRT	001	0000	1363	
\$\$#ERM	001	0928	1563	
\$\$#FSP	001	1880	1659	
\$\$#INV	001	212C	1803	
\$\$#PWR	001	2300	1807	
\$\$#RSP	001	1780	1639	
\$\$#SAV	001	1180	1627	
\$\$#SSA	001	1128	1623	
\$\$#VUF	001	0B08	1583	
\$\$#0TR	001	0000	1355	
\$\$#1TR	001	0080	1359	
\$\$@#BL	001	0001	1673	
\$\$@#CK	001	0004	1801	
\$\$@#CN	001	0001	1769	
\$\$@#CO	001	003A	1561	
\$\$@#CS	001	003A	1621	
\$\$@#DR	001	0008	1365	
\$\$@#ER	001	0032	1565	
\$\$@#FS	001	0030	1661	
\$\$@#IN	001	003A	1805	
\$\$@#PW	001	00C0	1809	
\$\$@#RS	001	0030	1641	
\$\$@#SA	001	0108	1629	
\$\$@#SS	001	0001	1625	
\$\$@#VU	001	0002	1585	
\$\$@#0T	001	0018	1357	
\$\$@#1T	001	0018	1361	
\$\$@BCO	001	0018	1373	
\$\$@BOV	001	0018	1645	
\$\$@DPR	001	0005	1381	
\$\$@DRE	001	0001	1397	
\$\$@DSP	001	0004	1417	
\$\$@ECM	001	0006	1677	
\$\$@EFK	001	0002	1697	
\$\$@ERR	001	0003	1669	
\$\$@EXM	001	0003	1557	
\$\$@FIL	001	0009	1637	
\$\$@FIS	001	0009	1633	
\$\$@FML	001	0052	1765	
\$\$@FMS	001	0052	1605	
\$\$@GRA	001	0003	1529	
\$\$@GUF	001	0010	1665	
\$\$@INL	001	0010	1745	
\$\$@INS	001	0010	1369	
\$\$@KAL	001	000F	1533	
\$\$@KCA	001	000C	1749	
\$\$@KCH	001	000C	1501	
\$\$@KCN	001	0010	1617	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 74

#\$@KCT	001	0009	1469
#\$@KDE	001	0010	1465
#\$@KDI	001	0005	1545
#\$@KDN	001	0010	1453
#\$@KDO	001	000C	1549
#\$@KED	001	000E	1389
#\$@KEN	001	0006	1393
#\$@KEX	001	0003	1413
#\$@KGO	001	0002	1385
#\$@KHE	001	000C	1569
#\$@KKE	001	0006	1797
#\$@KLI	001	0011	1473
#\$@KLL	001	0001	1773
#\$@KLO	001	0008	1477
#\$@KME	001	0003	1457
#\$@KMO	001	0004	1401
#\$@KNA	001	0008	1513
#\$@KOV	001	0009	1433
#\$@KPA	001	0005	1409
#\$@KPO	001	000D	1497
#\$@KPR	001	0009	1521
#\$@KRE	001	0002	1441
#\$@KRL	001	0004	1537
#\$@KRM	001	0003	1405
#\$@KRN	001	0003	1425
#\$@KRO	001	000A	1429
#\$@KRS	001	000A	1753
#\$@KRU	001	0003	1449
#\$@KRV	001	000D	1541
#\$@KSA	001	0011	1485
#\$@KSE	001	0004	1525
#\$@KSO	001	0005	1577
#\$@KSS	001	000B	1509
#\$@KSV	001	0002	1505
#\$@KSY	001	000F	1517
#\$@KWI	001	0002	1445
#\$@KWR	001	0002	1437
#\$@LOA	001	0013	1377
#\$@MIP	001	000D	1573
#\$@SDS	001	0004	1685
#\$@SFF	001	0008	1689
#\$@SFL	001	0005	1681
#\$@SFO	001	0003	1653
#\$@SFS	001	0011	1649
#\$@SPA	001	0004	1489
#\$@SPO	001	0003	1493
#\$@SPS	001	0001	1481
#\$@STR	001	0002	1657
#\$@TDC	001	0003	1461
#\$@TSY	001	0003	1421
#\$@TVK	001	0001	1597
#\$@UAL	001	0011	1613
#\$@UAT	001	000C	1709
#\$@UCD	001	000B	1717
#\$@UCN	001	0009	1701
#\$@UCP	001	000F	1705

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 75

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$@UDE	001	000E	1721	
#\$@UDI	001	0008	1725	
#\$@UEX	001	000E	1609	
#\$@UIN	001	000F	1713	
#\$@UPA	001	0004	1693	
#\$@UPO	001	0005	1761	
#\$@UPT	001	0012	1757	
#\$@VCR	001	0008	1553	
#\$@VLO	001	0002	1589	
#\$@VOD	001	0016	1593	
#\$@VVM	001	0030	1601	
#\$@VXI	001	0002	1581	
#\$@ZDU	001	0008	1733	
#\$@ZLB	001	0002	1777	
#\$@ZLO	001	000C	1737	
#\$@ZLV	001	0006	1793	
#\$@ZL1	001	0007	1781	
#\$@ZL2	001	000D	1785	
#\$@ZL3	001	000A	1789	
#\$@ZTR	001	0001	1729	
#\$@ZUT	001	0014	1741	
#\$BCOM	001	0080	1371	
#\$BOLV	001	1780	1643	
#\$DPRI	001	014C	1379	
#\$DREA	001	0200	1395	
#\$DSPL	001	0240	1415	
#\$ECMA	001	1900	1675	
#\$EFKE	001	1990	1695	
#\$ERRP	001	18C0	1667	
#\$EXMS	001	07D4	1555	
#\$FILN	001	1724	1635	
#\$FIST	001	1700	1631	
#\$FMLN	001	1E00	1763	
#\$FMST	001	0D00	1603	
#\$GRAP	001	0690	1527	
#\$GUFU	001	1880	1663	
#\$INLN	001	1C84	1743	
#\$INST	001	0020	1367	
#\$KALL	001	06A4	1531	
#\$KCAL	001	1CC4	1747	
#\$KCHA	001	053C	1499	
#\$KCND	001	0F80	1615	
#\$KCTL	001	03BC	1467	
#\$KDEL	001	035C	1463	
#\$KDIS	001	0744	1543	
#\$KDNT	001	0300	1451	
#\$KDOV	001	0780	1547	
#\$KEDI	001	0188	1387	
#\$KENA	001	01C4	1391	
#\$KEXT	001	0234	1411	
#\$KGOS	001	0180	1383	
#\$KHEL	001	0A30	1567	
#\$KKEY	001	2100	1795	
#\$KLIS	001	0400	1471	
#\$KLLA	001	2004	1771	
#\$KLOG	001	0444	1475	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 76

#\$KMER	001	030C	1455
#\$KMOU	001	0204	1399
#\$KNAM	001	05C0	1511
#\$KOVN	001	0290	1431
#\$KPAS	001	0220	1407
#\$KPOO	001	0508	1495
#\$KPRT	001	063C	1519
#\$KREA	001	02BC	1439
#\$KRLA	001	0700	1535
#\$KRMO	001	0214	1403
#\$KRNU	001	0280	1423
#\$KROV	001	028C	1427
#\$KRSU	001	1D24	1751
#\$KRUN	001	02CC	1447
#\$KRVL	001	0710	1539
#\$KSAV	001	0488	1483
#\$KSET	001	0680	1523
#\$KSOV	001	0AC8	1575
#\$KSSP	001	0594	1507
#\$KSVL	001	058C	1503
#\$KSYM	001	0600	1515
#\$KWID	001	02C4	1443
#\$KWRI	001	02B4	1435
#\$LOAD	001	0100	1375
#\$MIPP	001	0A80	1571
#\$SDSY	001	192C	1683
#\$SFFI	001	193C	1687
#\$SFLO	001	1918	1679
#\$SFOV	001	1844	1651
#\$SFSY	001	1800	1647
#\$SPAC	001	04CC	1487
#\$SPOV	001	04DC	1491
#\$SPSY	001	0484	1479
#\$STRO	001	1850	1655
#\$TDCK	001	0350	1459
#\$TSYK	001	0250	1419
#\$TVKB	001	0BAC	1595
#\$UALL	001	0F00	1611
#\$UATR	001	1A38	1707
#\$UCDI	001	1AD8	1715
#\$UCNF	001	19B8	1699
#\$UCPL	001	19DC	1703
#\$UDEL	001	1B24	1719
#\$UDIS	001	1B5C	1723
#\$UEXL	001	0EA8	1607
#\$UINI	001	1A88	1711
#\$UPAC	001	1980	1691
#\$UPOV	001	1D24	1759
#\$UPTF	001	1D5C	1755
#\$VCRT	001	07B4	1551
#\$VLOA	001	0B80	1587
#\$VODK	001	0B88	1591
#\$VVMR	001	0C00	1599
#\$VXIT	001	0B00	1579
#\$ZDUM	001	1BA4	1731
#\$ZLBM	001	2008	1775

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 77

SYMBOL	LEN	VALUE	DEFN	REFERENCES
#\$ZLOA	001	1BC4	1735	
#\$ZLVR	001	20B0	1791	
#\$ZL1M	001	2010	1779	
#\$ZL2M	001	2030	1783	
#\$ZL3M	001	2088	1787	
#\$ZTRA	001	1B9C	1727	
#\$ZUTM	001	1C14	1739	
#KRVLA	001	0000	0001	
@@E001	001	0000	1259	1261
@@E003	001	0001	1261	1263
@@E004	001	0002	1263	1265
@@E005	001	0003	1265	1267
@@E006	001	0004	1267	1269
@@E007	001	0005	1269	1271
@@E008	001	0006	1271	1273
@@E009	001	0007	1273	1275
@@E010	001	0008	1275	1277
@@E011	001	0009	1277	1279
@@E012	001	000A	1279	1281
@@E013	001	000B	1281	1283
@@E014	001	000C	1283	1285
@@E015	001	000D	1285	1287
@@E016	001	000E	1287	1289
@@E017	001	000F	1289	1291
@@E018	001	0010	1291	1293
@@E019	001	0011	1293	1295
@@E020	001	0012	1295	1297
@@E021	001	0013	1297	1299
@@E023	001	0014	1299	1301
@@E024	001	0015	1301	1303
@@E025	001	0016	1303	1305
@@E026	001	0017	1305	1307
@@E027	001	0018	1307	1309
@@E028	001	0019	1309	1311
@@E029	001	001A	1311	1313
@@E030	001	001B	1313	1315
@@E031	001	001C	1315	1317
@@E032	001	001D	1317	1319
@@E035	001	001E	1319	1321
@@E036	001	001F	1321	1323
@@E037	001	0020	1323	1325
@@E038	001	0021	1325	1327
@@E039	001	0022	1327	1329
@@E040	001	0023	1329	1331
@@E041	001	0024	1331	1333
@@E042	001	0025	1333	1335
@@E043	001	0026	1335	1337
@@E044	001	0027	1337	1339
@@E045	001	0028	1339	1341
@@E046	001	0029	1341	1343
@@E060	001	002A	1343	1345
@@E080	001	002B	1345	
@@E100	001	0000	0731	0733
@@E101	001	0001	0733	0735
@@E102	001	0002	0735	0737
@@E103	001	0003	0737	0739

