

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

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



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	01/08/20	PAGE	2
	0000				1	#KGOSL	START 0					
					2		PRINT ON					
					3	*	@SYS EXP-N					
					212+		PRINT ON					
					213	*	@HDW EXP-N					
					397+		PRINT ON					
					398	*	@FXD EXP-N					
					802+		PRINT ON					
					803	*	@CAN EXP-N					
					906+		PRINT ON					
					907	*	@ERM EXP-N					
					1529+		PRINT ON					
	0C00				1530		ORG \$\$KLD3					
					1531	*	HDR #KGOSL,0					
					1532	*****						
					1533	*	PROGRAM HEADER FOR DISK LOAD					
					1534	*****						
				0180	1535	#\$KGOS EQU	X'0180'					DISK ADDR OF OKGOSL
			0C00		1536	\$\$\$KGO EQU	X'0C00'					CORE LOAD ADDRESS OF #KGOSL
			0002		1537	#\$@KGO EQU	002					SECTOR CNT OF #KGOSL
	0C00				1538		ORG \$\$\$KGO					CORE LOAD ADDRESS
			0C00		1539	\$\$\$\$\$\$ EQU	*					FIRST LOCATION IN PROGRAM
	0C00	7BD2C7D6E2D3		0C05	1540		DC CL6 '#KGOSL'					PROGRAM NAME
	0C06	07		0C06	1541		DC IL1 '007'					PROGRAM NUMBER OF #KGOSL
				0C07	1542	#KGOS EQU	*					ENTRY POINT TO PROGRAM
					1543	***	END OF EXPANSION ***					

@ERMEQ - GENERAL ERROR MESSAGE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	01/08/20	PAGE	3
1545				*****				
1546	*	5703-XM1		COPYRIGHT IBM CORP 1970				*
1547	*			REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083				*
1548	*							*
1549				*****				
1550	*			*STATUS -				*
1551	*			VERSION 1 MODIFICATION 0				*
1552	*							*
1553	*			*FUNCTION				*
1554	*	*		KGOSLO CAN RESUME THE EXECUTION OF A PROGRAM IN A PAUSE				*
1555	*			CONDITION IN ONE OF 3 MODES: STEP,TRACE,RUN. IF NO PARAMETER IS				*
1556	*			SPECIFIED WITH THE KEYWORD GO. EXECUTION CONTINUES IN THE SAME				*
1557	*			MODE AS WAS SPECIFIED BY THE COMMAND THAT LAST INITIATED				*
1558	*			EXECUTION.				*
1559	*	*		IF THE STEP PARAMETER IS SPECIFIED, EXECUTION IS CONTINUED IN				*
1560	*			'STEP' MODE. IF THE RUN PARAMETER IS SPECIFIED, EXECUTION IS				*
1561	*			CONTINUED IN THE NORMAL 'RUN' MODE. IF				*
1562	*			THE TRACE PARAMETER IS SPECIFIED, EXECUTION IS CONTINUED IN THE				*
1563	*			'TRACE' MODE ONLY IF THAT WAS THE ORIGINAL MODE OF EXECUTION,				*
1564	*			OTHERWISE. THE COMMAND IS REJECTED. IF 'ABORT' IS SPECIFIED AS				*
1565	*			AN OPERAND, THE PROGRAM IS ABORTED AND NOT EXECUTED.				*
1566	*							*
1567	*			*ENTRY POINTS				*
1568	*			THE ENTRY IS KGOSLO. THE BASE AND INDEX REGISTERS ARE NOT SAVED.				*
1569	*							*
1570	*			*INPUT				*
1571	*			THE INPUT IS THE PARAMETER FROM THE OPERATOR.				*
1572	*							*
1573	*			*OUTPUT				*
1574	*			NONE				*
1575	*							*
1576	*			*EXTERNAL REFERENCES				*
1577	*			\$CIMSK - ADDRESS OF INQUIRY REQUEST				*
1578	*			\$XIND2 - ADDRESS OF EXECUTION INDRS				*
1579	*			\$CAERR - ADDRESS OF ERROR CODE FOR ERROR PGM				*
1580	*			\$CAERK - ADDRESS OF ENTRY POINT TO ERROR PGM				*
1581	*			\$XRSAB - ADDRESS OF 2 BYTE SAVE AREA				*
1582	*			SCANIT - ADDRESS OF ENTRY POINT TO BLANK SCAN ROUTINE				*
1583	*			\$INDR3 - ADDRESS OF SYSTEM 1-BIT INDRS				*
1584	*			\$RSTR - ADDRESS OF ENTRY TO RESTORE CORE				*
1585	*			\$XINDI - ADDRESS OF EXECUTION INDRS				*
1586	*							*
1587	*			*EXITS, NORMAL				*
1588	*			NORMAL EXIT IS TO \$CARPL TO RETURN TO NORMAL INPUT MODE				*
1589	*							*
1590	*			*EXITS, ERROR				*
1591	*			SAME AS FOR NORMAL EXITS, BUT ABNORMAL TERMINATION				*
1592	*							*
1593	*			*TABLES/WORK AREAS				*
1594	*			THE CONSTANTS RESIDE AT THE END OF EXECUTABLE CODE.				*
1595	*							*
1596	*			*ATTRIBUTES				*
1597	*			RELOCATABLE				*
1598	*							*
1599	*			*CHARACTER CODE DEPENDENCY				*
1600	*			THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR				*

@ERMEQ - GENERAL ERROR MESSAGE EQUATES

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 01/08/20 PAGE 4
		1601	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		1602	*		*
		1603	*	NOTES	*
		1604	*	ERROR PROCEDURES	*
		1605	*	EXIT IS MADE TO THE ERROR PGM IF THE SYSTEM IS NOT IN A PAUSE	*
		1606	*	STATE, A SYNTAX ERROR IS FOUND. OR 'TRACE' IS THE PARAMETER	*
		1607	*	WHEN THE ORIGINAL MODE OF EXECUTION WAS NOT TRACE.	*
		1608	*		*
		1609	*	REGISTER USAGE	*
		1610	*	INDEX REGISTER 2 C@XR) IS USED TO SYNTAX CHECK.	*
		1611	*		*
		1612	*	SAVED/RESTORED AREAS	*
		1613	*	NONE	*
		1614	*		*
		1615	*	MODIFICATION CONSIDERATIONS	*
		1616	*	NONE	*
		1617	*		*
		1618	*	REQUIRED MODULES	*
		1619	*	@SYSEQ - COMMON SYSTEM EQUATES	*
		1620	*	@HDWEQ - SYSTEM HARDWARE EQUATES	*
		1621	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES	*
		1622	*	@CANEQ - SYSTEM LOCATION EQUATES	*
		1623	*	@CY0EQ - CYLINDER ZERO EQUATES	*
		1624	*	SCANIT - BLANK SCAN ROUTINE	*
		1625	*		*
		1626	*	OTHER	*
		1627	*	NONE	*
		1628	*	*****	*
0C07 C2 01 0C1E		1630		LA KGO110,@BR POINT BR TO ORIGIN	
	0C1E	1631		USING KGO110,@BR BASE REGISTER DISPLACEMENT	
0C0B 3C 80 0476		1632		MVI \$CIMSK,@NOP MASK PROG INTERRUPTS	
		1633	*		
		1634	*	DETERMINE IF IN EXECUTION PAUSE CONDITION	
		1635	*		
0C0F 38 02 03D1		1636	KGO100	TBN \$XIND2,\$PAUSE TEST FOR PAUSE STATEMENT	
0C13 F2 10 08		1637		JT KGO110 JUMP IF TRUE	
0C16 3C 2E 03CD		1638		MVI \$CAERR,@E225 SET ERROR CODE	
0C1A C0 87 0469		1639		B \$CAERK PROCESS ERROR CONDITION	
		1640	*		
		1641	*	SYNTACTICAL CHECK OF INPUT BUFFER	
		1642	*	TO DETERMINE TYPE OF GO COMMAND	
		1643	*		
0C1E 35 02 03C7		1644	KGO110	L \$XRSAV,@XR XR POINTS TO INPUT BUFFER	
0C22 C0 87 0D0E		1645		B SCANIT SCAN FIELD TO NON-BLANK	
0C26 34 02 0CEE		1646		ST KGO215+@OP1,@XR SAVE XR	
0C2A C0 01 0C34		1647		BNZ KGO120 CHARACTER POINTER MOVED	
0C2E BD 1E 00		1648		CLI 0(,@XR),@EOS CHECK FOR EOS CHARACTER	
0C31 F2 01 AC		1649		JNE KGO210 JUMP IF NOT EOS CHARACTER	
		1650	*		
		1651	*	INDEX REGISTER POINTS TO NON-BLANK	
		1652	*	DETERMINE IF 'ABORT' OPERAND	
		1653	*		
0C34 9D 04 04 D9		1654	KGO120	CLC KGOOL0-1(KGOOL0,@XR),KGOABT(,@BR) CHECK FOR 'ABORT'	
0C38 F2 01 13		1655		JNE KGO130 JUMP IF NOT 'ABORT'	
0C3B 36 02 0D05		1656		A KGOOP0,@XR INCREMENT XR PASSED FIELD	

@ERMEQ - GENERAL ERROR MESSAGE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 01/08/20 PAGE 5
	0C3F	C0	87	0CCE	1657	B	KGO180	BRANCH TO CHECK FOR EOS
	0C43	3C	00	03D1	1658	MVI	\$XIND2,@ZERO	SET ALL STATUS INDICATORS OFF
	0C47	3A	10	03D1	1659	SBN	\$XIND2,\$ABORT	SET ON 'ABORT' BIT
	0C4B	F2	87	0E	1660	J	KGO136	JUMP TO RESTORE CORE
	0C4E	BD	1E	00	1661	KGO130	CLI 0(,@XR),@EOS	CHECK FOR EOS
	0C51	F2	01	10	1662	JNE	KGO140	JUMP IF NOT EOS
	0C54	3A	01	03D1	1663	KGO135	SBN \$XIND2,\$EXCMD	
	0C58	3C	80	0476	1664	MVI	\$CIMSK,@NOP	MASK INTERRUPTS
	0C5C	3A	10	03D6	1665	KGO136	SBN \$INDR3,\$CLBFR	CLEAR INPUT BUFFER
	0C60	C0	87	04D6	1666	KGO137	B \$RSTR	RESTORE CORE FROM DISK
					1667	*		
					1668	*	DETERMINE IF 'STEP' OPERAND	
					1669	*		
	0C64	9D	03	03 DD	1670	KGO140	CLC KGOOL1-1(KGOOL1,@XR),KGOSTP(,@BR)	CHECK FOR 'STEP'
	0C68	F2	01	14	1671	JNE	KGO150	JUMP IF NOT 'STEP'
	0C6B	36	02	0D07	1672	A	KGOOP1,@XR	INCREMENT XR PASSED FIELD
	0C6F	C0	87	0CCE	1673	B	KGO180	BRANCH TO CHECK FOR EOS
	0C73	3B	05	03D0	1674	SBN	\$XIND1,\$RUNIT+\$TRACE	TURN OFF RUN AND TRACE INDR
	0C77	3A	02	03D0	1675	SBN	\$XIND1,\$STEPT	SET ON STEP INDR
	0C7B	C0	87	0C54	1676	B	KGO135	RESTORE CORE FROM DISK
					1677	*		
					1678	*	DETERMINE IF 'RUN' OPERAND	
					1679	*		
	0C7F	9D	02	02 E5	1680	KGO150	CLC KGOOL3-1(KGOOL3,@XR),KGORUN(,@BR)	CHECK FOR 'RUN'
	0C83	F2	01	14	1681	JNE	KGO160	JUMP IF NOT 'RUN'
	0C86	36	02	0D0B	1682	A	KGOOP3,@XR	INCREMENT XR PASSED FIELD
	0C8A	C0	87	0CCE	1683	B	KGO180	BRANCH TO CHECK FOR EOS
	0C8E	3B	06	03D0	1684	KGO155	SBN \$XIND1,\$STEPT+\$TRACE	TURN OFF STEP AND TRACE INDR
	0C92	3A	01	03D0	1685	SBN	\$XIND1,\$RUNIT	SET ON RUN INDR
	0C96	C0	87	0C54	1686	B	KGO135	JUMP AND RESTORE CORE FROM DIS
					1687	*		
					1688	*	DETERMINE IF 'TRACE' OPERAND	
					1689	*		
	0C9A	9D	04	04 E2	1690	KGO160	CLC KGOOL2-1(KGOOL2,@XR),KGOTRC(,@BR)	CHECK FOR 'TRACE'
	0C9E	F2	81	07	1691	JE	KGO170	JUMP IF 'TRACE' FOUND
	0CA1	3C	11	03CD	1692	MVI	\$CAERR,@E131	INVALID PARAMETER CODE
	0CA5	F2	87	47	1693	J	KGO220	JUMP TO ERROR PROGRAM
	0CA8	36	02	0D09	1694	KGO170	A KGOOP2,@XR	INCREMENT XR PASSED FIELD
	0CAC	C0	87	0CCE	1695	B	KGO180	BRANCH TO CHECK EOS
	0CB0	39	38	03D0	1696	TBF	\$XIND1,\$TFLOW+\$TRALL+\$TRVAR	CHECK ANY TRACE INDRS ON
	0CB4	F2	10	0C	1697	JT	KGO175	JUMP IF ORIG MODE NOT TRACE
	0CB7	3B	03	03D0	1698	SBN	\$XIND1,\$STEPT+\$RUNIT	TURN OFF STEP AND RUN INDRS
	0CBB	3A	04	03D0	1699	SBN	\$XIND1,\$TRACE	SET ON TRACE INDR
	0CBF	C0	87	0C54	1700	B	KGO135	JUMP TO PROCESS
	0CC3	D2	02	00	1701	KGO175	LA 0(,@BR),@XR	POINT XR OUT OF BUFFER
	0CC6	3C	36	03CD	1702	MVI	\$CAERR,@E237	MOVE ERROR CODE
	0CCA	C0	87	0469	1703	B	\$CAERK	BRANCH TO ERROR PROGRAM
					1704	*		
					1705	*	CHECK FOR EOS CHARACTER AFTER PARAMETER FIELD	
					1706	*	EXIT:	
					1707	*	EOS NOT FOUND - ERROR CODE PROCESSING	
					1708	*	EOS FOUND - NORMAL RETURN	
					1709	*		
	0CCE	34	08	0D0D	1710	KGO180	ST KGOOP4,@ARR	SAVE ARR FOR RETURN
	0CD2	C0	87	0D0E	1711	B	SCANIT	SCAN FIELD TO NON-BLANK
					1712	*		

@ERMEQ - GENERAL ERROR MESSAGE EQUATES

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	01/08/20	PAGE	6
0CD6	BD	1E	00		1713	KGO190	CLI 0(,@XR),@EOS				CHARACTER EOS ?
0CD9	F2	01	04		1714		JNE KGO210				JUMP IF NOF EOS
0CDC	35	10	0D0D		1715	KGO200	L KGOOP4,@IAR				RETURN TO CALL
0CE0	3C	11	03CD		1716	KGO210	MVI \$CAERR,@E131				INVALID PARAMETER
0CE4	3D	00	0D4E		1717		CLI SCACNT,@ZERO				POINTER MOVED ?
0CE8	F2	01	04		1718		JNE KGO220				NO
0CEB	C2	02	0000		1719	KGO215	LA *-*,@XR				RESTORE XR
0CEF	C0	87	0469		1720	KGO220	B \$CAERK				ERROR PROCESS
					1721	*					
					1722	*	GO COMMAND OPERANDS USED IN KGOSLO				
					1723	*					
				0CF3	1724	KGOEQ0	EQU *				
0CF3	C1C2D6D9E3			0CF7	1725	KGOABT	DC CL5 'ABORT'				
				0CF8	1726	KGOEQ1	EQU *				
0CF8	E2E3C5D7			0CFB	1727	KGOSTP	DC CL4 'STEP'				
				0CFC	1728	KGOEQ2	EQU *				
0CFC	E3D9C1C3C5			0D00	1729	KGOTRC	DC CL5 'TRACE'				
				0D01	1730	KGOEQ3	EQU *				
0D01	D9E4D5			0D03	1731	KGORUN	DC CL3 'RUN'				
				0D04	1732	KGOEQ4	EQU *				
				1733	*						
				1734	*		CONSTANTS USED IN KGOSLO				
				1735	*						
0D04	0005			0D05	1736	KGOOP0	DC AL2(KGOEQ1-KGOEQ0)				'ABORT' LENGTH
0D06	0004			0D07	1737	KGOOP1	DC AL2(KGOEQ2-KGOEQ1)				'STEP' LENGTH
0D08	0005			0D09	1738	KGOOP2	DC AL2(KGOEQ3-KGOEQ2)				'TRACE' LENGTH
0D0A	0003			0D0B	1739	KGOOP3	DC AL2(KGOEQ4-KGOEQ3)				'RUN' LENGTH
				1740	*						
0D0C	0000			0D0D	1741	KGOOP4	DC AL2(*-*)				ARR SAVE AREA
				1742	*						
				1743	*		EQUATES USED IN KGOSLO				
				1744	*						
				0005	1745	KGOOL0	EQU KGOEQ1-KGOEQ0				ONE BYTE 'ABORT' LENGTH
				0004	1746	KGOOL1	EQU KGOEQ2-KGOEQ1				ONE BYTE 'STEP' LENGTH
				0005	1747	KGOOL2	EQU KGOEQ3-KGOEQ2				ONE BYTE 'TRACE' LENGTH
				0003	1748	KGOOL3	EQU KGOEQ4-KGOEQ3				ONE BYTE 'RUN' LENGTH
				0D0E	1749	KGOEND	EQU *				ADDRESS OF ARR SAVE
				1751	*		\$CANI				

## SCANIT - DELIMETER SCAN MODULE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	01/08/20	PAGE	7
		1753+		*****				
		1754+	*	5703-XM1	COPYRIGHT IBM CORP. 1970			*
		1755+	*		REFER TO INSTRUCTIONS ON COPYRIGHT NOTICE, 120-2083			*
		1756+	*					*
		1757+		*****				
		1758+	*	STATUS				*
		1759+	*	VERSION 1 MODIFICATION 0				*
		1760+	*					*
		1761+	*	FUNCTION				*
		1762+	*	THE FUNCTION OF SCANIT IS TO SCAN PAST VALID DELIMITERS AND				*
		1763+	*	RETURN A POINTER TO THE FIRST CHARACTER THAT'S NOT A DELIMITER.				*
		1764+	*					*
		1765+	*	ENTRY POINTS				*
		1766+	*	* THE ENTRY POINT IS SCANIT.				*
		1767+	*	* THE CALLING SEQUENCE IS AS FOLLOWS:				*
		1768+	*	B SCANIT				*
		1769+	*	WITH REGISTER 2 (@XR) POINTING TO THE FIRST CHARACTER TO BE				*
		1770+	*	EXAMINED.				*
		1771+	*					*
		1772+	*	INPUT				*
		1773+	*	NONE				*
		1774+	*					*
		1775+	*	OUTPUT				*
		1776+	*	NONE				*
		1777+	*					*
		1778+	*	EXTERNAL REFERENCES				*
		1779+	*	\$CAERR - ERROR CODE SAVE AREA				*
		1780+	*					*
		1781+	*	EXITS, NORMAL				*
		1782+	*	NORMAL EXIT FROM SCANIT IS TO THE BYTE FOLLOWING THE BRANCH TO				*
		1783+	*	SCANIT IN THE CALLING ROUTINE. THE PSR (REGISTER 4) WILL CONTAIN				*
		1784+	*	A ZERO IF NO DELIMITERS WERE FOUND OR A HIGH CONDITION IF ONE OR				*
		1785+	*	MORE DELIMITERS WERE SCANNED.				*
		1786+	*					*
		1787+	*	EXITS, ERROR				*
		1788+	*	ERROR EXIT FROM SCANIT IS TO THE BYTE FOLLOWING THE BRANCH TO				*
		1789+	*	SCANIT IN THE CALLING ROUTINE. THE PSR WILL CONTAIN A LOW				*
		1790+	*	CONDITION.				*
		1791+	*					*
		1792+	*	TABLES/WORKAREAS				*
		1793+	*	* SCACNT - AREA CONTAINING NUMBERS OF DELIMITERS SCANNED				*
		1794+	*	* SCAMMA - LOCATION WHERE SCACOM MAY BE MOVED IF ONE COMMA IS ALSO				*
		1795+	*	TO BE CONSIDERED A DELIMITER. MOVING SCACOF BACK INTO SCAMMA				*
		1796+	*	INDICATES THAT ONLY BLANKS SHOULD BE CONSIDERED DELIMITERS.				*
		1797+	*					*
		1798+	*	ATTRIBUTES				*
		1799+	*	RELOCATABLE AND RE-USABLE				*
		1800+	*					*
		1801+	*	CHARACTER CODE DEPENDENCY				*
		1802+	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR				*
		1803+	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.				*
		1804+	*					*
		1805+	*	NOTES				*
		1806+	*	ERROR PROCEDURES				*
		1807+	*	THE ONLY ERROR CONDITION DETECTED BY SCANIT IS THE CASE WHERE				*
		1808+	*	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 01/08/20 PAGE 8

```

1809+*      CALLING ROUTINE, @PSR WILL BE SET TO A LOW CONDITION, THE      *
1810+*      ERROR CODE IS SET IN $CAERR, AND MG WU BE POINTING TO THE      *
1811+*      CARRIAGE-RETURN CHARACTER.                                     *
1812+*                                                                 *
1813+*      REGISTER USAGE                                                 *
1814+*      REGISTER 2 (@XR) IS USED AS A POINTER ACROSS THE AREA BEING    *
1815+*      SCANNED FOR DELIMITERS.                                         *
1816+*                                                                 *
1817+*      SAVED/RESTORED AREAS                                           *
1818+*      UPON ENTRY TO SCANIT, REGISTER 8 (@ARR) IS SAVED AND USED AS   *
1819+*      THE RETURN ADDRESS.                                             *
1820+*                                                                 *
1821+*      MODIFICATION CONSIDERATIONS                                     *
1822+*      NONE                                                             *
1823+*                                                                 *
1824+*      REQUIRED MODULES                                                 *
1825+*      * @SYSEQ - COMMON SYSTEM EQUATES                               *
1826+*      * @FXDEQ - FIXED NUCLEUS ADDRESSES EQUATES                     *
1827+*                                                                 *
1828+*      OTHER                                                             *
1829+*      SCANIT IS INITIALIZED TO BYPASS BLANKS ONLY. IF SCACOM IS      *
1830+*      MOVED TO SCAMMA, ONE COMMA WILL BE SCANNED ALONG WITH BLANKS.    *
1831+*      THE INSTRUCTION TO DO THIS IS AS FOLLOWS:                      *
1832+*      MVI    SCAMMA,SCACOM                                             *
1833+*                                                                 *
1834+*      TO DROP THE COMMA FROM ITS DELIMITER STATUS, SCACOF SHOULD BE    *
1835+*      MOVED TO SCAMMA, USING THE FOLLOWING INSTRUCTION:                *
1836+*      MVI    SCAMMA,SCACOF                                             *
1837+*****
1839+*
1840+*      EQUATES USED IN THIS SUBROUTINE
1841+*
0001 1842+SCAINC EQU    1          TO INCREMENT POINTER
0001 1843+SCACOM EQU   @BNE        SWITCH TO ALLOW SCANNING COMMA
0087 1844+SCACOF EQU   @UCB        SWITCH TO SET OFF THE INDICATON
1845+*      * FOR SCANNING A COMMA
0D0E 1846+SCANIT EQU   *          ENTRY POINT TO THIS SUBROUTINE
0D0E 34 08 0D4A      1847+      ST    SCA500+@OP1,@ARR          SAVE RETURN ADDRESS
0D12 34 02 0D4C      1848+      ST    SCASVE,@XR              SAVE POINTER VALUE
0D16 3C 04 03CD      1849+      MVI    $CAERR,@E110           SET ERROR CODE
0D1A F2 87 03      1850+      J      SCA200                   GO TO PROCESS
0D1D E2 02 01      1851+SCA100 LA    SCAINC(,@XR),@XR          INCREMENT POINTER TO NEXT CHAR
0D20 BD 40 00      1852+SCA200 CLI    0(,@XR),@BLANK          IS THIS CHAR BLANK ?
0D23 C0 81 0D1D      1853+      BE     SCA100                  YES, FETCH NEXT ONE
0D27 BD 6B 00      1854+      CLI    0(,@XR),@COMMA           IS IT A COMMA ?
0D2A F2 87 10      1855+SCA250 JC    SCA400,@UCB              UCS TO RETURN -- OR NOP IF
1856+*      * SCAMMA IS ACTIVE AND CHAR
0D2D E2 02 01      1857+SCA300 LA    SCAINC(,@XR),@XR          INCREMENT POINTER TO NEXT CHAR
0D30 BD 40 00      1858+      CLI    0(,@XR),@BLANK          IS THIS CHAR A BLANK ?
0D33 C0 81 0D2D      1859+      BE     SCA300                  YES, FETCH NEXT ONE
0D37 BD 1F 00      1860+      CLI    0(,@XR),@EOS+1           IS THIS EOS ?
0D3A F2 82 0A      1861+      JL     SCA500                   IF NOT, SKIP ERROR ROUTINE
0D3D 34 02 0D4E      1862+SCA400 ST    SCACNT,@XR              SAVE NEW POINTER VALUE
0D41 0F 01 0D4E 0D4C 1863+      SLC    SCACNT(2),SCASVE        SET PSR TO EQUAL IF POINTER

```

[illegible]

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 10

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$\$\$	001	0C00	1539	
\$\$\$\$\$1	177	0DFF	1885	
\$\$\$\$L1	001	0D4F	1880	1883 1885
\$\$\$\$T1	001	0E00	1882	1885
\$\$\$CMD	001	0020	0841	
\$\$\$DAT	001	0040	0840	
\$\$\$EPL	001	0091	0837	
\$\$\$ERN	001	0080	0891	
\$\$\$FUN	001	0010	0842	
\$\$\$NLN	001	00A0	0887	
\$\$\$STD	001	0081	0836	
\$\$BNLN	001	0605	0817	0819
\$\$CDBS	001	08C0	0867	
\$\$CDND	001	0666	0826	
\$\$CDRD	001	0890	0865	0867
\$\$CKEY	001	0603	0815	
\$\$CKFF	001	0B3D	0847	
\$\$COFF	001	0B44	0846	
\$\$CSNS	001	209C	0876	
\$\$DATB	001	0BBF	0848	
\$\$EOSA	001	0AFE	0845	
\$\$ERSK	001	1C00	0886	
\$\$FITS	001	1D00	0894	
\$\$FLIB	001	06FF	0893	
\$\$ILEN	001	0601	0811	0813 0817
\$\$ILHD	001	0600	0809	0811
\$\$INLN	001	0607	0824	0826 0828
\$\$INND	001	06FA	0828	
\$\$KBDT	001	09E1	0835	0839
\$\$KBSN	001	09E2	0839	0844
\$\$KLD1	001	0600	0899	
\$\$KLD2	001	0700	0901	
\$\$KLD3	001	0C00	0903	1530
\$\$LPOS	001	09EB	0844	
\$\$PCNT	001	07E9	0860	
\$\$PLYN	001	2004	0874	
\$\$PRES	001	0890	0833	0835 0845 0846 0847 0848 0865
\$\$PRFL	001	2143	0878	
\$\$PRNT	001	0707	0854	0855 0859 0860
\$\$PRTN	001	0782	0855	
\$\$PSIO	001	07CE	0859	
\$\$PYCD	001	2200	0880	
\$\$PYMP	001	2000	0872	0874 0876 0878 0880
\$\$SLIB	001	1C00	0889	
\$\$TPCD	001	0606	0819	0824
\$\$UPAR	001	0602	0813	0815
\$\$WSPB	001	1E00	0892	
\$\$XIND	001	06FF	0890	0893
\$\$ZERO	001	0000	0406	0407 0409 0410 0411 0415 0872
\$ABORT	001	0010	0518	1659
\$BASIC	001	0080	0576	
\$BIGCD	001	0080	0652	
\$BLDPL	001	0579	0785	0787
\$BLNOE	001	0569	0775	
\$BLOAD	001	0522	0766	0768 0771 0784 0785
\$BLRTN	001	0550	0774	0775

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 11

\$BRSAV	001	03C5	0463	0464						
\$BSADR	001	0587	0790	0792						
\$BUFPT	001	03E3	0671	0672						
\$CABLD	001	04B4	0744	0745						
\$CAERK	001	0469	0721	0724	1639	1703	1720			
\$CAERR	001	03CD	0469	0471	1638*	1692*	1702*	1716*	1849*	
\$CAIPL	001	049D	0740	0742						
\$CALLI	001	0008	0661							
\$CARDI	001	0001	0432							
\$CARPL	001	04A1	0742	0744						
\$CIENT	001	0483	0731	0732						
\$CIEXT	001	0480	0730	0731						
\$CIMSK	001	0476	0727	0730	1632*	1664*				
\$CISUS	001	0496	0735	0740						
\$CLBFR	001	0010	0619	1665						
\$CMDKY	001	0008	0531							
\$CMODE	001	0002	0581							
\$CONFIG	001	03DD	0644	0654						
\$CRPOS	001	03E2	0670	0671						
\$CRTAD	001	044D	0709	0710						
\$CRTAV	001	0002	0525							
\$CRTDN	001	0002	0549							
\$CRTIN	001	03D3	0546	0553						
\$CRTNO	001	0004	0528							
\$CRTPU	001	0004	0550							
\$CRTSP	001	0008	0551							
\$CRTUP	001	0001	0548							
\$CRUSH	001	0080	0657							
\$CSDPL	001	050E	0756	0757						
\$C0001	001	0464	0713	0719						
\$DATE	001	043A	0694	0695						
\$DBGUF	001	03E0	0656	0665						
\$DBLOK	001	0001	0606							
\$DFDET	001	03E8	0677	0678						
\$DISKN	001	0025	0409							
\$DKERR	001	0008	0587							
\$DKSIZ	001	03D7	0631	0639	0680					
\$DK100	001	0001	0633							
\$DK200	001	0002	0634							
\$DK400	001	0004	0635							
\$DK600	001	0008	0636							
\$DK800	001	0010	0637							
\$DPLSV	001	0449	0705	0707						
\$DTNMB	001	0040	0452							
\$DTRDR	001	0040	0540							
\$ENDNU	001	0600	0799	0809	0833	0854	0890	0899	0901	0903
\$ERDPL	001	046F	0724	0726						
\$ERFIL	001	0040	0479							
\$ERHRD	001	0004	0611							
\$ERKEY	001	0080	0483							
\$ERLOG	001	0345	0414							
\$ERMAD	001	0472	0726	0727						
\$ERPND	001	0004	0584							
\$ERRCT	001	03CF	0485							
\$ERRPG	001	03CE	0473							
\$ERSFL	001	0035	0478							

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 12

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$ERSTK	001	0030	0476	
\$ER050	001	0363	0415	
\$ER1N2	001	0050	0481	
\$EXADR	001	0517	0759	0761
\$EXCMD	001	0001	0513	1663
\$EXFTR	001	043B	0695	0700
\$FCIND	001	0010	0591	
\$FDIND	001	0040	0598	
\$FEARR	001	0004	0407	
\$FEMAP	001	0588	0792	0793
\$FILIB	001	03DA	0642	0643
\$FITIN	001	0010	0567	
\$FUIND	001	0020	0596	
\$GUFIO	001	0583	0789	0790
\$GUFIR	001	0008	0441	
\$HISTE	001	042E	0692	0693
\$HIST1	001	0435	0693	0694
\$HRDER	001	0020	0537	
\$INDR1	001	03D4	0553	0579
\$INDR2	001	03D5	0579	0604
\$INDR3	001	03D6	0604	0631 1665*
\$INLNO	001	03CF	0471	0473 0485 0492
\$INRPT	001	0020	0449	
\$IOIND	001	03D2	0520	0546
\$IOPGS	001	0010	0660	
\$IOYES	001	0002	0435	
\$IPLDV	001	05FF	0796	0799
\$IRKEY	001	0020	0659	
\$KEYBD	001	03E1	0665	0670
\$KEYCD	001	03C3	0429	0463
\$KEYDT	001	0040	0573	
\$KE090	001	00DE	0410	
\$KE130	001	01D5	0411	
\$KYBSY	001	0010	0446	
\$LDRTN	001	0571	0784	
\$LEVEL	001	03DF	0654	0656
\$LIST	001	0002	0608	
\$LMRGN	001	03C1	0424	0426
\$LNPTR	001	0080	0543	
\$LOADB	001	054A	0768	
\$LOADR	001	051A	0761	0764
\$LPRIO	001	03E9	0678	
\$LPROS	001	03E5	0673	0675
\$LPRP3	001	03E4	0672	0673
\$MOUNT	001	0020	0622	
\$MPDWN	001	0001	0522	
\$NEXTB	001	03E6	0675	0676
\$NEXTL	001	03E7	0676	0677
\$NOENB	001	0008	0614	
\$NOLST	001	0004	0438	
\$NUCBS	001	03C0	0421	0422
\$NWRKF	001	0080	0627	
\$NWRKR	001	0040	0624	
\$PASWD	001	042D	0691	0692
\$PAUSD	001	04BA	0745	0747
\$PAUSE	001	0002	0515	1636

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 13

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$PGMDT	001	0020	0570	
\$PGMST	001	0010	0534	
\$PKERT	001	0419	0689	0691
\$PLST1	001	0454	0710	0711
\$PLST2	001	045B	0711	0712
\$PLST3	001	0462	0712	0713
\$PRDEV	001	044B	0707	0709
\$PRESN	001	0002	0558	
\$PROCI	001	0001	0555	
\$PRPOS	001	03C2	0426	0429
\$PSDBR	001	04FA	0750	
\$PSDXR	001	04F2	0749	0750
\$PSTEP	001	0004	0516	
\$PSTMT	001	0008	0517	
\$PTCH1	001	03F5	0680	0684
\$READY	001	0080	0600	
\$REORD	001	0040	0658	
\$RLOAD	001	051E	0764	0766
\$RMGRN	001	03C0	0422	0424
\$RSTR	001	04D6	0747	0749 0751 0756 1666
\$RUNIT	001	0001	0494	1674 1685 1698
\$SFAID	001	050D	0752	
\$SPRNT	001	0465	0719	0721
\$SRTRN	001	04FE	0751	0752
\$STEPT	001	0002	0495	1675 1684 1698
\$SWPCR	001	0511	0757	0759
\$TABLN	001	03CB	0466	0469
\$TFLOW	001	0008	0501	1696
\$TRACE	001	0004	0496	1674 1684 1699
\$TRALL	001	0010	0502	1696
\$TROVR	001	054E	0771	0774
\$TRUNK	001	0080	0454	
\$TRVAR	001	0020	0503	1696
\$UNMSK	001	048D	0732	0735
\$USRDR	001	03DC	0643	0644
\$VMDEF	001	0080	0507	
\$VOLF1	001	03FE	0686	0687
\$VOLF2	001	040E	0688	
\$VOLID	001	03F6	0684	0685 0689
\$VOLR1	001	03F6	0685	0686
\$VOLR2	001	0406	0687	0688
\$WAITF	001	057F	0787	0789
\$WFDEF	001	0040	0701	
\$WFLOK	001	0008	0564	
\$WFNME	001	0443	0700	0705
\$WSIND	001	0004	0561	
\$XIND1	001	03D0	0492	0511 1674* 1675* 1684* 1685* 1696 1698* 1699*
\$XIND2	001	03D1	0511	0520 1636 1658* 1659* 1663*
\$XIND3	001	03D8	0639	0642
\$XPREC	001	0040	0504	
\$XRSAB	001	03C7	0464	0466 1644
\$ZTRAD	001	05A2	0793	
\$12K	001	0004	0648	
\$16CKY	001	0008	0650	
\$16K	001	0002	0647	
\$22IMP	001	0001	0645	

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 14

SYMBOL	LEN	VALUE	DEFN	REFERENCES
###\$KGO	001	0C00	1536	1538
##\$@KGO	001	0002	1537	
##\$KGOS	001	0180	1535	
#KGOS	001	0C07	1542	
#KGOSL	001	0000	0001	
@@E001	001	0000	1441	1443
@@E003	001	0001	1443	1445
@@E004	001	0002	1445	1447
@@E005	001	0003	1447	1449
@@E006	001	0004	1449	1451
@@E007	001	0005	1451	1453
@@E008	001	0006	1453	1455
@@E009	001	0007	1455	1457
@@E010	001	0008	1457	1459
@@E011	001	0009	1459	1461
@@E012	001	000A	1461	1463
@@E013	001	000B	1463	1465
@@E014	001	000C	1465	1467
@@E015	001	000D	1467	1469
@@E016	001	000E	1469	1471
@@E017	001	000F	1471	1473
@@E018	001	0010	1473	1475
@@E019	001	0011	1475	1477
@@E020	001	0012	1477	1479
@@E021	001	0013	1479	1481
@@E023	001	0014	1481	1483
@@E024	001	0015	1483	1485
@@E025	001	0016	1485	1487
@@E026	001	0017	1487	1489
@@E027	001	0018	1489	1491
@@E028	001	0019	1491	1493
@@E029	001	001A	1493	1495
@@E030	001	001B	1495	1497
@@E031	001	001C	1497	1499
@@E032	001	001D	1499	1501
@@E035	001	001E	1501	1503
@@E036	001	001F	1503	1505
@@E037	001	0020	1505	1507
@@E038	001	0021	1507	1509
@@E039	001	0022	1509	1511
@@E040	001	0023	1511	1513
@@E041	001	0024	1513	1515
@@E042	001	0025	1515	1517
@@E043	001	0026	1517	1519
@@E044	001	0027	1519	1521
@@E045	001	0028	1521	1523
@@E046	001	0029	1523	1525
@@E060	001	002A	1525	1527
@@E080	001	002B	1527	
@@E100	001	0000	0913	0915
@@E101	001	0001	0915	0917
@@E102	001	0002	0917	0919
@@E103	001	0003	0919	0921
@@E110	001	0004	0921	0923 1849
@@E112	001	0005	0923	0925
@@E113	001	0006	0925	0927



## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   01/08/20   PAGE   15

@@E114	001	0007	0927	0929	
@@E115	001	0008	0929	0931	
@@E116	001	0009	0931	0933	
@@E117	001	000A	0933	0935	
@@E120	001	000B	0935	0937	
@@E122	001	000C	0937	0939	
@@E123	001	000D	0939	0941	
@@E124	001	000E	0941	0943	
@@E129	001	000F	0943	0945	
@@E130	001	0010	0945	0947	
@@E131	001	0011	0947	0949	1692   1716
@@E133	001	0012	0949	0951	
@@E134	001	0013	0951	0953	
@@E135	001	0014	0953	0955	
@@E136	001	0015	0955	0957	
@@E137	001	0016	0957	0959	
@@E138	001	0017	0959	0961	
@@E139	001	0018	0961	0963	
@@E142	001	0019	0963	0965	
@@E143	001	001A	0965	0967	
@@E150	001	001B	0967	0969	
@@E151	001	001C	0969	0971	
@@E160	001	001D	0971	0973	
@@E162	001	001E	0973	0975	
@@E163	001	001F	0975	0977	
@@E164	001	0020	0977	0979	
@@E200	001	0021	0979	0981	
@@E205	001	0022	0981	0983	
@@E210	001	0023	0983	0985	
@@E211	001	0024	0985	0987	
@@E212	001	0025	0987	0989	
@@E213	001	0026	0989	0991	
@@E215	001	0027	0991	0993	
@@E216	001	0028	0993	0995	
@@E217	001	0029	0995	0997	
@@E220	001	002A	0997	0999	
@@E221	001	002B	0999	1001	
@@E222	001	002C	1001	1003	
@@E223	001	002D	1003	1005	
@@E225	001	002E	1005	1007	1638
@@E226	001	002F	1007	1009	
@@E227	001	0030	1009	1011	
@@E228	001	0031	1011	1013	
@@E229	001	0032	1013	1015	
@@E230	001	0033	1015	1017	
@@E232	001	0034	1017	1019	
@@E234	001	0035	1019	1021	
@@E237	001	0036	1021	1023	1702
@@E240	001	0037	1023	1025	
@@E241	001	0038	1025	1027	
@@E242	001	0039	1027	1029	
@@E248	001	003A	1029	1031	
@@E249	001	003B	1031	1033	
@@E250	001	003C	1033	1035	
@@E251	001	003D	1035	1037	
@@E252	001	003E	1037	1039	



## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 16

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E253	001	003F	1039	1041
@@E254	001	0040	1041	1043
@@E255	001	0041	1043	1045
@@E256	001	0042	1045	1047
@@E300	001	0043	1047	1049
@@E301	001	0044	1049	1051
@@E302	001	0045	1051	1053
@@E303	001	0046	1053	1055
@@E304	001	0047	1055	1057
@@E305	001	0048	1057	1059
@@E308	001	0049	1059	1061
@@E310	001	004A	1061	1063
@@E315	001	004B	1063	1065
@@E316	001	004C	1065	1067
@@E320	001	004D	1067	1069
@@E325	001	004E	1069	1071
@@E330	001	004F	1071	1073
@@E335	001	0050	1073	1075
@@E338	001	0051	1075	1077
@@E340	001	0052	1077	1079
@@E350	001	0053	1079	1081
@@E351	001	0054	1081	1083
@@E352	001	0055	1083	1085
@@E360	001	0056	1085	1087
@@E361	001	0057	1087	1089
@@E362	001	0058	1089	1091
@@E371	001	0059	1091	1093
@@E380	001	005A	1093	1095
@@E390	001	005B	1095	1097
@@E400	001	005C	1097	1099
@@E410	001	005D	1099	1101
@@E415	001	005E	1101	1103
@@E417	001	005F	1103	1105
@@E420	001	0060	1105	1107
@@E430	001	0061	1107	1109
@@E432	001	0062	1109	1111
@@E433	001	0063	1111	1113
@@E450	001	0064	1113	1115
@@E451	001	0065	1115	1117
@@E460	001	0066	1117	1119
@@E461	001	0067	1119	1121
@@E464	001	0068	1121	1123
@@E465	001	0069	1123	1125
@@E466	001	006A	1125	1127
@@E467	001	006B	1127	1129
@@E469	001	006C	1129	1131
@@E470	001	006D	1131	1133
@@E471	001	006E	1133	1135
@@E473	001	006F	1135	1137
@@E474	001	0070	1137	1139
@@E475	001	0071	1139	1141
@@E476	001	0072	1141	1143
@@E477	001	0073	1143	1145
@@E478	001	0074	1145	1147
@@E479	001	0075	1147	1149
@@E480	001	0076	1149	1151

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 17

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E481	001	0077	1151	1153
@@E482	001	0078	1153	1155
@@E483	001	0079	1155	1157
@@E484	001	007A	1157	1159
@@E485	001	007B	1159	1161
@@E486	001	007C	1161	1163
@@E487	001	007D	1163	1165
@@E488	001	007E	1165	1167
@@E489	001	007F	1167	1169
@@E490	001	0080	1169	1171
@@E491	001	0081	1171	1173
@@E492	001	0082	1173	1175
@@E493	001	0083	1175	1177
@@E494	001	0084	1177	1179
@@E495	001	0085	1179	1181
@@E496	001	0086	1181	1183
@@E497	001	0087	1183	1185
@@E498	001	0088	1185	1187
@@E500	001	0089	1187	1189
@@E501	001	008A	1189	1191
@@E530	001	008B	1191	1193
@@E531	001	008C	1193	1195
@@E535	001	008D	1195	1197
@@E540	001	008E	1197	1199
@@E541	001	008F	1199	1201
@@E542	001	0090	1201	1203
@@E543	001	0091	1203	1205
@@E544	001	0092	1205	1207
@@E545	001	0093	1207	1209
@@E546	001	0094	1209	1211
@@E547	001	0095	1211	1213
@@E548	001	FFFF	1417	
@@E549	001	0096	1213	1215
@@E550	001	0097	1215	1217
@@E551	001	0098	1217	1219
@@E552	001	0099	1219	1221
@@E553	001	009A	1221	1223
@@E554	001	009B	1223	1225
@@E555	001	009C	1225	1227
@@E556	001	009D	1227	1229
@@E558	001	009E	1229	1231
@@E570	001	009F	1231	1233
@@E571	001	00A0	1233	1235
@@E572	001	00A1	1235	1237
@@E573	001	00A2	1237	1239
@@E574	001	00A3	1239	1241
@@E575	001	FFFF	1419	
@@E578	001	00A4	1241	1243
@@E579	001	FFFF	1421	
@@E580	001	FFFF	1423	
@@E585	001	00A5	1243	1245
@@E595	001	FFFF	1425	
@@E597	001	FFFF	1427	
@@E598	001	FFFF	1429	
@@E600	001	00A6	1245	1247
@@E601	001	00A7	1247	1249

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 18

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E602	001	00A8	1249	1251
@@E603	001	00A9	1251	1253
@@E604	001	00AA	1253	1255
@@E606	001	00AB	1255	1257
@@E607	001	00AC	1257	1259
@@E608	001	00AD	1259	1261
@@E609	001	00AE	1261	1263
@@E610	001	00AF	1263	1265 1273
@@E611	001	00B0	1265	1267
@@E612	001	00B1	1267	1269
@@E613	001	00B2	1269	1271
@@E614	001	00B3	1271	
@@E700	001	00B0	1273	1275
@@E701	001	00B1	1275	1277
@@E710	001	00B2	1277	1279
@@E712	001	00B3	1279	1281
@@E713	001	00B4	1281	1283
@@E714	001	00B5	1283	1285
@@E715	001	00B6	1285	1287
@@E716	001	00B7	1287	1289
@@E717	001	00B8	1289	1291
@@E718	001	00B9	1291	1293
@@E720	001	00BA	1293	1295
@@E721	001	00BB	1295	1297
@@E723	001	00BC	1297	1299
@@E724	001	00BD	1299	1301
@@E725	001	00BE	1301	1303
@@E726	001	00BF	1303	1305
@@E727	001	00C0	1305	1307
@@E728	001	00C1	1307	1309
@@E729	001	00C2	1309	1311
@@E730	001	00C3	1311	1313
@@E732	001	00C4	1313	1315
@@E752	001	00C5	1315	1317
@@E753	001	00C6	1317	1319
@@E754	001	00C7	1319	1321
@@E755	001	00C8	1321	1323
@@E756	001	00C9	1323	1325
@@E757	001	00CA	1325	1327
@@E758	001	00CB	1327	1329
@@E759	001	00CC	1329	1331
@@E760	001	00CD	1331	1333
@@E761	001	00CE	1333	1335
@@E762	001	00CF	1335	1337
@@E763	001	00D0	1337	1339
@@E764	001	00D1	1339	1341
@@E765	001	00D2	1341	1343
@@E766	001	00D3	1343	1345
@@E767	001	00D4	1345	1347
@@E768	001	00D5	1347	1349
@@E769	001	00D6	1349	1351
@@E770	001	00D7	1351	1353
@@E771	001	00D8	1353	1355
@@E772	001	00D9	1355	1357
@@E773	001	00DA	1357	1359
@@E774	001	00DB	1359	1361

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 19

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E775	001	00DC	1361	1363
@@E776	001	00DD	1363	1365
@@E777	001	00DE	1365	1367
@@E778	001	00DF	1367	1369
@@E779	001	00E0	1369	1371
@@E780	001	00E1	1371	1373
@@E781	001	00E2	1373	1375
@@E782	001	00E3	1375	1377
@@E783	001	00E4	1377	1379
@@E784	001	00E5	1379	1381
@@E785	001	00E6	1381	1383
@@E786	001	00E7	1383	1385
@@E790	001	00E8	1385	1387
@@E791	001	00E9	1387	1389
@@E792	001	00EA	1389	1391
@@E793	001	00EB	1391	1393
@@E794	001	00EC	1393	1395
@@E795	001	00ED	1395	1397
@@E796	001	00EE	1397	1399
@@E797	001	00EF	1399	1401
@@E798	001	00F0	1401	1403
@@E800	001	FFFF	1431	
@@E801	001	FFFF	1433	
@@E802	001	FFFF	1435	
@@E803	001	FFFF	1437	
@@E804	001	FFFF	1439	
@@E900	001	00F1	1403	1405
@@E901	001	00F2	1405	1407
@@E902	001	00F3	1407	1409
@@E903	001	00F4	1409	1411
@@E905	001	00F5	1411	1413
@@E906	001	00F6	1413	1415
@@E910	001	00F7	1415	
@ALTFL	001	0001	0249	
@ARR	001	0008	0016	1710 1847
@ASIGN	001	007C	0071	
@ASTER	001	005C	0069	
@BCRDL	001	0050	0088	
@BE	001	0081	0043	
@BF	001	0090	0052	
@BH	001	0084	0041	
@BKSPC	001	0010	0345	
@BL	001	0082	0042	
@BLANK	001	0040	0065	1852 1858
@BM	001	0082	0054	
@BNE	001	0001	0046	1843
@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	
@BOZ	001	0088	0047	
@BP	001	0084	0053	

CROSS REFERENCE										
SYMBOL	LEN	VALUE	DEFN	REFERENCES	VER 15, MOD 00	01/08/20	PAGE	20		
@BR	001	0001	0013	1630*	1631	1654	1670	1680	1690	1701
@BT	001	0010	0051							
@BZ	001	0081	0055							
@BZ37B	001	00F2	0358							
@B1	001	0001	0063							
@CADDR	001	0002	0141							
@CARDL	001	0060	0087	0826						
@CC37B	001	0000	0354							
@CD37B	001	00F0	0372							
@CHARA	001	00C1	0072							
@CHARF	001	00C6	0073							
@CHARR	001	00D9	0074							
@CHARZ	001	00E9	0075							
@CKY01	001	0001	0307							
@CKY02	001	0002	0308							
@CKY03	001	0003	0309							
@CKY04	001	0004	0310							
@CKY05	001	0005	0311							
@CKY06	001	0006	0312							
@CKY07	001	0007	0313							
@CKY08	001	0008	0314							
@CKY09	001	0009	0315							
@CKY10	001	000A	0316							
@CKY11	001	000B	0317							
@CKY12	001	000C	0318							
@CKY13	001	000D	0319							
@CKY14	001	000E	0320							
@CKY15	001	000F	0321							
@CKY16	001	0010	0322							
@CLOFF	001	0010	0094							
@CLON	001	0011	0093							
@CMLON	001	0001	0325							
@CMOFF	001	0000	0324							
@COMMA	001	006B	0066	1854						
@CPLUS	001	004E	0079							
@CP37B	001	0004	0385							
@CRERR	001	0090	0340							
@CRPRY	001	0004	0344							
@CRTDS	001	0092	0337							
@CRTQ	001	0090	0339							
@CURSR	001	0040	0341							
@DADDR	001	0002	0139							
@DBFR1	001	0004	0128							
@DBFR2	001	0005	0129							
@DBUSY	001	0002	0243							
@DCALK	001	0001	0081							
@DCBCY	001	0009	0114							
@DCBT1	001	0050	0116							
@DCFLN	001	0004	0227							
@DCNT	001	0003	0127							
@DCRID	001	0001	0241							
@DCST1	001	0040	0115							
@DCTRL	001	0000	0124							
@DCTRW	001	0000	0240							

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 01/08/20 PAGE 21

@DCYMV	001	0001	0228	
@DD2	001	0003	0030	
@DEFLG	001	0002	0250	
@DERCE	001	0020	0280	
@DERD2	001	0008	0273	
@DEREQ	001	0010	0272	
@DERIN	001	0040	0270	
@DERMA	001	0020	0271	
@DERNR	001	0004	0274	
@DERR	001	0000	0244	
@DERSC	001	0001	0276	
@DERTC	001	0002	0275	
@DFCR	001	0006	0230	
@DFDR	001	0004	0231	
@DGET	001	0001	0133	
@DHARD	001	0000	0258	
@DLNCT	001	000F	0343	
@DLNLG	001	0040	0342	
@DOLAR	001	005B	0068	
@DOP2	001	0004	0028	
@DPLNG	001	0006	0131	
@DPOS	001	0000	0132	
@DPUT	001	0002	0134	
@DREAD	001	0001	0234	
@DSAD	001	0002	0126	
@DSBCY	001	0004	0105	
@DSBSY	001	0092	0338	
@DSCS1	001	0000	0106	
@DSEEK	001	0000	0233	
@DSIVF	001	0003	0137	
@DSPIN	001	0002	0130	
@DTRSZ	001	0018	0085	
@DUNSF	001	0080	0269	
@DVBCY	001	0007	0107	
@DVERY	001	0003	0239	
@DVRFY	001	0031	0135	
@DVST1	001	0002	0245	
@DVST2	001	0003	0246	
@DWAIT	001	00FF	0136	
@DWBCY	001	0005	0102	
@DWBIT	001	0002	0235	
@DWSIZ	001	00C0	0104	
@DWTB1	001	0003	0103	
@DZERO	001	00F0	0064	
@D1	001	0002	0026	
@EOF	001	001C	0077	
@EOFTC	001	0075	0160	
@EOS	001	001E	0076	1648 1661 1713 1860
@ER37B	001	00F0	0359	
@FDDBC	001	0000	0193	
@FDE1	001	000C	0198	
@FDFNA	001	000B	0196	
@FDHLN	001	0002	0206	
@FDLNC	001	0002	0191	
@FDNSC	001	0003	0208	
@FDSD	001	0000	0204	

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 22

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@FLACE	001	0009	0195	
@FLDBC	001	0001	0194	
@FLDIN	001	0012	0332	
@FLENT	001	0004	0199	
@FLFNA	001	0002	0197	
@FLHLN	001	0002	0207	
@FLLNC	001	0002	0192	
@FLNSC	001	0001	0209	
@FLSD	001	0001	0205	
@HDRLN	001	0007	0092	0854
@HSTAD	001	0009	0256	
@HSTEN	001	0007	0255	
@HSTPE	001	0006	0254	
@HSTQR	001	0001	0252	
@HSTSN	001	0005	0253	
@HSTVI	001	000F	0257	
@IAR	001	0010	0017	1715*
@ID37B	001	0040	0395	
@INDEX	001	0001	0154	0155
@INST3	001	0003	0032	
@INST4	001	0004	0033	
@INST5	001	0005	0034	
@INST6	001	0006	0035	
@IP37B	001	00C0	0394	
@I1IAR	001	00C0	0020	
@KCMDK	001	0020	0306	
@KELOK	001	001B	0305	
@KENAB	001	001E	0303	
@KEXIT	001	001F	0304	
@KEYBD	001	0010	0323	
@KFUNK	001	0010	0326	
@KHARD	001	0011	0331	
@KLEAR	001	000D	0327	
@LINSZ	001	00F4	0084	0828
@LO37B	001	00F0	0363	
@MAPEN	001	0005	0089	
@MINCR	001	2000	0083	
@MINUS	001	0060	0080	
@NOP	001	0080	0040	1632 1664
@NORFL	001	0000	0251	
@NTRDY	001	00A0	0387	
@NUMBR	001	007B	0070	
@OPD2	001	0004	0029	
@OP1	001	0003	0027	1646* 1847*
@OP2	001	0005	0031	
@OVRUN	001	0004	0281	
@PBUSY	001	00E2	0293	
@PCAR	001	00E6	0290	
@PCNT	001	0003	0225	
@PCTRL	001	0000	0147	
@PCYL	001	0001	0223	
@PC37B	001	00F2	0379	
@PDAR	001	00E4	0289	
@PDATA	001	0003	0149	
@PD37B	001	0080	0393	
@PERR	001	00E0	0296	

## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 23

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@PFLAG	001	0000	0222	
@PFORM	001	00E1	0294	
@PGCSZ	001	0020	0082	0083
@PLITE	001	00E2	0295	
@PLNGH	001	0004	0286	
@PMGCK	001	0020	0297	
@PN37B	001	00F0	0378	
@PPLNG	001	0004	0146	
@PRCNT	001	0001	0148	
@PRETR	001	00C0	0152	
@PRINT	001	0040	0150	0152
@PRITY	001	0080	0330	
@PSAD	001	0002	0224	
@PSIOQ	001	00E0	0292	
@PSIOR	001	0000	0291	
@PSNSQ	001	00E2	0298	
@PSR	001	0004	0015	
@PWAIT	001	00FF	0156	
@P1IAR	001	0020	0018	
@P2IAR	001	0040	0019	
@Q	001	0001	0024	1866
@RD37B	001	00F1	0373	
@REGL	001	0002	0012	
@RETRN	001	0080	0151	0152
@RLDWN	001	004F	0157	
@RTCNT	001	0003	0288	
@RTRNC	001	0080	0159	
@RT37B	001	0005	0386	
@SBLNL	001	0002	0182	
@SCTSZ	001	0100	0099	
@SDFLN	001	0007	0090	
@SDF0	001	0000	0164	
@SDF1	001	0001	0165	
@SDF2	001	0002	0166	
@SDF3	001	0003	0167	
@SDLN	001	0005	0168	
@SECCY	001	0030	0086	
@SIST	001	0001	0179	
@SKCTL	001	0000	0238	
@SLASH	001	0061	0067	
@SLAST	001	0002	0181	
@SMIDL	001	0003	0180	
@SNSB0	001	0000	0262	
@SNSB1	001	0001	0263	
@SNSB2	001	0002	0264	
@SNSB3	001	0003	0265	
@SNULL	001	0080	0171	
@SN37B	001	00F2	0367	
@SONLY	001	0000	0178	
@SPINA	001	00A0	0247	
@SPINB	001	00B0	0248	
@STEXT	001	0007	0170	
@STYPE	001	0006	0169	
@SYCNT	001	0002	0287	
@TBCNT	001	0000	0158	
@TBLEF	001	0010	0153	0155



## CROSS REFERENCE

VER 15, MOD 00 01/08/20 PAGE 24

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@TBLIX	001	0011	0155	
@TJ37B	001	0040	0384	
@TYPAM	001	0002	0329	
@TYPO	001	001C	0328	
@UCB	001	0087	0039	1844 1855
@UPARW	001	005A	0078	
@VADDR	001	0002	0140	
@VENTA	001	0056	0112	
@VMDDV	001	00FE	0113	
@VMFD1	001	0000	0108	
@VMFD2	001	0001	0109	
@VMRS3	001	0002	0111	
@VMTRL	001	0001	0110	
@VOLID	001	0006	0091	
@VQ	001	0001	0025	
@WA37B	001	00FF	0392	
@WSFIT	001	0500	0100	
@WSTBL	001	0503	0101	
@XR	001	0002	0014	1644* 1646 1648 1654 1656* 1661 1670 1672* 1680 1682* 1690 1694* 1701* 1713 1719* 1848 1851 1851* 1852 1854 1857 1857* 1858 1860 1862
@ZERO	001	0000	0062	1658 1717
@4K	001	0010	0346	
KGOABT	005	0CF7	1725	1654
KGOEND	001	0D0E	1749	
KGOEQ0	001	0CF3	1724	1736 1745
KGOEQ1	001	0CF8	1726	1736 1737 1745 1746
KGOEQ2	001	0CFC	1728	1737 1738 1746 1747
KGOEQ3	001	0D01	1730	1738 1739 1747 1748
KGOEQ4	001	0D04	1732	1739 1748
KGOOL0	001	0005	1745	1654 1654
KGOOL1	001	0004	1746	1670 1670
KGOOL2	001	0005	1747	1690 1690
KGOOL3	001	0003	1748	1680 1680
KGOOP0	002	0D05	1736	1656
KGOOP1	002	0D07	1737	1672
KGOOP2	002	0D09	1738	1694
KGOOP3	002	0D0B	1739	1682
KGOOP4	002	0D0D	1741	1710* 1715
KGORUN	003	0D03	1731	1680
KGOSTP	004	0CFB	1727	1670
KGOTRC	005	0D00	1729	1690
KGO100	004	0C0F	1636	
KGO110	004	0C1E	1644	1630 1631 1637
KGO120	004	0C34	1654	1647
KGO130	003	0C4E	1661	1655
KGO135	004	0C54	1663	1676 1686 1700
KGO136	004	0C5C	1665	1660
KGO137	004	0C60	1666	
KGO140	004	0C64	1670	1662
KGO150	004	0C7F	1680	1671
KGO155	004	0C8E	1684	
KGO160	004	0C9A	1690	1681
KGO170	004	0CA8	1694	1691
KGO175	003	0CC3	1701	1697
KGO180	004	0CCE	1710	1657 1673 1683 1695

[illegible]

START ADDRESS	CATEGORY	NAME AND ENTRY	CODE LENGTH HEXADECIMAL DECIMAL
---------------	----------	----------------	------------------------------------

0C00	0	#KGOSL	0E00 3584
------	---	--------	-----------

OL100 I THE TOTAL CORE USED BY #KGOSL IS 3584 DECIMAL.  
 OL101 I THE START CONTROL ADDRESS OF THIS MODULE IS 0C00.  
 OL104 I TOTAL NUMBER OF LIBRARY SECTORS REQUIRED IS 15  
 NAME-#KGOSL,PACK-R1R1R1,UNIT-R1,RETAIN-P,LIBRARY-O  
 CTION AT RUN-TIME. \*

3249	*	MATRIX FUNCTION BUCKET - 3 BYTES (B\$MFBK), FOR THE EXTERNAL	*
3250	*	CORE-RESIDENT BUCKET, USED TO ACCUMULATE MATRIX EXPRESSION	*
3251	*	FUNCTION CHARACTERS.	*
3252	*		*
3253	*	*ATTRIBUTES	*
3254	*	BMMATA IS NATURALLY RELOCATABLE AND REUSABLE.	*
3255	*		*
3256	*	*CHARACTER CODE DEPENDENCY	*
3257	*	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA-	*
3258	*	TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE	*
3259	*	USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT	*
3260	*	REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT	*
3261	*	IN A CORRECT MODULE FOR THE NEW DEFINITIONS.	*
3262	*		*
3263	*	*NOTES	*
3264	*	ERROR PROCEDURES	*
3265	*	N/A	*
3266	*		*
3267	*	REGISTER USAGE	*
3268	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
3269	*		*
3270	*	SAVED/RESTORED AREAS	*
3271	*	N/A	*
3272	*		*
3273	*	MODIFICATION CONSIDERATIONS	*
3274	*	BMMATA RESIDES ON TWO SECTORS AND IS CO-RESIDENT ON THE	1-4*
3275	*	SECOND SECTOR WITH BPREAD. ANY MODIFICATIONS MUST MAINTAIN	1-4*
3276	*	LINKAGE BETWEEN THE TWO SECTORS, CONSIDER ANY CHANGE IN THE	1-4*
3277	*	ENTRY ADDRESS OF BPREAD, AND REALIZE THE LIMITATION OF THE	1-4*
3278	*	SECTOR BOUNDARY UPON SIZE.	1-4*
3279	*		*
3280	*	REQUIRED MODULES	*
3281	*	@SYSEQ - COMMON SYSTEM EQUATES.	*
3282	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
3283	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
3284	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
3285	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
3286	*	@ERMEQ - ERROR MESSAGE EQUATES.	*
3287	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
3288	*	\$B\$EQU - COMPILER FIXED EQUATES.	*
3289	*	\$B@EQU - COMPILER SYSTEM EQUATES	*
3290	*		*
3291	*	OTHER	*
3292	*	BMMATA IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
3293	*	*****	*

0A00

3295	ORG	*,256,0
0A00 3296	USING	*,@BR

BEGIN AT CORE PAGE BOUNDARY
DEFINE BASE ADDR FOR CORE PAGE

3297 \*  
3298 \* ENTER BMMATA - MAT ASSIGNMENT STATEMENT ROUTINE  
3299 \*

0A00 3300 BMMATA EQU \*

## S/3 BASIC COMPILER -MAT- ASSIGNMENT STMT RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 17

```

0A00 74 01 F4      3301      ST      BMMCA2(,@BR),@BR
                   3302 *
                   3303 * SET MATRIX PROCESSING ROUTINE NOT TO GENERATE PMC AND ADVANCE POINTER
                   3304 * TO REFERENCE CHAR BEFORE 1ST MAT REFERENCE
                   3305 *
0A03 3C 02 0873    3306      MVI      B$NUMC,B@LMAT-1      SET GET TO SKIP TO 'T' IN MAT
0A07 C0 87 0867    3307      B        B$GETC              LINK TO ADVANCE POINTER
0A0B 3B 07 1981    3308      SBF      B$MPSW,B$MPMK        SET PUT SWITCH OFF
0A0F 3C 00 0A39    3309      MVI      B$PERC,@ZERO          INITIALIZE ERROR CODE TO ZERO
0A13 C0 87 18F3    3310      B        B$MATR              LINK TO PROCESS MAT REFERENCE
0A17 C0 87 0867    3311      B        B$GETC              LINK TO GET NEXT CHAR
0A1B 3A 07 1981    3312      SBN      B$MPSW,B$MPMK        SET PUT SWITCH ON
                   3313 *
                   3314 * TEST CHAR FOR INDICATION OF MAT MULTIPLICATION BY A SCALAR VALUE
                   3315 *
0A1F BD 4D 00      3316      CLI      B@CHAR(,@XR),B@LPAR      IF SCALAR MULTIPLICATION
0A22 F2 81 9C      3317      JE       BMM060                * GO PROCESS EXPRESSION
                   3318 *
                   3319 * SET UP FUNCTION SAVE BUCKET FOR COMPARISON
                   3320 *
0A25 2C 00 1B8F 00 3321      MVC      B$MFBK+BMMBK0,B@CHAR(1,@XR)  MOVE CHAR TO 1ST BUCKET BYT,.
0A2A C0 87 0867    3322      B        B$GETC              LINK TO GET NEXT CHAP
0A2E BD 1E 00      3323      CLI      B@CHAR(,@XR),B@EOST        IF CHAR IS NOT AN EOS
0A31 F2 01 09      3324      JNE      BMM005                * GO SET 2ND CHAR IN BUCKET
0A34 7C 6F F2      3325      MVI      BMMPB(,@BR),BMM160-BMMAT2  SET BR ADDR TO 4TH ENTRY PT
0A37 7C 6F EE      3326      MVI      BMM095+@D1(,@BR),BMM160-BMMAT2  SET RR ADDR TO 4TH ENT PT
0A3A F2 87 8A      3327      J        BMM070                GO CALL SECOND SEGMENT
0A3D 2C 00 1B90 00 3328 BMM005 MVC      B$MFBK+BMMBK1,B@CHAR(1,@XR)  MOVE CHAR TO 2ND BUCKET 'ME
0A42 C0 87 0867    3329      B        B$GETC              LINK TO GET NEXT CHAR
N04 0A46 00 00 0000 00 3330      MVC      B$MFCK+BMMBK2,B@CHAR(1,@XR)  MOVE CHAR TO 3RD BUCKET BYTE
0A4B C0 87 0867    3331      B        B$GETC              LINK TO GET NEXT CHAR
                   3332 *
                   3333 * SET POINTER TO 2ND BUCKET BYTE AND TEST FOC CHAR BEING '.', '-' OR 'A'
                   3334 *
0A4F C2 02 1B90    3335      LA       B$MFBK+BMMBK1,@XR        SET POINTER TO 2ND CHAR OF FUNC
0A53 BD 4E 00      3336      CLI      B@CHAR(,@XR),B@PLUS        IF CHAR IS A
0A56 F2 81 0C      3337      JE       BMM010                * GO SET AN!) CALL 2ND SEGMENT
0A59 BD 60 00      3338      CLI      B@CHAR(,@XR),B@MINS        IF CHAR LI A '-'
0A5C F2 81 06      3339      JE       BMM010                * GO SET ALD CALL 2ND SEGMENT
0A5F BD 5C 00      3340      CLI      B@CHAR(,@XR),B@MULT        IF CHAR NOT
0A62 F2 01 09      3341      JNE      BMM020                * GO SET FUNC TYPE
                   3342 *
                   3343 * SET SECOND SEGMENT BRANCH ADDRESS FOR MAIN ENTRY POINT
                   3344 *
0A65 7C 00 F2      3345 BMM010 MVI      BMMPB(,@BR),BMM100-BMMAT2  SET BR ADDR TO MAIN ENTRY PT
0A68 7C 00 EE      3346      MVI      BMM095+@D1(,@BR),BMM100-BMMAT2  SET BR ADDR TO MAIN ENT PT
0A6B F2 87 59      3347      J        BMM070                GO CALL SECOND SEGMENT
                   3348 *
                   3349 * SET BRANCH ADDRESS IN CALLING SEQUENCE FOR SEG-2 SECONDARY ENTRY PT
                   3350 *
0A6E 7C 2C F2      3351 BMM020 MVI      BMMPB(,@BR),BMM110-BMMAT2  SET BR ADDR TO MAIN ENTRY PT
0A71 7C 2C EE      3352      MVI      BMM095+@D1(,@BR),BMM110-BMMAT2  SET 3R ADDR TO 2ND ENT PT
                   3353 *
                   3354 * TEST DELIMITEP FOR BEING A STATEMENT TERMINATOR
                   3355 *
0A74 35 02 0878    3356      L        B$GPTR,@XR                RESTORE TEXT POINTER

```

## S/3 BASIC COMPILER -MAT- ASSIGNMENT STMT RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 18

0A78	BD 1E 00	3357	CLI	B@CHAR(,@XR),B@EOST	IF DELIMITER IS NOT AN EOS
0A7B	F2 01 07	3358	JNE	BMM030	* GO PROCESS FUNC SUBSCRIPT
0A7E	C0 87 093A	3359	B	B\$PUTC	LINK TO GENERATE .SDO. PMC
0A82	F2 87 42	3360	J	BMM070	GO CALL SECOND SEGMENT
		3361	*		
		3362	*	TEST IF FUNCTION IS 'INV' OR 'TRN'	
		3363	*		
0A85	3D D5 1B90	3364	BMM030 CLI	B\$MFBK+BMMBK1,BMMINV	IF FUNC IS 'INV'
0A89	F2 81 07	3365	JE	BMM040	* GO PROCESS NAT REFERENCE
0A8C	3D D9 1B90	3366	CLI	B\$MFBK+BMMBK1,BMMTRN	IF FUNC IS 'TRN'
0A90	F2 01 0F	3367	JNE	BMM050	* GO PROCESS OTHER FUNCTIONS
		3368	*		
		3369	*	PROCESS MATRIX REFERENCED 'INV' OR 'TRN'.	
		3370	*		
0A93	C0 87 093A	3371	BMM040 B	B\$PUTC	LINK TO GENERATE .SDO. PMC
0A97	C0 87 18F3	3372	B	B\$MATR	LINK TO PROCESS MAT REFERENCE
0A9B	C0 87 0867	3373	B	B\$GETC	LINK TO GET NEXT CHAR
0A9F	F2 87 25	3374	J	BMM070	GO CALL SECOND SEGMENT
		3375	*		
		3376	*	PROCESS MATRIX FOR 'IDN', 'CON', OR 'ZER' FUNC	
		3377	*		
0AA2	3D 00 0A39	3378	BMM050 CLI	B\$PERC,@ZERO	IF ERROR IS FOR UNDEFINED ARRAY
0AA6	C0 01 1AE6	3379	BNE	B\$RMK	* NRURN TO DIST VIA REMARK
0AAA	3B 07 18FF	3380	SBF	B\$MGSW,B\$MGMK	SET MAT RTN NOT TO CALL GET RTN
N04 0AAE	00 00 0000	3381	SBN	B\$MBSW,BSMBMK	SET TO SKIP DOPE VECTOR STK
0AB2	C0 87 18F3	3382	B	B\$MATR	LINK TO REDIM AND GENERATE PMC
0AB6	3B 07 1903	3383	SBF	B\$MBSW,B\$MBMK	SET SN NOT TO SKIP D.V. STK
0ABA	3A 07 18FF	3384	SBN	B\$MGSW,B\$MGMK	ENABLE MAT RTN TO CALL GET RTN
0ABE	F2 87 06	3385	J	BMM070	GO CALL SECOND SEGMENT
		3386	*		
		3387	*	SET BRANCH ADDRESS FOR 3RD ENTRY POINT BEFORE GOING TO CALLING SEG	
		3388	*		
0AC1	7C 4C F2	3389	BMM060 MVI	BMPBA(,@BR),BMM140-BMMAT2	SET BR ADDR FOR 3RD ENTRY PT
0AC4	7C 4C EE	3390	MVI	BMM095+@D1(,@BR),BMM140-BMMAT2	SET BR ADDR TO 3RD ENT PT
		3392	*****		
		3393	*	MAT ASSIGNMENT 2ND SEGMENT CALLING SEQUENCE ROUTINE	
		3394	*****		
		3395	*		
		3396	*	TEST WHETHER CURRENT SEGMENT WAS CORE OR DISK RESIDENT	
		3397	*		
0AC7	5D 00 F3 F1	3398	BMM070 CLC	BMMCA2-1(,@BR),BMPBA-1(@CADDR-1,@BR)	IF CURR SEG FR DISK
0ACB	F2 81 10	3399	JE	BMM080	* GO LOAD & EXEC 2ND SEGMENT
		3400	*		
		3401	*	CURRENT SEGMENT WAS CORE RESIDENT TEST WHETHER 2ND SEGMENT HAS	
		3402	*	ALSO BEEN LOADED INTO CORE	
		3403	*		
0ACE	5C 01 F7 F9	3404	MVC	BMMFCP(,@BR),BMMFPE(@CADDR,@BR)	SET FINAL CORE PAGE ADDR
0AD2	4E 00 F6 043B	3405	ALC	BMMFCP-1(,@BR),\$EXFTR(1)	CALC MAX PROCESSOR CORE PAGE
		3406	*		
0AD7	5D 01 F4 F7	3407	CLC	BMMCA2(,@BR),BMMFCP(@CADDR,@BR)	IF 2ND SEGMENT IN CORE
0ADB	F2 82 0B	3408	JL	BMM090	* GO SET TO EXEC 2ND SEGMENT
		3409	*		
		3410	*	2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR PARAMETERS FOR	
		3411	*	CORE-LOADING AND EXECIITING DE 2ND SEGMENT	
		3412	*		

## S/3 BASIC COMPILER -MAT- ASSIGNMENT STMT RTN

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 19
	0ADE	5C 01 F4 F2		3413	BMM080 MVC	BMMCA2(, @BR), B MMPBA(@CADDR, @BR) SET UP DISKLOAD ADDR	
				3414	*		
				3415	* EXIT TO DISTRIBUTOR FOR 2ND SEGMENT CORELOAD AND EXECUTION		
				3416	*		
	0AE2	D2 02 F3		3417	LA	BMMAD2(, @BR), @XR LOAD DISTRIBUTOR PARM CADDR	
	0AE5	C0 87 073A		3418	B	B\$DST2 GO LOAD & EXECUTE 2ND SEGMENT	
				3419	*		
				3420	* 2ND SEGMENT IS CORE RESIDENT- BRANCH TO NEXT CONSECUTIVE CORE PAGE		
				3421	* AND CONTINUE MAT ASSIGNMENT EXECUTION		
				3422	*		
	0AE9	76 01 F0		3423	BMM090 A	BMMBLS(, @BR), @BR ADJUST BASE ADDR FOR 2ND SEG	
	0AEC	D0 87 00		3424	BMM095 B	*-(, @BR) GO EXECUTE 2ND SEGMENT	
				3426	*****		
				3427	* MAT ASSIGNMENT SEGMENT-1 CONSTANTS AND WORK AREAS, AND EQUATES		
				3428	*****		
				3429	*		
	0AEF	0100	0AF0	3430	BMMBLS DC	AL(@CADDR)(B@BLSZ) * REFERNECE NEXT PAGE BOUNDARY	
				3431	*		
				3432	*		
	0AF1		0AF2	3433	B MMPBA DS	CL(@CADDR) PROCESSOR DISK BUFFER CADDR	
	0AF1			3434	ORG	*-@CADDR INITIALIZE DISK BUFFER CADDR TO	
	0AF1	0600	0AF2	3435	DC	AL(@CADDR)(B\$CSBF) * REFERENCE PAGE BOUNDARY	
				3436	*		
			00D5	3437	BMMINV EQU	C'N' COMPARISON FOR FUNC 'INV'	
			00D9	3438	BMMTRN EQU	C'R' COMPARISON FOR FUNC 'TRN'	
				3439	*		
			0AF3	3440	BMMAD2 EQU	* DISTR PARAMS FOR SEG-2 EXEC	
	0AF3		0AF4	3441	BMMCA2 DS	CL(@CADDR) MAT ASSIGNMENT SEG CORE ADDRESS	
	0AF5	0C	0AF5	3442	BMMIA2 DC	AL1(B@DMAT+BMMPSI) BMMATA SEG-2 PHYS SECTOR ADDR	
				3443	*		
	0AF6		0AF7	3444	BMMFCP DS	CL(@CADDR) FINAL AVAILABLE CORE PAGE ADDR	
	0AF8	1F00	0AF9	3445	BMMFPE DC	AL(@CADDR)(B\$CSXA-B@BLSZ) FINAL PAGE BEFORE EXTENSION	
				3446	*		
				3447	* EQUATES		
				3448	*		
			0000	3449	BMMBK0 EQU	0 DISP TO 1ST BUCKET BYTE	
			0001	3450	BMMBK1 EQU	1 DISP TO 2ND BUCKET BYTE	
			0002	3451	BMMBK2 EQU	2 DISP TO 3RD BUCKET BYTE	
				3452	*		
			0000	3453	BMMMSG2 EQU	0 DISP FOR BMMATA SEG-2 ENTRY PT	
			0004	3454	BMMPSI EQU	X'04' PHYS SECTOR ADM INCREMENT	
				3456	*****		
				3457	* MAT ASSIGNMENT SECOND SEGMENT		
				3458	*****		
				3459	*		
				3460	* ESTABLISH MAT ASSIGNMENT SEGMENT-2 ADDRESSABILITY		
				3461	*		
	0B00			3462	ORG	BMMATA+B@BLSZ BEGIN SEG-2 AT PAGE BOUNDARY	
			0B00	3463	USING	*, @BR DEFINE SEG-2 BASE ADDRESS	
			0B00	3464	BMMAT2 EQU	* BMMATA - SEG-2 MAIN ENTRY PT	
				3465	*		
				3466	* GENERATE THE 'SD0' PMC IN VIRTUAL MEMORY		
				3467	*		
	0B00	C0 87 093A		3468	BMM100 B	B\$PUTC LINK TO GENERATE 'SD0. PMC	



## S/3 BASIC COMPILER -MAT- ASSIGNMENT STMT RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 20
					3469	*		
					3470	*	PROCESS FIRST MATRIX REFERENCE IN MAT OPERATION	
					3471	*		
	0B04	3D	00	0873	3472	CLI	B\$NUMC,B@GETS	SET GET NOT TO SKIP CHAR
	0B08	C2	02	1B8F	3473	LA	B\$MFBK+BMMBK0,@XR	SET PTR TO 15T BUCKET BYTE
	0B0C	3B	07	18FF	3474	SBF	B\$MGSW,B\$MGMK	DISABLE BMATXR TO CALL GET RTN
	0B10	C0	87	18F3	3475	B	B\$MATR	LINK TO PROCESS MAT REFERENCE
					3476	*		
					3477	*	PROCESS THE SECOND MATRIX REFERENCE IN MAT OPERATION	
					3478	*		
	0B14	3C	00	0873	3479	MVI	B\$NUMC,B@GETS	SET GET NOT TO SKIP CHAR
	0B18	C2	02	1B91	3480	LA	B\$MFBK+BMMBK2,@XR	SET PTR TO 3RD BUCKET BYTE
	0B1C	C0	87	18F3	3481	B	B\$MATR	LINK TO PROCESS MAT REFERENCE
	0B20	3A	07	18FF	3482	SBN	B\$MGSW,B\$MGMK	ENABLE BMATXR TO CALL GET RTN
					3483	*		
					3484	*	MOVE BLANKS INTO THE 1ST AND 3RD BYTES OF THE SAVE BUCKET	
					3485	*		
	0B24	3C	40	1B8F	3486	MVI	B\$MFBK+BMMBK0,B@BLNK	SET 15T BUCKET BYTE TO BLANK
	0B28	3C	40	1B91	3487	MVI	B\$MFBK+BMMBK2,B@BLNK	SET 3RD BUCKET BYTE TO BLANK
					3488	*		
					3489	*	SEARCH TABLE FOR MATCHING FUNCTION - 2ND ENTRY PT FOR 2ND SEGMENT	
					3490	*		
N04	0B2C	00	00	00	3491	BMM110	LA BMMTB5(,@BR),@BR	LOAD FUNC TBL POINTER
	0B2F	D2	02	06	3492	BMM120	LA BMMTEL(,@BR),@XR	INCREMENT POINTER TO NEXT ENTRY
N04	0B32	00	00	0000 00	3493	CLC	B\$MFBK+BMKBK2,BMMFND(B@LIFN,@XR)	IF FUNC = TBL ENTRY
	0B37	D0	01	2F	3494	BNE	BMM120(,@BR)	GO COMPARE FUNC TO NXT TBL ENT
					3495	*		
					3496	*	GENERATE THE PMC ASSOCIATED WITH THE TABLE ENTRY FUNCTION	
					3497	*		
	0B3A	E2	02	03	3498	BMM130	LA B@LIFN(,@XR),@XR	LOAD CADDR OF .11F1. INSTR
	0B3D	34	02	0A40	3499	ST	B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'MF1'
	0B41	3C	02	0A41	3500	MVI	B\$PNBY,B@LMF1-1	SET LNG PARM OF PUT FOR 'MF1'
	0B45	C0	87	093A	3501	B	B\$PUTC	LINK TO GENERATE PMC
					3502	*		
					3503	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
					3504	*		
	0B49	F2	87	1F	3505	J	BMM150	GO CALL DISTRIBUTOR
					3506	*		
					3507	*		
					3508	*	GENERATE THE 'SDO' PMC IN VIRT MEM BEFORE PROCESSING THE EXPRESSION	
					3509	*		
	0B4C	C0	87	093A	3510	BMM140	B B\$PUTC	LINK TO GENERATE 'SD0' PMC
					3511	*		
					3512	*	PROCESS ARITHMETIC EXPRESSION AND MAT REFERENCE	
					3513	*		
	0B50	C0	87	1514	3514	B	B\$SCAN	LINK TO PROCESS ARITH DPP
	0B54	C0	87	0867	3515	B	B\$GETC	LINK TO GET NEXT CHAR
	0B58	C0	87	18F3	3516	B	B\$MATR	LINK TO PROCESS MAT REFERENCE
					3517	*		
					3518	*	GENERATE AN 'MSM' INSTR IN VIRTUAL MEMORY	
					3519	*		
	0B5C	D2	02	99	3520	LA	BMMMSC(,@BR),@XR	LOAD CADDR OF 'MSM' INSTR
	0B5F	34	02	0A40	3521	ST	B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'MSM'
	0B63	3C	02	0A41	3522	MVI	B\$PNBY,B@LMSM-1	SET LNG PARM OF PUT FOR 'MSM'
	0B67	C0	87	093A	3523	B	B\$PUTC	LINK TO GENERATE 'NSM' PMC
					3524	*		



			3525	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
			3526	*		
0B6B	C0	87	0700	3527	BMM150 B B\$DIST	RETURN TO DISTRIBUTOR
			3528	*		
			3529	*	GENERATE 'SDO' FOR 1ST MAT REFERENCE AND PROCESS 2ND MAT REFERENCE	
			3530	*		
0B6F	C0	87	093A	3531	BMM160 B B\$PUTC	LINK TO GENERATE 'SDO' PMC
0B73	3C	00	0873	3532	MVI B\$NUMC,B@GETS	DISABLE GET RTN TO GET CHARS
0B77	3B	07	18FF	3533	SBF B\$MGSW,B\$MGMK	SET GET RTN NOT TO ADVANCE PTR
N04 0B7B	00	00	0000	3534	LA B\$MFBK+BMKBK0,@XR	SET PTR TO MAT REFERENCE
0B7F	C0	87	18F3	3535	B B\$MATR	LINK TO PROCESS MAT REFERENCE
0B83	3A	07	18FF	3536	SBN B\$MGSW,B\$MGMK	ENABLE GET RTN TO GET CHARS
			3537	*		
			3538	*	GENERATE AN 'MF2' INSTR IN VIRTUAL MEMORY	
			3539	*		
0B87	D2	02	9C	3540	LA BMMM2C(,@BR),@XR	LOAD CADDR OF 'MF2' INSTR
0B8A	34	02	0A40	3541	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'MF2'
0B8E	3C	02	0A41	3542	MVI B\$PNBY,B@LMF2-1	SET LNG PARM OF PUT FOR 'MF2'
0B92	C0	87	093A	3543	B B\$PUTC	LINK TO GENERATE 'MF2' PMC
0B96	D0	87	6B	3544	B BMM150(,@BR)	RETURN TO DIST