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 78

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E110	001	0004	0739	0741 5060
@@E112	001	0005	0741	0743
@@E113	001	0006	0743	0745
@@E114	001	0007	0745	0747
@@E115	001	0008	0747	0749
@@E116	001	0009	0749	0751
@@E117	001	000A	0751	0753
@@E120	001	000B	0753	0755
@@E122	001	000C	0755	0757
@@E123	001	000D	0757	0759
@@E124	001	000E	0759	0761
@@E129	001	000F	0761	0763
@@E130	001	0010	0763	0765
@@E131	001	0011	0765	0767
@@E133	001	0012	0767	0769
@@E134	001	0013	0769	0771
@@E135	001	0014	0771	0773
@@E136	001	0015	0773	0775
@@E137	001	0016	0775	0777
@@E138	001	0017	0777	0779
@@E139	001	0018	0779	0781
@@E142	001	0019	0781	0783
@@E143	001	001A	0783	0785
@@E150	001	001B	0785	0787
@@E151	001	001C	0787	0789
@@E160	001	001D	0789	0791
@@E162	001	001E	0791	0793
@@E163	001	001F	0793	0795
@@E164	001	0020	0795	0797
@@E200	001	0021	0797	0799
@@E205	001	0022	0799	0801
@@E210	001	0023	0801	0803
@@E211	001	0024	0803	0805
@@E212	001	0025	0805	0807
@@E213	001	0026	0807	0809
@@E215	001	0027	0809	0811
@@E216	001	0028	0811	0813
@@E217	001	0029	0813	0815
@@E220	001	002A	0815	0817
@@E221	001	002B	0817	0819
@@E222	001	002C	0819	0821
@@E223	001	002D	0821	0823
@@E225	001	002E	0823	0825
@@E226	001	002F	0825	0827
@@E227	001	0030	0827	0829
@@E228	001	0031	0829	0831
@@E229	001	0032	0831	0833
@@E230	001	0033	0833	0835
@@E232	001	0034	0835	0837
@@E234	001	0035	0837	0839
@@E237	001	0036	0839	0841
@@E240	001	0037	0841	0843
@@E241	001	0038	0843	0845
@@E242	001	0039	0845	0847
@@E248	001	003A	0847	0849
@@E249	001	003B	0849	0851

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 79

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E250	001	003C	0851	0853
@@E251	001	003D	0853	0855
@@E252	001	003E	0855	0857
@@E253	001	003F	0857	0859
@@E254	001	0040	0859	0861
@@E255	001	0041	0861	0863
@@E256	001	0042	0863	0865
@@E300	001	0043	0865	0867
@@E301	001	0044	0867	0869
@@E302	001	0045	0869	0871
@@E303	001	0046	0871	0873
@@E304	001	0047	0873	0875
@@E305	001	0048	0875	0877
@@E308	001	0049	0877	0879
@@E310	001	004A	0879	0881
@@E315	001	004B	0881	0883
@@E316	001	004C	0883	0885
@@E320	001	004D	0885	0887
@@E325	001	004E	0887	0889
@@E330	001	004F	0889	0891
@@E335	001	0050	0891	0893
@@E338	001	0051	0893	0895
@@E340	001	0052	0895	0897
@@E350	001	0053	0897	0899
@@E351	001	0054	0899	0901
@@E352	001	0055	0901	0903
@@E360	001	0056	0903	0905
@@E361	001	0057	0905	0907
@@E362	001	0058	0907	0909
@@E371	001	0059	0909	0911
@@E380	001	005A	0911	0913
@@E390	001	005B	0913	0915
@@E400	001	005C	0915	0917
@@E410	001	005D	0917	0919
@@E415	001	005E	0919	0921
@@E417	001	005F	0921	0923
@@E420	001	0060	0923	0925
@@E430	001	0061	0925	0927
@@E432	001	0062	0927	0929
@@E433	001	0063	0929	0931
@@E450	001	0064	0931	0933
@@E451	001	0065	0933	0935
@@E460	001	0066	0935	0937
@@E461	001	0067	0937	0939
@@E464	001	0068	0939	0941
@@E465	001	0069	0941	0943
@@E466	001	006A	0943	0945
@@E467	001	006B	0945	0947
@@E469	001	006C	0947	0949
@@E470	001	006D	0949	0951
@@E471	001	006E	0951	0953
@@E473	001	006F	0953	0955
@@E474	001	0070	0955	0957
@@E475	001	0071	0957	0959
@@E476	001	0072	0959	0961
@@E477	001	0073	0961	0963

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 80

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E478	001	0074	0963	0965
@@E479	001	0075	0965	0967
@@E480	001	0076	0967	0969
@@E481	001	0077	0969	0971
@@E482	001	0078	0971	0973
@@E483	001	0079	0973	0975
@@E484	001	007A	0975	0977
@@E485	001	007B	0977	0979
@@E486	001	007C	0979	0981
@@E487	001	007D	0981	0983
@@E488	001	007E	0983	0985
@@E489	001	007F	0985	0987
@@E490	001	0080	0987	0989
@@E491	001	0081	0989	0991
@@E492	001	0082	0991	0993
@@E493	001	0083	0993	0995
@@E494	001	0084	0995	0997
@@E495	001	0085	0997	0999
@@E496	001	0086	0999	1001
@@E497	001	0087	1001	1003
@@E498	001	0088	1003	1005
@@E500	001	0089	1005	1007 2971 3040
@@E501	001	008A	1007	1009 3000 3043
@@E530	001	008B	1009	1011
@@E531	001	008C	1011	1013
@@E535	001	008D	1013	1015
@@E540	001	008E	1015	1017
@@E541	001	008F	1017	1019
@@E542	001	0090	1019	1021
@@E543	001	0091	1021	1023
@@E544	001	0092	1023	1025
@@E545	001	0093	1025	1027
@@E546	001	0094	1027	1029
@@E547	001	0095	1029	1031
@@E548	001	FFFF	1235	
@@E549	001	0096	1031	1033
@@E550	001	0097	1033	1035 4178
@@E551	001	0098	1035	1037 4351
@@E552	001	0099	1037	1039
@@E553	001	009A	1039	1041
@@E554	001	009B	1041	1043
@@E555	001	009C	1043	1045
@@E556	001	009D	1045	1047
@@E558	001	009E	1047	1049
@@E570	001	009F	1049	1051
@@E571	001	00A0	1051	1053
@@E572	001	00A1	1053	1055
@@E573	001	00A2	1055	1057
@@E574	001	00A3	1057	1059
@@E575	001	FFFF	1237	
@@E578	001	00A4	1059	1061
@@E579	001	FFFF	1239	
@@E580	001	FFFF	1241	
@@E585	001	00A5	1061	1063
@@E595	001	FFFF	1243	
@@E597	001	FFFF	1245	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 81

@@E598	001	FFFF	1247	
@@E600	001	00A6	1063	1065
@@E601	001	00A7	1065	1067
@@E602	001	00A8	1067	1069
@@E603	001	00A9	1069	1071
@@E604	001	00AA	1071	1073
@@E606	001	00AB	1073	1075
@@E607	001	00AC	1075	1077
@@E608	001	00AD	1077	1079
@@E609	001	00AE	1079	1081
@@E610	001	00AF	1081	1083
@@E611	001	00B0	1083	1085
@@E612	001	00B1	1085	1087
@@E613	001	00B2	1087	1089
@@E614	001	00B3	1089	1091
@@E700	001	00B4	1091	1093
@@E701	001	00B5	1093	1095
@@E710	001	00B6	1095	1097
@@E712	001	00B7	1097	1099
@@E713	001	00B8	1099	1101
@@E714	001	00B9	1101	1103
@@E715	001	00BA	1103	1105
@@E716	001	00BB	1105	1107
@@E717	001	00BC	1107	1109
@@E718	001	00BD	1109	1111
@@E720	001	00BE	1111	1113
@@E721	001	00BF	1113	1115
@@E723	001	00C0	1115	1117
@@E724	001	00C1	1117	1119
@@E725	001	00C2	1119	1121
@@E726	001	00C3	1121	1123
@@E727	001	00C4	1123	1125
@@E728	001	00C5	1125	1127
@@E729	001	00C6	1127	1129
@@E730	001	00C7	1129	1131
@@E732	001	00C8	1131	1133
@@E752	001	00C9	1133	1135
@@E753	001	00CA	1135	1137
@@E754	001	00CB	1137	1139
@@E755	001	00CC	1139	1141
@@E756	001	00CD	1141	1143
@@E757	001	00CE	1143	1145
@@E758	001	00CF	1145	1147
@@E759	001	00D0	1147	1149
@@E760	001	00D1	1149	1151
@@E761	001	00D2	1151	1153
@@E762	001	00D3	1153	1155
@@E763	001	00D4	1155	1157
@@E764	001	00D5	1157	1159
@@E765	001	00D6	1159	1161
@@E766	001	00D7	1161	1163
@@E767	001	00D8	1163	1165
@@E768	001	00D9	1165	1167
@@E769	001	00DA	1167	1169
@@E770	001	00DB	1169	1171
@@E771	001	00DC	1171	1173

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 82

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E772	001	00DD	1173	1175
@@E773	001	00DE	1175	1177
@@E774	001	00DF	1177	1179
@@E775	001	00E0	1179	1181
@@E776	001	00E1	1181	1183
@@E777	001	00E2	1183	1185
@@E778	001	00E3	1185	1187
@@E779	001	00E4	1187	1189
@@E780	001	00E5	1189	1191
@@E781	001	00E6	1191	1193
@@E782	001	00E7	1193	1195
@@E783	001	00E8	1195	1197
@@E784	001	00E9	1197	1199
@@E785	001	00EA	1199	1201
@@E786	001	00EB	1201	1203
@@E790	001	00EC	1203	1205
@@E791	001	00ED	1205	1207
@@E792	001	00EE	1207	1209
@@E793	001	00EF	1209	1211
@@E794	001	00F0	1211	1213
@@E795	001	00F1	1213	1215
@@E796	001	00F2	1215	1217
@@E797	001	00F3	1217	1219
@@E798	001	00F4	1219	1221
@@E800	001	FFFF	1249	
@@E801	001	FFFF	1251	
@@E802	001	FFFF	1253	
@@E803	001	FFFF	1255	
@@E804	001	FFFF	1257	
@@E900	001	00F5	1221	1223
@@E901	001	00F6	1223	1225
@@E902	001	00F7	1225	1227
@@E903	001	00F8	1227	1229
@@E905	001	00F9	1229	1231
@@E906	001	00FA	1231	1233
@@E910	001	00FB	1233	
@ARR	001	0008	0016	3174 3488 3732 3756 3838 4157 4272 4368 4594 5058 5170* 5171 5172* 5173
@ASIGN	001	007C	0071	3231 3528 3808
@ASTER	001	005C	0069	
@BCRDL	001	0050	0088	
@BE	001	0081	0043	
@BF	001	0090	0052	
@BH	001	0084	0041	
@BL	001	0082	0042	
@BLANK	001	0040	0065	2932 2938 3843 5063 5069
@BM	001	0082	0054	
@BNE	001	0001	0046	5054
@BNH	001	0004	0044	
@BNL	001	0002	0045	
@BNM	001	0002	0057	
@BNOL	001	0020	0050	
@BNOZ	001	0008	0049	
@BNP	001	0004	0056	
@BNZ	001	0001	0058	
@BOL	001	00A0	0048	

CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	30/10/23	PAGE	83
@BOZ	001	0088	0047																
@BP	001	0084	0053																
@BR	001	0001	0013	2862*	2864	2867	2867*	2868	2881	2890	2900	2900*	2902	2944	2945				
				2948	2950	2953	3175	3176	3177*	3178	3179	3186	3186	3188	3188				
				3189	3189	3190	3190	3191*	3192*	3193	3194	3194*	3195	3222*	3223				
				3238	3251*	3255	3260	3263	3281*	3285	3291	3292	3297	3298	3299				
				3300	3310	3312	3313	3314	3315	3316	3320	3320	3322	3322	3327				
				3330	3331	3332	3338	3338	3340	3340	3342	3342	3362*	3366	3368				
				3384*	3388	3393	3394	3406*	3410	3416	3431*	3435	3440	3441	3448				
				3456	3457	3459	3460*	3473	3484	3493	3498	3502	3503	3537	3539				
				3546	3546	3547	3548	3554	3565	3565	3566	3578	3586	3598	3598				
				3600	3600	3604	3604	3606	3606	3611	3612	3612	3617	3617	3619				
				3619	3628	3630	3635	3641	3641	3646	3653	3653	3658	3661	3662				
				3663	3670	3672	3673	3682	3688	3689	3698	3699	3699	3700	3741				
				3741	3757	3766	3771	3772	3773	3779	3782	3794	3794	3798	3798				
				3804	3816	3818	3824	3825	3838	3839	3848	3849	3849	4153	4155				
				4156*	4158	4163	4165	4171	4172	4173	4173	4174	4175	4175	4178				
				4179	4179	4182	4183	4184	4184	4191	4193	4194	4200*	4204	4206				
				4209	4210	4211	4219	4225	4228	4229	4230	4231	4237	4238	4241				
				4242	4243	4244	4248	4248	4254	4254	4257	4259	4259	4261	4261				
				4262	4266	4266	4267	4268	4272	4279	4280	4280	4281	4282	4285				
				4286	4287	4287	4290	4370	4371*	4385	4392	4394	4396	4396*	4398				
				4399	4399*	4405	4407*	4589	4591	4592*	4612	4612	4614	4630	4633				
				4640	4645	4647	4648	4648	4649	4651	4652	4653	4654	4655	4658				
				4662	4665	4666	4674	4676*	4772	4773	4774	4774	4776	4777	4778				
				4780	4782	4792	4794	4796	4796	4799	4802	4812	4813	4814	4814				
				4816	4826	4826	4827	4827	4829	4829	4830	4830	4832	4834	4839				
				4840	4840	4842	4843	4843	4853	4855	4860	4861	4869	4870	4872				
				4873	4874	4875	4876	4884	4895	4895	4896	4898	4901	4903	4903				
				4904	4905	4905	4906	4906	4907	4909	4909	4911	4911	4912	4923				
				4926	4929	4939	4943</												

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 84

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@CPLUS	001	004E	0079	2868
@DADDR	001	0002	0140	2834 3015 3019 4173 5175
@DBFR1	001	0004	0129	5242*
@DBFR2	001	0005	0130	4697
@DCALK	001	0001	0081	
@DCBCY	001	0009	0115	2397
@DCBT1	001	0050	0117	2400
@DCNT	001	0003	0128	5224
@DCST1	001	0040	0116	2398
@DCTRL	001	0000	0125	
@DCYL	001	0001	0126	4695 5212
@DD2	001	0003	0030	
@DGET	001	0001	0134	3014 4302
@DOLAR	001	005B	0068	2955 3233 3530 3804
@DOP2	001	0004	0028	4662* 4663* 4814* 4816* 4826 4826* 4827* 4830 5171* 5175* 5176* 5247
				5248
@DPLNG	001	0006	0132	5177 5211
@DPOS	001	0000	0133	
@DPUT	001	0002	0135	4294 4690
@DSAD	001	0002	0127	4696 4906* 5213
@DSBCY	001	0004	0106	2335
@DSCS1	001	0000	0107	2336
@DSIVF	001	0003	0138	
@DSPIN	001	0002	0131	
@DTRSZ	001	0018	0085	
@DVBCY	001	0007	0108	2394
@DVRFY	001	0031	0136	
@DWAIT	001	00FF	0137	
@DWBCY	001	0005	0103	2391
@DWSIZ	001	00C0	0105	
@DWTB1	001	0003	0104	2392
@DZERO	001	00F0	0064	
@D1	001	0002	0026	3635*
@EOF	001	001C	0077	2846 4222 4632 4710
@EOFTC	001	0075	0162	4335 4709
@EOS	001	001E	0076	2407 2853 2962 3787 4376 5071
@FDDBC	001	0000	0195	
@FDE1	001	000C	0200	
@FDFNA	001	000B	0198	
@FDHLN	001	0002	0208	
@FDLNC	001	0002	0193	
@FDNSC	001	0003	0210	
@FDSD	001	0000	0206	
@FLACE	001	0009	0197	
@FLDBC	001	0001	0196	
@FLENT	001	0004	0201	
@FLFNA	001	0002	0199	
@FLHLN	001	0002	0209	
@FLLNC	001	0002	0194	
@FLNSC	001	0001	0211	
@FLSD	001	0001	0207	
@HDRLN	001	0007	0092	0672 3003
@IAR	001	0010	0017	
@INDEX	001	0001	0156	0157
@INST3	001	0003	0032	
@INST4	001	0004	0033	

CROSS REFERENCE																					
SYMBOL	LEN	VALUE	DEFN	REFERENCES													VER 15, MOD 00	30/10/23	PAGE	85	
@INST5	001	0005	0034																		
@INST6	001	0006	0035																		
@I1IAR	001	00C0	0020																		
@LINSZ	001	00F4	0084	0646																	
@MAPEN	001	0005	0089																		
@MINCR	001	2000	0083																		
@MINUS	001	0060	0080																		
@NOP	001	0080	0040	2838	4228	4604	4656	4677	4858	4931	4952	4953	5180								
@NUMBR	001	007B	0070	3229	3526	3806															
@OPD2	001	0004	0029																		
@OP1	001	0003	0027	2885*	2898*	2913	2915*	2916*	2921	2957	2961*	3174*	3175*	3297*	3311*						
				3329*	3455*	3488*	3494*	3625*	3626*	3627*	3660*	3669	3669*	3732*	3756*						
				3838*	4198*	4201	4203	4256	4264	4292	4368*	4369*	4370*	4591*	4593*						
				4594*	4830*	4896*	5058*	5167*	5173*												
@OP2	001	0005	0031	2911*	2915	2961															
@PCTRL	001	0000	0149																		
@PDATA	001	0003	0151																		
@PGCSZ	001	0020	0082	0083																	
@PPLNG	001	0004	0148																		
@PRCNT	001	0001	0150																		
@PRETR	001	00C0	0154																		
@PRINT	001	0040	0152	0154																	
@PSR	001	0004	0015																		
@PWAIT	001	00FF	0158																		
@P1IAR	001	0020	0018																		
@P2IAR	001	0040	0019																		
@Q	001	0001	0024	2914*	2931*	4178*	4225*	4228*	4241*	4247	4618*	4656*	4842*	4843*	4844						
				4857*	4858*	4859*	4952*	4956*	5077	5179*	5180*	5190*	5196*	5222	5223						
				5225	5234*	5236															
@REGL	001	0002	0012	3028	3030																
@RETRN	001	0080	0153	0154																	
@RLDWN	001	004F	0159																		
@RTRNC	001	0080	0161																		
@SBLN	001	0005	0170	2992	4332	4780	4901														
@SBLNL	001	0002	0184																		
@SCTSΖ	001	0100	0100																		
@SDFLN	001	0007	0090	3001																	
@SDF0	001	0000	0166	4336																	
@SDF1	001	0001	0167	4337	4405*	4406*	4639*	4640*	4657	4658	4661	4663	4773	4782	4792						
				4792*	4794*	4796*	4800	4813	4816	4827	4829*	4839	4840	4843							
@SDF2	001	0002	0168	4338	4631*	4801*	4802*	4876													
@SDF3	001	0003	0169	4620*	4869	4875															
@SECCY	001	0030	0086																		
@SIST	001	0001	0181	4801																	
@SLASH	001	0061	0067																		
@SLAST	001	0002	0183	4239	4802																
@SMIDL	001	0003	0182																		
@SNULL	001	0080	0173	4196	4205	4877															
@SONLY	001	0000	0180	4226																	
@STEXT	001	0007																			