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 22

```

3546 *****
3547 * MAT ASSIGNMENT SEGMENT-2 STORAGE AND PARAMETER AREA
3548 *****
3549 *
N04 0B99 00          0B99 3550 BMMMSC DC    AL(B@LCOP)(B@CMSN)      CADDR OF 'MSM' INSTR OPCODE
    0B9A 4264        0B9B 3551 BMMMSO DC    AL(B@LCVA)(V$MSMY)      CADDR OF 'MSM' INSTR OPERAND
                                3552 *
    0B9C 1A          0B9C 3553 BMMM2C DC    AL(B@LCOP)(B@CMF2)      CADDR OF 'MF2' INSTR OPCODE
    0B9D 43A0        0B9E 3554 BMMM2O DC    AL(B@LCVA)(V$MASN)      CADDR OF 'MF2' INSTR OPERAND

                                3556 *****
                                3557 * 'MAT' ASSIGNMENT STATEMENT MATRIX FUNCTION TABLE
                                3558 *****
                                3559 *
                                0006 3560 BMMTEL EQU    6              LENGTH OF TABLE ENTRY
                                0003 3561 BMMPID EQU    3              LENGTH OF PSEUDO INSTR DISP
                                0002 3562 BMMFND EQU    2              LENGTH OF FUNCTION DISP
                                3563 *
                                0B9F 3564 BMMTAB EQU    *              BEGINNING OF MAT FUNCTION TBL
                                3565 *
    0B9F 404E40        0BA1 3566          DC    CL(B@LIFN)' + '        FUNC FOR MATRIX ADDITION
    0BA2 1C            0BA2 3567          DC    AL(B@LCOP)(B@CMF3)      CADDR OF 'MF3' INSTR OPCODE
    0BA3 4007          0BA4 3568          DC    AL(B@LCVA)(V$MADD)      CADDR OF 'MF3' INSTR OPERAND
                                3569 *
    0BA5 406040        0BA7 3570          DC    CL(B@LIFN)' - '        FUNC FOR MATRIX SUBTRACTION
    0BA8 1C            0BA8 3571          DC    AL(B@LCOP)(B@CMF3)      CADDR FOR 'MF3' INSTR OPCODE
    0BA9 4000          0BAA 3572          DC    AL(B@LCVA)(V$MSUB)      CADDR FOR 'MF3' INSTR OPERAND
                                3573 *
    0BAB 405C40        0BAD 3574          DC    CL(B@LIFN)' * '        FUNC FOR MATRIX MULTIPLICATION
    0BAE 1C            0BAE 3575          DC    AL(B@LCOP)(B@CMF3)      CADDR FOR 'MF3' INSTR OPCODE
    0BAF 4100          0BB0 3576          DC    AL(B@LCVA)(V$MMPY)      CADDR FOR 'MF3' INSTR OPERAND
                                3577 *
    0BB1 C9D5E5        0BB3 3578          DC    CL(B@LIFN)' INV'        FUNC FOR MATRIX INVERSION
    0BB4 1A            0BB4 3579          DC    AL(B@LCOP)(B@CMF2)      CADDR FOR 'MF2' INSTR OPCODE
    0BB5 4500          0BB6 3580          DC    AL(B@LCVA)(V$MINV)      CADDR FOR 'MF2' INSTR OPERAND
                                3581 *
    0BB7 E3D9D5        0BB9 3582          DC    CL(B@LIFN)' TRN'        FUNC FOR MATRIX TRANSPOSITION
    0BBA 1A            0BBA 3583          DC    AL(B@LCOP)(B@CMF2)      CADDR FOR 'MF2' INSTR OPCODE
    0BBB 4400          0BBC 3584          DC    AL(B@LCVA)(V$MTRN)      CADDR FOR 'MF2' INSTR OPERAND
                                3585 *
    0BBD E9C5D9        0BBF 3586          DC    CL(B@LIFN)' ZER'        FUNC FOR MAT INITIALLY ZEROES
    0BC0 18            0BC0 3587          DC    AL(B@LCOP)(B@CMF1)      CADDR OF 'MF1' INSTR OPCODE
    0BC1 432B          0BC2 3588          DC    AL(B@LCVA)(V$MZER)      CADDR OF 'MF1' INSTR OPERAND
                                3589 *
    0BC3 C3D6D5        0BC5 3590          DC    CL(B@LIFN)' CON'        FUNC FOR MAT INITIALLY ONE'S
    0BC6 18            0BC6 3591          DC    AL(B@LCOP)(B@CMF1)      CADDR OF 'MF1' INSTR OPCODE
    0BC7 4324          0BC8 3592          DC    AL(B@LCVA)(V$MCON)      CADDR OF 'MF1' INSTR OPERAND
                                3593 *
    0BC9 C9C4D5        0BCB 3594          DC    CL(B@LIFN)' IDN'        FUNC FOR MATRIX IDENTITY
    0BCC 18            0BCC 3595          DC    AL(B@LCOP)(B@CMF1)      CADDR FOR 'MF1' INSTR OPCODE
    0BCD 4300          0BCE 3596          DC    AL(B@LCVA)(V$MIDN)      CADDR FOR 'MF1' INSTR OPERAND
                                3597 *
                                0B99 3598 BMMTBS EQU    BMMTAB-BMMTEL    INITIAL FUNC TOL ENTRY POINT
                                3599 *****
                                3600 *
                                3601 * END OF 'MAT ASSIGNMENT' STATEMENT CODING

```



## S/3 BASIC COMPILER -READ- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 24
		3604		*****	
		3605	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		3606	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		3607	*		*
		3608		*****	
		3609	*	*STATUS	*
		3610	*	VERSION 1 MODIFICATION 0	*
		3611	*		*
		3612	*	*FUNCTION	*
		3613	*	BPREAD IS EXECUTED TO TRANSLATE READ STATEMENTS AS THEY OCCUR IN	*
		3614	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE	*
		3615	*	PSEUDOCODE IN VIRTUAL MEMORY.	*
		3616	*		*
		3617	*	*ENTRY POINTS	*
		3618	*	BPREAD HAS ONLY ONE ENTRY POINT:	*
		3619	*	BPREAD - TRANSLATE READ STATEMENT	*
		3620	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		3621	*	B BPREAD	*
		3622	*		*
		3623	*	*INPUT	*
		3624	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		3625	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE	*
		3626	*	LEADING KEYWORD, READ.	*
		3627	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		3628	*	FIRST CHARACTER IN THE LEADING KEYWORD, READ.	*
		3629	*		*
		3630	*	*OUTPUT	*
		3631	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		3632	*	GENERATED BY BPREAD IS STORED IN THE PEST AVAILABLE VIRTUAL	*
		3633	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		3634	*	SEQUENCES.	*
		3635	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		3636	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		3637	*		*
		3638	*	*EXTERNAL REFERENCES	*
		3639	*	BSGETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.	*
		3640	*	BSPUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL	*
		3641	*	MEMORY OUTPUT ROUTINE.	*
		3642	*	B\$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.	*
		3643	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		3644	*		*
		3645	*	*EXITS, NORMAL	*
		3646	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		3647	*		*
		3648	*	*EXITS, ERROR	*
		3649	*	N/A	*
		3650	*		*
		3651	*	*TABLES/WORK AREAS	*
		3652	*	N/A	*
		3653	*		*
		3654	*	*ATTRIBUTES	*
		3655	*	BPREAD IS NATURALLY RELOCATABLE AND REUSABLE	*
		3656	*		*
		3657	*	*CHARACTER CODE DEPENDENCY	*
		3658	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		3659	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*

## S/3 BASIC COMPILER -READ- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 25
				3660	*				*
				3661	*NOTES				*
				3662	* ERROR PROCEDURES				*
				3663	* N/A				*
				3664	*				*
				3665	* REGISTER USAGE				*
				3666	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION				*
				3667	*				*
				3668	* SAVED/RESTORED AREAS				*
				3669	* N/A				*
				3670	*				*
				3671	* MODIFICATION CONSIDERATIONS				*
				3672	* BPREAD IS CO-RESIDENT ON A SECTOR WITH BMMATA.			1-4	*
				3673	* ANY MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND			1-4	*
				3674	* THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.			1-4	*
				3675	*				*
				3676	* REQUIRED MODULES				*
				3677	* @SYSEQ - COMMON SYSTEM EQUATES.				*
				3678	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.				*
				3679	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
				3680	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
				3681	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
				3682	* @ERMEQ - ERROR MESSAGE EQUATES.				*
				3683	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.				*
				3684	* \$B\$EQU - COMPILER FIXED EQUATES.				*
				3685	* \$B@EQU - COMPILER SYSTEM EQUATES.				*
				3686	*				*
				3687	* OTHER				*
				3688	* BPREAD IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
				3689	*****				*
				3691	*				*
				3692	* ENTER BPREAD - 'READ' STATEMENT ROUTINE				*
				3693	*				*
			0BCF	3694	BPREAD EQU *	BPREAD ENTRY POINT			*
				3695	*				*
				3696	* SET INPUT PARAMETER TO SKIP TO 'D' IN KEYWORD 'READ'				*
				3697	*				*
		0BCF 3C 03 0873		3698	BPR010 MVI B\$NUMC,B@LREA-1	SKIP TO 'D' IN 'READ'			*
		0BD3 C0 87 0867		3699	B B\$GETC	LINK TO ADVANCE POINTER			*
				3700	*				*
				3701	* ADVANCE POINTER TO GET NEXT CHARACTER				*
				3702	*				*
		0BD7 C0 87 0867		3703	BPR020 B B\$GETC	LINK TO GET NEXT CHARACTER			*
				3704	*				*
				3705	* CALL LIST ROUTINE TO PROCESS CURRENT LIST ELEMENT				*
				3706	*				*
		0BDB C0 87 1853		3707	BPR030 B B\$LIST	LINK TO PROCESS LIST ELEMENT			*
				3708	*				*
				3709	* GENERATE A GET INSTRUCTION PMC IN VIRTUAL MEMORY WHICH REFERENCES				*
				3710	* THE VIRTUAL ENTRY ADDRESS OF THE RUN-TIME READ ROUTINE				*
				3711	*				*
		0BDF D2 02 FC		3712	BPR040 LA BPRGTC(,@BR),@XR	LOAD CADDR OF 'GET' INSTR			*
		0BE2 34 02 0A40		3713	ST B\$PCAD,@XR	SET PUT RTN VADDR FOR 'GET'			*
		0BE6 3C 02 0A41		3714	MVI B\$PNBY,B@LGET-1	SET PUT RTN LNG FOR 'GET'			*
		0BEA C0 87 093A		3715	B B\$PUTC	LINK TO GENERATE 'GET' PNC			*

## S/3 BASIC COMPILER -READ- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 26
				3716	*				
				3717	*	TEST FOR STATEMENT TERMINATOR			
				3718	*				
0BEE	35	02 0878		3719	BPR050 L	B\$GPTR,@XR			RESTORE TEXT POINTER
0BF2	BD	1E 00		3720		CLI B@CHAR(,@XR),B@EOST			IF ANOTHER LIST ELEMENT FOLLOWS
0BF5	D0	01 D7		3721		BNE BPR020(,@BR)			* GO PROCESS NEXT ELEMENT
				3722	*				
				3723	*	RETURN CONTROL TO COMPILER DISTRIBUTOR			
				3724	*				
0BF8	C0	87 0700		3725	BPR060 B	B\$DIST			RETURN TO DISTRIBUTOR
				3727	*****				
				3728	*	'READ' STATEMENT ROUTINE STORAGE AND PARAMETER AREAS			
				3729	*****				
				3730	*				
0BFC	52		0BFC	3731	BPRGTC DC	AL(B@LCOP)(B@CGET)			'GET' OPCODE
0BFD	3300		0BFE	3732	BPRGTO DC	AL(B@LCVA)(V\$XSRD)			'GET' OPERAND
				3734	*****				
				3735	*				
				3736	*	END OF 'READ' STATEMENT ROUTINE CODING			
				3737	*				

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 27
			3739		*****			
			3740	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			3741	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			3742	*				*
			3743		*****			*
			3744	*	STATUS			*
			3745	*	VERSION 1 MODIFICATION 0			*
			3746	*				*
			3747	*	FUNCTION			*
			3748	*	BSTRLT IS EXECUTED TO TRANSLATE LET STATEMENTS WITH SUB-STRING			*
			3749	*	OPERANDS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE			*
			3750	*	PSEUDO INSTRUCTION SEQUENCE AND TO PLACE THE PSEUDO INSTRUCTION			*
			3751	*	SEQUENCE IN VIRTUAL MEMORY.			*
			3752	*				*
			3753	*	ENTRY POINTS			*
			3754	*	BSTRLT HAS TWO ENTRY POINTS:			*
			3755	*	BSTRLT - TRANSLATE LET STATEMENTS			*
			3756	*	BSTRAS - TRANSLATE ASSIGNMENT STMT (KEYWORD-LET MISSING)			*
			3757	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			3758	*	B BSTRLT			*
			3759	*	B BSTRAS			*
			3760	*				*
			3761	*	INPUT			*
			3762	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			3763	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			3764	*	KEYWORD LET, OR THE FIRST CHARACTER IN THE ASSIGNMENT LIST			*
			3765	*	IF THE KEYWORD, LET, IS MISSING.			*
			3766	*	* A TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			3767	*	FIRST CHARACTER IN THE LEADING KEYWORD, LET, OR THE FIRST			*
			3768	*	CHARACTER IN THE ASSIGNMENT LIST IF THE KEYWORD, LET, IS			*
			3769	*	MISSING.			*
			3770	*				*
			3771	*	OUTPUT			*
			3772	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			3773	*	GENERATED BY BSTRLT IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			3774	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			3775	*	SEQUENCES.			*
			3776	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			3777	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			3778	*				*
			3779	*	EXTERNAL REFERENCES			*
			3780	*	* B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			3781	*	* B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRTUAL			*
			3782	*	MEMORY OUTPUT ROUTINE.			*
			3783	*	* B\$LIST - (B\$LSTR, B\$LVSV, B\$LRN, B\$LBAS) - ENTRY TO BASIC			*
			3784	*	COMPILER LIST ADDRESS ROUTINE.			*
			3785	*	* B\$SCAN - ENTRY TO COMPILER ARITHMETIC EXPRESSION SCAN ROUTINE.			*
			3786	*	* B\$SCSN - (B\$CSTP, B\$CRAD, B\$CDAS, B\$CRBS) - COMPILER CHARACTER			*
			3787	*	EXPRESSION SCAN ROUTINE.			*
			3788	*	* B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH			*
			3789	*	TABLE ROUTINE.			*
			3790	*	* B\$DIST - (B\$DST2) - ENTRY TO BASIC COMPILER DISTRIBUTOR ROUTINE.			*
			3791	*	* B\$COMN - (B\$PRM1, B\$RTRN, B\$BROP, B\$CADR) - COMPILER CORE			*
			3792	*	RESIDENT COMMON SECTION			*
			3793	*	* B\$SYMB - (B\$CRSW, B\$BCKT) - COMPILER SYMBOL TRANSLATION ROUTINE.			*
			3794	*				*



## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 28
		3795	*	*EXITS, NORMAL	*
		3796	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR	*
		3797	*		*
		3798	*	*EXITS, ERROR	*
		3799	*	N/A	*
		3800	*		*
		3801	*	*TABLES/WORK AREAS	*
		3802	*	N/A	*
		3803	*		*
		3804	*	*ATTRIBUTES	*
		3805	*	BSTRLT IS NATURALLY RELOCATABLE AND REUSABLE.	*
		3806	*		*
		3807	*	*CHARACTER CODE DEPENDENCY	*
		3808	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		3809	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		3810	*		*
		3811	*	*NOTES	*
		3812	*	ERROR PROCEDURES	*
		3813	*	N/A	*
		3814	*		*
		3815	*	REGISTER USAGE	*
		3816	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
		3817	*		*
		3818	*	SAVED/RESTORED AREAS	*
		3819	*	N/A	*
		3820	*		*
		3821	*	MODIFICATION CONSIDERATIONS	*
		3822	*	BSTRLT IS DIVIDED INTO THREE SECTIONS. OCCUPYING THREE	*
		3823	*	SECTORS. ANY MODIFICATIONS MUST MAINTAIN LINKAGE BETWEEN	*
		3824	*	THE THREE SECTORS AND REALIZE THE LIMITATION OF THE SECTOR	*
		3825	*	BOUNDARY ON THE SIZE OF EACH SECTION.	*
		3826	*		*
		3827	*	REQUIRE MODULES	*
		3828	*	@SYSEQ - COMMON SYSTEM EQUATES.	*
		3829	*	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES.	*
		3830	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
		3831	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
		3832	*	@ERMEQ - ERROR MESSAGE EQUATES.	*
		3833	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
		3834	*	\$B\$EQU - COMPILER FIXED ADDRESS EQUATES.	*
		3835	*	\$B@EQU - COMPILER SYSTEM EQUATES.	*
		3836	*		*
		3837	*	OTHER	*
		3838	*	BSTRLT IS ASSEMBLED WITH ALL THE STATEMENT PROCESSORS.	*
		3839	*	*****	*
0C00		3841		ORG *,256,0	PLACE MODULE AT PAGE BOUNDARY
	0C00	3842		USING *,@BR	ESTABLISH BASE ADDRESSING
		3843	*	*****	*
		3844	*	FIRST DETERMINE IF THIS SEGMENT HAS BEEN ACCESSED	*
		3845	*	PREVIOUSLY IN THE PROCESSING OF THIS STATEMENT.	*
		3846	*	*****	*
	0C00	3847	BSTRLT EQU *		LET ENTRY POINT ADDRESS
0C00 74 01 F6		3848	ST	CNTCA2(,@BR),@BR	SAVE THE CADDR OF THIS SECTION
0C03 3D 00 1AF5		3849	CLI	B\$RTRN,@ZERO	IF THIS FIELD IS ZERO WE ARE
0C07 F2 81 09		3850	JE	BST020	* ENTERING FOR THE ?INST TIME



## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 29

```

0C0A 4C 01 12 1AF5      3851      MVC      BST010+@OP1(@CADDR,@BR),B$RTRN  ELSE BRANCH TO THE SAVED
0C0F C0 87 0000          3852 BST010 B      *-*              * RETURN ADDRESS
3853 *****
3854 *                  LET ENTRY POINT (KEYWORD, LET, IS PRESENT).  THIS ENTRY *
3855 *                  POINT WILL ADVANCE THE TEXT CHARACTER POINTER TO THE *
3856 *                  'T' IN THE KEYWORD LET.                                *
3857 *****
0C13 3C 03 0873          3858 BST020 MVI      B$NUMC,B@LLET      SET GET ROUTINE TO SKIP KEYWORD
0C17 C0 87 0867          3859      B      B$GETC      ADVANCE TEXT CHARACTER POINTER
3860 *****
3861 *                  ASSIGNMENT ENTRY POINT (KEYWORD, LET, IS MISSING).  THIS *
3862 *                  ENTRY POINT WILL ADVANCE THE TEXT CHARACTER POINTER TO *
3863 *                  THE LEADING CHARACTER OF THE FIRST ASSIGNMENT LIST *
3864 *                  ELEMENT.                                              *
3865 *****
0C1B 74 01 F6            0C1B 3866 BSTRAS EQU      *                  ASSIGNMENT ENTRY POINT ADDRESS
3867      ST      CNTCA2(,@BR),@BR      SAVE THE CADDR OF THIS SECTION
3868 *****
3869 *                  THE TEXT CHARACTER POINTER IS POSITIONED. NOW INITIALIZE *
3870 *                  ALL SWITCHES AND GENERATE A BRANCH INSTRUCTION IMAGE SO *
3871 *                  THAT AT EXECUTION TIME THE RIGHT SIDE OF THE EQUAL SIGN *
3872 *                  HILL BE PROCESSED FIRST AND THE RESULT SAVED IN THE *
3873 *                  TEMPORARY VARIABLE, ECWRK.                            *
3874 *****
0C1E D2 02 E9            3875 BST080 LA      CNTBRA(,@BR),@XR      LOAD CADDR OF BRANCH INSTR
0C21 D0 87 CF            3876      B      BST150(,@BR)      GO GENERATE BRANCH INSTR IMAGE
0C24 0C 01 1AF7 0A43      3877      MVC      B$BROP(@VADDR),B$PVAD      SAVE RETURN ADDR FOR RTRN BRNCH
0C2A 35 02 0878          3878      L      B$GPTR,@XR      LOAD THE TEXT CHARACTER POINTER
3879 *****
3880 *                  INITIALIZE MODULE SWITCHES AND BEGIN PROCESSING *
3881 *                  ASSIGNMENT LIST ELEMENTS IN SEQUENCE.                *
3882 *****
0C2E 3C 01 1BAC          3883 BST100 MVI      B$SSTA,@B1      ENABLE BDSYMB DETECTION OF 'STR'
0C32 C0 87 0DBC          3884      B      B$SYMB      TRANSLATE CURRENTLY REED SYMBOL
0C36 3C 00 159E          3885      MVI      B$KWSW,@ZERO      TURN OFF KEYWCOK SWITCH
3886 *****
3887 *                  IF SYMBOL JUST TRANSLATED WAS A CHARACTER REFERENCE. *
3888 *                  THE SWITCH, BSCRSW, WILL BE ON AND THE VADDR OF THE *
3889 *                  REFERENCE WILL BE AT BSBCKT.  THE TEXT CHARACTER POINTER *
3890 *                  REFERENCES THE CHARACTER FOLLOWING THE CHARACTER *
3891 *                  REFERENCE(THE OPENING PARENTHESIS OF AN ARRAY REFERENCE). *
3892 *                  IF THE SYMBOL WAS A STRING REFERENCE, THE TEXT CHARACTER *
3893 *                  POINTER REFERENCES THE 'T' IN STR.                      *
3894 *****
0C3A 3D 00 0E42          3895 BST120 CLI      B$CRSW,@ZERO      IF THE SYMBOL WAS A CHAR REF
0C3E D0 01 4B            3896      BNE      BST130(,@BR)      * GO ACCESS CHAR PROCESSOR SEG
3897 *****
3898 *                  THE SYMBOL JUST PROCESSED WAS A STRING FUNCTION *
3899 *                  SET UP TO ACCESS STR PROCESSOR SEGMENT *
3900 *****
0C41 7C 14 F7            3901      MVI      CNTSAD(,@BR),CNTSTR      SET DISK ADDR PARM FOR STR PROC
0C44 5C 01 F9 F6          3902      MVC      CNTWRK(@CADDR,@BR),CNTCA2(,@BR)  SET UP CORE RES TEST
0C48 F2 87 25            3903      J      BST132      GO TO ACCESSING ROUTINE
3904 *****
3905 *                  THE SYMBOL JUST PROCESSED WAS A CHARACTER REFERENCE. *
3906 *                  SET UP TO ACCESS CHAR PROCESSOR SEGMENT. *

```

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 30
				3907		*****	*****			
	0C4B	D2	02	62	3908	BST130	LA BST131(,@BR),@XR			LOAD RETURN ADDR
	0C4E	34	02	18EB	3909		ST B\$LRTN,@XR			SAVE RETURN ADDRESS IN BLISTA
	0C52	34	01	18E7	3910		ST B\$LBSV,@BR			SAVE BASE REG IN BLISTA
	0C56	C2	01	185E	3911		LA B\$LBAS,@BR			LOAD BLISTA BASE ADDRESS
	0C5A	35	02	0878	3912		L B\$GPTR,@XR			LOAD TEXT CHARACTER POINTER
	0C5E	C0	87	1862	3913		B B\$LSTR			GO GENERATE CHAR ADDR STACK PMC
				3914		*****	*****			
				3915	*		COMPLETE CHARACTER REFERENCE PROCESSING BY STACKING			*
				3916	*		THE CONTENT OF &CWRK.			*
				3917		*****	*****			
	0C62	D2	02	EE	3918	BST131	LA CNTCWR(,@BR),@XR			LOAD CADDR OF 'STC' &CWRK INSTR
	0C65	4C	00	EF 159F	3919		MVC CNTCWR+@B1(,@BR),B\$WORK-@B1(@B1)			SET VADDR OF &CWRK
	0C6A	D0	87	CF	3920		B BST150(,@BR)			GO GENERATE PMC
	0C6D	F2	87	36	3921		J BST140			GO CHECK NEXT LIST ELEMENT

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20	PAGE 31
				3923		*****			
				3924	*		NEXT REQUIRED SEGMENT ACCESSING SECTION.		*
				3925		*****			
N04	0C70	00	00	0000	00	3926	BST132 MVC B\$CADDR(@CADDR),CNICA2(,@BR) SAVE CADDR OF CNTRL SECTION		
	0C75	D2	02	A6		3927	LA BST140(,@BR),@XR SAVE THE RETURN ADDRESS FOR		
	0C78	34	02	1AF5		3928	ST B\$RTRN,@XR * RE ENTERING THE CNTRL SECTION		
	0C7C	5D	01	F6 F4		3929	BST134 CLC CNTCA2(@CADDR,@BR),CNTPBA(,@BR) IF CURR SEG CAME FR DISK		
	0C80	F2	81	0F		3930	JE BST136 * GO LOAD & EXEC SEG FR DISK		
				3931		*****			
				3932	*		CONTROL SECTION WAS CORE RESIDENT - TEST WHETHER THE		*
				3933	*		REQUIRED SECTION IS ALSO CORE RESIDENT.		*
				3934		*****			
	0C83	7C	1F	FC		3935	MVI CNTFCP-@B1(,@BR),CNTFPE SET FINAL CORE PAGE		
	0C86	4E	00	FC 043B		3936	ALC CNTFCP-1(,@BR),\$EXFTR(@B1) CALC MAX PROCESSOR CORE PAGE		
	0C8B	5D	01	F9 FD		3937	CLC CNTWRK(,@BR),CNTFCP(@CADDR,@BR) IF NEXT SEGMENT IN CORE		
	0C8F	F2	82	0B		3938	JL BST138 * GO SET TO EXEC NEXT SEGMENT		
				3939		*****			
				3940	*		REQUIRED SECTION IS DISK RESIDENT - ESTABLISH		*
				3941	*		DISTRIBUTOR PARAMETERS FOR CORELOADING AND EXECUTING		*
				3942	*		THE REQUIRED SECTION.		*
				3943		*****			
	0C92	5C	01	F6 F4		3944	BST136 MVC CNTCA2(,@BR),CNTPBA(@CADDR,@BR) SET UP DISKLOAD CADDR		
	0C96	D2	02	F5		3945	LA CNTAD2(,@BR),@XR LOAD DIST PARAMETERS CADDR		
	0C99	C0	87	073A		3946	B B\$DST2 GO LOAD & EXEC NEXT SEGMENT		
				3947		*****			
				3948	*		REQUIRED SEGMENT IS CORE RESIDENT BRANCH TO THE		*
				3949	*		REQUIRED SEGMENT'S ENTRY POINT.		*
				3950		*****			
	0C9D	75	01	F9		3951	BST138 L CNTWRK(,@BR),@BR LOAD THE BASE ADDRESS FOR		
	0CA0	76	01	F2		3952	A CNTBLS(,@BR),@BR * NEXT SEGMENT		
	0CA3	D0	87	00		3953	B CNTENT(,@BR) GO EXECUTE NEXT SEGMENT		

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 32

```

3955 *****
3956 *          LIST ELEMENT HAS BEEN PROCESSED, NOW CHECK TO SEE          *
3957 *          IF THE ENTIRE LIST HAS BEEN PROCESSED.  IF NOT GO GET      *
3958 *          THE NEXT LIST ELEMENT, IF IT HAS BEEN, GO PROCESS          *
3959 *          THE RIGHT SIDE.                                             *
3960 *****
N04 0CA6 00 00 00          3961 BST140 LA      CNTUSC(, @BR), @XR          LOAD CADDR OF USC INSTRUCTION
      0CA9 7C 01 D7          3962          MVI    BST160+@Q(, @BR), B@LUSC-1  SET LNGTH PARM FOR PUT RTN
      0CAC D0 87 CF          3963          B      BST150(, @BR)          GO GENERATE PMC
      0CAF BD 7E 00          3964          CLI    B@CHAR(, @XR), B@EQL      IF THE NEXT CHARACTER IS AN'+.
      0CB2 F2 81 07          3965          JE     BST145          * THEN GO ACCESS TERM SECTION
      0CB5 C0 87 0867        3966          B      B$GETC          ELSE ADVANCE TEXT POINTER AND
      0CB9 D0 87 2E          3967          B      BST100(, @BR)        * PROCESS NEXT LIST ELEMENT
3968 *****
3969 *          THE ENTIRE ASSIGNMENT LIST HAS BEEN PROCESSED, NOW          *
3970 *          SET UP TO ACCESS THE TERMINATION SECTION                    *
3971 *****
N04 0CBC 00 00 00          3972 BST145 MVI    CNTSAD(, @BR), CNITRM          SET DISK ADDR PARR FOR TAM SCTN
      0CBF 5C 01 F9 F6        3973          MVC    CNTWRK(@CADDR, @BR), CNTCA2(, @BR)  SET UP CORE RES TEST
      0CC3 5E 01 F9 FB        3974          ALC    CNTWRK(@CADDR, @BR), CNTBL1(, @BR)  INCREMENT TO CADDR-1 PAGE
      0CC7 1C 01 1AF5 EB      3975          MVC    B$RTRN(@CADDR), CNTBOP(, @BR)  CLEAR RETURN ADDRESS
      0CCC D0 87 7C          3976          B      BST134(, @BR)          GO ACCRDS TERMINATION SECTION
3977 *****
3978 *          THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY,          *
3979 *          THE PSEUDO INSTRUCTION POINTED TO BY @XR.                  *
3980 *          THE INPUT PARAMETERS ARE AS FOLLOWS:                        *
3981 *          1. XR REFERENCES THE INSTRUCTION TO BE                      *
3982 *          GENERATED.                                                  *
3983 *          2. IF THE LENGTH OF THE INSTRUCTION IS NOT                  *
3984 *          THREE, THE LENGTH MUST BE STORED IN A                       *
3985 *          MVI INSTRUCTION (BST160+@Q).                                *
3986 *****
      0CCF 74 08 E8          3987 BST150 ST      BST170+@OP1(, @BR), @ARR      SAVE THE RETURN ADDRESS
      0CD2 34 02 0A40        3988          ST      B$PCAD, @XR          SET CADDR PARM FOR THE PUT RTN
      0CD6 3C 02 0A41        3989 BST160 MVI    B$PNBY, B@LLET-1          SET LENGTH FARAH FOR THE PUT RTN
      0CDA C0 87 093A        3990          B      B$PUTC          GENERATE PMC IN VIRTUAL MEMORY
      0CDE 7C 02 D7          3991          MVI    BST160+@Q(, @BR), B@LLET-1  MAKE SUBROUTINE REUSABLE
      0CE1 35 02 0878        3992          L      B$GPTR, @XR          LOAD THE TEXT CHARACTER POINTER
      0CE5 C0 87 0000        3993 BST170 B      *- *          RETURN TO CALLING SECTION

```

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 33

```
3995 *****
3996 * SUBSTRING ASSIGNMENT - CONTROL SECTION CONSTANTS *
3997 * AND WORKAREAS. *
3998 *****
0CE9 46 0CE9 3999 CNTBRA DC AL(B@LCOP)(B@CBRA) BRANCH OPCODE
0CEA 0000 0CEB 4000 CNTBOP DC AL(@VADDR)(@ZERO) BRANCH OPERAND
4001 *
0CEC 2C 0CEC 4002 CWTU## DC AL(B@LCOP)(B@CUSC) UNSTACK CHAR OPCODE
0CED 01 0CED 4003 DC XL1'01' UNSTACK CHAR OPERAND
0CEE 28 0CEE 4004 CNTCWR DC AL(B@LCOP)(B@CSTC) STACK CHAR OPCODE
0CEF F500 0CF0 4005 DC AL2(B$CWRK) STACK CHAR OPERAND
0004 4006 CNTPSI EQU X'04' PHYSICAL SECTOR INCREMENT
0000 4007 CNTENT EQU 0 DISP TO ENTRY PTS OF OTHER SCTNS
0014 4008 CNTSTR EQU B@DSML+CNTPSI STR PROC SECTION-PHYS SCTR ADDR
0018 4009 CNTTRM EQU CNTSTR+CNTPSI TERM SECTION-PHYS SCTR ADDR
4010 *
0CF1 0100 0CF2 4011 CNTBLS DC AL(@CADDR)(B@BLSZ) LENGTH OF CORE PAGE
0CF3 0600 0CF4 4012 CNTPBA DC AL(@CADDR)(B$CSBF) PROCESSOR DISK BUFFER CORE ADDR
4013 *
4014 *
0CF5 4015 CNTAD2 EQU * DIST PARMS FOR EXEC NEXT SECTION
0CF5 0CF6 4016 CNTCA2 DS CL(@CADDR) CONTROL SECTION CORE ADDRESS
0CF7 0CF7 4017 CNTSAD DS CL1 PHYSICAL SECTOR ADDRESS
0CF8 0CF9 4018 CNTWRK DS CL2 CONTROL SECTION WORKAREA
0CFA 0200 0CFB 4019 CNTBL1 DC AL(@CADDR)(2*B@BLSZ) LENGTH OF 2 CORE PAGES
0CFC 0000 0CFD 4020 CNTFCP DC AL(@CADDR)(@ZERO) FINAL AVAILABLE CORE PAGE ADDR
001F 4021 CNTFPE EQU X'1F' FINAL PAGE BEFORE EXTENSION
4022 *****
4023 * END OF LET-CONTROL SECTION *
4024 *****
```

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 34

```

0D00          4026      ORG   BSTRLT+B@BLSZ          PLACE SEGMENT AT PAGE BOUNDARY
                0D00 4027      USING *,@BR          ESTABLISH BASE ADDRESS
                4028 *****
                4029 *              SYMBOL JUST TRANSLATED WAS A STRING FUNCTION - ADVANCE *
                4030 *              TEXT CHARACTER POINTER TO OPENING CHARACTER OF CHAR *
                4031 *              REFERENCE WITHIN THE STRING FUNCTION. *
                4032 *****
0D00 3C 03 0873 4033 BST200 MVI   B$NUMC,B@LLET      SKIP TO LEADING CHAR IN STRING
0D04 C0 87 0867 4034          B   B$GETC          * FUNCTION CHARACTER REFERENCE
0D08 C0 87 0DBC 4035          B   B$SYMB          TRANSLATE THE CHAR REFERENCE
                4036 *****
                4037 *              THE VADDR OF THE TRANSLATED CHARACTER REFERENCE IS *
                4038 *              AT B$BCKT. *
                4039 *****
0D0C 4C 01 DF 1590 4040          MVC   STRAOP(@VADDR,@BR),B$BCKT  SAVE VADDR IN 'STA' OPERAND
0D11 BD 4D 00 4041          CLI   B@CHAR(,@XR),B@LPAR      IF CHAR REF IS AN ARRAY REF
0D14 D0 81 55 4042          BE    BST240(,@BR)          * GO PROCESS ARRAY REFERENCE
                4043 *****
                4044 *              STRING FUNCTION CHARACTER REFERENCE IS A CHARACTER *
                4045 *              VARIABLE. *
                4046 *****
0D17 D2 02 DD 4047          LA     STRSTA(,@BR),@XR        LOAD CADDR OF STA INSTRUCTION
0D1A D0 87 83 4048          B      BST300(,@BR)          GO GENERATE PMC
0D1D 5C 01 E2 DF 4049          MVC   STRCOP(@VADDR,@BR),STRAOP(,@BR)  SET VADDR OPERND OF 'STC'
0D21 D2 02 E0 4050          LA     STRSTC(,@BR),@XR        LOAD CADDR OF 'STC' INSTRUCTION
0D24 D0 87 83 4051          B      BST300(,@BR)          GO GENERATE PMC
0D27 C0 87 1514 4052 BST210 B      B$SCAN          PROCESS 1ST 'STR' ARITH OPERAND
0D2B BD 5D 00 4053          CLI   B@CHAR(,@XR),B@RPAR      IF LENGTH PARM IS NOT PRESENT
0D2E D0 81 38 4054          BE    BST220(,@BR)          * GO GENERATE 'STX' INSTRUCTION
0D31 C0 87 1514 4055          B      B$SCAN          ELSE PROCESS LENGTH PARAMETER
0D35 D0 87 41 4056          B      BST230(,@BR)          GO COMPLETE 'STR' PROCESSING
0D38 D2 02 E3 4057 BST220 LA     STRSTX(,@BR),@XR        LOAD CADDR OF 'STX' INSTRUCTION
0D3B 7C 01 8B 4058          MVI   BST310+@Q(,@BR),B@LSTX-1  SET LENGTH PARM FOR PUT ROUTINE
0D3E D0 87 83 4059          B      BST300(,@BR)          GO GENERATE PMC
                4060 *****
                4061 *              STRING FUNCTION IS PROCESSED. NOW GENERATE CHARACTER *
                4062 *              STACKING FOR ECWRK AND FUNCTION CALL THEN RETURN TO *
                4063 *              PROCESS NEXT ASSIGNMENT LIST ELEMENT. *
                4064 *****
0D41 D2 02 E5 4065 BST230 LA     STRCWR(,@BR),@XR        LOAD CADDR OF 'STC' &CWRK INSTR
0D44 4C 00 E6 159F 4066          MVC   STRWOP-@B1(,@BR),B$WORK-@B1(@B1)  SET VADDR OF &CWRK
0D49 D0 87 83 4067          B      BST300(,@BR)          GO GENERATE PMC
0D4C D2 02 E8 4068          LA     STRFN2(,@BR),@XR        LOAD CADDR OF FNO #2 INSTR
0D4F D0 87 83 4069          B      BST300(,@BR)          GO GENERATE PMC
0D52 D0 87 5F 4070          B      BST250(,@BR)          RETURN TO PROCESS NEXT LIST ELMT
                4071 *****
                4072 *              STRING FUNCTION CHARACTER REFERENCE IS A CHARACTER *
                4073 *              ARRAY REFERENCE. *
                4074 *****
0D55 D0 87 99 4075 BST240 B      BST340(,@BR)          GO PROCESS STR CHAR ARRAY REF
0D58 C0 87 0867 4076          B      B$GETC          LINK TO ADVANCE TEXT CHAR PNTR
0D5C D0 87 27 4077          B      BST210(,@BR)          GO PROCESS 'STR' ARITH OPERANDS
                4078 *****
                4079 *              WHEN THE STRING OPERAND HAS BEEN PROCESSED. *
                4080 *              THIS SECTION WILL RETURN TO THE CONTROL SECTION TO *
                4081 *              CONTINUE PROCESSING THE ASSIGNMENT LIST ELEMENTS. *

```

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 35
			4082		*****				
0D5F	1D	01 1AF9 F9	4083	BST250	CLC	B\$CADR(@CADDR),STRPBA(,@BR) IF CONTROL SECTION CAME FROM			
0D64	F2	81 0D	4084		JE	BST270 * DISK-GO LD & EXEC CNTL SECTION			
			4085		*****				
			4086	*		CONTROL SECTION IS CORE RESIDENT - LOAD BASE REGISTER *			
			4087	*		AND RETURN. *			
			4088		*****				
0D67	4C	01 73 1AF5	4089		MVC	BST260+@OP1(@CADDR,@BR),B\$RTRN SET UP RETURN BRANCH ADDR			
0D6C	35	01 1AF9	4090		L	B\$CADR,@BR LOAD CONTROL SECTION BASE ADDR			
0D70	C0	87 0000	4091	BST260	B	*-* RETURN TO CONTROL SECTION			
			4093		*****				
			4094	*		CONTROL SECTION IS DISK RESIDENT - SET DISTRIBUTOR *			
			4095	*		PARAMETERS TO LOAD AND EXECUTE CONTROL SECTION. *			
			4096		*****				
0D74	5C	01 F6 F9	4097	BST270	MVC	STRCA2(@CADDR,@BR),STRPBA(,@BR) SET UP DISKLOAD CADDR			
0D78	C0	87 0867	4098		B	B\$GETC ADVANCE THE TEXT CHAR POINTER			
0D7C	D2	02 F5	4099		LA	STRAD2(,@BR),@XR LOAD DIST PARAMETERS CADDR			
0D7F	C0	87 073A	4100		B	B\$DST2 GO LOAD & EXEC CONTROL SECTION			



## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 36

```

4102 *****
4103 * THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY, *
4104 * THE PSEUDO INSTRUCTION POINTED TO BY @XR. *
4105 * THE INPUT PARAMETERS ARE AS FOLLOWS: *
4106 * 1. XR REFERENCES THE INSTRUCTION TO BE *
4107 * GENERATED. *
4108 * 2. IF THE LENGTH OF THE INSTRUCTION IS NOT *
4109 * THREE. THE LENGTH MUST BE STORED IN *
4110 * MVI INSTRUCTION (BST310+@Q). *
4111 *****
0D83 74 08 98 4112 BST300 ST BST320+@OP1(,@BR),@ARR SAVE THE RETURN ADDRESS
0D86 34 02 0A40 4113 ST B$PCAD,@XR SET CADDR PARM FOR THE PUT RTN
0D8A 3C 02 0A41 4114 BST310 MVI B$PNBY,B@LLET-1 SET LENGTH PARM FOR THE PUT RTN
0D8E C0 87 093A 4115 B B$PUTC GENERATE PMC IN VIRTUAL MEMORY
0D92 7C 02 8B 4116 MVI BST310+@Q(,@BR),B@LLET-1 MAKE THE SUBROUTINE REUSABLE
0D95 C0 87 0000 4117 BST320 B *-* RETURN TO CALLING SECTION

4119 *****
4120 * THIS SUBROUTINE WILL GENERATE PSEUDO INSTRUCTIONS *
4121 * TO PROCESS A CHARACTER ARRAY REFERENCE. THE INPUT *
4122 * PARAMETERS ARE AS FOLLOWS: *
4123 * 1. THE VIRTUAL ADDRESS OF THE ARRAY DESCRIPTOR *
4124 * IS AT BSBCKT. *
4125 * 2. THE TEXT CHARACTER POINTER REFERENCES THE *
4126 * OPENING PARERTHESIS OF THE ARRAY INDEX. *
4127 *****
0D99 74 08 DC 4128 BST340 ST BST360+@OP1(,@BR),@ARR SAVE THE RETURN ADDRESS
0D9C 4C 01 ED 1590 4129 MVC STR1OP(@VADDR,@BR),B$BCKT SAVE VADDR OF ARRAY DESCRIPTOR
0DA1 4C 01 DF 15A0 4130 MVC STRAOP(@VADDR,@BR),B$WORK SET VADDR OF @WRK IN 'STA' PMC
0DA6 D2 02 DD 4131 LA STRSTA(,@BR),@XR LOAD CADDR OF 'STA' INSTR
0DA9 D0 87 83 4132 B BST300(,@BR) GO GENERATE 'STA' PMC
0DAC C0 87 1514 4133 B B$SCAN GO PROCESS ARRAY INDEX
0DB0 7C 00 8B 4134 MVI BST310+@Q(,@BR),B@LUSF-1 SET LENGTH PARM OF PUT ROUTINE
0DB3 D2 02 F4 4135 LA STRUSF(,@BR),@XR LOAD CADDR OF 'USF' INSTR
0DB6 D0 87 83 4136 B BST300(,@BR) GO GENERATE 'USF' INSTR
0DB9 5C 01 F3 DF 4137 MVC STRFOP(@VADDR,@BR),STRAOP(,@BR) SET VADDR OPRND FOR 'STF'
0DBD D2 02 F1 4138 LA STRSTF(,@BR),@XR LOAD CADDR OF 'STF' INSTR
0DC0 D0 87 83 4139 B BST300(,@BR) GO GENERETE 'STF' INSTR
0DC3 5C 01 F0 ED 4140 MVC STRBOP(@VADDR,@BR),STR1OP(,@BR) SET VADDR OPRND FOR 'STF'
0DC7 D2 02 EE 4141 LA STRSB1(,@BR),@XR LOAD CADDR OF 'SB1' INSTR
0DCA D0 87 83 4142 B BST300(,@BR) GO GENERATE 'SB1' INSTR
0DCD D2 02 F1 4143 LA STRSTF(,@BR),@XR LOAD CADDR OF 'STF' INSTR
0DD0 D0 87 83 4144 B BST300(,@BR) GO GENERATE 'STF &WRK' PMC
0DD3 D2 02 EB 4145 LA STRSC1(,@BR),@XR LOAD CADDR OF 'SC1' INSIR
0DD6 D0 87 83 4146 B BST300(,@BR) GO GENERATE 'SC1' INSTR
0DD9 C0 87 0000 4147 BST360 B *-* RETURN TO CALLING SECTION

```

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 37
				4149	*****				
				4150	*	STRING PROCESSOR SECTION EQUANS,CONSTANTS, AND			*
				4151	*	WORKAREAS.			*
				4152	*****				
0DDD	34		0DDD	4153	STRSTA DC	AL(B@LCOP)(B@CSTA)			STACK ADDRESS OPCODE
0DDE			0DDF	4154	STRAOP DS	CL2			STACK ADDRESS OPERAND
				4155	*				
0DE0	28		0DE0	4156	STRSTC DC	AL(B@LCOP)(B@CSTC)			STACK CHARACTER FIELD OPCODE
0DE1			0DE2	4157	STRCOP DS	CL2			STACK CHARACTER FIELD OPERAND
				4158	*				
0DE3	3C		0DE3	4159	STRSTX DC	AL(B@LCOP)(B@CSTX)			STACK EXEC CTRL CODE OPCODE
0DE4	FF		0DE4	4160	STRXOP DC	XL1 'FF'			STACK EXEC CTRL CODE OPERAND
				4161	*				
0DE5	28		0DE5	4162	STRCWR DC	AL(B@LCOP)(B@CSTC)			STACK CHAR OF CWRK OPCODE
0DE6	F500		0DE7	4163	STRWOP DC	AL2(B\$CWRK)			STACK CHAR OF CWRK OPERAND
				4164	*				
0DE8	12		0DE8	4165	STRFN2 DC	AL(B@LCOP)(B@CFN0)			FUNCT CALL-NO ARGUMENT OPCODE
0DE9	5120		0DEA	4166	DC	AL2(V\$CCON)			FUNCT CALL-NO ARGUMENT OPERAND
				4167	*				
0DEB	2A		0DEB	4168	STRSC1 DC	AL(B@LCOP)(B@CSC1)			STACK CHAR ARRAY ELEMENT OPCODE
0DEC			0DED	4169	STR1OP DS	CL2			STACK CHAR ARRAY ELEMENT OPERAND
				4170	*				
0DEE	3A		0DEE	4171	STRSB1 DC	AL(B@LCOP)(B@CSB1)			STACK CHAR ARRAY ADDR OPCODE
0DEF			0DF0	4172	STRBOP DS	CL2			STACK CHAR ARRAY ADDR OPERAND
				4173	*				
0DF1	20		0DF1	4174	STRSTF DC	AL(B@LCOP)(B@CSTF)			STACK FLOATING PT VALUE OPCODE
0DF2			0DF3	4175	STRFOP DS	CL2			STACK FLOATING PT VALUE OPERAND
				4176	*				
0DF4	26		0DF4	4177	STRUSF DC	AL(B@LCOP)(B@CUSF)			UNSTACK FLTING PT VALUE OPCODE
				4178	*				
			0DF5	4179	STRAD2 EQU	*			DIST PARAMETER ADDR
0DF5			0DF6	4180	STRCA2 DS	CL(@CADDR)			CONTROL SECTION CORE ADDRESS
0DF7	10		0DF7	4181	DC	AL1(B@DSML)			PHYSICAL SECTOR ADDRESS
N04 0DF8	0000		0DF9	4182	STRPBA DC	AL(@CADDR)(B\$CSIF)			PROCESSOR DISK BUFFER CADDR
				4183	*****				
				4184	*	END OF LET-STRING PROCESSOR SECTION			*
				4185	*****				

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 38

```

0E00          4187      ORG   BSTRLT+2*B@BLSZ          PLACE SEGMENT AT PAGE BOUNDARY
0E00          4188      USING *,@BR                  ESTABLISH BASE ADDRESS
0E00          4189      *****
0E00          4190      *          THE ASSIGNMENT LIST HAS BEEN PROCESSED. NOW GENERATE A *
0E00          4191      *          BRANCH INSTRUCTION IMAGE IN VIRTUAL MEMORY(AT EXECUTION *
0E00          4192      *          TIME THIS BRANCH WILL TRANSFER CONTROL BEYOND THE SET *
0E00          4193      *          UP FOR THE RIGHT SIDE TO THE NEXT SEQUENTIAL STATEMENT. *
0E00          4194      *****
0E00 D2 02 D8    4195 BST400 LA      TRMBIC(,@BR),@XR          LOAD CADDR OF 'BRA' INSTRUCTION
0E03 D0 87 C2    4196      B      BST550(,@BR)          GO GENERATE PMC
0E03          4197      *****
0E03          4198      *          ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND *
0E03          4199      *          IN THE FIRST BRANCH INSTRUCTION IMAGE (BST080) *
0E03          4200      *****
0E06 0C 01 19EF 1AF7 4201      MVC   B$BRVA,B$BROP(@VADDR)      SET BRANCH TABLE VADDR PARM
N04 0E0C 00 00 0000 00 4202      SLC   B$BRVA,TRMBNI(@VADDR,@BR) * FOR THE BRA IMAGE OPERAND
0E11 0C 01 19F1 0A43 4203      MVC   B$BRLN,B$PVAD(@VADDR)      SET BRANCH TABLE LINE NO. PARM
0E17 C0 87 1996    4204      B      B$BTAB          LINK TO SET UP RESOLUTION
0E17          4205      *****
0E17          4206      *          GENERATE PSEUDO INSTRUCTIONS TO UNSTACK THE SOURCE *
0E17          4207      *          CHARACTERS INTO ECWRK. THE FIRST BRANCH INSTRUCTION *
0E17          4208      *          PASSES CONTROL TO THIS INSTRUCTION SEQUENCE. *
0E17          4209      *****
0E1B D2 02 DD    4210      LA      TRMSTA(,@BR),@XR          LOAD CADDR OF 'STA' INSTRUCTION
0E1E 4C 00 DE 159F 4211      MVC   TRMAOP-@B1(,@BR),B$WORK-@B1(@B1) SET VADDR OF &CWRK
0E23 D0 87 C2    4212      B      BST550(,@BR)          GO GENERATE PMC
0E26 C0 87 0867    4213      B      B$GETC          ADVANCE TEXT CHARACTER POINTER
0E2A BD 7D 00    4214      CLI     B@CHAR(,@XR),B@SQUO        IF THE OPERAND IS A LITERAL
0E2D F2 01 0B    4215      JNE     BST410          * BYPASS BDSYMB CALL
0E30 3C 00 0873    4216      MVI     B$NUMC,B@GETS          DISABLE THE GET ROUTINE
0E34 C0 87 14B0    4217      B      B$CSCN          GO PROCESS CHAR LITERAL OPERAND
0E38 F2 87 5C     4218      J      BST600          CONTINUE PROCESSING
0E3B 3C 01 1BAC    4219 BST410 MVI     B$SSTA,@B1          ENABLE BDSYMB DETECTION OF 'STR'
0E3F C0 87 0DBC    4220      B      B$SYMB          TRANSLATE SOURCE SYMBOL
0E43 3C 00 159E    4221      MVI     B$KWSW,@ZERO          TURN OFF KEYWORD SWITCH
0E47 3D 00 0E42    4222      CLI     B$CRSW,@ZERO          IF SOURCE SYMBOL IS NOT A CHAR
0E4B D0 81 65     4223      BE      BST500(,@BR)          * REF GO SET UP 'STR' PROCESSING
0E4B          4224      *****
0E4B          4225      *          SOURCE SYMBOL IS A CHARACTER REFERENCE (ARRAY, VARIABLE, *
0E4B          4226      *          OR CONSTANT). *
0E4B          4227      *****
0E4E D2 02 97     4228 BST440 LA      BST600(,@BR),@XR          LOAD CADDR OF RETURN ADDR
0E51 34 02 150D    4229 BST460 ST      B$CRAD,@XR          SET RETURN ADDR IN BECSCN
0E55 34 01 1509    4230      ST      B$CRBS,@BR          SAVE BASE REG CONTENT IN BECSCN
0E59 C2 01 14BB    4231      LA      B$CBAS,@BR          LOAD BECSCN BASE ADDRESS
N04 0E5D 00 00 0000 4232      L      B$GPIR,@XR          LOAD TEXT CHARACTER POINTER
0E61 C0 87 14CC    4233      B      B$CSTR          GO TO CHAR EXPRSSN SCAN ROUTINE

```

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 39
				4235		*****		
				4236	*	SOURCE SYMBOL IS A STRING FUNCTION	*	
				4237		*****		
	0E65	3C	03	0873	4238	BST500 MVI	B\$NUMC,B@LLET	SET GET RTN TO SKIP 'STRC'
	0E69	C0	87	0867	4239	B	B\$GETC	ADVANCE TEXT CHARACTER POINTER
	0E6D	C0	87	0DBC	4240	B	B\$SYMB	TRANSLATE STRING CHARACTER REF
	0E71	D2	02	77	4241	LA	BST540(, @BR), @XR	LOAD RETURN ADDRESS
	0E74	D0	87	51	4242	B	BST460(, @BR)	GO TO CHAR EXPRSSN SCAN ROUTINE
	0E77	C0	87	1514	4243	BST540 B	B\$SCAN	PROCESS 1ST ARITH OPERAND
	0E7B	BD	5D	00	4244	CLI	B@CHAR(, @XR), B@RPAR	IF NEXT OPERAND IS PRESENT THEN
	0E7E	D0	01	8D	4245	BNE	BST545(, @BR)	* PROCESS IT
	0E81	D2	02	E0	4246	LA	TRMSTX(, @BR), @XR	ELSE LOAD CADDR OF STX INSTRUCTN
	0E84	7C	01	CA	4247	MVI	BST560+@Q(, @BR), B@LSTX-1	SET LENGTH PARM FOR PUT RTN
	0E87	D0	87	C2	4248	B	BST550(, @BR)	* AND GEN PMC
	0E8A	D0	87	91	4249	B	BST547(, @BR)	GO FINISH STR PROCESSING
	0E8D	C0	87	1514	4250	BST545 B	B\$SCAN	PROCESS LAST OPERAND
	0E91	D2	02	E5	4251	BST547 LA	TRMFN1(, @BR), @XR	LOAD CADDR OF PMC FOR 'FNO' #1
	0E94	D0	87	C2	4252	B	BST550(, @BR)	GO GENERATE PMC
	0E97	D2	02	E8	4253	BST600 LA	TRMUSC(, @BR), @XR	LOAD CADDR OF UNSTACK PMC
	0E9A	7C	01	CA	4254	MVI	BST560+@Q(, @BR), B@LUSC-1	SET LENGTH PARM FOR PUT RTN
	0E9D	D0	87	C2	4255	B	BST550(, @BR)	GO GENERATE PMC
				4257		*****		
				4258	*	INSTRUCTIONS TO PROCESS THE SOURCE VALUE ARE COMPLETE.	*	
				4259	*	NON GENERATE THE RETURN BRANCH INSTRUCTION. THIS	*	
				4260	*	INSTRUCTION WILL TRANSFER CONTROL TO THE LIST	*	
				4261	*	ASSIGNMENT SEQUENCE AFTER THE SOURCE VALUE IS STORED	*	
				4262	*	INTO ECNRK.	*	
				4263		*****		
N04	0EA0	00	00	00 0000	4264	MVC	TRNBOP(@VADDR, @BR), B\$BROP	SET VADDR OPRND OF RTRN BRANCH
	0EA5	D2	02	E2	4265	LA	TRMBRC(, @BR), @XR	LOAD CADDR OF 'BRA' INSTRUCTION
	0EA8	D0	87	C2	4266	B	BST550(, @BR)	GO GENERATE PMC
				4267		*****		
				4268	*	RESOLVE SECOND BRANCH INSTRUCTION IMAGE (BST500).	*	
				4269		*****		
	0EAB	0C	01	19EF 19F1	4270	MVC	B\$BRVA, B\$BRLN(@VADDR)	SET BRANCH TABLE VADDR PARM
	0EB1	1F	01	19EF DC	4271	SLC	B\$BRVA, TRMBN1(@VADDR, @BR)	* FOR 'BRA' IMAGE INSTR
	0EB6	3A	07	071D	4272	SBN	B\$NXSW, B\$NXMK	SET NXT STMT SWCH ON TO ESTBLSH
				4273	*			* LINE NO. PARM
				4274		*****		
				4275	*	RETURN TO COMPILER DISTRIBUTOR	*	
				4276		*****		
	0EBA	C0	87	0867	4277	B	B\$GETC	LINK TO ADVANCE TEXT CHAR PTR
	0EBE	C0	87	0700	4278	B	B\$DIST	

## S/3 BASIC COMPILER -LET- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 40

```

4280 *****
4281 * THIS SUBROUTINE WILL GENERATE, IN VIRTUAL MEMORY, *
4282 * THE PSEUDO INSTRUCTION POINTED TO BY @XR. THE *
4283 * INPUT PARAMETERS ARE AS FOLLOWS: *
4284 * 1. XR REFERENCES THE INSTRUCTION TO BE *
4285 * GENERATED. *
4286 * 2. IF THE LENGTH OF THE INSTRUCTION IS NOT *
4287 * THREE, THE LENGTH MUST BE STORED IN A *
4288 * MVI INSTRUCTION (BST560+@Q). *
4289 *****
0EC2 74 08 D7 4290 BST550 ST BST570+@OP1(,@BR),@ARR SAVE THE RETURN ADDRESS
0EC5 34 02 0A40 4291 ST B$PCAD,@XR SET CADDR PARM FOR THE PUT RTN
0EC9 3C 02 0A41 4292 BST560 MVI B$PNBY,B@LLET-1 SET LENGTH PARM FOR THE PUT RTN
0ECD C0 87 093A 4293 B B$PUTC GENERATE PMC IN VIRTUAL MEMORY
0ED1 7C 02 CA 4294 MVI BST560+@Q(,@BR),B@LLET-1 MAKE SUBROUTINE REUSABLE
0ED4 C0 87 0000 4295 BST570 B *- * RETURN TO CALLING SECTION

4297 *****
4298 * TERMINATION SECTION CONSTANTS, EQUATES AND WORKAREAS *
4299 *****
0ED8 46 0ED8 4300 TRMBIC DC AL(B@LCOP)(B@CBRA) UNCONDITIONAL BRANCH OPCODE
0ED9 0000 0EDA 4301 DC AL(@VADDR)(@ZERO) BRANCH IMAGE OPERAND
4302 *
0EDB 0001 0EDC 4303 TRMBN1 DC IL(@VADDR)'1' BINARY ONE
4304 *
0EDD 34 0EDD 4305 TRMSTA DC AL(B@LCOP)(B@CSTA) STACK ADDRESS OPCODE
0EDE F500 0EDF 4306 TRMAOP DC AL2(B$CWRK) STACK ADDRESS OPERAND
4307 *
0EE0 3C 0EE0 4308 TRMSTX DC AL(B@LCOP)(B@CSTX) STACK EXEC CTRL CODE OPCODE
0EE1 FF 0EE1 4309 DC XL1'FF' STACK EXEC CTRL CODE OPERAND
4310 *
0EE2 46 0EE2 4311 TRMBRC DC AL(B@LCOP)(B@CBRA) UNCONDITIONAL BRANCH OPCODE
0EE3 0EE4 4312 TRMBOP DS CL2 UNCONDITIONAL BRANCH OPERAND
4313 *
N04 0EE5 00 0EE5 4314 TRMFN1 DC AL(B@LCOP)(B@CFNO) FUNC CALL-NO ARGUMENT OPCODE
0EE6 5100 0EE7 4315 DC AL(@VADDR)(V$CSSR) FUNC CALL-NO ARGUMENT OPERAND
4316 *
0EE8 2C 0EE8 4317 TRMUSC DC AL(B@LCOP)(B@CUSC) UNSTACK CHAR ELEMENT OPCODE
0EE9 01 0EE9 4318 DC XL1'01' UNSTACK CHAR ELEMENT OPERAND
4319 *****
4320 * END OF LET-TERMINATION SECTION *
4321 *****

```

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 41
		4323		*****			
		4324	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		4325	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		4326	*				*
		4327		*****			*
		4328	*	*STATUS			*
		4329	*	VERSION 1 MODIFICATION 4			*
		4330	*				*
		4331	*	*FUNCTION			*
		4332	*	BSTRIF IS EXECUTED TO TRANSLATE IF STATEMENTS WITH SUB-STRING			*
		4333	*	OPERANDS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE			*
		4334	*	PSEUDO INSTRUCTION SEQUENCE AND TO PLACE THE PSEUDO INSTRUCTION			*
		4335	*	SEQUENCE IN VIRTUAL MEMORY.			*
		4336	*				*
		4337	*	*ENTRY POINTS			*
		4338	*	BSTRIF HAS ONLY ONE ENTRY POINT:			*
		4339	*	BSTRIF - TRANSLATE IF STATEMENTS			*
		4340	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		4341	*	B BSTRIF			*
		4342	*				*
		4343	*	*INPUT			*
		4344	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		4345	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		4346	*	LEADING KEYWORD, IF.			*
		4347	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		4348	*	FIRST CHARACTER IN THE LEADING KEYWORD, IF.			*
		4349	*				*
		4350	*	*OUTPUT			*
		4351	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		4352	*	GENERATED BY BSTRIF IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		4353	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		4354	*	SEQUENCES.			*
		4355	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		4356	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		4357	*				*
		4358	*	*EXTERNAL REFERENCES			*
		4359	*	* B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		4360	*	* B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRTUAL			*
		4361	*	MEMORY OUTPUT ROUTINE.			*
		4362	*	* B\$CSCN - (B\$CSTR) - ENTRY TO COMPILER CHARACTER EXPRESSION			*
		4363	*	SCAN ROUTINE.			*
		4364	*	* B\$SCAN - ENTRY TO COMPILER ARITHMETIC EXPRESSION SCAN ROUTINE.			*
		4365	*	* B\$DIST - (BSDST2) - ENTRY TO COMPILER DISTRIBUTOR ROUTINE.			*
		4366	*	* B\$SYMB - (B\$CRSW, B\$SSTA) - ENTRY TO COMPILER SYMBOL			*
		4367	*	TRANSLATION ROUTINE.			*
		4368	*	* B\$ZDBN - ENTRY TO COMPILER DECIMAL TO BINARY CONVERSION			*
		4369	*	ROUTINE.			*
		4370	*	* B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO COMPILER BRANCH ADDRESS			*
		4371	*	TABLE ROUTINE.			*
		4372	*	* B\$COMN - (B\$PRM1, B\$CADR) - COMPILER CORE RESIDENT COMMON SCTN.			*
		4373	*				*
		4374	*	*EXITS, NORMAL			*
		4375	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		4376	*				*
		4377	*	*EXITS, ERROR			*
		4378	*	N/A			*



## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 42
		4379	*		*
		4380	*	*TABLES/WORK AREAS	*
		4381	*	* * RELATIONAL OPERATOR - CONDITION CODE TABLE - EXTERNAL TO	*
		4382	*	BSTRIF, THIS 14-BYTE TABLE IS USED TO DETERMINE THE ONE BYTE	*
		4383	*	BRANCH-ON-CONDITION CONDITION CODE WHICH CORRESPONDS TO THE	*
		4384	*	RELATIONAL OPERATOR PRESENT IN THE SOURCE STATEMENT. THE	*
		4385	*	ENTRIES ARE TWO BYTES IN LENGTH, EACH TWO-BYTE ENTRY CONSISTS	*
		4386	*	ONE-BYTE HEXIDECIMAL REPRESENTATION OF THE RELATIONAL	*
		4387	*	OPERATOR AND A ONE-BYTE BRANCH-ON-CONDITION CONDITION CODE.	*
		4388	*	THE TABLE IS LOCATED IN THE COMPILER CORE RESIDENT COMMON	*
		4389	*	SECTION, BZCOMN.	*
		4390	*		*
		4391	*	*ATTRIBUTES	*
		4392	*	BSTRIF IS NATURALLY RELOCATABLE AND REUSABLE.	*
		4393	*		*
		4394	*	*CHARACTER CODE DEPENDENCY	*
		4395	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A	*
		4396	*	PARTICULAR INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER	*
		4397	*	SET.	*
		4398	*		*
		4399	*	*NOTES	*
		4400	*	ERROR PROCEDURES	*
		4401	*	N/A	*
		4402	*		*
		4403	*	REGISTER USAGE	*
		4404	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
		4405	*		*
		4406	*	SAVED/RESTORED AREAS	*
		4407	*	N/A	*
		4408	*		*
		4409	*	MODIFICATION CONSIDERATIONS	*
		4410	*	BSTRIF CROSSES A SECTOR BOUNDARY AND RESIDES ON TWO SECTORS.	*
		4411	*	CO-RESIDENT ON THE SECOND ONE WITH BKSUBG. ANY MODIFICATIONS	*
		4412	*	MUST MAINTAIN LINKAGE BETWEEN THE TWO SECTORS. CONSIDER	*
		4413	*	CHANGE IN THE ENTRY ADDRESS OF BKSUBG, AND REALIZE THE	*
		4414	*	LIMITATION OF THE SECTOR BOUNDARY UPON SIZE.	*
		4415	*		*
		4416	*	REQUIRED MODULES	*
		4417	*	@SYSEQ - COMMON SYSTEM EQUATES.	*
		4418	*	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES.	*
		4419	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE THE NUCLEUS EQUATES.	*
		4420	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
		4421	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
		4422	*	@ERMEQ - ERROR MESSAGE EQUATES.	*
		4423	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
		4424	*	\$B\$EQU - COMPILER FIXED ADDRESS EQUATES.	*
		4425	*	\$B@EQU - COMPILER SYSTEM EQUATES.	*
		4426	*		*
		4427	*	OTHER	*
		4428	*	BSTRIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
		4429	*	*****	*
0F00		4431		ORG *,256,0	PLACE MODULE AT PAGE BOUNDARY
	0F00	4432		USING *,@BR	ESTABLISH BASE ADDRESSING
	0F00	4433		BSTRIF EQU *	ENTRY POINT
		4434		*****	



## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 43
			4435	*	THIS IS THE RETURN ENTRY POINT FROM PROCESSING THE	*
			4436	*	RELATIONAL OPERATOR IN SECTION TWO OF BSTRIF. RESET	*
			4437	*	THE LOOP COUNTER AND CONTINUE TO PROCESS THE SECOND	*
			4438	*	OPERAND OF THE SUB-STRING IF STATEMENT.	*
			4439	*****	*****	*****
0F00	7C	01 F4	4440	MVI	BITLSW(, @BR), @B1	RESTORE LOOP COUNTER
0F03	D0	87 0D	4441	B	BIT100(, @BR)	GO PROCESS SECOND OPERAND
			4442	*****	*****	*****
			4443	*	ADVANCE TEXT CHARACTER POINTER TO FIRST CHARACTER	*
			4444	*	OF IDENTIFIER AND INITIALIZE LOOP COUNTER TO ZERO.	*
			4445	*****	*****	*****
			0F06 4446	BITRE1 EQU	*	PRIMARY ENTRY POINT
0F06	3C	02 0873	4447	MVI	B\$NUMC, B@LKIF	SET GET ROUTINE TO SKIP KEYWORD
0F0A	7C	00 F4	4448	MVI	BITLSW(, @BR), @ZERO	INITIALIZE LOOP SWITCH TO ZERO
0F0D	74	01 E8	4449	BIT100 ST	BITCA2(, @BR), @BR	SAVE BSTRIF CORE ADDRESS
0F10	C0	87 0867	4450	B	B\$GETC	SET TEXT CHARACTER POINTER
			4451	*****	*****	*****
			4452	*	PROCESS THE IDENTIFIER VIA A CALL TO BDSYMB. IF THE	*
			4453	*	IDENTIFIER IS A CHARACTER REF. THE SWITCH BSCRSW WILL	*
			4454	*	BE ON AND THE VADDR OF THE REF WILL BE LOCATED AT	*
			4455	*	B\$BCKT.	*
			4456	*****	*****	*****
0F14	BD	7D 00	4457	CLI	B@CHAR(, @XR), B@SQUO	IF THE OPERAND IS A LITERAL
0F17	F2	01 0B	4458	JNE	BIT110	* BYPASS BDSYMB CALL
			4459	*****	*****	*****
			4460	*	OPERAND IS A CHARACTER LITERAL, DON'T USE BDSYMB	*
			4461	*****	*****	*****
0F1A	3C	00 0873	4462	MVI	B\$NUMC, B@GETS	DISABLE THE GET ROUTINE
0F1E	C0	87 14B0	4463	B	B\$CSCN	GO PROCESS CHAR LITERAL OPERAND
0F22	F2	87 70	4464	J	BIT200	CONTINUE PROCESSING
0F25	3C	01 1BAC	4465	BIT110 MVI	B\$SSTA, @B1	ENABLE DETECTION OF 'STR'
0F29	C0	87 0DBC	4466	B	B\$SYMB	TRANSLATE THE IDENTIFIER
0F2D	3C	00 159E	4467	MVI	B\$KWSW, @ZERO	CLEAR KEYWORD SWITCH
0F31	3D	00 0E42	4468	CLI	B\$CRSW, @ZERO	IS CHARACTER REF SWITCH ON ?
0F35	D0	01 7E	4469	BNE	BIT160(, @BR)	YES-GO PROCESS CHAR REF

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 44

```

4471 *****
4472 * THE IDENTIFIER IS A STR FUNCTION. SO ADVANCE TEXT *
4473 * CHARACTER POINTER TO THE LEADING CHARACTER OF THE *
4474 * CHARACTER REF IN THE STR FUNCTION AND PROCESS THE *
4475 * REST OF THE STRING. *
4476 *****
N04 0F38 00 00 0000 4477 MVI B$NUMC,B@LIET-1 SET GET ROUTINE TO SKIP 'STR'
0F3C C0 87 0867 4478 B B$GETC ADVANCE TEXT CHARACTER POINTER
0F40 C0 87 14B0 4479 B B$CSCN PROCESS CHAR REF WITHIN 'STR'
0F44 3C 00 159E 4480 MVI B$KWSW,@ZERO TURN OFF KETWORK SWITCH
0F48 C0 87 1514 4481 B B$SCAN PROCESS FIRST 'STR' PARAMETER
0F4C BD 5D 00 4482 CLI @ZERO(,@XR),B@RPAR IS 2ND PARAMETER MISSING ?
0F4F D0 01 64 4483 BNE BIT120(,@BR) NO-GO PROCESS 2ND PARAMETER
0F52 D2 02 F6 4484 LA BITSTX(,@BR),@XR SET CADDR PARAMETER FOR PUT RTN
0F55 34 02 0A40 4485 ST B$PCAD,@XR * WITH 'STX' INSTR ADDR
0F59 3C 01 0A41 4486 MVI B$PNBY,B@LSTX-1 SET LNGTH PARAMETER FOR PUT RTN
0F5D C0 87 093A 4487 B B$PUTC GO GENERATE PMC
0F61 D0 87 68 4488 B BIT140(,@BR) GO CONTINUE PROCESSING
0F64 C0 87 1514 4489 BIT120 B B$SCAN PROCESS LAST 'STR' PARAMETER
0F68 D2 02 F8 4490 BIT140 LA BITFNO(,@BR),@XR LOAD CADDR OF 'FNO' INSTRUCTION
0F6B 34 02 0A40 4491 ST B$PCAD,@XR SET CADDR PARM FOR PUT ROUTINE
N04 0F6F 00 00 0000 4492 MVI B$PNBY,B@LFNO-1 SET LENGTH PARM FOR PUT ROUTINE
0F73 C0 87 093A 4493 BIT150 B B$PUTC LINK TO GENERATE PMC
0F77 C0 87 0867 4494 B B$GETC ADVANCE TEXT CHARACTER POINTER
0F7B D0 87 95 4495 B BIT200(,@BR) GO SET LOOP SWITCH VALUE

```

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 45
			4497		*****	
			4498	*	THE OPERAND JUST PROCESSED BY BDSYMB WAS A CHARACTER	*
			4499	*	REFERENCE. THE VADDR OF THE REFERENCE IS CONTAINED	*
			4500	*	AT RSBCKT AND THE TEXT CHARACTER POINTER REFERENCES	*
			4501	*	THE CHARACTER FOLLOWING THE LAST CHARACTER OF THE	*
			4502	*	IDENTIFIER.	*
			4503		*****	
0F7E	D2 02 95		4504	BIT160 LA	BIT200(, @BR), @XR	SAVE RETURN ADDRESS
0F81	34 02 150D		4505	ST	B\$CRAD, @XR	* IN BECSCN
0F85	34 01 1509		4506	ST	B\$CRBS, @BR	SAVE BASE REGISTER FOR RETURN
			4507	*		* FROM BECSCN
0F89	C2 01 14BB		4508	LA	B\$CBAS, @BR	LOAD BECSCN BASE REGISTER
0F8D	35 02 0878		4509	L	B\$GPTR, @XR	LOAD TEXT CHARACTER POINTER
0F91	C0 87 14CC		4510	B	B\$CSTR	GO PROCESS CHAR REF
			4511		*****	
			4512	*	THE OPERAND HAS BEEN PROCESSED. NOW INCREMENT THE	*
			4513	*	LOOP SWITCH AND DETERMINE IF PROCESSING IS FINISHED.	*
			4514		*****	
0F95	5E 00 F4 F5		4515	BIT200 ALC	BITLSW(@B1, @BR), BIT001(, @BR)	INCREMENT LOOP SWITCH BY 1
0F99	7D 02 F4		4516	CLI	BITLSW(, @BR), @CADDR	IS LOOP SWITCH * 2 ?
0F9C	D0 81 A9		4517	BE	BIT300(, @BR)	YES-GO TO TERMINATION CODE
			4518		*****	
			4519	*	LOOP SWITCH = 1, SO WE NOW MUST COMPUTE THE CONDITION	*
			4520	*	CODE WHICH CORRESPONDS TO THE RELATIONAL OPERATOR(S)	*
			4521	*	IN THE BASIC STATEMENT. WE MUST ACCESS SECTION TWO	*
			4522	*	IN ORDER TO PROCESS THE RELATIONAL OPERATOR.	*
			4523		*****	
0F9F	34 01 1AF9		4524	ST	B\$CADR, @BR	SAVE OPERAND PROC SECTION CADDR
0FA3	7C 00 E6		4525	MVI	BIT390+@D1(, @BR), @ZERO	SAVE DISP INTO SEGMENT 2
0FA6	F2 87 0F		4526	J	BIT340	GO ACCESS SEGMENT 2
			4527		*****	
			4528	*	SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN'	*
			4529	*	TO ADVANCE THE TEXT CHARACTER POINTER TO THE LINE NO.	*
			4530		*****	
0FA9	3C 04 0873		4531	BIT300 MVI	B\$NUMC, B@LTHN	SET GET RTN TO SKIP KEYWORD
0FAD	C0 87 0867		4532	B	B\$GETC	ADVANCE TEXT CHAR POINTER
0FB1	7C 4A E6		4533	MVI	BIT390+@D1(, @BR), BITTRM	SAVE TERMINATION DISPLACEMENT
			4534		*****	
			4535	*	CONVERT LINE NO. FROM DECIMAL TO BINARY	*
			4536		*****	
0FB4	C0 87 19F2		4537	B	B\$ZDBN	LINK TO CONVERT LINE NUMBER

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 46
			4539		*****	
			4540	*	ACCESS PART 2 OF IF STATEMENT PROCESSOR TO	*
			4541	*	COMPLETE PSEUDOCODE GENERATION.	*
			4542		*****	
0FB8	5D	01 E8 F1	4543	BIT340 CLC	BITCA2(,@BR),BITPBA(@CADDR,@BR) IF CURR SEG CAME FR DISK	
0FBC	F2	81 10	4544	JE	BIT360 * GO LOAD & EXEC 2ND SEGMENT	
			4545		*****	
			4546	*	CURRENT SEGMENT WAS CORE RESIDENT - TEST WHETHER 2ND	*
			4547	*	SEGMENT HAS ALSO BEEN LOADED INTO CORE.	*
			4548		*****	
0FBF	5C	01 EB ED	4549	BIT350 MVC	BITFCP(,@BR),BITFPE(@CADDR,@BR) SET FINAL CORE PAGE	
0FC3	4E	00 EA 043B	4550	ALC	BITFCP-1(,@BR),\$EXFTR(@B1) CALC MAX PROCESSOR CORE PAGE	
0FC8	5D	01 E8 EB	4551	CLC	BITCA2(,@BR),BITFCP(@CADDR,@BR) IF 2ND SEGMENT IN CORE	
0FCC	F2	82 0F	4552	JL	BIT380 * GO SET TO EXEC 2ND SEGMENT	
			4553		*****	
			4554	*	2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR	*
			4555	*	PARAMETERS FOR CORELOADING & EXECUTING 210 SEGMENT	*
			4556		*****	
0FCF	5C	01 E8 F1	4557	BIT360 MVC	BITCA2(,@BR),BITPBA(@CADDR,@BR) SET UP DISKLOAD CADDR	
			4558		*****	
			4559	*	EXIT TO DISTRIBUTOR TO ACCESS 2ND SEGMENT	*
			4560		*****	
0FD3	D2	02 E7	4561	BIT370 LA	BITAD2(,@BR),@XR LOAD DIST PARM CADDR	
0FD6	5C	00 E8 E6	4562	MVC	BITCA2(@B1,@BR),BIT390+@D1(,@BR) SET CADDR TERM SECTION	
0FDA	C0	87 073A	4563	B	B\$DST2 GO LOAD & EXEC 2ND SEGMENT	
			4564		*****	
			4565	*	2ND SEGMENT IS CORE RESIDENT - BRANCH TO NEST	*
			4566	*	CONSECUTIVE CORE APGE & CONTINUE EXECUTION	*
			4567		*****	
0FDE	75	01 E8	4568	BIT380 L	BITCA2(,@BR),@BR LOAD THE BASE ADDRESS FOR	
0FE1	76	01 EF	4569	A	BITBLS(,@BR),@BR * 2ND SEGMENT	
0FE4	D0	87 00	4570	BIT390 B	BITSG2(,@BR) GO EXECUTE THE 2ND SEGMENT	

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 47
			4572	*****	*****	
			4573	*	CONSTANTS & MORKAREAS TO ACCESS THE 2ND SEGMENT	*
			4574	*****	*****	
			0000	4575	BITSG2 EQU 0	PAGE 2 ENTRY PT DISP
			004A	4576	BITTRM EQU X'4A'	DISP TO TERM ENTRY IN SECTION 2
			0FE7	4577	BITAD2 EQU *	DISTR PARMS FOR SEG-2 EXEC
0FE7			0FE8	4578	BITCA2 DS CL(@CADDR)	IF SEGMENT CORE ADDRESS
0FE9	20		0FE9	4579	DC AL1(B@DSIF+4)	BSTRIF SEG-2 PHYS SECTOR ADDR
0FEA			0FEB	4580	BITFCP DS CL(@CADDR)	FINAL AVAILABLE CORE PAGE ADDR
0FEC	1F00		0FED	4581	BITFPE DC AL(@CADDR)(B\$CSXA-B@BLSZ)	FINAL PAGE BEFORE EXTENSION
0FEE	0100		0FEF	4582	BITBLS DC AL(@CADDR)(B@BLSZ)	LENGTH OF CORE PAGE
0FF0	0600		0FF1	4583	BITPBA DC AL(@CADDR)(B\$CSBF)	PROCESSOR DISK BUFFER CADDR
0FF2	0001		0FF3	4584	BITBN1 DC IL(@VADDR)'1'	BINARY 1
			4585	*****	*****	
			4586	*	CONSTANTS, PSUEDO INSTRUCTION IMAGES AND WORKAREAS	*
			4587	*****	*****	
0FF4			0FF4	4588	BITLSW DS CL1	LOOP SWITCH
0FF5	01		0FF5	4589	BIT001 DC XL1'01'	INCR FOR LOOP SWITCH VALUE
			4590	*		
0FF6	3C		0FF6	4591	BITSTX DC AL(B@LCOP)(B@CSTX)	STACK EXEC CTRL CODE OPCODE
0FF7	FF		0FF7	4592	DC XL1'FF'	STACK EXEC CTRL CODE OPERAND
			4593	*		
N04	0FF8 00		0FF8	4594	BITFNO DC AL(B@LCOP)(B@CFNO)	FUNCTION CALL-NO ARGUMENT OPCODE
	0FF9 5100		0FFA	4595	BITOOP DC AL2(V\$CSSR)	FUNCTION CALL-NO ARGUMENT OPERAND

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 48
			4597		*****	
			4598	*	ESTABLISH IF SEGMENT 2 ADDRESSABILITY	*
			4599		*****	
	1000		4600	ORG	BSTRIF+B@BLSZ BEGIN SEGMENT 2 AT PAGE BNDRY	
			4601	USING	*,@BR DEFINE SEGMENT 2 BASE ADDRESS	
			4602		*****	
			4603	*	THIS SECTION WILL SEARCH THE RELATIONAL OPERATOR	*
			4604	*	TABLE FOR THE CONDITION CODE AND SAVE THE CONDITION	*
			4605	*	CODE AT ISPRM1 FOR LATER USE IN A BRANCH-ON-CONDITION	*
			4606	*	PSEUDO INSTRUCTION.	*
			4607		*****	
			1000 4608	BITREL EQU *	RELATIONAL OPERATOR ENTRY POINT	
	1000 35 02 0878		4609	L	B\$GPTR,@XR LOAD TEXT CHARACTER POINTER	
			4610		*****	
			4611	*	STORE 1ST RELATIONAL OPERATOR IN OPERAND OF CLI INSTR	*
			4612		*****	
	1004 6C 00 2B 00		4613	MVC	BIT280+@Q(@B1,@BR),B@CHAR(,@XR) STORE 1ST RELATNL OPTR	
			4614		*****	
			4615	*	CHECK FOR COMPOUND RELATIONAL OPERATOR	*
			4616		*****	
	1008 C0 87 0867		4617	B	B\$GETC ADVANCE TEXT CHARACTER PTR	
	100C BD 7E 00		4618	CLI	B@CHAR(,@XR),B@EQU	IF CHARACTER IS '='
	100F D0 81 1F		4619	BE	BIT240(,@BR)	* GO COMPUTE OPERATOR
N04	1012 00 00 00		4620	CLI	B@CHAR(,@XR),BAGRTR	IF CHARACTER IS '>'
	1015 D0 81 1F		4621	BE	BIT240(,@BR)	* GO COMPUTE OPERATOR
			4622		*****	
			4623	*	THE OPERATOR IS NOT COMPOUND-DISABLE GET ROUTINE	*
			4624		*****	
	1018 3C 00 0873		4625	MVI	B\$NUMC,B@GETS DISABLE THE GET ROUTINE	
	101C D0 87 23		4626	B	BIT260(,@BR) GO SEARCH OPERATOR TABLE	
			4627		*****	
			4628	*	IF THE RELATIONAL OPERATOR IS COMPOUND. ADD TIE TWO	*
			4629	*	RELATIONAL OPERATORS TO DERIVE A CHARACTER CODE	*
			4630		*****	
	101F 6E 00 2B 00		4631	BIT240 ALC	BIT280+@Q(@B1,@BR),B@CHAR(,@XR) ADD OPERATORS	
			4632		*****	
			4633	*	SEARCH THE RELATIONAL OPERATOR TABLE FOR THE	*
			4634	*	CORRESPONDING CONDITION CODE TO BE PLACED IN THE	*
			4635	*	BRANCH ON CONDITION PSEUDO INSTRUCTION	*
			4636		*****	
	1023 C2 02 1AF8		4637	BIT260 LA	B\$TOTB,@XR LOAD TABLE BASE ADDRESS	
	1027 E2 02 02		4638	BIT270 LA	B\$TLTH(,@XR),@XR ADD LENGTH TO ADDR	
	102A BD 00 00		4639	BIT280 CLI	B\$TOD1(,@XR),*-*	IF TEXT OPERATOR = TABLE ENTRY
	102D D0 01 27		4640	BNE	BIT270(,@BR)	* FALL THROUGH
			4641		*****	
			4642	*	SAVE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTR	*
			4643		*****	
	1030 2C 00 1AF3 01		4644	MVC	B\$PRM1(@B1),B\$TCD2(,@XR) SAVE BRC CONDITION CODE	

## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 49
		4646		*****	
		4647	*	RETURN TO PROCESS NEXT CHARACTER EXPRESSION	*
		4648		*****	
1035	3D 06 1AF8	4649	CLI	B\$CADR-@B1,BITEN2	IF OPERAND SECTION IS ON DISK
1039	F2 81 07	4650	JE	BIT290	* GO LOAD AND EXEC FROM DISK
		4651		*****	
		4652	*	OPERAND PROCESSOR SECTION IS CORE RESIDENT - RESTORE	*
		4653	*	STATUS AND BRANCH TO OPERAND PROCESSOR SECTION.	*
		4654		*****	
103C	35 01 1AF9	4655	L	B\$CADR,@BR	RESTORE OPERAND SECTN BASE ADDR
1040	D0 87 00	4656	B	@ZERO(,@BR)	GO TO OPERAND PROC SECTION
		4657		*****	
		4658	*	OPERAND PROCESSOR SECTION IS DISK RESIDENT - LOAD	*
		4659	*	AND RETURN.	*
		4660		*****	
1043	D2 02 8D	4661	BIT290 LA	TWOAD2(,@BR),@XR	LOAD DIST PARAMETER CADDR
1046	C0 87 073A	4662	B	B\$DST2	LOAD & RTRN TO OPRND PROC SECTN



## S/3 BASIC COMPILER SUB-STRING -IF- ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 50
			4664	*****	*****	
			4665	*	THIS SECTION WILL GENERATE A COMPARE CHARACTERS	*
			4666	*	PSEUDO INSTRUCTION, A BRANCH ON CONDITION PSEUDO	*
			4667	*	INSTRUCTION, ESTABLISH CONDITIONS FOR BRANCH TABLE	*
			4668	*	RESOLUTION AND EXIT TO THE COMPILER DISTRIBUTOR.	*
			4669	*****	*****	
			104A 4670	BITERM EQU *	TERMINATION SECTION ENTRY POINT	
	104A	D2 02 8C	4671	LA	BITCMC(, @BR), @XR	LOAD CADDR OF 'CMC' INSTRUCTION
	104D	34 02 0A40	4672	ST	B\$PCAD, @XR	SET CADDR PARM FOR PUT RTN
N04	1051	00 00 0000	4673	MVI	B\$PNBY, B\$LCMC-1	SET LENGTH PARM FOR PUT RTN
	1055	C0 87 093A	4674	B	B\$PUTC	LINK TO GENERATE PMC
			4675	*****	*****	
			4676	*	GENERATE BRANCH ON CONDITION INSTRUCTION IMAGE	*
			4677	*****	*****	
	1059	4C 00 89 1AF3	4678	MVC	BITB02(@B1, @BR), B\$PRM1	GET CONDITION CODE 'FRM' SEG-1
	105E	D2 02 86	4679	LA	BITBRC(, @BR), @XR	LOAD CADDR OF 'BRC' INSTRUCTION
	1061	34 02 0A40	4680	ST	B\$PCAD, @XR	SET CADDR PARM FOR PUT RTN
	1065	3C 03 0A41	4681	MVI	B\$PNBY, B@LBRC-1	SET LENGTH PARAMETER FOR PUT RTN
	1069	C0 87 093A	4682	B	B\$PUTC	LINK TO GENERATE PMC
			4683	*****	*****	
			4684	*	ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR	*
			4685	*	BRANCH TABLE RESOLUTION	*
			4686	*****	*****	
	106D	0C 01 19EF 0A43	4687	MVC	B\$BRVA, B\$PVAD(@VADDR)	SET VADDR PARAMETER
	1073	1F 01 19EF 8B	4688	SLC	B\$BRVA, BITLNG(@VADDR, @BR)	SET PARM FOR VADDR OF 'BRC'
	1078	0C 01 19F1 1A6A	4689	MVC	B\$BRLN, B\$BINO(B@LCLN)	SET LINE NO. PARM
	107E	C0 87 1996	4690	B	B\$BTAB	LINK TO SET RESOLUTION COND.
			4691	*****	*****	
			4692	*	PROCESSING IS FINISHED RETURN TO DISTRIBUTOR	*
			4693	*****	*****	
	1082	C0 87 0700	4694	B	B\$DIST	RETURN TO DISTRIBUTOR
			4695	*****	*****	
			4696	*	SEGMENT2 CONSTANTS ANC WORK AREAS	*
			4697	*****	*****	
	1086	44	1086 4698	BITBRC DC	AL(B@LCOP)(B@CBRC)	BRANCH ON CONDITION OPCODE
	1087	0000	1088 4699	BITB01 DC	XL(B@LCVA)'00'	BRANCH ON COND VADDR OPERAND
	1089		1089 4700	BITB02 DS	CL(B@LCCC)	BRANCH ON COND COND CODE OPERAND
	108A	0002	108B 4701	BITLNG DC	AL(@VADDR)(B@LCCC+1)	LENGTH OF COND CODE + 1
	108C	42	108C 4702	BITCMC DC	AL(B@LCOP)(B@CCMC)	COMPARE CHARACTER OPCODE
			0006 4703	BITEN2 EQU	X'06'	CORE PGE NO. OF DISK BUFFER
			108D 4704	TWOAD2 EQU	*	CONSTANTS AND WORK AREAS USED
	108D	0600	108E 4705	TWOCA2 DC	AL(@CADDR)(B\$CSBF)	* BY THE RELATIONAL OPERATOR
	108F	1C	108F 4706	DC	AL1(B@DSIF)	* SECTION TO RETURN TO THE
			4707	*		* OPERAND PROCESSOR SECTION
			4708	*****	*****	
			4709	*	END OF SUBSTRING IF STATEMENT PROCESSOR	*
			4710	*****	*****	

## S/3 BASIC COMPILER -GOSUB- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 51
			4712		*****			
			4713	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			4714	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			4715	*				*
			4716		*****			*
			4717	*	*STATUS			*
			4718	*	VERSION 1 MODIFICATION 0			*
			4719	*				*
			4720	*	*FUNCTION			*
			4721	*	BKSUBG IS EXECUTED TO TRANSLATE GOSUB STATEMENTS AS THEY OCCUR			*
			4722	*	IN A BASIC PROGRAM INTO THE APROPRIATE PSEUDOCODE AND TO PLACE			*
			4723	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
			4724	*				*
			4725	*	*ENTRY POINTS			*
			4726	*	BKSUBG HAS ONLY ONE ENTRY POINT:			*
			4727	*	BKSUBG - TRANSLATE GOSUB STATEMENT			*
			4728	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			4729	*	B BKSUBG			*
			4730	*				*
			4731	*	*INPUT			*
			4732	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			4733	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			4734	*	LEADING KEYWORD, GOSUB.			*
			4735	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			4736	*	CHARACTER IN THE LEADING KEYWORD, GOSUB.			*
			4737	*				*
			4738	*	*OUTPUT			*
			4739	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			4740	*	GENERATED BY BKSUBG IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			4741	*	MEMORY LOCATION, FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			4742	*	SEQUENCES.			*
			4743	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			4744	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			4745	*	* BSRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
			4746	*	ADDRESS OPERAND FIELD IN THE RETURN-ADDRESS STACKING			*
			4747	*	INSTRUCTION.			*
			4748	*	* BSNXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE RETURN-			*
			4749	*	ADDRESS STACKING INSTRUCTION OPERAND ADDRESS.			*
			4750	*				*
			4751	*	*EXTERNAL REFERENCES			*
			4752	*	* B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			4753	*	* B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRTUAL			*
			4754	*	MEMORY OUTPUT ROUTINE.			*
			4755	*	* B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH			*
			4756	*	TABLE ROUTINE.			*
			4757	*	* B\$ZDBN - (B\$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL TO			*
			4758	*	BINARY CONVERSION ROUTINE.			*
			4759	*	* B\$DIST - (B\$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
			4760	*				*
			4761	*	*EXITS, NORMAL			*
			4762	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
			4763	*				*
			4764	*	*EXITS, ERROR			*
			4765	*	N/A			*
			4766	*				*
			4767	*	*TABLES/WORK AREAS			*

## S/3 BASIC COMPILER -GOSUB- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 52
				4768 *	N/A	*
				4769 *		*
				4770 *	*ATTRIBUTES	*
				4771 *	BKSUBG IS NATURALLY RELOCATABLE AND REUSABLE.	*
				4772 *		*
				4773 *	*CHARACTER CODE DEPENDENCY	*
				4774 *	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
				4775 *	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
				4776 *		*
				4777 *	*NOTES	*
				4778 *	ERROR PROCEDURES	*
				4779 *	N/A	*
				4780 *		*
				4781 *	REGISTER USAGE	*
				4782 *	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				4783 *		*
				4784 *	SAVED/RESTORED AREAS	*
				4785 *	N/A	*
				4786 *		*
				4787 *	MODIFICATION CONSIDERATIONS	*
				4788 *	BKSUBG IS CO-RESIDENT ON A SECTOR WITH BSTRIF. ANY	1-4*
				4789 *	MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND THE	1-4*
				4790 *	LIMITATION OF THE SECTOR BOUNDARY ON SIZE.	1-4*
				4791 *		*
				4792 *	REQUIRED MODULES	*
				4793 *	@SYSEQ - COMMON SYSTEM EQUATES	*
				4794 *	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES	*
				4795 *	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES	*
				4796 *	@VMDEQ VIRTUAL MEMORY DIRECTORY EQUATES	*
				4797 *	@SPFEQ - SYSTEM PROGRAM FILE EQUATES	*
				4798 *	@ERMEQ - ERROR MESSAGE EQUATES	*
				4799 *	\$V\$EQ - FIXED VIRTUAL ADDRESS EQUATES	*
				4800 *	\$B\$EQ - COMPILER FIXED EQUATES	*
				4801 *	\$B@EQ COMPILER SYSTEM EQUATES	*
				4802 *		*
				4803 *	OTHER	*
				4804 *	BKSUBG IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS	*
				4805 *	*****	*
				4807 *		*
				4808 *	ENTER BKSUBG - 'GOSUB' STATEMENT ROUTINE	*
				4809 *		*
			1090	4810	BKSUBG EQU * BKSUBG ENTRY POINT	
				4811 *		
				4812 *	SET INPUT PARAMETER TO SKIP KEYWORD 'GOSUB'	
				4813 *		
1090	3C	05	0873	4814	BKS010 MVI B\$NUMC,B@LGSB SET GET RTN TO SKIP 'GOSUB'	
1094	C0	87	0867	4815	B B\$GETC LINK TO ADVANCE POINTER	
				4816 *		
				4817 *	CONVERT 'GOSUB' LINE NUMBER TO BINARY FROM ITS DECIMAL FORM	
				4818 *		
1098	C0	87	19F2	4819	BKS020 B B\$ZDBN LINK TO CONVERT LINE NUMBER	

## S/3 BASIC COMPILER -GOSUB- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 53

```

4821 *
4822 * GENERATE AN ADDRESS STACKING INSTRUCTION IMAGE FOR RETURN ADDRESS
4823 *
109C D2 02 E6 4824 BKS030 LA BKSTAC(,@BR),@XR LOAD CADDR OF 'STA' INSTR
109F 34 02 0A40 4825 ST B$PCAD,@XR SET PUT RTN FOR VADDR OF 'STA'
10A3 3C 02 0A41 4826 MVI B$PNBY,B@LSTA-1 SET PUT RTN FOR LENGTH OF 'STA'
10A7 C0 87 093A 4827 B B$PUTC LINK TO GENERATE 'STA' IMAGE
10AB 4C 01 EF 0A43 4828 MVC BKSVAS(,@BR),B$PVAD(@VADDR) SAVE VADDR AFTER 'STA' INST
4829 *
4830 * GENERATE AN UNCONDITIONAL BRANCH INSTRUCTION IMAGE IN VIRTUAL MEMORY
4831 *
10B0 D2 02 E9 4832 BKS040 LA BKSBR(,@BR),@XR LOAD CADDR OF 'BRA' INSTR
10B3 34 02 0A40 4833 ST B$PCAD,@XR SET PUT RTN FOR VADDR OF 'BRA'
10B7 3C 02 0A41 4834 MVI B$PNBY,B@LBRA-1 SET PUT RTN FOR LENGTH OF 'BRA'
10BB C0 87 093A 4835 B B$PUTC LINK TO GENERATE 'BRA' IMAGE
4836 *
4837 * ESTABLISH LINE NUMBER AND VIRTUAL ADDRESS FOR RESOLUTION OF 'BRA'
4838 * INSTRUCTION OPERAND
4839 *
10BF 0C 01 19F1 1A6A 4840 BKS050 MVC B$BRLN,B$BINO(@VADDR) ESTABLISH BRANCH LINE NUMBER
10C5 0C 01 19EF 0A43 4841 MVC B$BRVA,B$PVAD(@VADDR) SET BRANCH TABLE VADDR
10CB 1F 01 19EF ED 4842 SLC B$BRVA,BKSBN1(@VADDR,@BR) ADJUST VADDR FOR 'BRA' OPERAND
10D0 C0 87 1996 4843 B B$BTAB LINK TO RESOLVE 'BRA' OPERAND
4844 *
4845 * ESTABLISH VIRTUAL ADDRESS PARAMETER FOR 'STA' BRANCH TABLE RESOLUTION
4846 *
10D4 1C 01 19EF EF 4847 BKS060 MVC B$BRVA,BKSVAS(@VADDR,@BR) SET BRANCH TABLE VADDR
10D9 1F 01 19EF ED 4848 SLC B$BRVA,BKSBN1(@VADDR,@BR) ADJUST VADDR FOR 'STA' OPERAND
4849 *
4850 * SET SWITCH ON TO CAUSE THE DISTRIBUTOR TO SET UP ADDR RESOLUTION
4851 * CONDITIONS
4852 *
N04 10DE 00 00 0000 4853 BKS070 SBN B$NXSM,B$NXMK SET SW TO RESOLVE 'STA' ADDR
4854 *
4855 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
4856 *
10E2 C0 87 0700 4857 BKS080 B B$DIST RETURN TO DISTRIBUTOR
4859 *****
4860 * 'GOSUB' STMT ROUTINE PMC AND STORAGE PARAMETERS
4861 *****
4862 *
10E6 34 10E6 4863 BKSTAC DC AL(B@LCOP)(B@CSTA) STACK ADDRESS INSTR OPCODE
10E7 0000 10E8 4864 BKSTAO DC XL(B@LCVA)'00' STACK ADDRESS INSTR OPERAND
4865 *
10E9 46 10E9 4866 BKSBR( DC AL(B@LCOP)(B@CBRA) 'BRA' INSTR OPCODE
10EA 0000 10EB 4867 BKSBR( DC XL(B@LCVA)'00' 'BRA' INSTR OPERAND
4869 *****
4870 * 'GOSUB' STATEMENT ROUTINE CONSTANTS
4871 *****
4872 *
10EC 0001 10ED 4873 BKSBN1 DC IL(@VADDR)'1' BINARY 1
4875 *****
4876 * 'GOSUB' STMT ROUTINE WORK AREAS

```

		4877	*****	
		4878	*	
10EE	10EF	4879	BKSVAS DS	CL(@VADDR) VIRTUAL ADDRESS SAVE AREA
		4880	*****	
		4881	*	
		4882	* END OF 'GOSUB' STATEMENT ROUTINE CODING	
		4883	*	

## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 55
		4885		*****			
		4886	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		4887	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		4888	*				*
		4889		*****			*
		4890	*	STATUS			*
		4891	*	VERSION 1 MODIFICATION 0			*
		4892	*				*
		4893	*	FUNCTION			*
		4894	*	BNDATA IS EXECUTED TO TRANSLATE DATA STATEMENTS AS THEY OCCUR			*
		4895	*	IN A BASIC PROGRAM INTO APPROPRIATE PSEUDOCODE AND TO PLACE			*
		4896	*	THE PSEUDOCODE INTO VIRTUAL MEMORY.			*
		4897	*				*
		4898	*	ENTRY POINTS			*
		4899	*	BNDATA HAS ONLY ONE ENTRY POINT:			*
		4900	*	BNDATA - TRANSLATE DATA STATEMENT.			*
		4901	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		4902	*	B BNDATA			*
		4903	*				*
		4904	*	INPUT			*
		4905	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		4906	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		4907	*	LEADING KEYWORD, DATA.			*
		4908	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		4909	*	CHARACTER IN THE LEADING KEYWORD, DATA.			*
		4910	*	* \$INLNO - CONTAINS A VALUE OF ZERO WHEN NO PREVIOUS DATA			*
		4911	*	STATEMENTS HAVE BEEN PROCESSED.			*
		4912	*	* B\$CLNK - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
		4913	*	ADDRESS OPERAND FIELD IN THE LAST GENERATED DDL INSTRUCTION:			*
		4914	*	THIS IS ONLY REQUIRED WHEN \$INLNO IS NON-ZERO.			*
		4915	*				*
		4916	*	OUTPUT			*
		4917	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		4918	*	GENERATED BY BNDATA IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		4919	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		4920	*	SEQUENCES.			*
		4921	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		4922	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		4923	*	* \$INLNO - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST DCA			*
		4924	*	INSTRUCTION GENERATED FOR THE DATA STATEMENT WHEN THIS IS THE			*
		4925	*	FIRST SUCH STATEMENT TO BE PROCESSED IN THE PROGRAM.			*
		4926	*	* B\$DLNK - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
		4927	*	ADDRESS OPERAND FIELD IN THE DDL INSTRUCTION GENERATED FOR THE			*
		4928	*	CURRENT STATEMENT.			*
		4929	*	* B\$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
		4930	*	ADDRESS OPERAND FIELD IN THE BYPASS BRANCH INSTRUCTION			*
		4931	*	GENERATED FOR THE CURRENT STATEMENT.			*
		4932	*	* BSNXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE BYPASS			*
		4933	*	BRANCH INSTRUCTION OPERAND ADDRESS.			*
		4934	*				*
		4935	*	EXTERNAL REFERENCES			*
		4936	*	B\$GETC - (B\$NUNC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		4937	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRTUAL			*
		4938	*	MEMORY OUTPUT ROUTINE.			*
		4939	*	B\$FCON - (B\$CTYP, B\$BCKT) - ENTRY TO BASIC COMPILER CONSTANT			*
		4940	*	ROUTINE.			*



## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 56
		4941	*	B\$BTAB - (B\$BRVA, B\$BRIN) - ENTRY TO BASIC COMPILER BRANCH	*	
		4942	*	TABLE ROUTINE.	*	
		4943	*	B\$SCAN - (B\$FVPP, B\$FVPP, B\$FVPS, BIFVME, B\$FVMP, B\$FVMS) -	*	
		4944	*	ENTRY TO BASIC COMPILER SCAN ROUTINE.	*	
		4945	*	B\$DLNK - AREA CONTAINING VIRTUAL ADDRESS OF THE RIGHT BYTE OF	*	
		4946	*	ADDRESS OPERAND FIELD OF 'DCA' INSTRUCTIONS.	*	
		4947	*	\$INLNO - AREA CONTAINING VIRTUAL ADDRESS OF 'DCA' INSTRUCTIONS.	*	
		4948	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*	
		4949	*		*	
		4950	*	*EXITS, NORMAL	*	
		4951	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*	
		4952	*		*	
		4953	*	*EXITS, ERROR	*	
		4954	*	N/A	*	
		4955	*		*	
		4956	*	*TABLES/WORK AREAS	*	
		4957	*	* INTERNAL CONSTANT BUCKET - 2 BYTES. INTERNAL TO BNDA; FOR	*	
		4958	*	ACCUMULATING INTERNAL CONSTANT SYMBOL CHARACTERS IN PREPARATION	*	
		4959	*	FOR A TABLE SEARCH.	*	
		4960	*	* INTERNAL CONSTANT TABLE - INTERNAL TO BNDA, THIS TABLE	*	
		4961	*	CONTAINS THE CORE ADDRESSES OF VIRTUAL ADDRESS VALUES	*	
		4962	*	ASSOCIATED WITH EACH INTERNAL CONSTANT, AND A LENGTH CODE WHICH	*	
		4963	*	REPRESENTS ONE LESS THAN THE CONSTANT SYMBOL LENGTH. SYMBOL	*	
		4964	*	MATCHING IS BASED ON THE SIGN CR THE CONSTANT AND THE LETTER	*	
		4965	*	CHARACTER FOLLOWING THE '&' IDENTIFIER.	*	
		4966	*		*	
		4967	*	*ATTRIBUTES	*	
		4968	*	BNDA IS NATURALLY RELOCATABLE AND REUSABLE.	*	
		4969	*		*	
		4970	*	*CHARACTER CODE DEPENDENCY	*	
		4971	*	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA-	*	
		4972	*	TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE	*	
		4973	*	ONE USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT	*	
		4974	*	REDEFINITION OF CHARACTER CONSTANIS, BY REASSEMBLY, WILL RESULT IN	*	
		4975	*	A CORRECT MODULE FOR THE NEW DEFINITIONS.	*	
		4976	*		*	
		4977	*	*NOTES	*	
		4978	*	ERROR PROCEDURES	*	
		4979	*	N/A	*	
		4980	*		*	
		4981	*	REGISTER USAGE	*	
		4982	*	BOTH THE INDEX AND BASE REGISTERS ARE USER DURING EXECUTION.	*	
		4983	*		*	
		4984	*	SAVED/RESTORED AREAS	*	
		4985	*	N/A	*	
		4986	*		*	
		4987	*	MODIFICATION CONSIDERATIONS	*	
		4988	*	BNDA MUST RESIDE ON ONE SECTOR OR BE LINKED PROPERLY IF IT	*	
		4989	*	CROSSES A SECTOR BOUNDARY. AS IT APPROACHES THE SECTOR	*	
		4990	*	LIMITATION, EXCEEDING THIS SIZE MUST BE A CONSIDERATION IN ANY	*	
		4991	*	MODIFICATIONS.	*	
		4992	*		*	
		4993	*	REQUIRED MODULES	*	
		4994	*	@SYSEQ - COMMON SYSTEM EQUATES	*	
		4995	*	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES	*	
		4996	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS	*	



## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 57
		4997	*		@VMDEQ - VIRTUAL NEWRY DIRECTORY EQUATES			*
		4998	*		@SPFEQ - SYSTEM PROGRAM FILE EQUATES			*
		4999	*		@ERMEQ - ERROR MESSAGE EQUATES			*
		5000	*		\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES			*
		5001	*		\$B\$EQU - COMPILER FIXED EQUATES			*
		5002	*		\$B@EQU - COMPILER SYSTEM EQUATES			*
		5003	*					*
		5004	*	OTHER				*
		5005	*		BNDATA IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			*
		5006	*		*****			*
1100		5008		ORG	*,256,0			BEGIN AT CORE PAGE BOUNDARY
		1100 5009		USING	*,@BR			DEFINE USE ADDR FOR CORE PAGE
		5010	*					
		5011	*	ENTER	BNDATA 'DATA' STATEMENT ROUTINE			
		5012	*					
		1100 5013		BNDATA	EQU *			
		5014	*					
		5015	*	SET	GET ROUTINE PARAMETER TO SKIP TO 1ST CHARACTER FOLLOWING 'DATA'			
		5016	*					
1100 3C 04 0873		5017		BND010 MVI	B\$NUMC,B@LDAT			SET GET TO SKIP 'DATA'
		5018	*					
		5019	*	GENERATE	A 'BRA' IMAGE IN VIRTUAL MEMORY			
		5020	*					
1104 D2 02 D1		5021		BND020 LA	BNDBRC(,@BR),@XR			LOAD CADDR OF 'BRA' INSTR
1107 34 02 0A40		5022		ST	B\$PCAD,@XR			SET PUT RTN VADDR FOR 'BRA'
		5023	*					
		5024	*	SET	THE LENGTH PARAMETER IN PUT TO BE USED IN THE GENERATION OF THE			
		5025	*	FOLLOWING	INSTRUCTIONS: 'BRA', 'DCA' AND 'DDL'.			
		5026	*					
110B 3C 02 0A41		5027		MVI	B\$PNBY,B@LCOP+B@LCVA-1			SET LENGTH PARM OF PUT RTN
110F C0 87 093A		5028		B	B\$PUTC			LINK TO GENERATE 'BRA' PMC
		5029	*					
		5030	*	SAVE	THE NEXT AVAILABLE VADDR IN THE BRANCH TABLE LINE NUMBER PARM			
		5031	*					
1113 0C 01 19F1 0A43		5032		BND030 MVC	B\$BRLN,B\$PVAD(@VADDR)			SAVE THE NEXT AVAILABLE VADDR
		5033	*					
		5034	*	TEST	THE CURRENT STATEMENT FOR BEING THE FIRST DATA STATEMENT			
		5035	*					
1119 3D 56 03CE		5036		BND040 CLI	\$INLNO-1,B@DVC1			IF THIS IS NOT 1ST DATA STMT
111D F2 02 09		5037		JNL	BND060			* GO SET ADDR RESOLUTION COND
		5038	*					
		5039	*	IF	THIS IS THE FIRST DCA ESTABLISH THE NEXT AVAILABLE VADDR AS THE			
		5040	*	VALUE	OF THE LINE NUMBER COMMUNICATION PARAMETER			
		5041	*					
1120 0C 01 03CF 0A43		5042		BND050 MVC	\$INLNO,B\$PVAD(@VADDR)			SAVE NEXT VADDR IN LN NO PARM
1126 F2 87 0A		5043		J	BND070			JUMP TO SET PUT RTN PARAMETERS
		5044	*					
		5045	*	SET	UP ADDRESS RESOLUTION CONDITIONS TO LINK PREVIJUS ADOR DEFINITII			
		5046	*	SEQUENCE	WITH THE SEQUENCE FOR THE CURRENT STATEMENT			
		5047	*					
1129 0C 01 19EF 1B37		5048		BND060 MVC	B\$BRVA,B\$DLNK(@VADDR)			SET VADDR OF LAST DOL OPND AS
		5049	*					* INPUT PARM
112F C0 87 1996		5050		B	B\$BTAB			LINK TO RESOLVE BRANCH ADDRESS
		5051	*					
		5052	*	SET	INPUT PARAMETERS FOR THE PUT ROUTINE			

## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 58
					5053	*		
N04	1133	00 00 00			5054	BND070 LA	BNODAC(, @BR), @XR	LOAD CADDR OF 'DCA' INSTR
	1136	34 02 0A40			5055	ST	B\$PCAD, @XR	SET PUT RTN VADDR FOR 'DCA'
					5056	*		
					5057	*	ADVANCE THE TEXT POINTER TO THE 1ST CHAR OF DATA LIST ITEM	
					5058	*		
	113A	C0 87 0867			5059	BND080 B	B\$GETC	LINK TO GET 1ST ITEM CHAR
					5060	*		
					5061	*	TEST FOR CHARACTER DATA	
					5062	*		
	113E	BD 7D 00			5063	BND090 CLI	B@CHAR(, @XR), B@SQUO	IF ELEMENT IS NOT CHAR DATA
	1141	F2 01 07			5064		JNE BND100	* GO TEST FOR INTERNAL CONSTANT
	1144	3C 1F 0A5F			5065		MVI B\$CTYP, B\$CCON	SET CONSTANT RTN FOR CHAR DATA
	1148	F2 87 4A			5066		J BND170	GO PROCESS DATA CONSTANT
					5067	*		
					5068	*	TEST FOR INTERNAL CONSTANT DATA ELEMENT	
					5069	*		
	114B	7C 4E DA			5070	BND100 MVI	BNDBKT+BNDBK0(, @BR), B@PLUS	SET SIGN OF CONSTANT TO PLUS
	114E	BD 6C 00			5071		CLI B@CHAR(, @XR), B@ICON	IF CHAR IS NOT INTERNAL CON
	1151	F2 01 2C			5072		JNE BND130	* GO SET BUCKET SIGN BYTE
					5073	*		
					5074	*	SET 2ND BYTE OF COMPARE BUCKET AND SEARCH TABLE FOR INTERNAL CONSTANT	
					5075	*		
	1154	C0 87 0867			5076	BND110 B	B\$GETC	LINK TO GET NEXT CHAR
	1158	6C 00 DB 00			5077		MVC BNDBKT+BNDBK1(, @BR), B@CHAR(1, @XR)	SET 2ND BUCKET BYTE
	115C	D2 02 D7			5078		LA BNDTAB-BNDEL(, @BR), @XR	LOAD TABLE BASE ADDR IN XR
	115F	E2 02 05			5079	BND120 LA	BNDEL(, @XR), @XR	INCREMENT POINTER TO NEXT ENTRY
	1162	6D 01 DB 01			5080		CLC BNDBKT+BNDBK1(, @BR), BNDEL1(BNDBKL, @XR)	IF ICON NOT = ENT
	1166	D0 01 5F			5081		BNE BND120(, @BR)	* GO SEARCH TABLE AGAIN
	1169	2C 00 0873 04			5082		MVC B\$NUMC, BNDEL4(1, @XR)	SET GET TO ADVANCE POINTER
	116E	B5 02 03			5083		L BNDEL3(, @XR), @XR	LOAD INTERNAL CON VADDR CADDR
	1171	6C 01 D6 00			5084		MVC BNDDAO(, @BR), BNDELCA(@VADDR, @XR)	SET 'DCA' INST OPERAND
	1175	C0 87 093A			5085		B B\$PUTC	LINK TO GENERATE 'DCA' PMC
	1179	C0 87 0867			5086		B B\$GETC	LINK TO GET CONSTANT DELIMITER
	117D	F2 87 22			5087		J BND190	GO TEST FOR END OF DATA LIST
					5088	*		
					5089	*	MOVE CHAR TO 1ST BUCKET BYTE AND TEST FOR INTERNAL CONSTANT	
					5090	*		
	1180	6C 00 DA 00			5091	BND130 MVC	BNDBKT+BNDBK0(, @BR), B@CHAR(1, @XR)	SET BUCKET SIGN BYTE
	1184	C0 87 0867			5092		B B\$GETC	LINK TO GET NEXT CHAR
	1188	BD 6C 00			5093		CLI B@CHAR(, @XR), B@ICON	IF ELEMENT IS AN INTERNAL CON
	118B	D0 81 54			5094		BE BNDEL10(, @BR)	* GO GET NEXT CHAR IN SEARCH TBL
					5095	*		
					5096	*	DISABLE BAGETC TO GET NEXT CHAR AND RESTORE TEXT POINTER	
					5097	*		
	118E	D2 02 DA			5098		LA BNDBKT+BNDBK0(, @BR), @XR	RESTORE TEXT POINTER
	1191	3C 00 0873			5099		MVI B\$NUMC, B@GETS	DISABLE GET RTN TO GET CHARS
					5100	*		
					5101	*	CALL CONSTANT SCAN ROUTINE TO PROCESS THE DATA ELEMENT	
					5102	*		
	1195	C0 87 0A46			5103	BND170 B	B\$FCON	LINK TO PROCESS DATA CONSTANT
					5104	*		
					5105	*	GENERATE A 'DCA' PMC WITH THE VADDR OF THE DATA CONSTANT AS OPERAND	
					5106	*	IN VIRTUAL MEMORY	
					5107	*		
	1199	4C 01 D6 1590			5108	BND180 MVC	BNDDAO(, @BR), B\$BCKT(@VADDR)	SET DATA CON VADDR 'DCA' OPND

## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 59
119E	C0 87 093A	5109	B	B\$PUTC	LINK TO GENERATE 'DCA' PMC			
		5110	*					
		5111	*	TEST FOR A STATEMENT TERMINATOR				
		5112	*					
11A2	BD 1E 00	5113	BND190	CLI B@CHAR(,@XR),B@EOST	IF THERE IS ANOTHER ELEMENT			
11A5	D0 01 3A	5114		BNE BND080(,@BR)	* GO REPEAT PROCESSING			
		5115	*					
		5116	*	GENERATE A SEQUENCE LINKAGE INSTR (DDL) IN VIRTUAL MEMORY				
		5117	*					
11A8	D2 02 D7	5118	BND200	LA BNDDL(,@BR),@XR	LOAD CADDR OF 'DDL' INSTR			
11AB	34 02 0A40	5119		ST B\$PCAD,@XR	SET PUT RTN VADDR FOR 'DDL.			
11AF	C0 87 093A	5120	B	B\$PUTC	LINK TO GENERATE 'DDL' PMC			
		5121	*					
		5122	*	SAVE THE VADDR OF THE OPERAND FIELD OF THE DDL INSTR				
		5123	*					
11B3	0C 01 1B37 0A43	5124	BND210	MVC B\$DLNK,B\$PVAD(@VADDR)	SET PARM WITH NEXT VADDR			
11B9	1F 00 1B37 FA	5125		SLC B\$DLNK,BNDBN1(@VADDR-1,@BR)	ADJUST VADDR TO OPND OF 'DDL'			
		5126	*					
		5127	*	SET UP ADDRESS RESOLUTION CONDITIONS FOR THE BYPASS BRANCH INSTR				
		5128	*					
11BE	0C 01 19EF 19F1	5129	BND220	MVC B\$BRVA,B\$BRLN(@VADDR)	SET PARM WITH VADDR AFTER BRA			
11C4	1F 00 19EF FA	5130		SLC B\$BRVA,BNDBN1(@VADDR-1,@BR)	ADJUST VADDR TO OPND OF 'BRA'			
11C9	3A 07 071D	5131		SBN B\$NXSW,B\$NXMK	SET SW FOR LINE RESOLUTION			
		5132	*					
		5133	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR				
		5134	*					
11CD	C0 87 0700	5135	BND230	B B\$DIST	RETURN TO DISTRIBUTOR			

## S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 60

```

5137 *****
5138 * 'DATA' STATEMENT ROUTINE PARAMETER AND STORAGE AREAS
5139 *****
5140 *
11D1 46      11D1 5141 BNDBRC DC    AL(B@LCOP)(B@CBRA)      'BRA' INSTR OPCODE
11D2 0000    11D3 5142 BNDBRO DC    XL(B@LCVA)'00'          'BRA' INSTR OPERAND
5143 *
11D4 6A      11D4 5144 BNDDAC DC    AL(B@LCOP)(B@CDCA)      'DCA' INSTR OPCODE
11D5         11D6 5145 BNDDAO DS    CL(B@LCVA)              'DCA' INSTR OPERAND
5146 *
11D7 6C      11D7 5147 BNDDL C DC    AL(B@LCOP)(B@CDDL)      'DDL' INSTR OPCODE
11D8 0000    11D9 5148 BNDDL O DC    XL(B@LCVA)'00'          'DDL' INSTR OPERAND
5150 *****
5151 * 'DATA' STATEMENT INTERNAL CONSTANT TABLE
5152 *****
5153 *
0000 5154 BNDBK0 EQU    0      LENGTH TO 1ST BUCKET BYTE
0001 5155 BNDBK1 EQU    1      LENGTH TO 2ND BUCKET BYTE
5156 *
0005 5157 BNDBTEL EQU    5      LNG OF INTERNAL CON TBL ENTRY
0001 5158 BNDBTB1 EQU    1      DISP TO FIELD FOR BUCKET COMP
0003 5159 BNDBTB3 EQU    3      DISP TO CADDR OF CON VADDR
0004 5160 BNDBTB4 EQU    4      DISP TO CONSTANT LENGTH
5161 *
0000 5162 BNDBICA EQU    0      DISP FOR INTERNAL CON VADDR
0002 5163 BNDBK L EQU    2      LNG OF INT CON COMP AREA
5164 *
11DA         11DA 5165 BNDBKT EQU    *      INTERNAL CON COMPARE AREA ADDR
11DB         11DB 5166          DS    CL(BNDBK L)          COMPARE AREA FOR INTERNAL CON
5167 *
11DC         11DC 5168 BNDBTAB EQU    *
11DC 4E      11DC 5169          DC    AL1(B@PLUS)          POSITIVE SIGNED INTERNAL CON
11DD C5      11DD 5170          DC    AL1(B@CIEX)          2ND CHAR IN &E
11DE 15A8    11DF 5171          DC    AL(@CADDR)(B$FVPE)    CADDR OF VADDR OF +&E
11E0 01      11E0 5172          DC    AL1(B@LIEX-1)         LENGTH OF &E-1
5173 *
11E1 4E      11E1 5174          DC    AL1(B@PLUS)          POSITIVE SIGNED INTERNAL CON
11E2 D7      11E2 5175          DC    AL1(B@CIPI)          2ND CHAR IN &PI
11E3 15AA    11E4 5176          DC    AL(@CADDR)(B$FVPP)    CADDR OF VADDR OF +$PI
11E5 02      11E5 5177          DC    AL1(B@LIPI-1)         LENGTH OF &PI-1
5178 *
11E6 4E      11E6 5179          DC    AL1(B@PLUS)          POSITIVE SIGNED INTERNAL CON
11E7 E2      11E7 5180          DC    AL1(B@CIS2)          2ND CHAR IN &SQR2
11E8 15AC    11E9 5181          DC    AL(@CADDR)(B$FVPS)    CADDR OF VADDR OF +&SQR2
11EA 04      11EA 5182          DC    AL1(B@LIS2-1)         LENGTH OF &SQR2-1
5183 *
11EB 60      11EB 5184          DC    AL1(B@MINS)          NEGATIVE SIGNED INTERNAL CON
11EC C5      11EC 5185          DC    AL1(B@CIEX)          2ND CHAR IN &E
11ED 15A2    11EE 5186          DC    AL(@CADDR)(B$FVME)    CADDR OF VADDR OF -&E
11EF 01      11EF 5187          DC    AL1(B@LIEX-1)         LENGTH OF &E-1
5188 *
11F0 60      11F0 5189          DC    AL1(B@MINS)          NEGATIVE SIGNED INTERNAL CON
11F1 D7      11F1 5190          DC    AL1(B@CIPI)          2ND CHAR IN &PI
11F2 15A4    11F3 5191          DC    AL(@CADDR)(B$FVMP)    CADDR OF VADDR OF -&PI
11F4 02      11F4 5192          DC    AL1(B@LIPI-1)         LENGTH OF &PI-1

```

S/3 BASIC COMPILER -DATA- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 61
				5193	*				
	11F5	60	11F5	5194	DC	AL1(B@MINS)			NEGATIVE SIGNED INTERNAL CON
	11F6	E2	11F6	5195	DC	AL1(B@CIS2)			2ND CHAR IN &SQR2
	11F7	15A6	11F8	5196	DC	AL(@CADDR)(B\$FVMS)			CADDR OF VADDR OF -&SQR2
	11F9	04	11F9	5197	DC	AL1(B@LIS2-1)			LENGTH OF &SQR2-1
				5199	*****				
				5200	* 'DATA' STATEMENT ROUTINE CONSTANTS				
				5201	*****				
				5202	*				
11FA	01		11FA	5203	BNDBN1 DC	IL(@VADDR-1)'1'			BINARY 1
				5204	*				
				5205	*****				
				5206	*				
				5207	* END OF 'DATA' STATEMENT ROUTINE CODING				
				5208	*				

## S/3 BASIC COMPILER -FOR- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 62
		5210		*****	
		5211	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		5212	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		5213	*		*
		5214		*****	
		5215	*	*STATUS	*
		5216	*	VERSION 1 MODIFICATION 0	*
		5217	*		*
		5218	*	*FUNCTION	*
		5219	*	BKFORX IS EXECUTED TO TRANSLATE FOR STATEMENTS AS THEY OCCUR IN A	*
		5220	*	BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE	*
		5221	*	PSEUDOCODE IN VIRTUAL MEMORY.	*
		5222	*		*
		5223	*	*INPUT	*
		5224	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		5225	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE	*
		5226	*	LEADING KEYWORD, FOR.	*
		5227	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		5228	*	FIRST CHARACTER IN THE LEADING KEYWORD, FOR.	*
		5229	*	* FOR TABLE - CONTAINS TEN 4-BYTE ENTRIES, EACH CONTAINING THE	*
		5230	*	VIRTUAL ADDRESSES OF A FOR-LOOP CONTROL VARIABLE AND OF THE	*
		5231	*	NXT INSTRUCTION IN THE ASSOCIATED FOR OBJECT CODE SEQUENCE.	*
		5232	*	* B\$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE OF THE	*
		5233	*	ENTRY LAST PLACED IN THE FOR TABLE.	*
		5234	*	* B\$FTND - CONTAINS THE CORE ADDRESS OF THE FINAL BYTE IN THE	*
		5235	*	FOR TABLE.	*
		5236	*		*
		5237	*	*OUTPUT	*
		5238	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		5239	*	GENERATED BY BKFORX IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		5240	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		5241	*	SEQUENCES.	*
		5242	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		5243	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		5244	*	* FOR TABLE - UPDATED WITH THE CURRENT STATEMENT FOR-LOOP ENTRY.	*
		5245	*	THE TABLE IS NOT AFFECTED WHEN AN ERROR OCCURS.	*
		5246	*	* B\$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE IN THE	*
		5247	*	FOR TABLE ENTRY GENERATED FOR THE CURRENT STATEMENT. THIS	*
		5248	*	IS NOT AFFECTED WHEN A COMPILER ERROR OCCURS.	*
		5249	*		*
		5250	*	*EXTERNAL REFERENCES	*
		5251	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.	*
		5252	*	B\$PUTC - (B\$PFNC, B\$PCAD, B\$PNBY, B\$PVAD, B\$PCDL, B\$PERC) -	*
		5253	*	ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.	*
		5254	*	B\$ECON - (B\$BCKT) - ENTRY TO BASIC COMPILER CONSTANT ROUTINE.	*
		5255	*	B\$SYKI - (B\$BCKT) - ENTRY TO BASIC SYMBOL TRANSLATION	*
		5256	*	ROUTINE	*
		5257	*	B\$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN	*
		5258	*	ROUTINE	*
		5259	*	B\$FTPT - FOR TABLE POINTER TO LAST BYTE PLACED IN TABLE.	*
		5260	*	\$XIND1 - INDICATOR FOR LONG OR SHORT PRECISION.	*
		5261	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		5262	*		*
		5263	*	*EXITS, NORMAL	*
		5264	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		5265	*		*