CROSS REFERENCE															
S Y M B O L	L E N	V A L U E	D E F N	R E F E R E N C E S								V E R 1 5 , M O D 0 0 3 0 / 1 0 / 2 3 P A G E 8 6			
@VADDR	001	0002	0141	2128	2564	2576	2577	2578	2578	2592	2595	2597	2621	2622	2623
				2661	2664	2667	2670	2673	2676	2679	2688	2691	2694	2697	2700
@VENTA	001	0056	0113	2395	2650										
@VMDDV	001	00FE	0114												
@VMFD1	001	0000	0109												
@VMFD2	001	0001	0110												
@VMRS3	001	0002	0112												
@VMTRL	001	0001	0111												
@VOLID	001	0006	0091												
@VQ	001	0001	0025	2919	2933	4660*	4661*	4664	4812*	4813*	4818				
@WSFIT	001	0500	0101												
@WSTBL	001	0503	0102	4298											
@XR	001	0002	0014	2849*	2853	2864	2873*	2885	2887	2887*	2890	2891*	2898	2930*	2932
				2933	2933	2938	2942*	2944	2948	2955	3207	3209*	3225	3227	3229
				3231	3233	3235	3235*	3301*	3306	3308	3311	3312	3314	3316	3327*
				3329	3333*	3344	3344*	3349*	3455	3458*	3493	3494	3503	3507	3512
				3517	3522	3524	3526	3528	3530	3570	3625	3626	3628*	3629	3629*
				3630	3640*	3647	3658*	3660	3671*	3686	3701	3711*	3719*	3737	3737*
				3761	3767	3769	3777	3779*	3780	3782*	3787	3792	3796	3806	3808
				3810	3812	3814	3824	3839*	3843	3848*	4162*	4171*	4172	4180	4183
				4189	4191	4192	4192*	4196	4198	4199	4199*	4205	4207	4217	4218
				4220	4226	4229	4230	4231	4232	4232*	4237	4239	4242	4243	4244
				4245	4245*	4246	4252	4255	4257	4263	4265	4265*	4279*	4281	4282*
				4283	4286	4369	4372*	4376	4378	4378	4380	4380	4386	4386*	4387
				4387	4397	4397*	4398	4400	4400*	4408*	4593	4614*	4616	4630*	4657*
				4664	4674	4675*	4800*	4818	4839*	4844	4869	4875	4876	4877	4912*
				4913	4944*	4946	4947	4947	4948	4949	4949*	5059	5062	5062*	5063
@ZERO	001	0000	0062	5065	5068	5068*	5069	5071	5073						
				2972	2977	3179	3448	3537	3685	4193	4237	4246*	4255	4376	4378
				4385*	4392	4394*	4398*	4616	4616*	4620	4631	4664*	4799	4818*	4844*
				4870	4913	4913*	4946	4947	5179						
B\$ADMK	001	0001	2032												
B\$ADSW	001	159D	2031												
B\$ARMK	001	0001	2017												
B\$ARSW	001	0A45	2016												
B\$BABF	001	1D00	1822												

CROSS REFERENCE																				
SYMBOL	LEN	VALUE	DEFN	REFERENCES													VER 15, MOD 00	30/10/23	PAGE	87
B\$CDIM	001	0673	1835	1870																
B\$CDUM	001	0000	1871																	
B\$CEND	001	0600	1869																	
B\$CEOF	001	0600	1870																	
B\$CFOR	001	0600	1842																	
B\$CGET	001	06A3	1850																	
B\$CGSB	001	0690	1848																	
B\$CGTO	001	06B3	1846																	
B\$CIFA	001	0600	1844																	
B\$CIFC	001	0600	1845																	
B\$CIMG	001	0600	1859																	
B\$CINP	001	0600	1854																	
B\$CLTA	001	0000	1836																	
B\$CLTC	001	0669	1840																	
B\$CLTM	001	0600	1838																	
B\$CMAT	001	0600	1860																	
B\$CMGT	001	0665	1861																	
B\$CMIN	001	06D3	1862																	
B\$CMPR	001	069B	1865																	
B\$CMPT	001	069B	1864																	
B\$CMPU	001	0600	1866																	
B\$CMRD	001	06D0	1863																	
B\$CNXT	001	0600	1843																	
B\$CPCT	001	0CA8	1925																	
B\$CPRT	001	0600	1857																	
B\$CPRU	001	0600	1858																	
B\$CPSE	001	06E7	1867																	
B\$CPUT	001	0600	1851																	
B\$CPWA	001	0CA6	1996																	
B\$CRAD	001	150D	1966																	
B\$CRBS	001	1509	1968																	
B\$CREA	001	06CF	1855																	
B\$CREM	001	0000	1832																	
B\$CRMK	001	0001	2044																	
B\$CRSR	001	06E3	1856																	
B\$CRST	001	06A6	1852																	
B\$CRSW	001	0E42	2043																	
B\$CRTN	001	06CF	1849																	
B\$CSBF	001	0600	1819	1833	1834	1835	1838	1839	1840	1841	1842	1843	1844	1845	1846					
				1847	1848	1849	1850	1851	1852	1853	1854	1855	1856	1857	1858					
				1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1872					
				1873	1874	1875	1876													
B\$CSCN	001	14B0	1941																	
B\$CSMK	001	0007	2047																	
B\$CSSW	001	14BC	2046																	
B\$CSTP	001	06D6	1868																	
B\$CSTR	001	14CC	1965																	
B\$CSXA	001	2000	1825																	
B\$CTYP	001	0A5F	1919																	
B\$CVPD	001	0C5D	1924																	
B\$CVPG	001	0CA5	1923																	
B\$CWRK	001	F500	1993																	
B\$DIST	001	0700	1885																	
B\$DLNK	001	1B37	1991																	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 88

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$DST2	001	073A	1886	
B\$ERMK	001	0007	2020	
B\$ERSW	001	0993	2019	
B\$FACA	001	0E53	1928	
B\$FAIS	001	15AC	1945	
B\$FAIW	001	15A0	1946	
B\$FCON	001	0A46	1918	
B\$FORT	001	1B0E	1987	
B\$FPWA	001	15AC	1998	
B\$FRMK	001	0007	2038	
B\$FRSW	001	16CC	2037	
B\$FSC1	001	0E4C	1929	
B\$FSC2	001	0E4D	1930	
B\$FSMK	001	0007	2029	
B\$FSSW	001	0E5C	2028	
B\$FSVA	001	0E4F	1931	
B\$FTND	001	1B0B	1989	
B\$FTPT	001	1B0D	1988	
B\$FVME	001	15A2	1950	
B\$FVMP	001	15A4	1951	
B\$FVMS	001	15A6	1952	
B\$FVPE	001	15A8	1947	
B\$FVPP	001	15AA	1948	
B\$FVPS	001	15AC	1949	
B\$GBSW	001	08AF	2022	
B\$GBWK	001	0001	2023	
B\$GETC	001	0867	1899	
B\$GPTR	001	0878	1901	
B\$GTBF	001	1E00	1823	
B\$IFMK	001	0007	2041	
B\$IFSW	001	16E5	2040	
B\$INVT	001	1B38	1981	
B\$KWMK	001	0001	2035	
B\$KWSW	001	159E	2034	
B\$LBAS	001	185E	1972	
B\$LBSV	001	18E7	1970	
B\$LDRP	001	1A00	1820	
B\$LINE	001	07D0	1887	
B\$LIST	001	1853	1954	
B\$LRTN	001	18EB	1971	
B\$LSTR	001	1862	1969	
B\$LTYP	001	18F2	1955	
B\$MATR	001	18F3	1957	
B\$MBMK	001	0007	2056	
B\$MBSW	001	1903	2055	
B\$MFBK	001	1B8F	1983	
B\$MGMK	001	0007	2053	
B\$MGSW	001	18FF	2052	
B\$MPMK	001	0007	2059	
B\$MPSW	001	1981	2058	
B\$MRMK	001	0007	2050	
B\$MRSW	001	0DDE	2049	
B\$NUMC	001	0873	1900	
B\$NXMK	001	0007	2026	
B\$NXSW	001	071D	2025	
B\$PARP	001	0A41	1908	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 89

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$PBNL	001	0A01	1914	
B\$PCAD	001	0A40	1909	
B\$PCDL	001	09D3	1913	
B\$PCPG	001	0A35	1912	
B\$PECT	001	0A44	1916	
B\$PERC	001	0A39	1915	
B\$PFAE	001	0033	1906	
B\$PFCL	001	009D	1907	
B\$PFNC	001	094E	1904	
B\$PFWP	001	0015	1905	
B\$PNBY	001	0A41	1910	
B\$PPWA	001	0A35	1995	
B\$PRM1	001	1AF3	1999	
B\$PTBF	001	1F00	1824	
B\$PUTC	001	093A	1903	
B\$PVAD	001	0A43	1911	
B\$RMRK	001	1AE6	1964	
B\$RTRN	001	1AF5	2000	
B\$SABF	001	1C00	1821	
B\$SCAN	001	1514	1943	
B\$SCAT	001	13C8	1938	
B\$SCON	001	001B	1921	
B\$SCVT	001	12E0	1936	
B\$SDPL	001	07DA	1889	
B\$SFAB	001	0E48	1933	
B\$SFNT	001	143C	1939	
B\$SLDT	001	109C	1935	
B\$SLVT	001	1062	1934	
B\$SNAT	001	131A	1937	
B\$SPAT	001	07E0	1890	
B\$SSTA	001	1BAC	1985	
B\$STAS	001	061B	1874	
B\$STIF	001	0606	1876	
B\$STMA	001	061B	1875	
B\$STML	001	0600	1873	
B\$STRL	001	0600	1872	
B\$SVRB	001	0E46	1932	
B\$SYMB	001	0DBC	1927	
B\$TCD2	001	0001	2005	
B\$TLTH	001	0002	2006	2007
B\$TOD1	001	0000	2004	
B\$TOTB	001	1AF8	2007	
B\$TTAB	001	1AFA	2003	2007
B\$TYPE	001	0739	1888	
B\$WORK	001	15A0	1992	
B\$ZDBN	001	19F2	1959	
B@ABAS	001	0007	2592	
B@ACD1	001	0001	2589	2590
B@ACD2	001	0003	2590	2591
B@AFLG	001	0000	2584	
B@ALLA	001	005C	2409	
B@AMAX	001	0005	2591	2592
B@BLNK	001	0040	2418	3308 3757
B@BLSZ	001	0100	2543	2682 2685 2688 2703 2706
B@BREQ	001	0084	2198	
B@BRHI	001	0088	2199	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 90

B@BRLO	001	0082	2197	
B@BRNE	001	0094	2201	
B@BRNH	001	0098	2202	
B@BRNL	001	0092	2200	
B@CADD	001	0006	2067	
B@CADF	001	0058	2108	
B@CBAS	001	0003	2595	
B@CBNX	001	004A	2101	
B@CBRA	001	0046	2099	
B@CBRC	001	0044	2098	
B@CBRD	001	0048	2100	
B@CBRS	001	004C	2102	
B@CCLS	001	005E	2111	
B@CCMC	001	0042	2097	
B@CCMF	001	0040	2096	
B@CCNT	001	001F	2521	
B@CCSA	001	003E	2095	
B@CDCA	001	006A	2117	
B@CDDL	001	006C	2118	
B@CDIV	001	000C	2070	
B@CDMN	001	0001	2594	2595
B@CDWA	001	006E	2119	
B@CEOF	001	0070	2120	
B@CEOP	001	0068	2116	
B@CFCI	001	0016	2075	
B@CFN0	001	0012	2073	
B@CFN1	001	0014	2074	
B@CFOR	001	004E	2103	
B@CGET	001	0052	2105	
B@CHAR	001	0000	2534	
B@CHLT	001	0004	2066	
B@CIEX	001	00C5	2494	3767
B@CIMH	001	0066	2115	
B@CINI	001	0056	2107	
B@CIPI	001	00D7	2497	3769
B@CIS2	001	00E2	2500	
B@CMF1	001	0018	2076	
B@CMF2	001	001A	2077	
B@CMF3	001	001C	2078	
B@CMA	001	006B	2429	
B@CMPY	001	000A	2069	
B@CMSM	001	001E	2079	
B@CNEG	001	0010	2072	
B@CNXT	001	0050	2104	
B@COLN	001	007A	2431	
B@CPMK	001	00FF	2339	2343 2347 2348 2382
B@CPRS	001	0060	2112	
B@CPRU	001	0062	2113	
B@CPUT	001	0054	2106	
B@CPWR	001	000E	2071	
B@CRSR	001	005A	2109	
B@CRST	001	005C	2110	
B@CSA1	001	0036	2091	
B@CSA2	001	0038	2092	
B@CSB1	001	003A	2093	
B@CSC1	001	002A	2085	

CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 30/10/23 PAGE 91

B@CSD0	001	002E	2087	
B@CSD1	001	0030	2088	
B@CSD2	001	0032	2089	
B@CSF1	001	0022	2081	
B@CSF2	001	0024	2082	
B@CSTA	001	0034	2090	
B@CSTC	001	0028	2084	
B@CSTF	001	0020	2080	
B@CSTH	001	0064	2114	
B@CSTX	001	003C	2094	
B@CSUB	001	0008	2068	
B@CSV	001	0002	2065	
B@CTYP	001	0020	2519	
B@CUSC	001	002C	2086	
B@CUSF	001	0026	2083	
B@CVAR	001	005B	2408	3517
B@DAMK	001	0080	2587	
B@DASA	001	00FF	2348	
B@DASC	001	0040	2352	
B@DASM	001	0038	2350	
B@DCGT	001	0050	2358	
B@DCLS	001	0054	2364	
B@DDAT	001	0024	2344	
B@DDEF	001	0034	2345	
B@DDIM	001	0004	2346	
B@DDUM	001	00FF	2382	
B@DEC0	001	00F0	2477	3507 3816
B@DEC1	001	00F1	2478	
B@DEC2	001	00F2	2479	
B@DEC3	001	00F3	2480	
B@DEC4	001	00F4	2481	
B@DEC5	001	00F5	2482	
B@DEC6	001	00F6	2483	
B@DEC7	001	00F7	2484	
B@DEC8	001	00F8	2485	
B@DEC9	001	00F9	2486	
B@DEND	001	0058	2380	2381
B@DEOF	001	0058	2381	
B@DFOR	001	0028	2353	
B@DGET	001	0040	2361	
B@DGSB	001	0020	2359	
B@DGTO	001	0044	2357	
B@DIFA	001	0048	2355	
B@DIFC	001	004C	2356	
B@DIGS	001	007B	2411	
B@DIMG	001	003C	2370	
B@DINP	001	0000	2365	
B@DIVD	001	0061	2428	
B@DLTA	001	00FF	2347	
B@DLTC	001	0040	2351	
B@DLTM	001	0038	2349	
B@DL01	001	0001	2662	2665
B@DL02	001	0003	2665	2668
B@DL03	001	0005	2668	2671
B@DL04	001	0007	2671	2674
B@DL05	001	0009	2674	2677

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 92

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@DL06	001	000B	2677	2680
B@DL07	001	0045	2680	2683
B@DL08	001	0145	2683	2686
B@DL09	001	0245	2686	2689
B@DL10	001	0289	2689	2692
B@DL11	001	02C3	2692	2695
B@DL12	001	02FD	2695	2698
B@DL13	001	0337	2698	2701
B@DL14	001	0371	2701	2704
B@DL15	001	0471	2704	2707
B@DL16	001	0507	2707	
B@DMAT	001	0008	2371	
B@DMGT	001	0044	2372	
B@DMIN	001	0038	2373	
B@DMPR	001	0048	2376	
B@DMPT	001	004C	2375	
B@DMPU	001	0054	2377	
B@DMRD	001	003C	2374	
B@DNXT	001	0044	2354	
B@DPNT	001	004B	2419	3818
B@DPRT	001	002C	2368	
B@DPRU	001	0030	2369	
B@DPSE	001	0050	2378	
B@DPUT	001	0040	2362	
B@DREA	001	000C	2366	
B@DREM	001	00FF	2343	
B@DRSR	001	005C	2367	
B@DRST	001	0050	2363	
B@DRTN	001	005C	2360	
B@DSCY	001	0004	2335	
B@DSIF	001	001C	2384	
B@DSLT	001	0010	2383	
B@DSML	001	0010	2385	
B@DSNS	001	0018	2337	
B@DSS1	001	0000	2336	
B@DSTP	001	0054	2379	
B@DTBN	001	0010	2401	
B@DTB1	001	0050	2400	
B@DTCY	001	0009	2397	
B@DTSN	001	0010	2399	
B@DTS1	001	0040	2398	
B@DTYP	001	0040	2513	
B@DVCY	001	0007	2394	
B@DVC1	001	0056	2395	
B@DWCY	001	0005	2391	
B@DWT1	001	0003	2392	
B@D1MK	001	0080	2585	
B@D2MK	001	00C0	2586	
B@EOST	001	001E	2407	3207
B@EQUL	001	007E	2433	3306
B@EXPC	001	00C5	2410	3686 3814
B@FOFL	001	005C	2412	
B@FVAD	001	0001	2597	
B@GETC	001	0001	2536	
B@GETE	001	00FF	2537	
B@GETS	001	0000	2535	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 93

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@GRTR	001	006E	2430	
B@ICON	001	0050	2492	3761
B@LADD	001	0001	2136	
B@LADF	001	0002	2177	
B@LADV	001	0008	2621	2642
B@LBIN	001	0002	2546	2547 2553
B@LBNX	001	0003	2170	
B@LBRA	001	0003	2168	
B@LBRC	001	0004	2167	
B@LBRD	001	0003	2169	
B@LBRS	001	0001	2171	
B@LCCA	001	0004	2577	
B@LCCC	001	0001	2129	2167
B@LCDV	001	0004	2622	2643
B@LCER	001	0001	2127	2191
B@LCFN	001	0004	2578	
B@LCLN	001	0002	2132	2183 2184 2191
B@LCLS	001	0001	2180	
B@LCMC	001	0001	2166	
B@LCMF	001	0001	2165	
B@LCNA	001	0006	2576	
B@LCNN	001	0001	2130	2155 2164 2176 2188
B@LCOP	001	0001	2126	2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189
B@LCRV	001	0013	2620	2640
B@LCSA	001	0002	2164	
B@LCVA	001	0002	2128	2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2153 2154 2156 2157 2158 2159 2160 2161 2162 2167 2168 2169 2170 2172 2173 2174 2186 2187
B@LCXX	001	0001	2131	2163 2175 2177 2181 2182
B@LDAT	001	0004	2290	3990
B@LDCA	001	0003	2186	
B@LDDL	001	0003	2187	
B@LDDM	001	0004	2550	
B@LDEF	001	0003	2291	3993
B@LDIM	001	0003	2292	3996
B@LDIN	001	0004	2549	2550 2551
B@LDIV	001	0001	2139	
B@LDMN	001	0002	2547	2576 2577 2589 2590 2591 2594 2621 2622
B@LDSN	001	0004	2551	
B@LDWA	001	0002	2188	
B@LELP	001	0010	2619	
B@LEND	001	0003	2319	4098
B@LEOF	001	0001	2189	
B@LEOP	001	0001	2185	
B@LERC	001	0003	2191	
B@LESP	001	0008	2618	
B@LESS	001	004C	2420	
B@LET\$	001	005B	2440	
B@LET#	001	007B	2441	
B@LET@	001	007C	2442	
B@LETA	001	00C1	2444	
B@LETB	001	00C2	2446	

CROSS REFERENCE																
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00 30/10/23 PAGE 94											
B@LETC	001	00C3	2447													
B@LETD	001	00C4	2448													
B@LETE	001	00C5	2449													
B@LETF	001	00C6	2450													
B@LETG	001	00C7	2451													
B@LETH	001	00C8	2452													
B@LETI	001	00C9	2453													
B@LETJ	001	00D1	2454													
B@LETK	001	00D2	2455													
B@LETL	001	00D3	2456													
B@LETM	001	00D4	2457													
B@LETN	001	00D5	2458													
B@LETO	001	00D6	2459													
B@LETP	001	00D7	2460													
B@LETQ	001	00D8	2461													
B@LETR	001	00D9	2462													
B@LETS	001	00E2	2463													
B@LETT	001	00E3	2464													
B@LETU	001	00E4	2465													
B@LETV	001	00E5	2466													
B@LETW	001	00E6	2467													
B@LETX	001	00E7	2468													
B@LETY	001	00E8	2469													
B@LETZ	001	00E9	2470													
B@LEXP	001	0008	2509													
B@LFCI	001	0003	2144													
B@LFNA	001	0002	2623	2644												
B@LFN0	001	0003	2142													
B@LFN1	001	0003	2143													
B@LFOR	001	0003	2172													
B@LFRT	001	0004	2564	2565												
B@LGET	001	0003	2174													
B@LGSB	001	0005	2298	4035												
B@LGTO	001	0004	2297	4029 4032												
B@LHLT	001	0001	2135													
B@LIEX	001	0002	2495													
B@LIFN	001	0003	2558	3320 3322 3338 3340 3342 3630 3641 3653 3921 3922 3927 3932 3933 3934 3935 3936 3937 3938 3939 3940 3941 3942 3943 3944 3945 3946 3947 3948 3949 3950 3951 3952 3953 3957 3958 3962												
				3963 3964												
B@LILP	001	0009	2617	2635 2636 2637												
B@LIMG	001	0001	2309	4068												
B@LIMH	001	0003	2184													
B@LINI	001	0002	2176													
B@LINP	001	0005	2304	4053												
B@LIPI	001	0003	2498													
B@LISP	001	0005	2616	2624 2630 2631 2632												
B@LIS2	001	0005	2501													
B@LIVT	001	0001	2574													
B@LKCL	001	0005	2303	4050												
B@LKFR	001	0003	2294	4017												
B@LKGT	001	0003	2300	4041												
B@LKIF	001	0002	2296	4023 4026 4119												
B@LKON	001	0002	2329	3263												

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 95

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LKRR	001	0007	2306	4059
B@LKRT	001	0005	2302	4047
B@LKTO	001	0002	2323	
B@LLET	001	0003	2293	3999 4005 4011 4107 4110
B@LL01	001	0002	2661	2662
B@LL02	001	0002	2664	2665
B@LL03	001	0002	2667	2668
B@LL04	001	0002	2670	2671
B@LL05	001	0002	2673	2674
B@LL06	001	0002	2676	2677
B@LL07	001	003A	2679	2680
B@LL08	001	0100	2682	2683
B@LL09	001	0100	2685	2686
B@LL10	001	0044	2688	2689
B@LL11	001	003A	2691	2692
B@LL12	001	003A	2694	2695
B@LL13	001	003A	2697	2698
B@LL14	001	003A	2700	2701
B@LL15	001	0100	2703	2704
B@LL16	001	0096	2706	2707
B@LMAT	001	0003	2310	4071
B@LMF1	001	0003	2145	
B@LMF2	001	0003	2146	
B@LMF3	001	0003	2147	
B@LMGT	001	0006	2311	4074
B@LMIN	001	0008	2312	4077
B@LMPR	001	0008	2315	4086
B@LMPT	001	0006	2314	4083
B@LMPU	001	000D	2316	4089
B@LMPY	001	0001	2138	
B@LMRD	001	0007	2313	4080
B@LMSM	001	0003	2148	
B@LNEG	001	0001	2141	
B@LNEX	001	0004	2295	4020
B@LNXT	001	0003	2173	
B@LPAR	001	004D	2421	3512 3570 3647 3701 3792
B@LPRS	001	0002	2181	
B@LPRT	001	0005	2307	4062
B@LPRU	001	0002	2182	
B@LPSE	001	0005	2317	4092
B@LPUT	001	0002	2175	
B@LPWR	001	0001	2140	
B@LREA	001	0004	2305	4056
B@LREM	001	0003	2289	3987
B@LRSR	001	0001	2178	
B@LRST	001	0001	2179	
B@LRTN	001	0006	2299	4038
B@LSA1	001	0003	2160	
B@LSA2	001	0003	2161	
B@LSB1	001	0003	2162	
B@LSC1	001	0003	2154	
B@LSDF	001	0004	2544	
B@LSD0	001	0003	2156	
B@LSD1	001	0003	2157	
B@LSD2	001	0003	2158	
B@LSF1	001	0003	2150	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 96

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LSF2	001	0003	2151	
B@LSKW	001	0002	2560	
B@LSNO	001	0002	2553	
B@LSPT	001	0003	2568	2571
B@LSTA	001	0003	2159	
B@LSTC	001	0003	2153	
B@LSTE	001	0004	2324	
B@LSTF	001	0003	2149	
B@LSTH	001	0003	2183	
B@LSTP	001	0004	2318	4095
B@LSTX	001	0002	2163	
B@LSUB	001	0001	2137	
B@LSVC	001	0001	2134	
B@LTHN	001	0004	2325	
B@LTYP	001	0001	2554	
B@LUFN	001	0002	2561	3617
B@LUSC	001	0002	2155	
B@LUSF	001	0001	2152	
B@LVPG	001	0100	2648	2651
B@MINS	001	0060	2427	
B@MULT	001	005C	2424	
B@NAAR	001	001D	2612	2642 2694
B@NCAR	001	001D	2613	2643 2697
B@NCRV	001	001D	2611	2640 2691
B@NDGT	001	000A	2604	2610
B@NEQL	001	007F	2434	
B@NFRT	001	000A	2563	2565
B@NICN	001	0006	2606	2608
B@NIEL	001	0007	2608	2624 2630 2635
B@NIFN	001	0018	2557	
B@NIVR	001	0001	2607	2608
B@NIVT	001	0057	2573	
B@NLDV	001	0122	2610	2632 2637 2688
B@NLRV	001	001D	2609	2631 2636 2679
B@NLTR	001	001D	2603	2609 2610 2611 2612 2613 2614
B@NSKW	001	0004	2559	
B@NSPT	001	0028	2567	
B@NUFN	001	001D	2614	2644 2700
B@NVPG	001	0100	2647	2651
B@NXHI	001	00E3	2528	
B@NXLO	001	001E	2527	
B@NXZR	001	0080	2526	2527 2528
B@PLUS	001	004E	2422	
B@POWR	001	005A	2423	
B@PREC	001	0020	2515	
B@PROD	001	0023	2624	
B@PRPL	001	0002	2211	
B@PRPN	001	0001	2210	
B@PRPR	001	0004	2213	
B@PRPS	001	0003	2212	
B@PRRC	001	0007	2216	
B@PRRL	001	0008	2217	
B@PRSL	001	0005	2214	
B@PRSS	001	0006	2215	
B@PTAB	001	0000	2569	
B@PTAD	001	0001	2570	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 97

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@PTSA	001	0002	2571	
B@PUD1	001	0006	2227	
B@PUD2	001	0007	2228	
B@PUI0	001	0001	2221	
B@PUI1	001	0004	2222	
B@PUI2	001	0005	2223	
B@PUNL	001	0002	2225	
B@PUNS	001	0003	2226	
B@PURE	001	0020	2231	
B@PUTM	001	0010	2230	
B@RPAR	001	005D	2425	3796
B@SADV	001	00E8	2642	2645
B@SAVL	001	0B76	2638	2655
B@SAVS	001	065E	2633	2654
B@SCDV	001	0074	2643	2645
B@SCLN	001	005E	2426	
B@SCRV	001	0227	2640	2654 2655
B@SDMK	001	0080	2555	
B@SEXP	001	0004	2508	
B@SFAT	001	0196	2645	2654 2655 2706
B@SFNA	001	003A	2644	2645
B@SFRT	001	0028	2565	
B@SIEL	001	003F	2635	2638
B@SIES	001	0023	2630	2633
B@SIGN	001	0010	2517	
B@SLDL	001	0A32	2637	2638
B@SLDS	001	05AA	2632	2633
B@SLVL	001	0105	2636	2638
B@SLVS	001	0091	2631	2633
B@SQUO	001	007D	2432	3777 3780
B@STAT	001	0000	2507	
B@TASA	001	0012	2242	
B@TASC	001	001E	2248	
B@TASM	001	0018	2244	
B@TASS	001	007B	2249	
B@TCGT	001	0030	2257	
B@TCLS	001	0042	2263	
B@TDAT	001	0006	2238	
B@TDEF	001	0009	2239	
B@TDIM	001	000C	2240	
B@TDUM	001	0078	2281	2960 3178
B@TEND	001	0072	2279	
B@TEOF	001	0075	2280	
B@TFOR	001	0021	2251	
B@TGET	001	0039	2260	
B@TGSB	001	0033	2258	
B@TGTO	001	002D	2256	
B@TIFA	001	0027	2253	
B@TIFC	001	002A	2254	
B@TIFS	001	007D	2255	
B@TIMG	001	0054	2269	
B@TINP	001	0045	2264	
B@TLTA	001	000F	2241	
B@TLTC	001	001B	2245	
B@TLTM	001	0015	2243	
B@TLTS	001	0079	2246	

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 98

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@TMAS	001	007C	2250	
B@TMAT	001	0057	2270	
B@TMGT	001	005A	2271	
B@TMIN	001	005D	2272	
B@TMLS	001	007A	2247	
B@TMPR	001	0066	2275	
B@TMPT	001	0063	2274	
B@TMPU	001	0069	2276	
B@TMRD	001	0060	2273	
B@TNXT	001	0024	2252	
B@TPRT	001	004E	2267	
B@TPRU	001	0051	2268	
B@TPSE	001	006C	2277	
B@TPUT	001	003C	2261	
B@TRAC	001	0080	2511	
B@TREA	001	0048	2265	
B@TREM	001	0003	2237	
B@TRSR	001	004B	2266	
B@TRST	001	003F	2262	
B@TRTN	001	0036	2259	
B@TSTP	001	006F	2278	
B@VMC1	001	0056	2650	
B@VMLB	001	F0CD	2655	
B@VMSB	001	F5E5	2654	
B@VMSZ	001	0000	2651	2653 2654 2655
B@VMTB	001	0000	2653	
B@ZNEG	001	00D0	2524	
B@ZPOS	001	00F0	2523	
DL4CYL	001	13AB	5212	5184*
DL4C01	002	13B1	5220	5170 5172 5184
DL4C05	002	13B3	5221	5176
DL4C24	003	1382	5223	5197
DL4C48	003	136F	5225	5191 5232 5238
DL4C96	003	135E	5222	5185
DL4DPL	006	13AF	5211	5177*
DL4EFD	001	0001	5218	5190 5236
DL4END	001	13F1	5249	
DL4ETB	001	0080	5219	5196
DL4E01	001	0001	5217	5192
DL4E24	001	0018	5216	5194
DL4E48	001	0030	5215	5188 5230
DL4E96	001	0060	5214	5182
DL4ICS	001	1335	5165	4288 4881
DL4LST	001	13AA	5210	5203 5212 5213 5224 5242*
DL4SAV	005	134C	5248	5235* 5238* 5241
DL4SCD	001	13AC	5213	5182 5185* 5188 5191* 5194 5197* 5198 5198* 5199 5199* 5200* 5229
DL4SCT	001	13AD	5224	5235 5241* 5243*
DL4SPT	004	13B4	5228	5192 5227 5233* 5242 5243 5244*
DL4SPT	004	13B4	5228	5193
DL4WRK	005	134D	5247	5227* 5229* 5230 5232* 5233 5244
DL4010	001	1339	5168	5166 5169
DL4020	005	1349	5175	5171* 5247 5248
DL4030	005	1352	5177	5175* 5176*
DL4035	003	1357	5179	5245
DL4040	003	135D	5182	5186 5222
DL4050	003	136E	5188	5183 5225

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 99

SYMBOL	LEN	VALUE	DEFN	REFERENCES
DL4060	003	137B	5192	5189
DL4070	003	1381	5194	5223 5231 5237 5239
DL4080	004	138E	5198	5195
DL4100	003	1396	5200	5179* 5190* 5196* 5236
DL4200	003	139F	5205	5180* 5234*
DL4500	004	13B4	5227	5228
DL4600	004	13DE	5241	5205
DL4900	004	13A2	5207	5167*
DL4920	004	13A6	5208	5173*
GCPACK	001	1041	4364	4643
GCPBFR	001	1A00	2996	4371 4372 4405* 4406* 4415
GCPMAX	001	001B	4419	4392
GCPONE	001	10AF	4414	4394
GCPSTL	002	10B1	4415	4406
GCPTWO	001	0002	4418	4380 4385 4387
GCP020	003	1055	4376	4401
GCP050	003	106C	4386	4395
GCP080	003	1085	4396	4393
GCP090	004	108B	4398	4379 4381
GCP100	003	108F	4399	4388
GCP110	004	1099	4405	4377
GCP120	004	10A3	4407	4370*
GCP130	004	10A7	4408	4369*
GCP140	004	10AB	4409	4368*
GPUADR	001	1D00	4578	4579
GPUBFR	001	1163	4697	4612 4614 4909* 4911* 4912
GPUBF1	001	1800	2986	2988 4693
GPUBRK	001	0080	4722	4652 4654 4834 4926
GPUCBL	001	1182	4755	4772* 4773* 4774 4945* 4946* 4950*
GPUCLA	002	1181	4751	4612* 4630 4674* 4752
GPUCNT	002	1176	4715	4648* 4716
GPUCYL	001	115F	4695	
GPUDBS	001	1174	4712	4645 4713 4903*
GPUDPL	001	115E	4690	4589 4592 4695 4696 4697 4882 4906*
GPUDSP	001	1178	4736	4737 4905* 4907
GPUDS0	001	0000	4559	
GPUDS1	001	0001	4560	4877*
GPUDS2	001	0002	4561	4876*
GPUDS3	001	0003	4562	4875*
GPUDS4	001	0004	4563	4869* 4947* 4948* 4949
GPUD11	001	000B	4564	4579
GPUECD	001	008A	3000	4646
GPUEOF	001	0040	4725	4665 4777 4853 4923
GPUERD	001	0020	4728	4776 4954
GPUERR	001	0963	2976	4677
GPUFIT	001	0001	4731	4884 4929
GPUFTS	001	1D0B	4579	4939*
GPUIDR	001	1177	4719	4652 4654* 4665 4720 4776* 4777* 4834* 4853 4884 4923 4926 4929
GPULIN	002	117A	4739	4954 4740 4780* 4901*
GPULN1	001	0001	4552	4658
GPULN2	001	0002	4553	4648 4780 4872 4876 4896 4901 4943
GPULN3	001	0003	4554	4875
GPULN4	001	0004	4555	4832 4860 4869 4898
GPULUD	002	117D	4746	
GPULUE	002	117F	4747	4748 4895* 4896 4939 4943* 4944