## S/3 BASIC COMPILER -FOR- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 63
		5266	*	EXITS, ERROR				*
		5267	*	N/A				*
		5268	*					*
		5269	*	TABLES/WORK AREAS				*
		5270	*	* FOR TABLE - EXTERNAL TO BKFORX, THIS "PUSH-DONN" TABLE				*
		5271	*	CONTAINS TEN 4-BYTE ENTRY LOCATIONS. THE FIRST ENTRY LOCATION				*
		5272	*	IS ALWAYS SET TO ZEROS, AND IS USED TO GUARD AGAINST A TABLE				*
		5273	*	REFERENCE WHEN THE TABLE IS EMPTY. THE FOLLOWING NINE ENTRY				*
		5274	*	LOCATIONS IN THE TABLE MAY EACH CONTAIN VIRTUAL ADDRESSES OF AN				*
		5275	*	UNFINISHED FOR-LOOP CONTROL VARIABLE AND ITS ASSOCIATED NXT				*
		5276	*	INSTRUCTION, DEPENDING ON THE CURRENT LOOP NESTING DEPTH.				*
		5277	*					*
		5278	*	ATTRIBUTES				*
		5279	*	BKFORX IS NATURALLY RELOCATABLE AND REUSABLE.				*
		5280	*					*
		5281	*	CHARACTER CODE DEPENDENCY				*
		5282	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR				*
		5283	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.				*
		5284	*					*
		5285	*	NOTES				*
		5286	*	ERROR PROCEDURES				*
		5287	*	IF MORE THAN NINE LEVELS OF FOR-LOOP NESTING ARE ATTEMPTED,				*
		5288	*	THE FOR TABLE STATUS REMAINS UNCHANGED AND THE ERROR CONDITION				*
		5289	*	CODE FOR MORE THAN 9 NESTED FOR/NXT LOOPS, IS LOGGED IN				*
		5290	*	VIRTUAL MEMORY USING OUTPUT ROUTINE BBPUTC, BKFORX EXECUTION				*
		5291	*	IS OTHERWISE UNAFFECTED.				*
		5292	*					*
		5293	*	REGISTER USAGE				*
		5294	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
		5295	*					*
		5296	*	SAVED/RESTORED AREAS				*
		5297	*	N/A				*
		5298	*					*
		5299	*	MODIFICATION CONSIDERATIONS				*
		5300	*	BKFORX RESIDES ON ONE SECTOR AND MUST NOT EXCEED ITS BOUNDARY.				*
		5301	*	ANY MODIFICATIONS MUST CONSIDER THIS SIZE LIMITATION.				*
		5302	*					*
		5303	*	REQUIRED MODULES				*
		5304	*	@SYSEQ - COMMON SYSTEM EQUATES				*
		5305	*	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES				*
		5306	*	@CANEQ - COMMON CORE LOCATIONS				*
		5307	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES				*
		5308	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES				*
		5309	*	@ERMEQ - ERROR MESSAGE EQUATES				*
		5310	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES				*
		5311	*	\$B\$EQU - COMPILER FIXED EQUATES				*
		5312	*	\$B@EQU - COMPILER SYSTEM EQUATES				*
		5313	*					*
		5314	*	OTHER				*
		5315	*	BKFORX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
		5316	*	*****				*
1200		5318		ORG	*,256,0			BEGIN AT CORE PAGE BOUNDARY
	1200	5319		USING	*,@BR			DEFINE BASE ADDR FOR CORE PAGE
		5320	*					
		5321	*	ENTER BKFORX - FOR STATEMENT ROUTINE				



## S/3 BASIC COMPILER -FOR- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 64

```

1200 5322 *
1200 5323 BKFORX EQU * BKFORX ENTRY POINT
1200 5324 *
1200 5325 * SET INPUT PARAMETER TO SKIP KEYWORD 'FOR'
1200 5326 *
N04 1200 00 00 0000 5327 BKFO10 MVI BINUMC,B@LKFR SET PARAMETER TO SKIP 'FOR'
1204 C0 87 0867 5328 B B$GETC LINK TO ADVANCE POINTER
1200 5329 *
1200 5330 * STORE CONTROL VARIABLE VIRTUAL ADDRESS
1200 5331 *
1208 C0 87 0DBC 5332 BKFO20 B B$SYMB LINK TO GET VADDR OF CTRL VAR
N04 120C 00 00 00 0000 5333 MVC BKFOF0(, @BR), B$BCKT(@VADDR) SAVE CTRL VARIABLE VADDR
1200 5334 *
1200 5335 * GENERATE PMC'S FOR INITIAL AND FINAL CONTROL VALUES
1200 5336 *
1211 C0 87 1514 5337 BKFO30 B B$SCAN LINK TO PROCESS INITIAL VALUE
1215 3C 00 0873 5338 MVI B$NUMC,B@LKTO-2 SET GET RTN NOT TO SKIP CHAR
1219 C0 87 0867 5339 B B$GETC LINK NOT TO SKIP CHARACTERS
121D 3C 01 1BAC 5340 MVI B$SSTA,@B1 SET SW TO ALLOW 'STEP' PARM
1221 C0 87 1514 5341 B B$SCAN LINK TO PROCESS FINAL VALUE
1225 3C 00 1BAC 5342 MVI B$SSTA,@ZERO SET SWITCH OFF FOR 'STEP'
1229 BD 1E 00 5343 CLI B@CHAR(, @XR), @EOS IF INCREMENT NOT SPECIFIED
122C F2 81 0F 5344 JE BKFO50 * SKIP TO SET INCREMENT = 1
1200 5345 *
1200 5346 * GENERATE PMC FOR SPECIFIED INCREMENT VALUE
1200 5347 *
122F 3C 02 0873 5348 BKFO40 MVI B$NUMC,BKFLSP+1 SET PARAMETER TO SKIP 'EP'
1233 C0 87 0867 5349 B B$GETC LINK TO ADVANCE POINTER
1237 C0 87 1514 5350 B B$SCAN LINK TO PROCESS INCREMENT
123B F2 87 1F 5351 J BKFO60 JUMP TO TEST PRECISION
1200 5352 *
1200 5353 * GENERATE PMC FOR DEFAULT INCREMENT VALUE
1200 5354 *
123E D2 02 E8 5355 BKFO50 LA BKFOC1(, @BR), @XR LOAD CADDR OF DECIMAL ONE
1241 3C 00 0873 5356 MVI B$NUMC,B@GETS SET GETC NOT TO GET NEXT CHAR
1245 C0 87 0A46 5357 B B$FCON LINK TO GET VADDR OF ONE
1249 4C 01 E3 1590 5358 MVC BKFO50(, @BR), B$BCKT(@VADDR) MOVE VADDR OF 1 TO PMC STRING
124E D2 02 E1 5359 LA BKFO5C(, @BR), @XR LOAD CADDR OF 'STF' INSTR
1251 34 02 0A40 5360 ST B$PCAD, @XR SET PUT RTN FOR VADDR OF 'STF'
1255 3C 02 0A41 5361 MVI B$PNBY, B@LSTF-1 SET PUT RTN FOR LENGTH OF 'STF'
1259 C0 87 093A 5362 B B$PUTC LINK TO WRITE INCREMENT PMC
1200 5363 *
1200 5364 * TEST FOR PRECISION BEFORE GENERATING FOR/NXT PMC SEQUENCE
1200 5365 *
125D 38 40 03D0 5366 BKFO60 TBN $XIND1, $XPREC IF PRECISION IS STANDARD
1261 F2 90 06 5367 JF BKFO70 * SKIP TO GENERATE FOR/NEXT PMC
1264 7C 27 E0 5368 MVI BKFOFA(, @BR), BKFLLP SET LENGTH FOR LONG PRECISION
1267 7C 20 BF 5369 MVI BKFDAN(, @BR), 2*B@LELP SET 'DWA' OPERAND FOR LONG PREC
1200 5370 *
1200 5371 * GENERATE FOR/NXT LOOP CONTROL PMC SEQUENCE
1200 5372 *
126A 1C 00 0A41 E0 5373 BKFO70 MVC B$PNBY, BKFOFA(1, @BR) SET PUT RTN FOR FOR LOOP LNG
126F D2 02 B8 5374 LA BKFOFC(, @BR), @XR LOAD CADDR FOR FOR LOOP INSTR
1272 34 02 0A40 5375 ST B$PCAD, @XR SET PUT BIN - FOR LOOP VADDR
1276 C0 87 093A 5376 B B$PUTC LINK TO GENERATE FOR/NXT STRING
1200 5377 *

```

## S/3 BASIC COMPILER -FOR- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 65

```

5378 * INCREMENT 'FOR' TABLE POINTER FOR CURRENT NEST DEPTH LEVEL
5379 *
127A 35 02 1B0D 5380 BKF080 L B$FTPT,@XR LOAD THE 'FOR' TABLE POINTER
127E E2 02 04 5381 LA B@LFRT(,@XR),@XR INCR POINTER TO NEXT LEVEL
1281 34 02 1B0D 5382 ST B$FTPT,@XR STORE THE 'FOR' TABLE POINTER
1285 0D 01 1B0D 1B0B 5383 CLC B$FTPT,B$FTND(@CADDR) IF NESTING LIMIT NOT EXCEEDED
128B F2 04 14 5384 JNH BKF100 * SKIP TO STORE CURRENT LEVEL
5385 *
5386 * GENERATE ERROR CODE FOR 'FOR' NESTING DEPTH EXCEPTION
5387 *
128E 1F 01 1B0D E5 5388 BKF090 SLC B$FTPT,BKFOTL(@CADDR,@BR) SET 'FOR' PT TO ORIGINAL ENTRY
1293 3C 33 094E 5389 MVI B$PFNC,B$PFAE SET PUT RTN FOR ERROR OUTPUT
1297 3C AD 0A39 5390 MVI B$PERC,@E608 SET ERROR CODE
129B C0 87 093A 5391 B B$PUTC LINK TO OUTPUT CHARACTER STRING
129F F2 87 12 5392 J BKF120 JUMP TO BKFORX EXIT
5393 *
5394 * STORE CURRENT LOOP VALUES IN FOR TABLE
5395 *
12A2 9C 01 01 BA 5396 BKF100 MVC BKFOCV(,@XR),BKFOFO(@VADDR,@BR) STORE CTRL VARIABLE VADDR
N04 12A6 00 00 00 0000 5397 MVC BKFOND(,@XR),BSRVAD(@VADDR) MOVE NEXT PMC VADDR TO TBL
N04 12AB 00 00 00 0000 5398 SLC BKFONI(,@XR),BSPCDL(@VADDR-1) SUBTRACT LENGTH OF LIST PMC
N04 12B0 00 00 00 00 5399 ALC BKFOND(,@XR),BKFOX3(@VADDR,@BR) SET NEXT PMC VADDR IN TBL
5400 *
5401 * RETURN CONTROL TO THE DISTRIBUTOR
5402 *
12B4 C0 87 0700 5403 BKF120 B B$DIST RETURN TO DISTRIBUTOR
5405 *****
5406 * 'FOR' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS
5407 *****
5408 *
12B8 4E 12B8 5409 BKFOFC DC AL(B@LCOP)(B@CFOR) 'BEGIN LOOP' OPCODE
12B9 12BA 5410 BKFOFO DS CL(B@LCVA) CONTROL VARIABLE VADDR
5411 *
12BB 50 12BB 5412 BKFONC DC AL(B@LCOP)(B@CNXT) 'CONTINUE LOOP' OPCODE
12BC 0000 12BD 5413 BKFONO DC XL(@VADDR)'00' LOOP EXIT BRANCH ADDR FIELD
5414 *
12BE 6E 12BE 5415 BKFDAC DC AL(B@LCOP)(B@CDWA) 'DWA' INSTRUCTION OPCODE
12BF 12BF 5416 BKFDAN DS CL(B@LCNN) 'DWA' INSTRUCTION OPERAND
12BF 5417 ORG BKFDAN INITIALIZE 'DMA' OPERAND FOR
12BF 10 12BF 5418 DC AL(B@LCNN)(2*B@LESP) * STANDARD PREC UNPACKED FLT PT
12C0 0000000000000000 12DF 5419 BKFOPR DC XL(2*B@LELP)'00' LOOP CONTROL PARAMETERS FIELD
12E0 12E0 5420 BKFOFA DS CL1 'FOR LOOP' PMC LENGTH - 1
12E0 5421 ORG BKFOFA LENGTH SET FOR SHORT PRECISION
12E0 17 12E0 5422 DC AL1(B@LFOR+B@LNXT+B@LDWA+2*B@LESP-1) CHANGE FOR LENGTH PR
5423 *
12E1 20 12E1 5424 BKFOSC DC AL(B@LCOP)(B@CSTF) STACK FLT VALUE OPCODE
12E2 12E3 5425 BKFOSO DS CL(B@LCVA) STACK FLT VALUE OPERAND
5427 *****
5428 * 'FOR' STATEMENT ROUTINE CONSTANTS AND EQUATES
5429 *****
5430 *
5431 * CONSTANTS
5432 *
12E4 0004 12E5 5433 BKFOTL DC AL(@CADDR)(B@LFRT) 'FOR' TABLE ENTRY LENGTH

```

S/3 BASIC COMPILER -FOR- STATEMENT ROUTINE							
ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 66
	12E6	0002	12E7	5434	BKFBN2 DC	IL(@VADDR)'2'	BINARY INTEGER *2
	12E8	F1	12E8	5435	BKFOC1 DC	CL1'1'	EBCDIC 1
	12E9	0003	12EA	5436	BKFOX3 DC	AL(@VADDR)(B@LFOR)	BINARY INTEGER *3
				5437	*		
				5438	* EQUATES		
				5439	*		
			0027	5440	BKFLLP EQU	B@LFOR+B@LNXT+B@LDWA+2*B@LELP-1	LONG PREC 'FOR' SEQ LNG
			0001	5441	BKFLSP EQU	1	LENGTH OF 'STEP'-2
			0001	5442	BKFOCV EQU	1	DISP FOR 'FOR' TABLE CTRL VAR.
			0003	5443	BKFONL EQU	3	DISP FOR 'FOR' TABLE NXT VADDR
				5444	*		
				5445	*****		
				5446	*		
				5447	* END OF 'FOR' STATEMENT ROUTINE CODING		
				5448	*		

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 67
		5450		*****			
		5451	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		5452	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		5453	*				*
		5454		*****			*
		5455	*	*STATUS			*
		5456	*	VERSION 1 MODIFICATION 0			*
		5457	*				*
		5458	*	*FUNCTION			*
		5459	*	BXDPRT IS EXECUTED TO TRANSLATE PRINT STATEMENTS AS THEY OCCUR,			*
		5460	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE			*
		5461	*	PSEUDOCODE IN VIRTUAL MEMORY.			*
		5462	*				*
		5463	*	*ENTRY POINTS			*
		5464	*	BXDPRT HAS ONLY ONE ENTRY POINT:			*
		5465	*	BXDPRT - TRANSLATE PRINT STATEMENT			*
		5466	*	THE FORMAT OF THE CALLII4 SEQUENCE IS;			*
		5467	*	B BXDPRT			*
		5468	*				*
		5469	*	*INPUT			*
		5470	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING,			*
		5471	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		5472	*	LEADING KEYWORD, PRINT.			*
		5473	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST,			*
		5474	*	CHARACTER IN THE LEADING KEYWORD, PRINT.			*
		5475	*				*
		5476	*	*OUTPUT			*
		5477	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		5478	*	GENERATED BY BXDPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		5479	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		5480	*	SEQUENCES.			*
		5481	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		5482	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		5483	*				*
		5484	*	*EXTERNAL REFERENCES			*
		5485	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		5486	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$ARSW) - ENTRY TO COMPILER			*
		5487	*	VIRTUAL MEMORY OUTPUT ROUTINE.			*
		5488	*	B\$FCON - (B\$CTYP, B\$BCKT, B\$@PCT) - ENTRY TO BASIC COMPILER			*
		5489	*	CONSTANT ROUTINE.			*
		5490	*	B\$CSCN - (B\$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN			*
		5491	*	ROUTINE.			*
		5492	*	B\$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN			*
		5493	*	ROUTINE.			*
		5494	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		5495	*				*
		5496	*	*EXITS, NORMAL			*
		5497	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		5498	*				*
		5499	*	*EXITS, ERROR			*
		5500	*	N/A			*
		5501	*				*
		5502	*	*TABLES/WORK AREAS			*
		5503	*	* PRINT CODE TABLE - INTERNAL TO BXDPRT, THIS TABLE CONTAINS PRS			*
		5504	*	INSTRUCTION CODES ASSOCIATED WITH PRINT LIST DELIMITERS.			*
		5505	*	DELIMITERS REQUIRE DIFFERENT CODES DEPENDING ON THE CLASS OF			*

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 68
				5506	* THE PRECEDING LIST ELEMENT.	*
				5507	*	*
				5508	*ATTRIBUTES	*
				5509	* * BXDPRT IS NATURALLY RELOCATABLE AND REUSABLE.	*
				5510	*	*
				5511	*CHARACTER CODE DEPENDENCY	*
				5512	* THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRESENTA-	*
				5513	* TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE ONE	*
				5514	* USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT	*
				5515	* REDEFINITION OF CHARACTER CONSTAN1S, BY REASSEMBLY, WILL RESULT	*
				5516	* IN A CORRECT MODULE FOR THE NEW DEFINITIONS.	*
				5517	*	*
				5518	*NOTES	*
				5519	* ERROR PROCEDURES	*
				5520	* N/A	*
				5521	*	*
				5522	* REGISTER USAGE	*
				5523	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				5524	*	*
				5525	* SAVED/RESTORED AREAS	*
				5526	* N/A	*
				5527	*	*
				5528	* MODIFICATION CONSIDERATIONS	*
				5529	* BXDPRT RESIDES ON ONE SECTOR AND HAS ONLY 9 BYTES AVAILABLE	*
				5530	* FOR MODIFICATION. IF A SIGNIFICANT CHANGE IN SIZE IS REQUIRED	*
				5531	* LINKAGE WOULD HAVE TO BE ESTABLISHED TO A SECOND SECTOR.	*
				5532	*	*
				5533	* REQUIRED MODULES	*
				5534	* @SYSEQ - COMMON SYSTEM EQUATES.	*
				5535	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
				5536	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
				5537	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
				5538	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
				5539	* @ERMEQ - ERROR MESSAGE EQUATES.	*
				5540	* \$V\$EQ - FIXED VIRTUAL ADDRESS EQUATES.	*
				5541	* \$B\$EQ - COMPILER FIXED EQUATES.	*
				5542	* \$B@EQ - COMPILER SYSTEM EQUATES.	*
				5543	*	*
				5544	* OTHER	*
				5545	* BXDPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS	*
				5546	*****	*
1300				5548	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
			1300	5549	USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
				5550	*	
				5551	* ENTER BXDPRT - 'PRINT' STATEMENT ROUTINE	
				5552	*	
			1300	5553	BXDPRT EQU *	
				5554	*	
				5555	* SKIP TO LETTER 'T' IN KEYWORD 'PRINT'	
				5556	*	
1300	3C 04 0873			5557	BXD010 MVI B\$NUMC,B@LPRT-1	SET GET RTN TO SKIP TO 'T'
1304	C0 87 0867			5558	B B\$GETC	LINK TO ADVANCE POINTER
				5559	*	
				5560	* INITIALIZE THE SUBROUTINE	
				5561	*	

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 69

```

1308 7B 07 A8          5562 BXD020 SBF   BXDRS1(,@BR),BXDRM1      SET PRINT LIST SWITCH OFF
                        5563 *
                        5564 * SET THE 'PRINT AND SPACE' CODE TABLE MODE TO 1
                        5565 *
130B 7C C8 2D          5566 BXD030 MVI   BXD090+@D1(,@BR),BXDMD1-BXDPRT-BXDLTH SET NODE TO 1
130E 7C 5B D6          5567          MVI   BXDM14(,@BR),BXD180-BXDPRT SET MODE 1 BRANCH ADDRESS
                        5568 *
                        5569 * ATTEMPT TO GENERATE PMC FOR ARITH EXPR BY CALLING ARITH SCAN ATM
                        5570 *
N04 1311 00 00 0000    5571 BXD040 B      B@SCAN              LINK TO ATTEMPT PMC GENERATION
                        5572 *
                        5573 * TEST LIST ELEMENT FOR BEING A CHARACTER VARIABLE
                        5574 *
1315 38 07 14BC        5575 BXD050 TBN   B$CSSW,B$CSMK              TEST FOR CHAR VARIAVE
                        5576 *
                        5577 * IF ELEMENT IS A CHARACTER VARIABLE CALL THE CHAR SCAN RWTINE
                        5578 *
1319 C0 10 14B0        5579 BXD060 BT     B$CSCN              LINK TO PROCESS CHAR VARIABLE
                        5580 *
                        5581 * TEST FOR ANY PMC HAVING BEEN GENERATED FOR THIS ELEMENT
                        5582 *
131D 38 01 0A45        5583 BXD065 TBN   B$ARSW,B$ARMK          IF PMC'S GENERATED
1321 F2 10 03          5584          JT     BXD080              * GO SEARCH TABLE
                        5585 *
                        5586 * SET THE 'PRINT AND SPACE' CODE TABLE MODE TO TWO
                        5587 *
1324 7C D4 2D          5588 BXD070 MVI   BXD090+@D1(,@BR),BXDMD2-BXDPRT-BXDLTH SET MODE TO 2
                        5589 *
                        5590 * SEARCH THE 'PRINT AND SPACE' CODE TABLE FOR PRS CODE AND BRANCH ADDR
                        5591 * FOR LIST DELIMITER
                        5592 *
1327 6C 00 32 00      5593 BXD080 MVC   BXD100+@Q(,@BR),B@CHAR(1,@XR) SAVE TEXT CHARACTER
132B D2 02 00          5594 BXD090 LA     *-*(,@BR),@XR          LOAD ADDR OF PB$ TABLE NODE
                        5595 *
132E E2 02 03          5596 BXD095 LA     BXDLTH(,@XR),@XR          INCREMENT TABLE BY ENTRY LENGTH
1331 BD 00 00          5597 BXD100 CLI   BXDDP0(,@XR),*-*        IF LIST AND TABLE DELIMITERS
1334 D0 81 3D          5598          BE     BXD110(,@BR)          * GO SET CODE AND BRANCH ADDR
1337 BD 00 00          5599          CLI   BXDDP0(,@XR),BXDDUM    IF DELIMITER IS NOT DUMMY ENTRY
133A D0 01 2E          5600          BNE   BXD095(,@BR)          BRANCH TO NEXT COMPARE
                        5601 *
                        5602 * SET PRS CODE AND BRANCH TO THE ADDRESS LISTED IN THE TABLE
                        5603 *
N04 133D 00 00 00 00    5604 BXD110 MVC   BXDPRO(,@BR),BXDOP1(1,@XR) SET PRS CODE IN PBS OPERAND
1341 6C 00 47 02      5605          MVC   BXD120+@D1(,@BR),BXDDP2(1,@XR) SET BRANCH DISPLACEMENT
1345 D0 87 00          5606 BXD120 B      *-*(,@BR)          BRANCH TO ADDR ACCORDING TO TBL
                        5607 *
                        5608 * GENERATE THE 'PRS' PMC INSTRUCTION IN VIRTUAL MEMORY
                        5609 *
1348 D0 87 B1          5610 BXD140 B      BXD300(,@BR)          LINK TO GENERATE 'PRS' PMC
                        5611 *
                        5612 * SET THE PRINT LIST SWITCH ON
                        5613 *
134B 7A 07 A8          5614 BXD150 SBN   BXDRS1(,@BR),BXDRM1      SET PRINT LIST SWITCH ON
134E D0 87 0B          5615          B      BXD030(,@BR)          BRANCH TO PROCESS NEXT ELEMENT
                        5616 *
                        5617 * GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY

```



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 70
					5618	*				
	1351	D0	87	B1	5619	BXD160 B	BXD300(, @BR)		LINK TO GENERATE 'PRS' PMC	
					5620	*				
					5621	*	DISABLE THE GET ROUTINE FOR THE NEXT EXECUTION OF ARM SCAN ROUTINE			
					5622	*				
N04	1354	00	00	0000	5623	BXD170 MVI	B\$NUMC, B\$GETS		SET GET RTN NOT TO SKIP CHAR	
	1358	D0	87	0B	5624	B	BXD030(, @BR)		BRANCH TO PROCESS NEXT ELEMENT	
					5625	*				
					5626	*	GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY			
					5627	*				
	135B	D0	87	B1	5628	BXD180 B	BXD300(, @BR)		LINK TO GENERATE 'PRS' PMC	
					5629	*				
					5630	*	CALL CONSTANT ROUTINE TO GENERATE CHARACTER STRING IN V.M.			
					5631	*				
	135E	3C	1B	0A5F	5632	BXD190 MVI	B\$CTYP, B\$SCON		SET CON RTN FOR CHAR STRING	
	1362	35	02	0878	5633	L	B\$GPTR, @XR		RESTORE TEXT POINTER	
	1366	C0	87	0A46	5634	B	B\$FCON		LINK TO GENERATE CHAR STRING	
					5635	*				
					5636	*	TEST FOR THIS ELEMENT BEING A NULL CHARACTER STRING			
					5637	*				
	136A	7C	E0	2D	5638	BXD200 MVI	BXD090+@D1(, @BR), BXDMD3-BXDPRT-BXDLTH		SET MODE TO 3	
	136D	3D	00	0CA8	5639	CLI	B\$CPCT, @ZERO		IF THIS IS A NULL STRING	
	1371	D0	81	27	5640	BE	BXD080(, @BR)		* GO SEARCH 'PRS' BRANCH TABLE	
					5641	*				
					5642	*	SET 'PRINT AND SPACE' CODE TABLE MODE TO FOUR			
					5643	*				
N04	1374	00	00	00	5644	BXD210 MVI	BXD090+@D1(, @BR), BXDMP1-BXDPRT-BXDLTH		SET MODE TO 4	
	1377	7C	51	D6	5645	MVI	BXDM14(, @BR), BXD160-BXDPRT		SET MODE 4 BRANCH ADDRESS	
	137A	7C	08	F3	5646	MVI	BXDPRO(, @BR), B@PRRL		SET CODE FOR PRINT LONG	
	137D	BD	6B	00	5647	CLI	B@CHAR(, @XR), B@CMA		IF DELIMITER IS A COMMA	
	1380	D0	81	B1	5648	BE	BXD300(, @BR)		* LINK TO GENERATE FMC	
	1383	7C	01	F3	5649	MVI	BXDPRO(, @BR), B@PRPN		SET CODE FOR PRINT AND NO SPACE	
					5650	*				
					5651	*	MOVE THE VADDR OF THE 1ST STRING SEGMENT TO AN 'STC' INSTRUCTION			
					5652	*				
N04	1386	00	00	00 0000	5653	BXD220 MVC	BXDSTO(, @BR), BSBCKT(@VADDR)		MOVE VADDR OF 1ST CON TO OPKD	
					5654	*				
					5655	*	GENERATE THE 'STC' INSTRUCTION IN VIRTUAL MEMORY			
					5656	*				
	138B	D2	02	F4	5657	BXD230 LA	BXDSTC(, @BR), @XR		LOAD CADDR OF 'STC' INSTR	
	138E	3C	02	0A41	5658	MVI	B\$PNBY, B@LSTC-1		SET PUT RTN LNG PARM FOR 'STC'	
	1392	D0	87	B8	5659	B	BXD310(, @BR)		LINK TO GENERATE 'STC' PMC	
					5660	*				
					5661	*	TEST FOR THE EXISTENCE OF ANOTHER SEGMENT IN THE CHARACTER STRING			
					5662	*				
	1395	1F	00	0CA8 EF	5663	BXD240 SLC	B\$CPCT, BXDBN1(1, @BR)		IF NO OTHER SEGMENTS EXIST	
	139A	D0	81	27	5664	BE	BXD080(, @BR)		* GO SEARCH PRS TABLE	
					5665	*				
					5666	*	IF ANOTHER SEGMENT DOES EXIST GENERATE THE 'PRS' PMC IN V.M.			
					5667	*				
	139D	D0	87	B1	5668	BXD250 B	BXD300(, @BR)		LINK TO GENERATE 'PRS' PMC	
					5669	*				
					5670	*	SUBTRACT THE LENGTH OF A STRING SEGMENT FROM THE 'STC' OPERAND			
					5671	*				
	13A0	5F	01	F6 F1	5672	BXD260 SLC	BXDSTO(, @BR), BXDSUB(@VADDR, @BR)		SUB SEGMENT LENGTH	
	13A4	D0	87	8B	5673	B	BXD230(, @BR)		BRANCH TO GENERATE 'STC' PMC	



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	20/07/20	PAGE 71
					5674	*					
					5675	*	TEST FOR THE PRINT LIST WNITCH BEING ON				
					5676	*					
13A7	F2	00	03		5677	BXD270	JC BXD290,*-*				IF LIST SWITCH IS ON
13A8					5678		ORG BXD270+@Q				* GO BRANCH TO DIST
13A8	80			13A8	5679		DC AL1(@NOP)				IF LIST SWITCH IS OFF
13AA					5680		ORG BXD270+@INST3				* GO BRANCH TO GENERATE PMC
					5681	*					
					5682	*	GENERATE THE 'PRS' INSTRUCTION IN VIRTUAL MEMORY				
					5683	*					
13AA	D0	87	B1		5684	BXD280	B BXD300(,@BR)				LINK TO GENERATE 'PRS' PMC
					5685	*					
					5686	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR				
					5687	*					
13AD	C0	87	0700		5688	BXD290	B B\$DIST				RETURN TO THE DISTRIBUTOR

## S/3 BASIC COMPILER -PRINT- STATEMENT RTN

```

ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT      VER 15, MOD 00  20/07/20  PAGE  72

5690 *****
5691 * SUBROUTINE FOR THE GENERATION OF PSEUDOCODE IN VIRTUAL MEMORY - *
5692 * * THE ENTIRE ROUTINE IS USED TO GENERATE THE 'PRS' INSTRUCTION *
5693 * * AND A SECOND ENTRY POINT ALLOWS THE ROUTINE TO COMPLETE THE *
5694 * * GENERATION OF THE 'STC' INSTRUCTION *
5695 *****
5696 *
5697 * ENTER GENERATE SUBROUTINE - FOR 'PRS' GENERATION
5698 *
13B1 D2 02 F2      5699 BXD300 LA      BXDPRC(,@BR),@XR      LOAD CADDR OF 'PRS' INSTC
13B4 3C 01 0A41    5700          MVI    B$PNBY,B@LPRS-1      SET PUT RTN LENGTH PARM
5701 *
5702 * SECONDARY ENTRY POINT TO GENERATE SUBROUTINE - FOR 'STC' GENERATION
5703 *
13B8 74 08 CA      5704 BXD310 ST      BXD320+@OP1(,@BR),@ARR    STORE RETURN ADDRESS
13BB 34 02 0A40    5705          ST      B$PCAD,@XR      SET PUT RTN VADDR PARM
13BF C0 87 093A    5706          B      B$PUTC      LINK TO GENERATE PMC
13C3 35 02 0878    5707          L      B$GPTR,@XR      RESTORE TEXT POINTER
13C7 C0 87 0000    5708 BXD320 B      *- *      BRANCH TO RETURN ADDRESS

5710 *****
5711 * PRINT STATEMENT 'PRINT AND SPACE' CODE TABLE
5712 *****
5713 *
0003 5714 BXDLTH EQU    3      LENGTH OF CODE TABLE ENTRY
0004 5715 BXDROM EQU    4      NUMBER OF ENTRIES PER MODE
5716 *
0000 5717 BXDDUM EQU    X'00'  TABLE DUMMY COMPARE

5719 *****
5720 * PRINT CODE TABLE MODE FOR LIST ELEMENT AND EXPRESSION PROCESSING
5721 *****
5722 *
13CB 6B          13CB 5723 BXDMD1 EQU    *      PRS TABLE - MODES I AND 4
13CB 6B          13CB 5724          DC      AL1(B@CMMA)    DELIMITER - COMMA
13CC 02          13CC 5725          DC      AL1(B@PRPL)    PRINT AND SPACE TO LONG ZONE
13CD 48          13CD 5726          DC      AL1(BXD140-BXDPRT) BRANCH ADDRESS
5727 *
N04 13CE 00      13CE 5728          DC      AL1(B$SCLN)    DELIMITER - SEMI-COLON
13CF 03          13CF 5729          DC      AL1(B@PRPS)    PRINT AND SPACE TO SHORT ZONE
13D0 48          13D0 5730          DC      AL1(BXD140-BXDPRT) BRANCH ADDRESS
5731 *
N04 13D1 00      13D1 5732          DC      AL1(B$EOST)    DELIMITER - END OF STATEMENT
13D2 04          13D2 5733          DC      AL1(B@PRPR)    PRINT AND RETURN CARRIAGE
13D3 AA          13D3 5734          DC      AL1(BXD280-BXDPRT) BRANCH ADDRESS
5735 *
13D4 00          13D4 5736          DC      AL1(BXDDUM)    DELIMITER - NOT , OR ; OR CR
13D5 01          13D5 5737          DC      AL1(B@PRPN)    PRINT AND NO SPACE
13D6          13D6 5738 BXDM14 DS      CL1      BRANCH ADDRESS

5740 *****
5741 * PRINT CODE TABLE MODE FOR CHARACTER STRING PROCESSING
5742 *****
5743 *
13D7 5744 BXDMD2 EQU    *      PRS TABLE - MODE 2
N04 13D7 00      13D7 5745          DC      AL1(B$CMMA)    DELIMITER - COMMA

```

## S/3 BASIC COMPILER -PRINT- STATEMENT RTN

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 73
N04	13D8	00	13D8	5746	DC	AL1(B\$PRSL)	SPACE TO LONG ZONE
	13D9	48	13D9	5747	DC	AL1(BXD140-BXDPRT)	BRANCH ADDRESS
				5748	*		
	13DA	5E	13DA	5749	DC	AL1(B@SCLN)	DELIMITER - SEMI-COLON
N04	13DB	00	13DB	5750	DC	AL1(B\$PRSS)	SPACE TO SHORT ZONE
	13DC	48	13DC	5751	DC	AL1(BXD140-BXDPRT)	BRANCH ADDRESS
				5752	*		
N04	13DD	00	13DD	5753	DC	AL1(B2EOST)	DELIMITER - END OF STATEMENT
	13DE	07	13DE	5754	DC	AL1(B@PRRC)	RETURN THE CARRIAGE
	13DF	A7	13DF	5755	DC	AL1(BXD270-BXDPRT)	BRANCH ADDRESS
				5756	*		
	13E0	00	13E0	5757	DC	AL1(BXDDUM)	DELIMITER - NOT , OR ; OR CR
	13E1	01	13E1	5758	DC	AL1(B@PRPN)	PRINT AND NO SPACE
	13E2	5E	13E2	5759	DC	AL1(BXD190-BXDPRT)	BRANCH ADDRESS
				5761	*****		
				5762	* PRINT CODE TABLE MODE FOR NULL STRING PROCESSING		
				5763	*****		
				5764	*		
	13E3	6B	13E3	5765	BXDMD3 EQU	*	PRS TABLE - MODE 3
	13E4	05	13E4	5766	DC	AL1(B@CMMA)	DELIMITER - COMMA
	13E5	48	13E5	5767	DC	AL1(B@PRSL)	SPACE TO LONG ZONE
				5768	DC	AL1(BXD140-BXDPRT)	BRANCH ADDRESS
				5769	*		
	13E6	5E	13E6	5770	DC	AL1(B@SCLN)	DELIMITER - SEMI-COLON
	13E7	01	13E7	5771	DC	AL1(B@PRPN)	PRINT AND NO SPACE
	13E8	0B	13E8	5772	DC	AL1(BXD030-BXDPRT)	BRANCH ADDRESS
				5773	*		
	13E9	1E	13E9	5774	DC	AL1(B@EOST)	DELIMITER - END OF STATEMENT
	13EA	07	13EA	5775	DC	AL1(B@PRRC)	RETURN THE CARRIAGE
	13EB	AA	13EB	5776	DC	AL1(BXD280-BXDPRT)	BRANCH ADDRESS
				5777	*		
	13EC	00	13EC	5778	DC	AL1(BXDDUM)	DELIMITER - NOT . OR ; OR CR
	13ED	01	13ED	5779	DC	AL1(B@PRPN)	PRINT AND NO SPACE
	13EE	54	13EE	5780	DC	AL1(BXD170-BXDPRT)	BRANCH ADDRESS
				5782	*****		
				5783	* PRINT STATEMENT ROUTINE CONSTANTS AND EQUATES		
				5784	*****		
				5785	*		
				5786	* EQUATES		
				5787	*		
			0000	5788	BXDDP0 EQU	0	PRS TABLE DISP FOR DELIMITER
			0001	5789	BXDDP1 EQU	1	PRS TABLE DISP FOR CODE
			0002	5790	BXDDP2 EQU	2	PRS TABLE DISP FOR BRANCH ADDR
			0009	5791	BXDDMY EQU	BXDLTH*3	PRS TABLE DISP TO DUMMY ENTRY
				5792	*		
				5793	* CONSTANT		
				5794	*		
	13EF	01	13EF	5795	BXDBN1 DC	IL(B@LCNN) '1'	BINARY 1
	13F0	0013	13F1	5796	BXDSUB DC	AL(@VADDR) (B@LCRV)	LENGTH OF SEGMENT TO SUB
				5798	*****		
				5799	* PRINT STATEMENT ROUTINE STORAGE AND PARAMETER AREA		
				5800	*****		
				5801	*		

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15,	MOD 00	20/07/20	PAGE	74
	13F2	60		13F2	5802	BXDPRC	DC	AL(B@LCOP)	(B@CPRS)		PRINT AND SPACE	OPCODE
	13F3			13F3	5803	BXDPRO	DS	CL(B@LCXX)			PRINT AND SPACE	OPERAND
					5804	*						
	13F4	28		13F4	5805	BXDSTC	DC	AL(B@LCOP)	(B@CSTC)		STACK CHARACTER	OPCODE
	13F5			13F6	5806	BXDSTO	DS	CL(@VADDR)			STACK CHARACTER	OPERAND
					5808	*****						
					5809	* PRINT STATEMENT ROUTINE PROGRAM SWITCHES						
					5810	*****						
					5811	*						
				13A8	5812	BXDRS1	EQU	BXD270+@Q			PRINT LIST SWITCH	
				0007	5813	BXDRM1	EQU	@UCB-@NOP			PRINT LIST SWITCH MASK	
					5814	*						
					5815	*****						
					5816	*						
					5817	* END OF 'PRINT' STATEMENT ROUTINE CODING						
					5818	*						

ERR LOC	OBJECT CODE	ADDR STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 75
		5820	*****			*
		5821	* 5703-XM1 COPYRIGHT IBM CORP. 1970			*
		5822	* REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		5823	*			*
		5824	*****			*
		5825	*STATUS			*
		5826	* VERSION 1 MODIFICATION 0			*
		5827	*			*
		5828	*FUNCTION			*
		5829	* BXUPRT IS EXECUTED TO TRANSLATE PRINT USING STATEMENTS AS THEY			*
		5830	* OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO			*
		5831	* PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		5832	*			*
		5833	*ENTRY POINTS			*
		5834	* BXUPRT HAS ONLY ONE ENTRY POINT:			*
		5835	* BXUPRT - TRANSLATE PRINT USING STATEMENT			*
		5836	* THE FORMAT OF THE CALLING SEQUENCE IS:			*
		5837	* B BXUPRT			*
		5838	*			*
		5839	*INPUT			*
		5840	* * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		5841	* THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE			*
		5842	* LEADING KEYWORD, PRINT USING.			*
		5843	* * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		5844	* CHARACTER IN THE LEADING KEYWORD. PRINT USING.			*
		5845	*			*
		5846	*OUTPUT			*
		5847	* * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		5848	* GENERATED BY EXUPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		5849	* MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		5850	* SEQUENCES.			*
		5851	* * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		5852	* CHARACTER WHICH TERMINATES THE STATEMENT.			*
		5853	*			*
		5854	*EXTERNAL REFERENCES			*
		5855	* B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		5856	* B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD, B\$ARSW) - ENTRY TO COMPILER			*
		5857	* VIRTUAL MEMORY OUTPUT ROUTINE.			*
		5858	* B\$FCON - (B\$CTYP, B\$CKCT, B\$CPCT) - ENTRY TO BASIC COMPILER			*
		5859	* CONSTANT ROUTINE.			*
		5860	* B\$CSCN - (B\$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN			*
		5861	* ROUTINE.			*
		5862	* B\$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN			*
		5863	* ROUTINE.			*
		5864	* B\$BTAB - (B\$BRVA, B\$IRLN) - ENTRY TO BASIC COMPILER BRANCH			*
		5865	* TABLE ROUTINE.			*
		5866	* B\$ZDBN - (B\$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO BINARY			*
		5867	* CONVERSION ROUTINE.			*
		5868	* B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		5869	*			*
		5870	*EXITS, NORMAL			*
		5871	* B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		5872	*			*
		5873	*EXITS, ERROR			*
		5874	* N/A			*
		5875	*			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 76
			5876	*TABLES/WORK AREAS	*
			5877	* N/A	*
			5878	*	*
			5879	*ATTRIBUTES	*
			5880	* BXUPRT IS NATURALLY RELOCATABLE AND REUSABLE.	*
			5881	*	*
			5882	*CHARACTER CODE DEPENDENCY	*
			5883	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
			5884	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
			5885	*	*
			5886	*NOTES	*
			5887	* ERROR PROCEDURES	*
			5888	* N/A	*
			5889	*	*
			5890	* REGISTER USAGE	*
			5891	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION	*
			5892	*	*
			5893	* SAVED/RESTORED AREAS	*
			5894	* N/A	*
			5895	*	*
			5896	* MODIFICATION CONSIDERATIONS	*
			5897	* BXUPRT RESIDES ON ONE SECTOR. THE LIMITATION OF THE SECTOR	*
			5898	* BOUNDARY ON SIZE SHOULD BE CONSIDERED IN MAKING MODIFICATIONS.	*
			5899	*	*
			5900	* REQUIRED MODULES	*
			5901	* @SYSEQ - COMMON SYSTEM EQUATES.	*
			5902	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
			5903	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
			5904	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
			5905	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
			5906	* @ERMEQ - ERROR MESSAGE EQUATES.	*
			5907	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
			5908	* \$B\$EQU - COMPILER FIXED EQUATES.	*
			5909	* \$B@EQU - COMPILER SYSTEM EQUATES.	*
			5910	*	*
			5911	* OTHER	*
			5912	* BXUPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
			5913	*****	*
1400			5915	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
		1400	5916	USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
			5917	*	
			5918	* ENTER BXUPRT - 'PRINT USING' STATEMENT ROUTINE	
			5919	*	
		1400	5920	BXUPRT EQU *	BXUPRT ENTRY POINT
			5921	*	
			5922	* SKIP TO CHARACTER FOLLOWING KEYWORDS 'PRINT USING'	
			5923	*	
1400 3C 0A 0873			5924	BXU010 MVI B\$NUMC,B@LKPU	SET GET RTN TO SKIP KEYWORDS
1404 C0 87 0867			5925	B B\$GETC	LINK TO ADVANCE POINTER
			5926	*	
			5927	* GENERATE AN 'STA' INSTRUCTION IMAGE IN VIRTUAL MEMORY	
			5928	*	
1408 D2 02 DC			5929	BXU020 LA BXUSTC(,@BR),@XR	LOAD CADDR OF 'STA' INSTR
140B 3C 02 0A41			5930	MVI B\$PNBY,B@LSTA-1	SET PUT RTN LNG PARM FOR STA
140F D0 87 C9			5931	B BXU360(,@BR)	LINK TO GENERATE 'STA' PMC