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 100

SYMBOL	LEN	VALUE	DEFN	REFERENCES
GPUL12	001	000C	4556	4939
GPUMOV	002	1184	4758	4640
GPUNUL	001	1168	4705	4869 4875 4876
GPUON1	001	0001	4574	4907
GPURCD	001	1173	4710	4778
GPUSCT	001	1160	4696	
GPUSDF	001	1164	4700	4702 4792* 4796* 4802* 4827 4829* 4832 4839 4840 4843
GPUSMT	001	1A00	2988	2990 2992 2993 2994 2996 4620* 4631* 4632 4639* 4640* 4657 4658 4661 4663 4757 4773 4778* 4780 4782 4792 4794* 4800 4801* 4813
				4816 4901
GPustr	001	117B	4742	4649 4658* 4743 4774 4782 4794 4796 4799* 4840* 4855 4860 4870 4872 4898 4904*
GPUSTT	002	1184	4757	4662 4758 4814
GPUTIT	001	10B2	4590	2856 2969
GPUXBC	001	00BC	4568	4945
GPUXFF	001	00FF	4566	4660 4812 4842 4904
GPUX08	001	0008	4576	4649 4855
GPU001	002	1186	4760	4648 4903 4905 4906 4909 4911 4948 4950
GPU003	002	1188	4761	4943
GPU004	002	118A	4762	4829 4895
GPU005	001	0005	4698	
GPU008	001	0008	4572	4772 4778
GPU050	004	10BA	4593	
GPU100	003	10C2	4604	4618*
GPU150	004	10C5	4612	
GPU188	001	00BC	4570	4645
GPU200	003	10D7	4630	4604
GPU210	004	10FA	4643	4635
GPU215	003	10FE	4645	4642
GPU220	004	1108	4648	4775 4779
GPU230	003	1112	4652	4924 4927
GPU240	004	111E	4656	4784 4953
GPU245	005	1140	4664	4660* 4661* 4662* 4663*
GPU247	003	1145	4665	4845
GPU250	003	114B	4674	
GPU260	004	114E	4675	4593* 4955 4957
GPU270	004	1152	4676	4591*
GPU275	004	1156	4677	4956*
GPU280	004	115A	4678	4594*
GPU300	003	118B	4772	4647
GPU320	003	119B	4776	
GPU340	003	119E	4777	4633
GPU360	005	11AA	4780	4653
GPU380	005	11B8	4792	
GPU390	005	11E5	4818	4812* 4813* 4814* 4816* 4826 4830
GPU395	005	11FA	4832	4830*
GPU396	003	1205	4839	4655
GPU398	005	1213	4844	4826* 4827* 4842* 4843*
GPU400	003	121C	4853	4651
GPU405	003	1234	4860	4666 4854 4856
GPU410	003	123E	4870	4861
GPU420	004	1251	4876	4874
GPU430	003	1255	4877	4873
GPU450	004	1258	4881	4835 4871
GPU455	005	126D	4898	4896*
GPU457	003	1272	4900	4858*

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 101

SYMBOL	LEN	VALUE	DEFN	REFERENCES
GPU460	004	127A	4903	4885 4900
GPU470	004	1296	4911	4908
GPU475	003	129A	4912	4910
GPU480	003	12B4	4931	4857* 4952*
GPU490	004	12CA	4947	4951
GPU500	004	12DD	4952	4930 4931
GPU502	004	12E1	4953	4656* 4859*
GRABIT	001	0EA3	4154	2836 2844
GRABOA	002	102C	4319	4248 4261 4266
GRABSE	004	0F81	4345	4153 4156
GRACCA	002	101D	4296	
GRACFN	001	101C	4294	
GRACPL	001	101C	4293	
GRACSC	001	101F	4299	4175*
GRAEBS	001	00FF	4327	4174 4290
GRAEDB	001	0002	4313	4182 4285
GRAEDC	001	0001	4344	
GRAEDL	001	0006	4332	4199 4217
GRAEDS	001	0005	4346	4280
GRAEDT	001	0007	4333	4189 4218 4220
GRAEET	001	0075	4335	4189 4220
GRAEFG	001	0004	4326	4211
GRAEFI	001	0000	4322	2833 4158
GRAEFR	001	0001	4324	2839 4163 4209
GRAEFS	001	0002	4325	4165
GRAEFW	001	0003	4323	
GRAELK	001	0000	4329	4180 4183 4283 4286
GRAELL	001	0002	4334	4217
GRAELN	001	0000	4330	4180 4283
GRAELP	001	0007	4340	4232
GRAELS	001	0004	4341	4245
GRAEMR	001	001B	4342	4252
GRAENC	001	0001	4343	4252 4257* 4263 4265
GRAERR	004	1035	4351	4178* 4194 4206 4210
GRAESC	001	0001	4328	
GRAES0	001	0001	4336	4196 4205
GRAES1	001	0002	4337	4191 4192 4229 4230* 4231 4242 4243* 4244
GRAES2	001	0003	4338	4207 4226 4239
GRAETP	001	0002	4339	4207
GRAEW2	001	0006	4347	
GRAEXA	001	0001	4331	4332 4333 4336 4337 4338
GRANCA	002	1027	4307	4172* 4179* 4280 4281*
GRANDA	002	1024	4303	4173* 4182* 4183* 4184* 4285* 4286* 4287*
GRANPB	002	102C	4312	4184 4287 4318 4319 4320
GRANPL	001	1022	4301	4289
GRANXC	002	102C	4320	
GRAONE	002	102C	4318	4257
GRAPSG	002	1031	4316	4230
GRASAR	004	0F24	4203	4157*
GRASBR	004	0F20	4201	4155*
GRASEG	001	1034	4321	4231* 4244* 4266*
GRASIZ	001	102D	4314	4174* 4191* 4193 4229* 4242* 4290*
GRASSG	002	1033	4317	4243
GRASSZ	002	102A	4311	4179
GRASVC	003	0FA5	4247	4237*
GRATND	005	0FBF	4256	4254* 4259 4261*

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 102

SYMBOL	LEN	VALUE	DEFN	REFERENCES
GRATXT	002	102F	4315	4219
GRA020	004	0EB5	4162	4198*
GRA100	003	0EC8	4171	4159
GRA140	003	0EE6	4180	
GRA150	004	0EF3	4184	4181
GRA200	003	0EFA	4189	4166
GRA210	004	0F00	4191	4167 4213
GRA220	003	0F07	4193	4234 4236
GRA230	004	0F16	4198	4190 4208 4212 4223
GRA240	004	0F1D	4200	4201
GRA245	004	0F21	4202	4203
GRA250	003	0F25	4204	4195 4197
GRA260	003	0F28	4205	4185
GRA300	005	0F46	4217	4164
GRA303	003	0F63	4225	4221
GRA305	004	0F6F	4229	4227
GRA310	004	0F81	4234	4225* 4228* 4235 4241* 4267 4345
GRA313	004	0F95	4242	4240
GRA315	003	0FA4	4246	4247
GRA316	004	0FA7	4248	4268
GRA317	001	0FAB	4249	4233
GRA320	005	0FBC	4255	4256 4262
GRA330	004	0FCF	4261	4258
GRA350	005	0FD6	4263	4251 4253 4264
GRA360	003	0FDB	4265	4260
GRA5SA	004	101B	4292	4272*
GRA500	003	0FE8	4272	4204 4238
GRA600	001	0FF1	4275	
GRA620	004	100B	4287	4284
GRA640	004	100F	4288	
GRA660	003	1015	4290	
GRA680	004	1018	4291	4292
GRBFRA	002	1021	4300	4171 4279 4280* 4282
GRBFR1	001	1B00	2990	3017 4300
GRLINE	001	1A05	2992	4217*
GRSCTR	001	1025	4304	4175
GRSRDA	002	101E	4295	2834* 4173 4296
GRTEND	005	0FD9	4264	2911 2913* 2914 2921* 4219* 4248* 4254 4259* 4639
GRTEXT	001	1A07	2994	2846 2849 4222* 4315
GRTYPE	001	1A06	2993	2960* 3183 4218*
GRWHAT	001	1028	4308	2833* 2839* 4158 4163 4165 4209 4211
KRVANY	001	0001	3008	2901
KRVCC1	002	0987	3032	2916 2918
KRVDPG	001	097A	3014	2831
KRVDUM	001	098D	3042	
KRVD02	001	0001	3005	2881 2890 2902 2945 2950 2953
KRVECT	001	098A	3039	2980
KRVERS	001	0983	3023	2963* 2972 2977 3024
KRVFOR	001	0004	3002	2867
KRVFVM	002	0981	3019	2834
KRVLAY	001	0807	2829	
KRVLLT	001	0002	3009	2905
KRVLNG	001	0982	3021	2901* 2905* 2906
KRVMAX	001	00FA	3001	2957
KRVONE	001	0001	3004	2926 2963
KRVSER	001	098F	3044	2981 2981 2981*