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 77
					5932	*				
					5933	*	ESTABLISH 'STA' OPERAND FOR ADDRESS RESOLUTION			
					5934	*				
	1412	0C	01	19EF	0A43	5935	BXU025 MVC B\$BRVA,B\$PVAD(@VADDR) SET ADDR FOR BRANCH TABLE			
	1418	1F	01	19EF	E8	5936	SLC B\$BRVA,BXUBN1(@VADDR,@BR) ADJUST VADDR TO 'STA' OPERAND			
					5937	*				
					5938	*	GENERATE A 'BNX' INSTRUCTION IMAGE IN VIRTUAL MEMORY			
					5939	*				
N04	141D	D2	02	DF		5940	BXU030 LA BXUBNC(,@BR),@XR LOAD CADDR OF 'BNX' INSTR			
	1420	00	00	0000		5941	MVI B\$PNBY,B@LINX-1 SET PUT RTN LNG PARM FOR 'BNX'			
	1424	D0	87	C9		5942	B BXU360(,@BR) LINK TO GENERATE 'BNX' PMC			
					5943	*				
					5944	*	ESTABLISH THE NEXT VADDR IN V.M.(BEGINNING OF DATA OUTPUT SEQUENCE,			
					5945	*	AS RESOLUTION ADDRESS			
					5946	*				
	1427	0C	01	19F1	0A43	5947	BXU040 MVC B\$BRLN,B\$PVAD(@VADDR) SET ADDR FOR BR TBL RESOLUTION			
					5948	*				
					5949	*	CALL BRANCH TABLE ROUTINE TO SET ADDRESS RESOLUTION CONDITIONS FOR			
					5950	*	THE 'STA' OPERAND			
					5951	*				
N04	142D	00	00	0000		5952	BXU050 B BDBTAB LINK TO SET RESOLUTION COND			
					5953	*				
					5954	*	ESTABLISH VADDR OF 'BNX' OPERAND FOR ADDRESS RESOLUTION			
					5955	*				
	1431	0C	01	19EF	0A43	5956	BXU060 MVC B\$BRVA,B\$PVAD(@VADDR) SET ADDRESS FOR BR TABLE			
	1437	1F	01	19EF	E8	5957	SLC B\$BRVA,BXUBN1(@VADDR,@BR) ADJUST VADDP TO 'BNX' OPERAND			
					5958	*				
					5959	*	CONVERT THE IMAGE LINE NUMBER TO BINARY FROM DECIMAL			
					5960	*				
	143C	C0	87	19F2		5961	BXU070 B B\$ZDBN LINK TO CONVERT LINE NO TO BIN			
					5962	*				
					5963	*	ESTABLISH THE IMAGE LINE NUMBER AS RESOLUTION LINE NUMBER			
					5964	*				
	1440	0C	01	19F1	1A6A	5965	BXU080 MVC B\$BRLN,B\$BINO(@VADDR) SET LN NO FOR BR TBL RESOLUTION			
					5966	*				
					5967	*	CALL BRANCH TABLE ROUTINE TO SET ADDRESS RESOLUTION CONDITIONS FOR			
					5968	*	THE 'BNX' OPERAND			
					5969	*				
	1446	C0	87	1996		5970	BXU090 B B\$BTAB LINK TO SET RESOLUTION COND			
					5971	*				
					5972	*	CHECK FOR THE PRESENCE OF LIST ELEMENTS			
					5973	*				
	144A	7D	1E	00		5974	BXU100 CLI B@CHAR(,@BR),B@EOST IF LIST ELEMENTS ARE PRESENT			
	144D	F2	01	10		5975	JNE BXU170 GO ATTEMPT PMC GENERATION			
					5976	*				
					5977	*	SET CODE FOR NO LIST ELEMENTS IN THE 'PRU' INSTRUCTION			
					5978	*				
	1450	7C	02	E3		5979	BXU110 MVI BXUPRO(,@BR),B@PUNL SET 'PRU' OPERAND FOR NO LIST			
					5980	*				
					5981	*	SET TERMINATOR FLAG TO INDICATE LAST OUTPUT FOR LIST			
					5982	*				
	1453	7A	10	E3		5983	BXU120 SBN BXUPRO(,@BR),B@PUTM SET LAST OUTPUT FOR LIST FLAG			
					5984	*				
					5985	*	GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY			
					5986	*				
	1456	D0	87	C2		5987	BXU130 B BXU350(,@BR) BRANCH TO GENERATE 'PRU' PMC			



ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 78
				5988	*	
				5989	* RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
				5990	*	
1459	C0	87 0700		5991	BXU140 B B\$DIST RETURN TO DISTRIBUTOR	
				5992	*	
				5993	* GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY	
				5994	*	
145D	D0	87 C2		5995	BXU150 B BXU350(,@BR) BRANCH TO GENERATE 'PRU' PMC	
				5996	*	
				5997	* CALL ARITH SCAN ROUTINE TO ATTEMPT PMC GENERATION OF ARUN EXPRESSION	
				5998	*	
1460	C0	87 1514		5999	BXU170 B B\$SCAN LINK TO ATTEMPT PMC GENERATION	
				6000	*	
				6001	* TEST FOR THIS LIST ELEMENT BEING A CHARACTER VARIABLE	
				6002	*	
1464	38	07 14BC		6003	BXU180 TBN B\$CSSW,B\$CSMK IF ELEMENT IS NOT A CHAR VAR	
1468	F2	90 04		6004	JF BXU200 * GO SET 'PRU' OPERAND	
				6005	*	
				6006	* IF THIS LIST ELEMENT IS A CHARACTER VARIABLE CALL THE CHAR SCAN RTN	
				6007	*	
146B	C0	87 14B0		6008	BXU190 B B\$CSCN LINK, GENERATE PMC FOR CHAR VAR	
				6009	*	
				6010	* SET 'PRU' OPERAND WITH CODE FOR ARITHMETIC OR CHARACTER EXPRESSION,	
				6011	* INCLUDING FIRST CONSTANT ESTABLISHED FOR A CHAR STRING BUT EXCLUDING	
				6012	* A NULL CHAR STRING	
				6013	*	
146F	7C	06 E3		6014	BXU200 MVI BXUPRO(,@BR),B@PUD1 SET 'PRU' OPERAND CODE	
				6015	*	
				6016	* TEST FOR ANY PMC HAVING BEEN GENERATED FOR THIS ELEMENT	
				6017	*	
1472	38	01 0A45		6018	BXU210 TBN B\$ARSW,B\$ARMK IF NO PMC GENERATED	
1476	F2	90 0D		6019	JF BXU230 * GO BRANCH TO CONSTANT RTN	
				6020	*	
				6021	* TEST FOR DELIMITER BEING AN END OF STATEMENT	
				6022	*	
1479	35	02 0878		6023	BXU220 L B\$GPTR,@XR RESTORE TEXT POINTER	
147D	BD	1E 00		6024	CLI B@CHAR(,@XR),B@EOST IF DELIMITER IS NOT TERMINATOR	
1480	D0	01 5D		6025	BNE BXU150(,@BR) * GO GENERATE 'PRU' PMC	
1483	D0	87 53		6026	B BXU120(,@BR) GO SET LAST LIST OUTPUT FLAG	
				6027	*	
				6028	* CALL CONSTANT ROUTINE TO GENERATE CHARACTER STRING IN V.M.	
				6029	*	
1486	3C	1B 0A5F		6030	BXU230 MVI B\$CTYP,B\$SCON SET CON RTN FOR CHAR STRING	
148A	C0	87 0A46		6031	B B\$FCON LINK TO GENERATE CHAR STRING	
				6032	*	
				6033	* TEST FOR THIS BEING A NULL STRING	
				6034	*	
148E	3D	00 0CA8		6035	BXU240 CLI B\$CPCT,@ZERO IF THIS IS A NOT A NULL STRING	
1492	F2	01 06		6036	JNE BXU260 * MOVE 1ST SEGMENT VADDR TO STC	
				6037	*	
				6038	* IF THIS IS A NULL CHARACTER STRING SET CODE IN 'PRU' OPERAND	
				6039	*	
1495	7C	03 E3		6040	BXU250 MVI BXUPRO(,@BR),B@PUNS SET 'PRU' OPND FOR NULL STRING	
1498	D0	87 79		6041	B BXU220(,@BR) GO CHECK FOR OTHER ELEMENTS	
				6042	*	
				6043	* MOVE THE VADDR OF THE FIRST STRING SEGMENT TO AN 'STC' INSTR OPWD	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 79
				6044	*	
	149B	4C 01 E6 1590		6045	BXU260 MVC BXUSCO(,@BR),B\$BCKT(@VADDR) SET 1ST SEGMENT VADDR IN OPND	
				6046	*	
				6047	* SET THE 'PRU' OPND CODE FOR ARITH AND CHAR EXPRESSIONS	
				6048	*	
	14A0	7C 06 E2		6049	BXU270 MVI BXUPRC(,@BR),B@PUD1 SET 'PRU' OPERAND CODE	
				6050	*	
				6051	* GENERATE THE 'STU' PMC INSTRUCTION IN VIRTUAL MEMORY	
				6052	*	
	14A3	D2 02 E4		6053	BXU280 LA BXUSCC(,@BR),@XR LOAD CADDR OF 'STC' INSTR	
	14A6	3C 02 0A41		6054	MVI B\$PNBY,B@LSTC-1 SET PUT RTN LNG PARM FOR 'STC'	
	14AA	D0 87 C9		6055	B BXU360(,@BR) LINK TO GENERATE 'STC' PMC	
				6056	*	
				6057	* TEST FOR THE EXISTENCE OF ANOTHER SEGMENT	
				6058	*	
	14AD	1F 00 0CA8 E8		6059	BXU290 SLC B\$CPCT,BXUBN1(1,@BR) IF NO OTHER SEGMENTS EXIST	
	14B2	D0 04 79		6060	BNH BXU220(,@BR) * GO TEST FOR OTHER ELEMENTS	
				6061	*	
				6062	* GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY	
				6063	*	
	14B5	D0 87 C2		6064	BXU300 B BXU350(,@BR) BRANCH TO GENERATE 'PRU' PMC	
				6065	*	
				6066	* SET 'PRU' OPERAND CODE FOR ANY CONSTANT ESTABLISHED FOR A CHAR STRING	
				6067	* EXCEPT FOR THE FIRST CONSTANT IN THAT STRING SERIES	
				6068	*	
	14B8	7C 07 E3		6069	BXU310 MVI BXUPRO(,@BR),B@PUD2 SET 'PRU' OPND CODE	
				6070	*	
				6071	* SUBTRACT LENGTH OF STRING SEGMENT FROM 'STC' INSTRUCTION OPERAND	
				6072	*	
	14BB	5F 01 E6 EA		6073	BXU320 SLC BXUSCO(,@BR),BXUSUB(@VADDR,@BR) SUB SEGMENT LENGTH	
				6074	*	
				6075	* BRANCH TO CONTINUE GENERATING THE 'STC'/'PRU' SEQ FOR THE CHAR STRING	
				6076	*	
	14BF	D0 87 A3		6077	BXU340 B BXU280(,@BR) BRANCH TO GENERATE 'STC' INSTR	
				6078	*	
				6079	*****	
				6080	* SUBROUTINE FOR THE GENERATION OF PSEUDOCODE IN VIRTUAL MEMORY - *	
				6081	* * THE ENTIRE ROUTINE IS USED TO GENERATE THE .PRU. INSTRUCTION *	
				6082	* * AND SECONDARY ENTRY POINT ALLOWS THE ROUTINE TO COMPLETE THE *	
				6083	* * GENERATION FOR THE 'STA', 'BNX' AND 'STC' INSTRUCTIONS. *	
				6084	*****	
				6086	*	
				6087	* ENTER THE GENERATE SUBROUTINE - FOR 'PRU' INSTRUCTION	
				6088	*	
	14C2	D2 02 E2		6089	BXU350 LA BXUPRC(,@BR),@XR LOAD CADDR OF 'PRU' INSTR	
	14C5	3C 01 0A41		6090	MVI B\$PNBY,B@LPRU-1 SET PUT RTN FOR LENGTH PARM	
				6091	*	
				6092	* SECONDARY ENTRY POINT TO GENERATE SUBROUTINE FOR 'STA', 'BNX', 'STC'	
				6093	*	
	14C9	74 08 DB		6094	BXU360 ST BXU370+@OP1(,@BR),@ARR STORE RETURN ADDRESS	
	14CC	34 02 0A40		6095	ST B\$PCAD,@XR SET PUT RTN VADDR PARM	
	14D0	C0 87 093A		6096	B B\$PUTC LINK TO GENERATE PMC	
	14D4	35 02 0878		6097	L B\$GPTR,@XR RESTORE TEXT POINTER	
	14D8	C0 87 0000		6098	BXU370 B *-* BRANCH TO RETURN ADDRESS	

```

        6100 *****
        6101 * PRINT USING STATEMENT RTN PARAMETER AND STORAGE AREAS
        6102 *****
        6103 *
14DC 34      14DC 6104 BXUSTC DC      AL(B@LCOP)(B@CSTA)      'STA' INSTR OPCODE
14DD 0000    14DE 6105 BXUSTO DC      XL(B@LCVA)'00'          'STA' INSTR OPERAND IMAGE
        6106 *
14DF 4A      14DF 6107 BXUBNC DC      AL(B@LCOP)(B@CBNX)      'INX' INSTR OPCODE
14E0 0000    14E1 6108 BXUBNO DC      XL(B@LCVA)'00'          'INX' INSTR OPERAND IMAGE
        6109 *
14E2 62      14E2 6110 BXUPRC DC      AL(B@LCOP)(B@CPRU)      'PRU' INSTR OPCODE
14E3          14E3 6111 BXUPRO DS      CL(B@LCXX)              'PRU' INSTR OPERAND
        6112 *
14E4 28      14E4 6113 BXUSCC DC      AL(B@LCOP)(B@CSTC)      'STC' INSTR OPCODE
14E5          14E6 6114 BXUSCO DS      CL(B@LCVA)              'STC' INSTR OPERAND
        6116 *****
        6117 * PRINT USING STATEMENT ROUTINE CONSTANTS
        6118 *****
        6119 *
14E7 0001    14E8 6120 BXUBN1 DC      IL(@VADDR)'1'          BINARY 1
        6121 *
14E9 0013    14EA 6122 BXUSUB DC      AL(@VADDR)(B@LCRV)      LENGTH OF STRING SEGMENT
        6123 *
        6124 *****
        6125 *
        6126 * END OF 'PRINT USING' STATEMENT ROUTINE CODING
        6127 *
```

## S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 81
	6129				*****			
	6130	*			5703-XM1 COPYRIGHT IBM CORP. 1970			*
	6131	*			REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
	6132	*						*
	6133				*****			*
	6134	*			*STATUS			*
	6135	*			VERSION 1 MODIFICATION 0			*
	6136	*						*
	6137	*			*FUNCTION			*
	6138	*			BNFDEF IS EXECUTED TO TRANSLATE DEF STATEMENTS AS THEY OCCUR IN A			*
	6139	*			BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE			*
	6140	*			PSEUDOCODE IN VIRTUAL MEMORY.			*
	6141	*						*
	6142	*			*ENTRY POINTS			*
	6143	*			BNFDEF HAS ONLY ONE ENTRY POINT:			*
	6144	*			BNFDEF - TRANSLATE DEF STATEMENT			*
	6145	*			THE FORMAT OF THE CALLING SEQUENCE IS:			*
	6146	*			B BNFDEF			*
	6147	*						*
	6148	*			*INPUT			*
	6149	*			* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
	6150	*			THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
	6151	*			LEADING KEYWORD, DEF.			*
	6152	*			* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
	6153	*			CHARACTER IN THE LEADING KEYWORD, DEF.			*
	6154	*			* FUNCTION ATTRIBUTE FIELDS - THE CORE-RESIDENT VIRTUAL ADDRESS			*
	6155	*			STORAGE LOCATIONS FOR EACH OF THE 29 POSSIBLE USER FUNCTIONS.			*
	6156	*			ATTRIBUTE FIELDS FOR PREVIOUSLY DEFINED USER FUNCTIONS CONTAIN			*
	6157	*			THE ENTRY POINT VIRTUAL ADDRESS ASSOCIATED WITH EACH FUNCTION.			*
	6158	*			UNDEFINED ATTRIBUTE FIELDS ARE CLEARED TO ZERO.			*
	6159	*						*
	6160	*			*OUTPUT			*
	6161	*			* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
	6162	*			GENERATED BY BNFDEF IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
	6163	*			MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
	6164	*			SEQUENCES.			*
	6165	*			* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
	6166	*			CHARACTER WHICH TERMINATES THE STATEMENT.			*
	6167	*			* FUNCTION ATTRIBUTE FIELDS - UPDATED WITH ENTRY POINT VIRTUAL			*
	6168	*			ADDRESS ASSOCIATED WITH THE USER FUNCTION DEFINED BY THE			*
	6169	*			CURRENT STATEMENT.			*
	6170	*			B\$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
	6171	*			ADDRESS OPERAND FIELD IN THE BYPASS BRANCH INSTRUCTION			*
	6172	*			GENERATED FOR THE CURRENT STATEMENT.			*
	6173	*			* B\$NXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE BYPASS			*
	6174	*			BRANCH INSTRUCTION OPERAND ADDRESS.			*
	6175	*						*
	6176	*			*EXTERNAL REFERENCES			*
	6177	*			B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIE,AL ROUTINE.			*
	6178	*			B\$PUTC - (B\$PFNC, B\$PCAD, B\$PNBY, B\$PCOL, B@PERC, B\$PVAD) -			*
	6179	*			ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.			*
	6180	*			B\$SYMB - (B\$BCKT, B\$FSVA, B\$FSSW, B\$FACA, B\$FSC1, B\$FSC2) -			*
	6181	*			ENTRY TO BASIC SYMBOL TRANSLATION ROUTINE.			*
	6182	*			B\$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.			*
	6183	*			B\$BTAB - (B\$BRVA) - ENTRY TO BASIC COMPILER BRANCH TABLE			*
	6184	*			ROUTINE.			*

## S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 82
		6185	*	\$XIND1	- INDICATOR FOR LONG CO SHOW PRECISION.	*
		6186	*	B\$DIST	- ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		6187	*			*
		6188	*	EXITS, NORMAL		*
		6189	*	B\$DIST	- ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		6190	*			*
		6191	*	EXITS, ERROR		*
		6192	*	N/A		*
		6193	*			*
		6194	*	TABLES/WORK AREAS		*
		6195	*	* FUNCTION ATTRIBUTE FIELDS	- EXTERNAL TO 1NFDEF, THESE FIELDS	*
		6196	*	CONTAIN VIRTUAL ADDRESSES	FOR THE 29 POSSIBLE USER FUNCTION	*
		6197	*	ENTRY POINTS	AS THEY ARE DEFINED IN A PROGRAM.	*
		6198	*			*
		6199	*	ATTRIBUTES		*
		6200	*	BNFDEF	IS NATURALLY RELOCATABLE AND REUSABLE.	*
		6201	*			*
		6202	*	CHARACTER CODE DEPENDENCY		*
		6203	*	THE OPERATION OF THIS MODULE	DOES NOT DEPEND UPON A PARTICULAR	*
		6204	*	INTERNAL REPRESENTATION	OF THE EXTERNAL CHARACTER SET.	*
		6205	*			*
		6206	*	NOTES		*
		6207	*	ERROR PROCEDURES		*
		6208	*	WHEN A DEF STATEMENT ATTEMPTS	TO DEFINE A USER FUNCTION WHICH	*
		6209	*	HAS BEEN PREVIOUSLY DEFINED	IN THE SAME PROGRAM, THE ERROR	*
		6210	*	CONDITION CODE FOR 'DUPLICATE	DEFINITION OF USER FUNCTION' IS	*
		6211	*	LOGGED IN VIRTUAL MEMORY	USING OUTPUT ROUTINE B@PUTC.	*
		6212	*	COMPILATION IS OTHERWISE	UNAFFECTED.	*
		6213	*			*
		6214	*	REGISTER USAGE		*
		6215	*	BOTH THE INDEX AND BASE	REGISTERS ARE USED DURING EXECUTION.	*
		6216	*			*
		6217	*	SAVED/RESTORED AREAS		*
		6218	*	N/A		*
		6219	*			*
		6220	*	MODIFICATION CONSIDERATIONS		*
		6221	*	BNFDEF RESIDES ON ONE SECTOR.	ANY MODIFICATION SHOULD CONSIDER	*
		6222	*	THE SIZE LIMITATION.		*
		6223	*			*
		6224	*	REQUIRED MODULES		*
		6225	*	@SYSEQ	- COMMON SYSTEM EQUATES.	*
		6226	*	@FXDEQ	- SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
		6227	*	@VMDEQ	- VIRTUAL MEMORY DIRECTORY EQUATES.	*
		6228	*	@SPFEQ	- SYSTEM PROGRAM FILE EQUATES.	*
		6229	*	@ERMEQ	- ERROR MESSAGE EQUATES.	*
		6230	*	\$V\$EQU	- FIXED VIRTUAL ADDRESS EQUATES.	*
		6231	*	\$B\$EQU	- COMPILER FIXED EQUATES.	*
		6232	*	\$B@EQU	- COMPILER SYSTEM EQUATES.	*
		6233	*			*
		6234	*	OTHER		*
		6235	*	BNFDEF	IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS	*
		6236	*	*****		*
1500		6238		ORG	*,256,0	PLACE MODULE AT PAGE BOUNDARY
	1500	6239		USING	*,@BR	ESTABLISH BASE ADDRESSING
		6240	*			

## S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 83
					6241	*	ENTER BNFDEF - 'DEF' STATEMENT ROUTINE			
					6242	*				
				1500	6243	BNFDEF EQU	*			
					6244	*				
					6245	*	SET INPUT PARAMETER TO SKIP KEYWORD 'DEF'			
					6246	*				
1500	3C	03	0873		6247	BNF010 MVI	B\$NUMC,B@LDEF SET GET RTN TO SKIP 'DEF'			
1504	C0	87	0867		6248	B	B\$GETC LINK TO ADVANCE POINTER			
					6249	*				
					6250	*	GENERATE A BYPASS BRANCH INSTRUCTION IMAGE			
					6251	*				
1508	D2	02	BC		6252	BNF020 LA	BNFBRC(,@BR),@XR LOAD CADDR OF 'BRA' INSTR			
150B	34	02	0A40		6253	ST	B\$PCAD,@XR SET PUT RTN VADDR FOR 'BRA'			
150F	C0	87	093A		6254	B	B\$PUTC LINK TO GENERATE 'BRA' PMC			
					6255	*				
					6256	*	SAVE NEXT AVAILABLE PMC VADDR FOR BRANCH RESOLUTIONS AND			
					6257	*	FUNCTION TABLE ENTRY			
					6258	*				
1513	0C	01	19EF 0A43		6259	BNF030 MVC	B\$BRVA,B\$PVAD(@VADDR) SAVE 'BRA' VADDR FOR RESOLUTION			
					6260	*				
					6261	*	CALL SYMBOL ROUTINE TO DETERMINE THE VIRTUAL ADDRESS OF THE FUNCTION			
					6262	*	TABLE LOCATION ASSOCIATED WITH THE CURRENT USER FUNCTION			
					6263	*				
1519	35	02	0878		6264	BNF040 L	B\$GPTR,@XR RESTORE TEXT POINTER			
151D	C0	87	0DBC		6265	B	B\$SYMB LINK TO GET CADDR OF USER FUNC			
					6266	*				
					6267	*	CHECK CADDR OF USER FUNC FOR INDICATION OF PREVIOUS DEFINITION			
					6268	*				
1521	35	02	0E93		6269	BNF050 L	B\$FACA,@XR LOAD CADDR OF USER FUNCTION			
1525	BD	56	00		6270	CLI	B@FVAD-1(,@XR),B@DVC1 IF FUNCTION NOT DEFINED			
1528	F2	82	0C		6271	JL	BNF070 * JUMP TO PROCESS USER FUNCTION			
					6272	*				
					6273	*	GENERATE ERROR MESSAGE IF FUNCTION HAS BEEN PREVIOUSLY DEFINED			
					6274	*				
152B	3C	33	094E		6275	BNF060 MVI	B\$PFNC,B\$PFAE SET PUT RTN FOR ERROR OUTPUT			
152F	3C	AA	0A39		6276	MVI	B\$PERC,@E604 SET PUT RTN FOR 'INVALID FUNC'			
1533	C0	87	093A		6277	B	B\$PUTC LINK TO GENERATE ERROR PMC			
					6278	*				
					6279	*	TEST FOR PRECISION BEFORE GENERATING FUNCTION LINKAGE SEQUENCE			
					6280	*				
1537	38	40	03D0		6281	BNF070 TBN	\$XIND1,\$XPREC IF PRECISION IS STANDARD			
153B	F2	90	06		6282	JF	BNF080 * SKIP TO GENERATE LINKAGE SEQ			
153E	7C	0D	CA		6283	MVI	BNFSPA(,@BR),BNFLIP SET LENGTH FOR LONG PREC			
1541	7C	09	C0		6284	MVI	BNFDAN(,@BR),B@LILP SET 'DWA' OPERAND FOR LONG PREC			
					6285	*				
					6286	*	GENERATE RETURN LINKAGE 'BRA' INSTR AND PARAMETER AREA			
					6287	*				
1544	1C	00	0A41 CA		6288	BNF080 MVC	B\$PNBY,BNFSPA(1,@BR) SET PUT RTN LNG FOR 'BRA' RET			
1549	C0	87	093A		6289	B	B\$PUTC LINK TO GENERATE RET LINK SEQ			
154D	4C	01	CD 0A43		6290	MVC	BNFBDO(,@BR),B\$PVAD(@VADDR) MOVE VIRTUAL ADDR OF LINKAGE			
1552	4F	00	CD 09D3		6291	SLC	BNFBDO(,@BR),B\$PCDL(@VADDR-1) * BRA INST TO 'BRD' OPERAND			
					6292	*				
					6293	*	ESTABLISH THE VADDR OF THE 'BRA' RETURN LINKAGE PMC AS THE FUNCTION			
					6294	*	TABLE ENTRY FOR THE USER FUNCTION CURRENTLY REFERENCED			
					6295	*				
1557	35	02	0E93		6296	BNF090 L	B\$FACA,@XR MOVE CADDR OF FUNC TBL ENTRY			



## S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 84

155B	9C 01 01 CD	6297	MVC	B@FVAD(, @XR), BNFBD0(@VADDR, @BR)	MOVE VADDR OF 'BRA' INSTR
		6298	*		
		6299	*	ADVANCE TEXT POINTER TO REFERENCE 1ST CHAR OF THE FUNC DUMMY ARG	
		6300	*		
155F	C0 87 0867	6301	BNF100 B	B\$GETC	LINK TO GET NEXT CHARACTER
		6302	*		
		6303	*	MOVE THE FIRST CHARACTER OF THE DUMMY ARG NAME INTO THE DUMMY SYMBOL	
		6304	*	WORD OF THE SYMBOL ROUTINE	
		6305	*		
1563	2C 00 0E4C 00	6306	BNF110 MVC	B\$FSC1, B@CHAR(1, @XR)	MOVE 1ST USER FUNC ARC CHAR
1568	C0 87 0867	6307	B	B\$GETC	LINK TO GET NEXT CHAR
		6308	*		
		6309	*	TEST FOR A SECOND USER FUNCTION CHARACTER	
		6310	*		
156C	BD 5D 00	6311	BNF120 CLI	B@CHAR(, @XR), B@RPAR	IF NO 2ND USER FUNC ARG CHAR
156F	F2 81 10	6312	JE	BNF140	* JUMP TO 'BLANK. SYMBOL WORD
		6313	*		
		6314	*	MOVE 2ND CHAR OF DUMMY IN NAME INTO DUMMY SYMBOL WORD OF SYMBOL RTN	
		6315	*		
1572	2C 00 0E4D 00	6316	BNF130 MVC	B\$FSC2, B@CHAR(1, @XR)	MOVE 2ND USER FUNC ARG CHAR
1577	3C 02 0873	6317	MVI	B\$NUMC, BNFSKP	SET GET RTN TO SKIP 10
157B	C0 87 0867	6318	B	B\$GETC	LINK TO ADVANCE TEXT POINTER
157F	F2 87 08	6319	J	BNF150	JUMP TO SET OTHER TM RTN PARM
		6320	*		
		6321	*	MOVE A BLANK AS 2ND CHAR OF USER FUNC DUMMY ARC NAME INTO THE DUMMY	
		6322	*	SYMBOL WORD OF THE SYMBOL ROUTINE	
		6323	*		
1582	3C 40 0E4D	6324	BNF140 MVI	B\$FSC2, B@BLNK	MOVE A BLNK INTO DUMMY SYM ND
1586	C0 87 0867	6325	B	B\$GETC	LINK TO GET NEXT CHARACTER
		6326	*		
		6327	*	MOVE THE VADDR OF THE 'BRA' RETURN LINKAGE PARAMATER AREA	
		6328	*	INTO THE SYMBOL ROUTINE INPUT PARAMETER	
		6329	*		
158A	1C 01 0E4F CD	6330	BNF150 MVC	B\$FSVA, BNFBD0(@VADDR, @BR)	MOVE THE VADDR OF LAST 'BRA'
158F	1E 00 0E4F CE	6331	ALC	B\$FSVA, BNFLTH(@VADDR-1, @BR)	ADJUST TO VADDR OF WORK AREA
		6332	*		
		6333	*	SET THE FUNCTION SCAN SWITCH ON TO INDICATE THE VARIABLE IS A USER	
		6334	*	FUNCTION DUMMY ARGUMENT NAME	
		6335	*		
1594	3A 07 0E5C	6336	BNF160 SBN	B\$FSSW, B\$FSMK	SET FUNCTION SCAN SWITCH
		6337	*		
		6338	*	CALL THE ARITH SCAN RTN TO GENERATE THE PMC'S FOR THE ARITH EXPR	
		6339	*		
1598	C0 87 1514	6340	BNF170 B	B\$SCAN	LINK TO PROCESS ARUM EXPR
159C	3B 07 0E5C	6341	SBF	B\$FSSW, B\$FSMK	SET FUNC SCAN SI41V-4 OFF
		6342	*		
		6343	*	GENERATE A 'BRD' INSTRUCTION TO COMPLETE THE TRANSFER OF CONTROL TO	
		6344	*	THE CALLING EXPRESSION	
		6345	*		
15A0	D2 02 CB	6346	BNF180 LA	BNFBD0(, @BR), @XR	LOAD CADDR OF 'ORD' INSIR
15A3	34 02 0A40	6347	ST	B\$PCAD, @XR	SET PUT RTN VADDR FOR 'BRD'
15A7	3C 02 0A41	6348	MVI	B\$PNBY, B@LBRD-1	SET LENGTH OF 'BRD'.
15AB	C0 87 093A	6349	B	B\$PUTC	LINK TO GENERATE 'BRD' PMC
		6350	*		
		6351	*	STORE THE VADDR OF THE FIRST 'BRA' INSTR OPERAND FOE ADDRESS	
		6352	*	RESOLUTION IN THE BRANCH ADDRESS TABLE	



S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE	85
			6353	*					
N04	15AF 00 00 0000 00		6354	BNF190 SLC	B\$BRVA,BNFBN1(@VADDR-1,@BR)	ADJUST	VADDR	TO	'BRA' OPRND
			6355	*					
			6356	*	SET 'NEXT' SWITCH TO CAUSE BRANCH TABLE LINE NUMBER RESOLUTION				
			6357	*					
15B4	3A 07 071D		6358	BNF200 SBN	B\$NXSW,B\$NXMK	SET	'NEXT'	SWITCH	ON
			6359	*					
			6360	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR				
			6361	*					
15B8	C0 87 0700		6362	BNF210 B	B\$DIST	RETURN	TO	DISTRIBUTOR	

## S/3 BASIC COMPILER -DEF- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 86
				6364	*****		
				6365	* 'DEF' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS		
				6366	*****		
				6367	*		
15BC	46		15BC	6368	BNFBRC DC	AL(B@LCOP)(B@CBRA)	'BRA' IMAGE OPCODE
15BD	0000		15BE	6369	BNFBRO DC	XL(B@LCVA)'00'	'BRA' IMAGE OPERAND
				6370	*		
N04 15BF	00		15BF	6371	BNFDAC DC	AL(B@LCOP)(B\$CDWA)	'DNA' INSTRUCTION OPCODE
15C0			15C0	6372	BNFDAN DS	CL(B@LCNN)	'DNA' INSTRUCTION OPERAND
15C0				6373	ORG	BNFDAN	INITIALIZE 'DMA' OPERAND FOR
15C0	05		15C0	6374	DC	AL(B@LCNN)(B@LISP)	* STANDARD PREC PACKED FLT PT
				6375	*		
15C1	0000000000000000		15C9	6376	BNFWKA DC	XL(B@LILP)'00'	USER FUNCTION ARGUMENT AREA
				6377	*		
15CA			15CA	6378	BNFSPA DS	CL1	'BRA' & ARG FIELD LENGTH - 1
15CA				6379	ORG	BNFSPA	LENGTH SET FOR SHORT PRECISION
15CA	09		15CA	6380	DC	AL1(B@LBRA+B@LDWA+B@LISP-1)	* CHANGE FOR LONG PRECISION
				6381	*		
15CB	48		15CB	6382	BNFBDC DC	AL(B@LCOP)(B@CBRD)	'BRD' INSTR OPCODE
15CC			15CD	6383	BNFBDO DS	CL(B@LCVA)	'BRD' INSTR OPERAND
				6385	*****		
				6386	* 'DEF' STATEMENT ROUTINE CONSTANTS AND EQUATES		
				6387	*****		
				6388	*		
				6389	* CONSTANTS		
				6390	*		
15CE	05		15CE	6391	BNFLTH DC	AL1(B@LBRA+B@LDWA)	LENGTH OF 'BRA' A 'DWA' PMC'S
15CF	01		15CF	6392	BNFBNI DC	IL(@VADDR-1)'1'	BINARY INTEGER +1
				6393	*		
				6394	* EQUATES		
				6395	*		
			0002	6396	BNFSKP EQU 2		LENGTH OF TWO CHARACTERS
				6397	*		
			000D	6398	BNFLIP EQU	B@LBRA+B@LDWA+B@LILP-1	LENGTH FOR LONG INTERNAL PREC
				6399	*		
				6400	*****		
				6401	*		
				6402	* END OF 'DEF' STATEMENT ROUTINE CODING		
				6403	*		

ERR LOC	OBJECT CODE	ADDR STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 87
6405			*****			*
6406	*		5703-XM1 COPYRIGHT IBM CORP. 1970			*
6407	*		REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
6408	*					*
6409			*****			*
6410			*STATUS			*
6411	*		VERSION 1 MODIFICATION 0			*
6412	*					*
6413			*FUNCTION			*
6414	*		BPMLET IS EXECUTED TO TRANSLATE MULTIPLE ARITHMETIC ASSIGNMENT			*
6415	*		AND LET STATEMENTS AS THEY OCCUR IN A BASIC PROGRAM INTO THE			*
6416	*		APPROPRIATE PSEUDOCODE AND TO PLACE THE PSEUDOCODE INTO VIRTUAL			*
6417	*		MEMORY.			*
6418	*					*
6419			*ENTRY POINTS			*
6420	*		BPMLET HAS TWO ENTRY POINTS:			*
6421	*		BPMASN - TRANSLATE MULTIPLE ARITHMETIC ASSIGNMENT STATEMENT			*
6422	*		BPMLET - TRANSLATE MULTIPLE ARITHMETIC LET STATEMENT			*
6423	*		THE FORMAT OF THE CALLING SEQUENCES IS AS FOLLOWS:			*
6424	*		B BPMASN			*
6425	*		B BPMLET			*
6426	*					*
6427			*INPUT			*
6428	*		* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
6429	*		THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
6430	*		LEADING KEYWORD, LET, OR IN THE ASSIGNMENT LIST IF THE			*
6431	*		OPTIONAL KEYWORD IS OMITTED.			*
6432	*		* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST			*
6433	*		CHARACTER IN THE LEADING KEYWORD, LET, OR IN THE ASSIGNMENT			*
6434	*		LIST IF THE OPTIONAL KEYWORD IS OMITTED.			*
6435	*					*
6436			*OUTPUT			*
6437	*		* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
6438	*		GENERATED BY BPMLET IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
6439	*		MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
6440	*		SEQUENCES. GENERATED PROGRAM CONSTANTS WILL BE STORED UNDER			*
6441	*		CONTROL OF THE COMPILER CONSTANT ROUTINE BCFCON.			*
6442	*		* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
6443	*		CHARACTER WHICH TERMINATES THE STATEMENT.			*
6444	*					*
6445			*EXTERNAL REFERENCES			*
6446	*		B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC TEXT RETRIEVAL			*
6447	*		ROUTINE.			*
6448	*		B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER			*
6449	*		VIRTUAL MEMORY OUTPUT ROUTINE.			*
6450	*		B\$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.			*
6451	*		B\$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.			*
6452	*		B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH			*
6453	*		TABLE ROUTINE.			*
6454	*		B\$DIST - (B\$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
6455	*		B\$WORK - ENTRY TO WORK AREA IN COMMON AREA OF CORE.			*
6456	*					*
6457			*EXITS, NORMAL			*
6458	*		B\$DIST - (B\$NXSW) - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
6459	*					*
6460			*EXITS, ERROR			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 88
			6461	* N/A	*
			6462	*	*
			6463	*TABLES/WORK AREAS	*
			6464	* * WORK AREA &WRK, WHOSE ADDRESS IS REFERENCED BY B\$WORK, IS	*
			6465	* USED FOR THE RUN-TIME STACKING AND UNSTACKING OF THE VALUE OF	*
			6466	* THE ARITHMETIC EXPRESSION ON THE RIGHT SIDE OF THE EQUAL SIGN.	*
			6467	*	*
			6468	*ATTRIBUTES	*
			6469	* BPMLET IS NATURALLY RELOCATABLE AND REUSABLE	*
			6470	*	*
			6471	*CHARACTER CODE DEPENDENCY	*
			6472	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
			6473	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
			6474	*	*
			6475	*NOTES	*
			6476	* ERROR PROCEDURES	*
			6477	* N/A	*
			6478	*	*
			6479	* REGISTER USAGE	*
			6480	* BOTH THE INDEX AND BASE REGISTERS ARE USED IN THE EXECUTION	*
			6481	* OF BPMLET.	*
			6482	*	*
			6483	* SAVED/RESTORED AREAS	*
			6484	* N/A	*
			6485	*	*
			6486	* MODIFICATION CONSIDERATIONS	*
			6487	* BPMLET IS CO-RESIDENT ON A SECTOR WITH BMINPT.	*
			6488	* ANY MODIFICATION TO BPMLET WILL CHANGE THE ENTRY ADDRESS	*
			6489	* OF BMINPT AND MUST CONSIDER THE LIMITATION OF THE SECTOR	*
			6490	* BOUNDARY ON SIZE.	*
			6491	*	*
			6492	* REQUIRED MODULES	*
			6493	* @SYSEQ - COMMON SYSTEM EQUATES	*
			6494	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES	*
			6495	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES	*
			6496	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES	*
			6497	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES	*
			6498	* @ERMEQ - ERROR MESSAGE EQUATES	*
			6499	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES	*
			6500	* \$B\$EQU - COMPILER FIXED EQUATES	*
			6501	* \$B@EQU - COMPILER SYSTEM EQUATES	*
			6502	*	*
			6503	* OTHER	*
			6504	* BPMLET IS ASSEMBLED WITH ALL OTHER STATEMENT PROCESSORS.	*
			6505	*****	*
1600			6507	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
		1600	6508	USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
			6509	*	
			6510	* ENTER BPMLET - MULTIPLE ARITHMETIC 'LET' STATEMENT PROCESSOR	
			6511	*	
		1600	6512	BPMLET EQU *	BPMLET ENTRY POINT
			6513	*	
			6514	* SKIP PAST 'LET' TO 1ST LIST ELEMENT SYMBOL CHARACTER	
			6515	*	
N04 1600 00 00 0000			6516	BPM010 MVI ISNUHC,BILLET	SET GET ROUTINE TO SKIP 'LET'