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 103

SYMBOL	LEN	VALUE	DEFN	REFERENCES
KRVSE1	001	0989	3038	2980
KRVSE2	001	098B	3040	2980 2981 2981*
KRVTBL	001	0707	3003	2862
KRVTB2	001	0002	3006	2900
KRVTMP	002	0985	3028	2851* 2873 2906 2918* 2926 2929* 2930 2931 2935*
KRVTM1	001	0984	3027	3029
KRVTWO	001	0988	3034	2929 2935
KRV002	001	0002	3007	
KRV380	004	0827	2844	2857
KRV390	004	0837	2850	2874
KRV392	004	0847	2856	2964
KRV395	004	084F	2862	2854
KRV400	004	0853	2864	2869
KRV405	003	085A	2867	2893
KRV407	004	0864	2873	2958
KRV410	004	086C	2878	2865
KRV420	004	087E	2885	2879
KRV430	004	088D	2891	2885*
KRV440	004	0891	2893	2882
KRV450	004	0895	2898	2883
KRV460	006	08AA	2906	2903
KRV470	006	08D7	2919	2911* 2914* 2915 2915* 2916* 2921 2957 2961
KRV500	004	08E3	2926	2907
KRV510	004	08FD	2933	2931*
KRV520	003	090A	2938	2927
KRV530	004	090D	2942	2898* 2913 2936
KRV550	003	0922	2950	2946
KRV555	003	092E	2955	2951
KRV560	004	0931	2957	2949 2954
KRV570	004	0943	2962	2961*
KRV600	004	094F	2969	2847
KRV650	004	0976	2982	2978
SCACNT	002	1334	5083	5073* 5074*
SCACOF	001	0087	5055	
SCACOM	001	0001	5054	
SCAINC	001	0001	5053	5062 5068
SCAMMA	003	1311	5077	
SCANIT	001	12F4	5057	2888
SCASVE	002	1332	5082	5059* 5074
SCASV1	001	1331	5081	
SCA100	003	1303	5062	5064
SCA200	003	1306	5063	5061
SCA250	003	1310	5066	5077
SCA300	003	1313	5068	5070
SCA400	004	1323	5073	5066
SCA500	004	132D	5076	5058* 5072
SVABRT	001	0E1C	3985	3191
SVABSW	001	0DA7	3887	3223 3255 3260* 3285 3291* 3366 3368* 3388 3393* 3410 3416* 3435
				3440* 3473*
SVACAC	001	0002	3883	2953 3586
SVACON	003	0E15	3962	3338
SVACVC	001	0004	3881	2950 3578
SVADET	003	0DC2	3921	3653
SVADIS	001	0080	3875	3184
SVADSW	001	0DAC	3899	3238* 3757* 3816 3818 3824*
SVAFIL	001	0DB7	3910	3635

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 104

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SVAFNC	002	0DC7	3926	3617
SVAFTD	001	0041	3874	3627
SVAIDN	003	0E18	3963	3340
SVAIFT	001	0DCB	3931	3628
SVAINV	003	0E0F	3957	3320
SVAI01	002	0DB6	3909	3209 3327 3546 3565 3658 3699 3741 3779 3782 3794 3798 3839
				3848 3849
SVAKGO	002	0DBF	3917	3606
SVAKLN	001	0002	3869	3598 3600 3604 3606
SVAKST	002	0DBB	3915	3600
SVAKTH	002	0DBD	3916	3604
SVAKTO	002	0DB9	3914	3598
SVAKWL	001	0DA9	3893	3193* 3263* 3741* 3894
SVALDC	001	0010	3880	2878 2902 2945 3548
SVALNG	001	0DB2	3903	2851 3298* 3330* 3456* 3498* 3546* 3565* 3661* 3672* 3849*
SVALSA	003	0DB0	3901	3320 3322 3338 3340 3342 3968 3969 3970
SVALS1	003	0DAE	3968	3312* 3493*
SVALS2	003	0DAF	3969	3314* 3503* 3598 3600 3604 3606 3617
SVALS3	003	0DB0	3970	3316* 3612* 3619 3630 3641 3653
SVALVC	001	0001	3879	3539 3670
SVAL15	001	000F	3872	3484
SVAMAG	001	0DB3	3904	3178* 3186* 3188
SVANAC	001	0008	3882	3300 3332 3459 3554 3663
SVAOFF	001	0000	3864	3223 3255 3285 3292 3366 3388 3394 3410 3435 3441 3473
SVAONN	001	0001	3865	3238 3260 3291 3368 3393 3416 3440
SVAPCT	001	0DAD	3900	3292* 3394* 3441* 3448 3484* 3537 3699* 3794* 3798*
SVAPD0	001	0000	3862	3195 3207 3225 3227 3229 3231 3233 3306 3308 3312 3314 3316
				3493 3503 3507 3512 3517 3522 3524 3526 3528 3530 3570 3612
				3630 3647 3777 3780 3787 3804 3806 3808 3810 3812 3814 3824
				3843
SVARAB	001	0990	3173	2850
SVARND	003	0DC5	3922	3641
SVASSS	001	0DA8	3890	3602* 3683 3685* 3891
SVASTC	002	0DAB	3896	3183* 3184* 3186 3190* 3192
SVASTR	003	0DCA	3927	3619
SVATRN	003	0E12	3958	3322
SVAVL1	001	0001	3867	3298 3330 3456 3498 3661 3672
SVAVL2	001	0002	3870	
SVAVL3	001	0003	3871	
SVAVTC	001	0DB1	3902	2878 2881 3300* 3332* 3459* 3539* 3548* 3554* 3578* 3586* 3663* 3670*
SVAZER	003	0E1B	3964	3342
SVAZRO	001	0DB4	3905	3179* 3188* 3189 3189* 3190
SVA0TD	001	0000	3863	3193
SVA020	004	09A8	3184	
SVA030	004	09BF	3191	3187
SVA050	003	09D2	3207	3210 3988 3991 4036 4039 4060 4069 4093 4096 4099 4102 4105
SVA070	004	09E1	3222	4003 4009 4015 4114 4117
SVA075	003	09EB	3225	3236
SVA080	003	09F7	3229	3226
SVA085	004	0A10	3237	3224 3228 3230 3232 3234
SVA090	004	0A1B	3251	4033
SVA095	003	0A1F	3255	
SVA1TD	001	0001	3866	
SVA100	003	0A25	3260	
SVA110	004	0A37	3268	3256
SVA120	004	0A3F	3281	4072

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 105

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SVA125	003	0A43	3285	
SVA130	004	0A49	3290	
SVA140	004	0A53	3296	
SVA144	003	0A5E	3299	
SVA148	004	0A64	3301	3297*
SVA150	003	0A6C	3306	3286
SVA151	003	0A78	3310	3307
SVA152	004	0A7B	3311	3309
SVA154	004	0A91	3320	
SVA155	004	0A98	3322	
SVA156	003	0A9F	3327	3321
SVA158	004	0AB3	3333	3329*
SVA160	004	0ABA	3338	3323
SVA163	003	0ACF	3344	3339 3341
SVA165	004	0AD5	3349	3311* 3343
SVA168	004	0AD9	3350	3345
SVA170	004	0ADD	3362	4042 4045 4048 4051
SVA175	003	0AE1	3366	
SVA190	004	0AEE	3373	3367
SVA192	004	0AF5	3384	4078 4081 4087 4090
SVA194	003	0AF9	3388	
SVA196	003	0AFF	3393	
SVA2TD	001	0002	3868	3194
SVA200	004	0B0C	3406	3994 3997 4000 4006 4012 4018 4021 4024 4027 4030 4054 4057 4063 4066 4108 4111 4120
SVA205	003	0B10	3410	
SVA210	004	0B16	3415	
SVA220	004	0B1D	3420	3411
SVA250	004	0B24	3431	4075 4084
SVA255	003	0B28	3435	
SVA260	003	0B2E	3440	
SVA270	004	0B3B	3447	3350 3389 3396 3436
SVA300	003	0B3F	3448	
SVA305	004	0B45	3450	3443
SVA310	004	0B4C	3455	3449
SVA312	004	0B56	3458	3455*
SVA315	004	0B5D	3460	3175* 3208 3239 3269 3302 3334 3374 3421 3451 3474
SVA320	004	0B61	3461	3174*
SVA330	003	0B65	3473	3788
SVA395	003	0B6C	3484	3237 3268 3373 3420
SVA400	004	0B6F	3488	3450
SVA410	004	0B73	3492	3649 3691 3702
SVA415	003	0B7F	3498	
SVA420	003	0B82	3502	
SVA430	003	0B89	3507	
SVA440	003	0B8F	3512	
SVA450	003	0B95	3517	
SVA460	003	0B9B	3522	
SVA465	003	0BA7	3526	3523
SVA470	003	0BB9	3537	
SVA480	004	0BC5	3546	3508
SVA490	003	0BD2	3554	3513 3538
SVA500	004	0BD8	3565	3518
SVA505	003	0BDF	3570	
SVA510	003	0BE6	3578	
SVA520	003	0BEC	3586	3571

CROSS REFERENCE

VER 15, MOD 00 30/10/23 PAGE 106

SYMBOL	LEN	VALUE	DEFN	REFERENCES
SVA525	003	0BEF	3587	3540 3549 3555 3579
SVA530	004	0BF2	3598	3525 3527 3529 3531
SVA535	004	0C07	3604	3601
SVA540	003	0C15	3611	
SVA545	004	0C18	3612	3687
SVA550	004	0C1C	3617	
SVA560	004	0C2A	3625	
SVA565	003	0C36	3628	3636
SVA570	003	0C39	3629	3627* 3635*
SVA580	004	0C4C	3640	3625*
SVA590	003	0C57	3646	
SVA600	004	0C64	3653	3642
SVA610	003	0C6B	3658	
SVA620	006	0C82	3669	3654
SVA625	004	0C8B	3671	3669*
SVA630	003	0C98	3682	3603 3605 3607
SVA632	003	0CAD	3688	3684
SVA635	003	0CB0	3689	3599
SVA640	003	0CB7	3698	3618 3620 3712
SVA645	004	0CBA	3699	3648 3703
SVA650	004	0CCC	3711	3626* 3631
SVA660	004	0CD4	3719	3494* 3587 3660* 3664 3669 3674
SVA670	004	0CD8	3720	3488*
SVA700	004	0CDC	3732	3261 3264 3290 3369 3395 3415 3442
SVA710	004	0CE0	3736	3742
SVA720	004	0CE7	3741	
SVA750	004	0CEF	3746	3732*
SVA900	004	0CF3	3756	3262 3296 3328 3447 3492 3659 3736
SVA902	003	0CFA	3761	3783 3826
SVA904	003	0D00	3766	
SVA906	003	0D12	3772	3770
SVA908	003	0D15	3773	3768
SVA910	003	0D18	3777	3762
SVA915	003	0D1E	3779	3781
SVA920	003	0D2F	3787	3778
SVA925	003	0D36	3792	
SVA928	003	0D43	3796	3793
SVA930	003	0D50	3804	3797
SVA940	004	0D80	3820	3756* 3805 3807 3809 3815
SVA950	004	0D84	3824	3795 3799 3811 3813 3817 3819
SVA960	003	0D8F	3838	3176 3177 3222 3251 3281 3299 3310 3313 3315 3331 3362 3384 3406 3431 3457 3502 3547 3566 3611 3646 3662 3673 3682 3688 3689 3698 3700 3766 3771 3772 3773 3825
SVA966	003	0D95	3843	3850
SVA970	004	0D98	3844	3838*
SVA975	003	0D9C	3848	

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 0

OL105 I THE CODE LENGTH OF #KRVLA IS 5120 DECIMAL.