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 89

1604	C0	87	0867	6517	B	B\$GETC	LINK TO GET 1ST SYMBOL CHAR
				6518	*		
				6519	*	ENTER BPMASN - MULTIPLE ARITHMETIC ASSIGNMENT STATEMENT PROCESSOR	
				6520	*		
			1608	6521	BPMASN EQU *		BPMASN ENTRY POINT
				6522	*		
				6523	*	GENERATE A BRANCH INSTRUCTION IMAGE - THIS INSTRUCTION IS REQUIRED	
				6524	*	TO TRANSFER CONTROL PAST THE ASSIGNMENT ADDRESS STACKING SEQUENCE	
				6525	*	TO THE SEQUENCE WHICH ESTABLISHES THE SOURCE FLOATING POINT VALUE	
				6526	*		
1608	D2	02	C5	6527	BPM020 LA	BPMBIC(,@BR),@XR	LOAD CADDR OF 'BRA' INSTR
160B	34	02	0A40	6528	ST	B\$PCAD,@XR	SET VADDR PARM FOR PUT RTN
160F	3C	02	0A41	6529	MVI	B\$PNBY,B@LBRA-1	SET LENGTH PARM FOR PUT RTN
1613	C0	87	093A	6530	B	B\$PUTC	LINK TO OUTPUT THE IMAGE
				6531	*		
				6532	*	STORE NEXT AVAILABLE PMC VIRTUAL ADDRESS (ADDRESS OF 1ST INSTRUCTION	
				6533	*	IN THE ADDRESS STACKING SEQUENCE) AS OPERAND IN A 'RETURN BRANCH'	
				6534	*	PSEUDO INSTRUCTION	
				6535	*		
N04	1617	00	00 00 0000	6536	BPM030 MVC	BPMBRO(,@BR),B\$PVAD(@VADDR)	SET 'RETURN BRANCH' OPERAND
				6537	*		
				6538	*	ESTABLISH &WRK AS OPERAND OF A 'STACK FLOATING VALUE' INSTRUCTION	
				6539	*		
N04	161C	00	00 00 0000	6540	BPM040 MVC	BPMSF0(,@BR),B@WORK(@VADDR)	SET 'STF' OPERAND &WRK
				6541	*		
				6542	*	GENERATE ADDRESS STACKING INSTRUCTIONS FOR AN ASSIGNMENT LIST ELEMENT	
				6543	*		
1621	35	02	0878	6544	BPM045 L	B\$GPTR,@XR	RESTORE TEXT POINTER
1625	C0	87	1853	6545	BPM050 B	B\$LIST	LINK TO PROCESS LIST ELEMENT
1629	6C	00	4C 00	6546	MVC	BPM070+@Q(,@BR),B@CHAR(1,@XR)	SAVE CADDR OF NEXT CHAR
				6547	*		
				6548	*	GENERATE PSEUDO INSTRUCTIONS TO STACK THE SOURCE VALUE AND UNSTACK	
				6549	*	IT TO THE ASSIGNMENT LIST ELEMENT ADDRESS	
				6550	*		
N04	162D	00	00 00	6551	BPM060 LA	BPMSFC(,@BR),@XR	LOAD CADDR OF 'STF' INSTR
1630	34	02	0A40	6552	ST	B\$PCAD,@XR	SET VADDR PARM FOR PUT RTN
1634	3C	02	0A41	6553	MVI	B\$PNBY,B@LSTF-1	SET LENGTH PARM FOR PUT RTN
1638	C0	87	093A	6554	B	B\$PUTC	LINK TO OUTPUT 'STF URIC'
163C	D2	02	D0	6555	LA	BPMUFC(,@BR),@XR	LOAD CADDR OF 'UV' INSTR
163F	34	02	0A40	6556	ST	B\$PCAD,@XR	SET VADDR PARM FOR PUT RTN
1643	3C	00	0A41	6557	MVI	B\$PNBY,B@LUSF-1	SET LENGTH PARM FOR PUT RTN
1647	C0	87	093A	6558	B	B\$PUTC	LINK TO OUTPUT 'USF' INST
				6559	*		
				6560	*	TEST FOR END OF THE MULTIPLE ASSIGNMENT LIST	
				6561	*		
164B	7D	00	D1	6562	BPM070 CLI	BPMIND(,@BR),*-*	IF LIST DELIMITER IS
164E	F2	81	07	6563	JE	BPM090	* EXIT LIST PROCESSING LOOP
				6564	*		
				6565	*	ADVANCE TEXT POINTER PAST LIST DELIMITER AND BRANCH TO PROCESS	
				6566	*	NEXT ELEMENT IN THE ASSIGNMENT LIST	
				6567	*		
1651	C0	87	0867	6568	BPM080 B	B\$GETC	LINK TO GET NEXT CHARACTER
1655	D0	87	25	6569	B	BPM050(,@BR)	GO PROCESS NEXT LIST ELEMENT
				6570	*		
				6571	*	GENERATE A BRANCH INSTRUCTION IMAGE - THIS INSTRUCTION IS REQUIRED	
				6572	*	TO TRANSFER CONTROL PAST THE SEQUENCE WHICH ESTABLISHES THE SOURCE	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 90
					6573	*	VALUE TO THE STATEMENT FOLLOWING THAT WHICH IS BEING PROCESSED	
					6574	*		
	1658	D2	02	C5	6575	BPM090	LA BPMBIC(, @BR), @XR	LOAD CADDR OF 'BRA' INSTR
	165B	34	02	0A40	6576		ST B\$PCAD, @XR	SET VADDR PARM FOR PUT RTN
	165F	3C	02	0A41	6577		MVI B\$PNBY, B@LBRA-1	SET LENGTH PARM FOR PUT RTN
	1663	C0	87	093A	6578		B B\$PUTC	LINK TO OUTPUT THE 'BRA' IMAGE
					6579	*		
					6580	*	ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND IN THE FIRST	
					6581	*	BRANCH INSTRUCTION IMAGE GENERATED ABOVE (BPM020)	
					6582	*		
N04	1667	00	00	0000 00	6583	BPM100	MVC B\$BRVA, BPMBRO(@VADDR, @BR)	SET BRANCH TABLE VADDR PARM
N04	166C	00	00	0000 00	6584		SLC B\$BRVA, BPMBN1(@VADDR, @BR)	* FOR THE 'BRA' IMAGE OPERAND
	1671	0C	01	19F1 0A43	6585		MVC B\$BRLN, B\$PVAD(@VADDR)	SET BRANCH TABLE LINE NO. PARM
					6586	*		* FOR BRANCH POINT VADDR
	1677	C0	87	1996	6587		B B\$BTAB	LINK TO SET UP RESOLUTION
					6588	*		
					6589	*	GENERATE INSTRUCTION TO STACK ADDRESS OF &WRK - THE FIRST BRANCH	
					6590	*	INSTRUCTION (BPM020) PASSES RUN-TIME CONTROL TO THIS INSTRUCTION	
					6591	*		
	167B	5C	01	CD CF	6592	BPM110	MVC BPMSAO(, @BR), BPMSFO(@VADDR, @BR)	SET 'STA' OPERAND &WRK
	167F	D2	02	CB	6593		LA BPMSAC(, @BR), @XR	LOAD CADDR OF 'STA' INSTR
	1682	34	02	0A40	6594		ST B\$PCAD, @XR	SET VADDR PARM FOR PUT RIP
N04	1686	00	00	0000	6595		MVI B\$PNBY, B\$LSTA-1	SET LENGTH PARM FOR PUT RTN
	168A	C0	87	093A	6596		B B\$PUTC	LINK TO OUTPUT 'STA MARK'
					6597	*		
					6598	*	GENERATE PSEUDO INSTRUCTIONS TO PROCESS THE STATEMENT EXPRESSION	
					6599	*	AND UNSTACK THE RESULTING VALUE INTO &WRK	
					6600	*		
	168E	C0	87	1514	6601	BPM120	B B\$SCAN	LINK TO GENERATE EXPRESSION PMC
	1692	D2	02	D0	6602		LA BPMUFC(, @BR), @XR	LOAD CADDR OF 'USF' INSTR
	1695	34	02	0A40	6603		ST B\$PCAD, @XR	SET VADDR PARM FOR PUT RTN
	1699	3C	00	0A41	6604		MVI B\$PNBY, B@LUSF-1	SET LENGTH PARM FUR PUT RTN
	169D	C0	87	093A	6605		B B\$PUTC	LINK TO OUTPUT 'USF' INST
					6606	*		
					6607	*	GENERATE THE RETURN BRANCH INSTRUCTION - THIS TRANSFERS CONTROL	
					6608	*	TO THE LIST ASSIGNMENT SEQUENCE AFIER THE SOURCE VALUE HAS BEEN	
					6609	*	STORED IN INTERNAL VARIABLE MIRK	
					6610	*		
	16A1	D2	02	C8	6611	BPM130	LA BPMBRC(, @BR), @XR	LOAD CADDR OF 'BRA' INSTR
	16A4	34	02	0A40	6612		ST B\$PCAD, @XR	SET VADDR PARM FOR PUT RTN
	16A8	3C	02	0A41	6613		MVI B\$PNBY, B@LBRA-1	SET LENGTH PARM FOR PUT RTN
	16AC	C0	87	093A	6614		B B\$PUTC	LINK TO OUTPUT RETURN 'BRA'
					6615	*		
					6616	*	ESTABLISH CONDITIONS TO RESOLVE THE ADDRESS OPERAND IN THE SECOND	
					6617	*	BRANCH INSTRUCTION IMAGE GENERATED ABOVE (BPM090)	
					6618	*		
	16B0	0C	01	19EF 19F1	6619	BPM140	MVC B\$BRVA, B\$BRLN(@VADDR)	SET BRANCH TABLE VADDR PARM
N04	16B6	00	00	0000 00	6620		SLC B\$BRVA, BPMBN1(@VADDR, @BR)	* FOR THE 'BRA' IMAGE OPERAND
N04	16BB	00	00	0000	6621		SBN B\$NXSN, B\$NXMK	SET 'NEXT STMT' SNITCH ON TO
					6622	*		* ESTABLISH LINE NO. PARM
					6623	*		
					6624	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
					6625	*		
	16BF	C0	87	0700	6626	BPM150	B B\$DIST	BRANCH TO DISTRIBUTOR

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 91
				6628	*****	
				6629	* MULTIPLE ARITHMETIC 'LET' ROUTINE CONSTANTS	
				6630	*****	
				6631	*	
16C3	0001		16C4	6632	BPMINI DC IL(@VADDR)'1' BINARY INTEGER 1	
				6634	*****	
				6635	* MULTIPLE ARITHMETIC 'LET' ROUTINE PMC AND STORAGE PARAMETERS	
				6636	*****	
				6637	*	
16C5	46		16C5	6638	BPMBIC DC AL(B@LCOP)(B@CBRA) BRANCH IMAGE 'BRA' OPCODE	
16C6	0000		16C7	6639	BPMBIO DC XL(B@LCVA)'00' BRANCH IMAGE NULL OPERAND	
				6640	*	
16C8	46		16C8	6641	BPMBRC DC AL(B@LCOP)(B@CBRA) RETURN BRANCH 'BRA' OPCODE	
16C9			16CA	6642	IPMBRO DS CL(B@LCVA) RETURN BRANCH OPERAND AREA	
				6643	*	
16CB	34		16CB	6644	BPMSAC DC AL(B@LCOP)(B@CSTA) STACK ADDRESS 'STA' OPCODE	
16CC			16CD	6645	BPMSAO DS CL(B@LCVA) STACK ADDRESS OPERAND AREA	
				6646	*	
N04				6647	BPMSFC DC AL(B@LCOP)(B@CSTF) STACK FLOATING 'STF' OPCODE	
16CE			16CF	6648	BPMSFO DS CL(B@LCVA) STACK FLOATING OPERAND AREA	
				6649	*	
16D0	26		16D0	6650	BPMUFC DC AL(B@LCOP)(B@CUSF) UNSTACK FLOATING 'USF' OPCODE	
				6651	*	
N04	16D1	00	16D1	6652	BPMIND DC AL1(BWOUL) DELIMITER COMPARE - '='	
				6653	*****	
				6654	*	
				6655	* END OF MULTIPLE ARITHMETIC 'LET' ROUTINE CODING	
				6656	*	



ERR LOC	OBJECT CODE	ADDR STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 92
		6658	*****			*
		6659	* 5703-XM1 COPYRIGHT IBM CORP. 1970			*
		6660	* REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		6661	*			*
		6662	*****			*
		6663	*STATUS			*
		6664	* VERSION 1 MODIFICATION 0			*
		6665	*			*
		6666	*FUNCTION			*
		6667	* BMINPT IS EXECUTED TO TRANSLATE MAT INPUT STATEMENTS AS THEY OCCUR*			*
		6668	* IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		6669	* THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		6670	*			*
		6671	*ENTRY POINTS			*
		6672	* BMINPT HAS ONLY ONE ENTRY POINT:			*
		6673	* BMINPT - TRANSLATE MAT INPUT STATEMENT			*
		6674	* THE FORMAT OF THE CALLING SEQUENCE IS:			*
		6675	* B BMINPT			*
		6676	*			*
		6677	*INPUT			*
		6678	* * COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		6679	* THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		6680	* LEADING KEYWORD, MAT INPUT.			*
		6681	* * TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		6682	* CHARACTER IN THE LEADING KEYWORD, MAT INPUT.			*
		6683	*			*
		6684	*OUTPUT			*
		6685	* * VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUE4CE			*
		6686	* GENERATED BY BMINPT IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		6687	* SEQUENCES.			*
		6688	* * TEXT CHARACTER POINTER - CONTAINS THE ARE ADDRESS OF THE			*
		6689	* CHARACTER WHICH TERMINATES THE STATEMENT.			*
		6690	*			*
		6691	*EXTERNAL REFERENCES			*
		6692	* B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		6693	* B\$PUTC - (B\$PCAI, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
		6694	* ROUTINE.			*
		6695	* B\$MATR - ENTRY TO BASIC MATRIX REFERENCE ROUTINE.			*
		6696	* B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		6697	*			*
		6698	*EXITS, NORMAL			*
		6699	* B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		6700	*			*
		6701	*EXITS, ERROR			*
		6702	* N/A			*
		6703	*			*
		6704	*TABLES/WORK AREAS			*
		6705	* N/A			*
		6706	*			*
		6707	*ATTRIBUTES			*
		6708	* BMINPT IS NATURALLY RELOCATABLE AND REUSABLE.			*
		6709	*			*
		6710	*CHARACTER CODE DEPENDENCY			*
		6711	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR			*
		6712	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
		6713	*			*

## S/3 BASIC COMPILER -MATH INPUT- STATEMENT RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 93
					6714	*NOTES				*
					6715	* ERROR PROCEDURES				*
					6716	* N/A				*
					6717	*				*
					6718	* REGISTER USAGE				*
					6719	* BOTH THE INDEX AND BASE REGISTERS ARE USED DUHNG EXECUTION.				*
					6720	*				*
					6721	* SAVED/RESTORED AREAS				*
					6722	* N/A				*
					6723	*				*
					6724	* MODIFICATION CONSIDERATIONS				*
					6725	* BMINPT RESIDES ON A SECTOR WITH OPITET. ANY MODIFICATION	1-4*			*
					6726	* SHOULD CONSIDER THE SECTOR BOUNDARY LIMITATION ON SIZE.	1-4*			*
					6727	*				*
					6728	* REQUIRED MODULES				*
					6729	* @SYSEQ - COMMON SYSTEM EQUATES.				*
					6730	* @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUE EQUATES.				*
					6731	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
					6732	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
					6733	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
					6734	* @ERMEQ - ERROR MESSAGE EQUATES.				*
					6735	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.				*
					6736	* \$B\$EQU - CCRPILER FIXED EQUATES.				*
					6737	* \$B@EQU - COMPILER SYSTEM EQUATES.				*
					6738	*				*
					6739	* OTHER				*
					6740	* BMINPT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
					6741	*****				*
					6743	*				*
					6744	* ENTER BMINPT - MAT INPUT STATEMENT ROUTINE				*
					6745	*				*
				16D2	6746	BMINPT EQU *	BMINPT ENTRY POINT			*
					6747	*				*
					6748	* SET GET ROUTINE TO SKIP TO 'T' IN KEYWORDS 'MAT INPUT'				*
					6749	*				*
					6750	BMI010 MVI B\$NUMC,B@LMIN-1	SET GET TO SKIP TO 'T' IN INPUT			*
					6751	B B\$GETC	LINK TO ADVANCE POINTER			*
					6752	*				*
					6753	* CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING				*
					6754	* INSTRUCTIONS IN VIRTUAL MEMORY				*
					6755	*				*
					6756	BMI020 B B\$MATR	LINK TO PROCESS MAT-REFERENCE			*
					6757	*				*
					6758	* GENERATE 'MF1' INSTRUCTION TO INDICATE INPUT IN VIRTUAL MEMORY				*
					6759	*				*
N04	16DE	00	00	00	6760	BMI030 LA BRIMFC(,@BR),@XR	LOAD CADDR OF 'MF1' INSTR			*
	16E1	34	02	0A40	6761	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR MF1			*
	16E5	3C	02	0A41	6762	MVI B\$PNBY,B@LMF1-1	SET LNG PARM OF PUT FOR MFT			*
	16E9	C0	87	093A	6763	B B\$PUTC	LINK TO GENERATE PMC			*
					6764	*				*
					6765	* TEST DELIMITER FOR BEING A STATEMENT TERMINATOR				*
					6766	*				*
					6767	BMI040 L B\$GPTR,@XR	RESTORE TEXT POINTER			*
	16F1	BD	1E	00	6768	CLI B@CHAR(,@XR),B@EOST	IF DELIMITER IS NOT AN EOS			*
	16F4	D0	01	DA	6769	BNE BMI020(,@BR)	* GO PROCESS NEXT MAT-REFERENCE			*

S/3 BASIC COMPILER -MATH INPUT- STATEMENT RTN						
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 94
			6770	*		
			6771	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
			6772	*		
16F7	C0 87 0700		6773	BMI050 B	B\$DIST RETURN TO DISTRIBUTOR	
			6775	*****		
			6776	*	MAT INPUT STATEMENT ROUTINE STORAGE AND PARAMETER AREAS	
			6777	*****		
			6778	*		
16FB	18	16FB	6779	BMIMFC DC	AL(B@LCOP)(B@CMF1) 'MF1' INSTR OP	CODE
16FC	3D00	16FD	6780	BMIMFO DC	AL(B@LCVA)(V\$XMIN)	'MF1' INSTR OPND - INPUT
			6781	*		
			6782	*****		
			6783	*		
			6784	*	END OF 'MAT INPUT' STATEMENT ROUTINE CODING	
			6785	*		

## S/3 BASIC COMPILER -IMAGE- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 95
		6787		*****			
		6788	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		6789	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		6790	*				*
		6791		*****			*
		6792	*	STATUS			*
		6793	*	VERSION 1 MODIFICATION 0			*
		6794	*				*
		6795	*	FUNCTION			*
		6796	*	BNIMAG IS EXECUTED TO TRANSLATE IMAGE STATEMENTS AS THEY OCCUR			*
		6797	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		6798	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		6799	*				*
		6800	*	ENTRY POINTS			*
		6801	*	BNIMAG HAS ONLY ONE ENTRY POINT:			*
		6802	*	BNIMAG - TRANSLATE IMAGE STATEMENT			*
		6803	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		6804	*	B BNIMAG			*
		6805	*				*
		6806	*	INPUT			*
		6807	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		6808	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		6809	*	LEADING KEYWORD, ': '.			*
		6810	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		6811	*	CHARACTER IN THE LEADING KEYWORD, ': '.			*
		6812	*	* B\$ERSW - THE COMPILER MODE SWITCH. THIS SWITCH, TESTED USING			*
		6813	*	MASK B\$ERMK, INDICATES COMPILER ERROR MODE WHEN ON.			*
		6814	*				*
		6815	*	OUTPUT			*
		6816	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		6817	*	GENERATED BY BNIMAG IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		6818	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		6819	*	SEQUENCES.			*
		6820	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		6821	*	CHARACTER WHICH FOLLOWS THE END-OF-STATEMENT CHARACTER IN THE			*
		6822	*	IMAGE STATEMENT.			*
		6823	*	* B\$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF THE			*
		6824	*	ADDRESS OPERAND FIELD IN THE STATEMENT BYPASS BRANCH			*
		6825	*	INSTRUCTION.			*
		6826	*	* B\$NXSN - SET TO ON STATUS TO CAUSE RESOLUTION OF THE STATEMENT			*
		6827	*	BYPASS BRANCH INSTRUCTION OPERAND BY THE COMPILER DISTRIBUTOR.			*
		6828	*				*
		6829	*	EXTERNAL REFERENCES			*
		6830	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		6831	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD, B\$PBNL, B\$ERSW) - ENTRY TO			*
		6832	*	COMPILER VIRTUAL MEMORY OUTPUT ROUTINE.			*
		6833	*	B\$FCON - (B\$CTYP, B\$BCKT, B\$CPCT) - ENTRY TO BASIC COMPILER			*
		6834	*	CONSTANT ROUTINE.			*
		6835	*	B UTAB - (B\$BRVA) - ENTRY TO COMPILER BRANCH TABLE ROUTINE.			*
		6836	*	B\$DIST - (B\$NISW, B\$LINE) - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
		6837	*				*
		6838	*	EXITS, NORMAL			*
		6839	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		6840	*				*
		6841	*	EXITS, ERROR			*
		6842	*	N/A			*

## S/3 BASIC COMPILER -IMAGE- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 96
			6843	*	*
			6844	*TABLES/WORK AREAS	*
			6845	* N/A	*
			6846	*	*
			6847	*ATTRIBUTES	*
			6848	* BNIMAG IS NATURALLY RELOCATABLE AND REUSABLE.	*
			6849	*	*
			6850	*CHARACTER CODE DEPENDENCY	*
			6851	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
			6852	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
			6853	*	*
			6854	*NOTES	*
			6855	* ERROR PROCEDURES	*
			6856	* N/A	*
			6857	*	*
			6858	* REGISTER USAGE	*
			6859	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
			6860	*	*
			6861	* SAVED/RESTORED AREAS	*
			6862	* N/A	*
			6863	*	*
			6864	* MODIFICATION CONSIDERATIONS	*
			6865	* BNIMAG IS CO-RESIDENT ON A SECTOR WITH BMREAD. ANY	1-4*
			6866	* MODIFICATION TO BNIMAG WILL CHANGE THE ENTRY ADDRESS OF	1-4*
			6867	* BMREAD AND MUST CONSIDER THE LIMITATION OF THE SECTOR	1-4*
			6868	* BOUNDARY ON SIZE.	1-4*
			6869	*	*
			6870	* REQUIRED MODULES	*
			6871	* @SYSEQ - COMMON SYSTEM EQUATES.	*
			6872	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
			6873	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.	*
			6874	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
			6875	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
			6876	* @ERMEQ - ERROR MESSAGE EQUATES.	*
			6877	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
			6878	* \$B\$EQU - COMPILER FIXED EQUATES.	*
			6879	* \$B@EQU - COMPILER SYSTEM EQUATES.	*
			6880	*	*
			6881	* OTHER	*
			6882	* BNIMAG IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
			6883	*****	
1700			6885	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
		1700	6886	USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
			6887	*	
			6888	* ENTER BNIMAG - 'IMAGE' STATEMENT ROUTINE	
			6889	*	
		1700	6890	BNIMAG EQU *	BNIMAG ENTRY POINT
			6891	*	
			6892	* REPLACE IMAGE STATEMENT 'STH' PSEUDO INSTRUCTION WITH SPECIAL	
			6893	* IMAGE STATEMENT HEADER ('IMH' INSTRUCTION - INSTRUCTION REPLACEMENT	
			6894	* IS NOT PERFORMED WHEN THE COMPILER IS OPERATING IN ERROR MODE	
			6895	*	
1700	38 07 0993		6896	TBN B\$ERSW,B\$ERMK	TEST ERROR SWITCH - BYPASS SIN
1704	F2 10 1E		6897	JT BNI005	* REPLACEMENT IF COMPILER ERRS
1707	1E 00 0A01 CC		6898	ALC B\$PBNL,BNISHL(1,@BR)	ADJUST INIC DUFFER POINTERS TO

## S/3 BASIC COMPILER -IMAGE- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 97

	170C	1F	00	0A43	CC	6899	SLC	B\$PVAD,BNISHL(1,@BR)	* DELETE LAST 'STH' PSEUDO INST
N04	1711	00	00	00	0000	6900	MVC	BNIIHE(,@BR),B\$LINE(B@LCLN)	SET 'IMH' OPERAND = LINE NO.
	1716	D2	02	BE		6901	LA	BNIIIMH(,@BR),@XR	LOAD 'IMH' INSTRUCTION CADDR
	1719	34	02	0A40		6902	ST	B\$PCAD,@XR	SET 'PUT' RTNPARM FOR 'INH'
	171D	3C	02	0A41		6903	MVI	B\$PNBY,B@LIMH-1	SET 'PUT' RTN LENGTHPARM
	1721	C0	87	093A		6904	B	B\$PUTC	LINK TO PUT THE 'IMH' INST
						6905	*		
						6906	*	'ADVANCE' CHARACTER POINTER TO LAST CHARACTER OF IMAGE 'KEYWORD'	
						6907	*		
N04	1725	00	00	0000		6908	BNI005 MVI	B\$NUMC,BOLIMG-1	SET GETPARM TO SKIP KEYWORD
	1729	C0	87	0867		6909	B	B\$GETC	LINK TO GET LAST KEYWORD CHAR
						6910	*		
						6911	*	GENERATE A 'BRA' IMAGE INSTRUCTION IN VIRTUAL MEMORY	
						6912	*		
	172D	D2	02	C1		6913	BNI010 LA	BNIBRC(,@BR),@XR	LOAD CADDR OF 'BRA' INSTR
	1730	34	02	0A40		6914	ST	B\$PCAD,@XR	SET PUT RTN VADDRPARM FOR BRA
	1734	3C	02	0A41		6915	MVI	B\$PNBY,B@LBRA-1	SET PUT RTN LENGTHPARM FOR BRA
	1738	C0	87	093A		6916	B	B\$PUTC	LINK TO GENERATE 'BRA' INSTR
						6917	*		
						6918	*	ESTABLISH 'BRA' OPERAND FOR ADDRESS RESOLUTION	
						6919	*		
	173C	0C	01	19EF	0A43	6920	BNI020 MVC	B\$BRVA,B\$PVAD(@VADDR)	SET BRA TABLE FOR 'BRA' VADOR
	1742	1F	01	19EF	CB	6921	SLC	B\$BRVA,BNIBN1(@VADDR,@BR)	ADJUST VADDR TO 'BRA' OPERAND
						6922	*		
						6923	*	SET THE TEXT POINTER TO REFERENCE A DUMMY TERMINATOR	
						6924	*		
	1747	D2	02	CD		6925	BNI030 LA	BNIEOS(,@BR),@XR	SET PTR TO DUMMY TERMINATOR
						6926	*		
						6927	*	CALL THE CONSTANT ROUTINE TO GENERATE THE CHARACTER STRING	
						6928	*		
	174A	3C	1B	0A5F		6929	BNI040 MVI	B\$CTYP,B\$SCON	SET CON RTN FOR CHAR STRING
	174E	C0	87	0A46		6930	B	B\$FCON	LINK TO GENERATE CHAR STRING
	1752	3C	00	0873		6931	MVI	B\$NUMC,B@GETS	DISABLE THE GET ROUTINE
						6932	*		
						6933	*	TEST FOR THIS BEING A NULL STRING	
						6934	*		
	1756	3D	00	0CA8		6935	BNI050 CLI	B\$CPCT,@ZERO	IF THIS WAS NOT A NULL STRING
	175A	F2	01	29		6936	JNE	BNI110	* GO GENERATE STC/PRU SEQUENCE
						6937	*		
						6938	*	MOVE A CODE OF '01' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE	
						6939	*	STATEMENT CONTAINS NO IMAGE SPECIFICATIONS	
						6940	*		
	175D	7C	01	C5		6941	BNI060 MVI	BNIPRO(,@BR),B@PUI0	SET 'PRU' CODE TO ONE
						6942	*		
						6943	*	GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY	
						6944	*		
	1760	D2	02	C4		6945	BNI070 LA	BNIPRC(,@BR),@XR	LOAD CADDR OF 'PRU' INSTR
	1763	34	02	0A40		6946	ST	B\$PCAD,@XR	SET PUT RTN VADDRPARM FOR PRU
	1767	3C	01	0A41		6947	MVI	B\$PNBY,B@LPRU-1	SET PUT RTN LNG PARM, FOR PRU
	176B	C0	87	093A		6948	B	B\$PUTC	LINK TO GENERATE 'PRU' INSTR
						6949	*		
						6950	*	GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY	
						6951	*		
	176F	D2	02	C9		6952	BNI080 LA	BNIBSC(,@BR),@XR	LOAD CADDR OF 'BRS' INSTR
	1772	34	02	0A40		6953	ST	B\$PCAD,@XR	SET PUT RTN VADDR PARM FOR 'BRS'
	1776	3C	00	0A41		6954	MVI	B\$PNBY,B@LBRS-1	SET PUT RTN LNGPARM FOR 'BRS'



## S/3 BASIC COMPILER -IMAGE- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 98

177A	C0 87 093A	6955	B	B\$PUTC	LINK TO GENERATE 'BRS' INSTR
		6956	*		
		6957	*	SET DISTRIBUTOR TO SET UP RESOLUTION CONDITIONS FOR 'BRA' OPERAND	
		6958	*		
177E	3A 07 071D	6959	BNI090 SBN	B\$NXSW,B\$NXMK	SET 'NEXT' SNITCH ON
		6960	*		
		6961	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
		6962	*		
1782	C0 87 0700	6963	BNI100 B	B\$DIST	RETURN TO DISTRIBUTOR
		6964	*		
		6965	*	IF THIS IS A CHARACTER STRING MOVE THE VADDR OF THE 1ST SEGMENT TO AN	
		6966	*	'STC' INSTRUCTION OPERAND	
		6967	*		
1786	4C 01 C8 1590	6968	BNI110 MVC	BNISTO(,@BR),B\$BCKT(@VADDR)	SET 'STC' OPERAND FOR VADDR
		6969	*		
		6970	*	MOVE A CODE OF '04' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE	
		6971	*	FIRST CHARACTER CONSTANT IS ESTABLISHED FOR THE IMAGE SPECIFICATION	
		6972	*		
178B	7C 04 C5	6973	BNI120 MVI	BNIPRO(,@BR),B@PUI1	SET 'PRU' CODE TO FOUR
		6974	*		
		6975	*	GENERATE AN 'STC' INSTRUCTION IN VIRTUAL MEMORY	
		6976	*		
178E	D2 02 C6	6977	BNI130 LA	BNISTC(,@BR),@XR	LOAD CADDR OF 'STC' INSTR
1791	34 02 0A40	6978		ST B\$PCAD,@XR	SET PUT RTN VADDRPARG FOR SIC
1795	3C 02 0A41	6979		MVI B\$PNBY,B@LSTC-1	SET PUT RTN LNGPARG FOR STC
1799	C0 87 093A	6980		B B\$PUTC	LINK TO GENERATE 'STC' INSTR
		6981	*		
		6982	*	GENERATE THE 'PRU' INSTRUCTION IN VIRTUAL MEMORY	
		6983	*		
179D	D2 02 C4	6984	BNI140 LA	BNIPRC(,@BR),@XR	LOAD CADDR OF 'PRU' INSTR
17A0	34 02 0A40	6985		ST B\$PCAD,@XR	SET PUT RTN VADDRPARG FOR PRU
17A4	3C 01 0A41	6986		MVI B\$PNBY,B@LPRU-1	SET PUT RTN LNGPARG FOR PRU
17A8	C0 87 093A	6987		B B\$PUTC	LINK TO GENERATE 'PRU' INSTR
		6988	*		
		6989	*	MOVE A CODE OF '05' TO THE 'PRU' INSTR OPERAND TO INDICATE THAT THE	
		6990	*	CHARACTER CONSTANT IS ANY ESTABLISHED FOR THE IMAGE SPECIFICATION	
		6991	*	EXCEPT THE FIRST	
		6992	*		
17AC	7C 05 C5	6993	BNI150 MVI	BNIPRO(,@BR),B@PUI2	SET THE PRU CODE TO FIVE
		6994	*		
		6995	*	SUBTRACT THE LENGTH OF A STRING SEGMENT FROM 'STC' INSIR OPERAND	
		6996	*		
17AF	5F 01 C8 CF	6997	BNI160 SLC	BNISTO(,@BR),BNISUB(@VADDR,@BR)	SUB LNG OF STRING SEGMENT
		6998	*		
		6999	*	TEST FOR THE PRESENCE OF OTHER STRING SEGMENTS	
		7000	*		
17B3	1F 00 0CA8 CB	7001	BNI170 SLC	B\$CPCT,BNIBN1(1,@BR)	IF OTHER SEGMENTS ARE PRESENT
17B8	D0 84 8E	7002		BH BNI130(,@BR)	* BRANCH TO GENERATE 'STC'
		7003	*		
		7004	*	IF OTHER SEGMENTS ARE NOT PRESENT BRANCH TO GENERATE THE 'BRS' INSTR	
		7005	*		
17BB	D0 87 6F	7006	BNI180 B	BNI080(,@BR)	BRANCH TO GENERATE 'BRS' INSTR



## S/3 BASIC COMPILER -IMAGE- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 99
				7008	*****		
				7009	* 'IMAGE' STATEMENT ROUTINE STORAGE AND PARAMETER AREA		
				7010	*****		
				7011	*		
17BE	66		17BE	7012	BNIIMH DC	AL(B@LCOP)(B@CIMH) 'IMH' INSTRUCTION OPCODE	
17BF			17C0	7013	BNIIHO DS	CL(B@LCLN) 'IMH' INSTRUCTION OPERAND	
				7014	*		
17C1	46		17C1	7015	BNIBRC DC	AL(B@LCOP)(B@CBRA) 'BRA' INSTR OPCODE	
17C2	0000		17C3	7016	BNIBRO DC	XL(B@LCVA) '00' 'BRA' INSTR OPERAND	
				7017	*		
17C4	62		17C4	7018	BNIPRC DC	AL(B@LCOP)(B@CPRU) 'PRU' INSTR OPCODE	
17C5			17C5	7019	BNIPRO DS	CL(B@LCXX) 'PRU' INSTR OPERAND	
				7020	*		
17C6	28		17C6	7021	BNISTC DC	AL(B@LCOP)(B@CSTC) 'STC' INSTR OPCODE	
17C7			17C8	7022	BNISTO DS	CL(@VADDR) 'STC' INSTR OPERAND	
				7023	*		
17C9	4C		17C9	7024	BNIBSC DC	AL(B@LCOP)(B@CBRS) 'BRS' INSTR OPCODE *	
				7026	*****		
				7027	* 'IMAGE' STATEMENT ROUTINE CONSTANTS		
				7028	*		
17CA	0001		17CB	7029	BNIBN1 DC	IL(@VADDR) '1' BINARY 1	
17CC	03		17CC	7030	BNISHL DC	AL1(B@LSTH) LENGTH OF 'STH' INSTRUCTION	
17CD	1E		17CD	7031	BNIEOS DC	AL1(B@EOST) DUMMY TERMINATOR	
17CE	0013		17CF	7032	BNISUB DC	AL(@VADDR)(B@LCRV) LENGTH OF STRING SEGMENT	
				7033	*		
				7034	*****		
				7035	*		
				7036	* END OF 'IMAGE' STATEMENT ROUTINE CODING		
				7037	*		

## S/3 BASIC COMPILER -MREAD- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 100
			7039		*****			
			7040	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			7041	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			7042	*				*
			7043		*****			*
			7044	*	STATUS			*
			7045	*	VERSION 1 MODIFICATION 0			*
			7046	*				*
			7047	*	FUNCTION			*
			7048	*	BNREAD IS EXECUTED TO TRANSLATE MAT READ STATEMENTS AS THEY OCCUR			*
			7049	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
			7050	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
			7051	*				*
			7052	*	ENTRY POINTS			*
			7053	*	BMREAD HAS ONLY ONE ENTRY POINT:			*
			7054	*	BMREAD - TRANSLATE MAT READ STATEMENT			*
			7055	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			7056	*	B BMREAD			*
			7057	*				*
			7058	*	INPUT			*
			7059	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			7060	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			7061	*	LEADING KEYWORD, MAT READ.			*
			7062	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			7063	*	CHARACTER IN TIE LEADING KEYWORD, MAT READ.			*
			7064	*				*
			7065	*	OUTPUT			*
			7066	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			7067	*	GENERATED BY BMREAD IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			7068	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			7069	*	SEQUENCES.			*
			7070	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			7071	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			7072	*				*
			7073	*	EXTERNAL REFERENCES			*
			7074	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			7075	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
			7076	*	OUTPUT ROUTINE.			*
			7077	*	B\$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.			*
			7078	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRUBUTOR.			*
			7079	*				*
			7080	*	EXITS, NORMAL			*
			7081	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTGE.			*
			7082	*				*
			7083	*	EXITS, ERROR			*
			7084	*	N/A			*
			7085	*				*
			7086	*	TABLES/WORK AREAS			*
			7087	*	N/A			*
			7088	*				*
			7089	*	ATTRIBUTES			*
			7090	*	BMREAD IS NATURALLY RELOCATABLE AND REUSABLE.			*
			7091	*				*
			7092	*	CHARACTER CODE DEPENDENCY			*
			7093	*	THE OPERATION OF THIS MODUE DOES NOT DEPEND UPON A PARTICULAR			*
			7094	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*

## S/3 BASIC COMPILER -MREAD- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 101
				7095	*				*
				7096	*NOTES				*
				7097	* ERROR PROCEDURES				*
				7098	* N/A				*
				7099	*				*
				7100	* REGISTER USAGE				*
				7101	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
				7102	*				*
				7103	* SAVED/RESTORED AREAS				*
				7104	* N/A				*
				7105	*				*
				7106	* MODIFICATION CONSIDERATIONS				*
				7107	* BMREAD IS CO-RESIDENT ON A SECTOR WITH BNIMAG. ANY			1-4	*
				7108	* MODIFICATION SHOULD CONSIDER THE CO-RESIDENCY AND THE			1-4	*
				7109	* LIMITATION OF THE SECTOR BOUNDARY ON SIZE.			1-4	*
				7110	*				*
				7111	* REQUIRED MODULES				*
				7112	* @SYSEQ - COMMON SYSTEM EQUATES.				*
				7113	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.				*
				7114	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
				7115	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
				7116	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
				7117	* @ERMEQ - ERROR MESSAGE EQUATES.				*
				7118	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.				*
				7119	* \$B\$EQU - COMPILER FIXED EQUATES.				*
				7120	* \$B@EQU - COMPILER SYSTEM EQUATES.				*
				7121	*				*
				7122	* OTHER				*
				7123	* BMREAD IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
				7124	*****				*
				7126	*				*
				7127	* ENTER BMREAD - MAT READ STATEMENT ROUTINE				*
				7128	*				*
			17D0	7129	BMREAD EQU *	BMREAD ENTRY POINT			*
				7130	*				*
				7131	* SET GET RTN TO SKIP TO 'D' IN KEYWORD 'MAT READ'				*
				7132	*				*
		17D0 3C 06 0873		7133	BMR010 MVI B\$NUMC,B@LMRD-1	SET GETC TO SKIP TO 'D'			*
		17D4 C0 87 0867		7134	B B\$GETC	LINK IT ADVANCE POINTER			*
				7135	*				*
				7136	* CALL MATRIX REFERENCE ROUTINE TO GENERATE DOPE VECTOR STACKING INSTR				*
				7137	*				*
		N04 17D8 00 00 0000		7138	BMR020 B B@MATR	LINK TO PROCESS MAT-REFERENCE			*
				7139	*				*
				7140	* GENERATE A MATRIX FUNCTION CALL INSTR WHICH REFERENCES THE VADDR OF				*
				7141	* THE RUN-TIME MATRIX DATA READ ROUTINE				*
				7142	*				*
		17DC D2 02 F9		7143	BMR030 LA BMRMFC(,@BR),@XR	LOAD CADDR OF 'MF1' INSTR			*
		17DF 34 02 0A40		7144	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'MF1'			*
		17E3 3C 02 0A41		7145	MVI B\$PNBY,B@LMF1-1	SET LNG PARM OF PUT FOR 'MF1'			*
		17E7 C0 87 093A		7146	B B\$PUTC	LINK TO GENERATE 'MF1' INSTR			*
				7147	*				*
				7148	* TEST DELIMITER FOR BEING A STATEMENT TERMINATOR				*
				7149	*				*
		17EB 35 02 0878		7150	BMR040 L B\$GPTR,@XR	RESTORE TEXT POINTER			*

S/3 BASIC COMPILER -MREAD- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 102
	17EF	BD 1E 00		7151	CLI	B@CHAR(,@XR),B@EOST			IF DELIMITER IS NOT EOS
	17F2	D0 01 D8		7152	BNE	BMR020(,@BR)			* GO PROCESS NEXT MAT-REFERENCE
				7153	*				
				7154	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR			
				7155	*				
	17F5	C0 87 0700		7156	BMR050 B	B\$DIST			RETURN TO DISTRIBUTOR
				7157	*****				
				7158	*	MAT READ STATEMENT ROUTINE STORAGE AND PARAMETER AREA			
				7159	*****				
				7160	*				
	17F9	18	17F9	7161	BMRMFC DC	AL(B@LCOP)(B@CMF1)			'MF1' INSTR OPCODE
	17FA	3E00	17FB	7162	BMRMFO DC	AL(B@LCVA)(V\$XMRD)			'MF1' INSTR OPERAND
				7163	*				
				7164	*****				
				7165	*				
				7166	*	END OF 'MAT READ' STATEMENT ROUTINE CODING			
				7167	*				

## S/3 BASIC COMPILER -PUT- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 103
			7169		*****			
			7170	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			7171	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			7172	*				*
			7173		*****			*
			7174	*	STATUS			*
			7175	*	VERSION 1 MODIFICATION 0			*
			7176	*				*
			7177	*	FUNCTION			*
			7178	*	BXPUTX IS EXECUTED TO TRANSLATE PUT STATEMENTS AS THEY OCCUR IN A			*
			7179	*	BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE			*
			7180	*	PSEUDOCODE IN VIRTUAL MEMORY.			*
			7181	*				*
			7182	*	ENTRY POINTS			*
			7183	*	BXPUTX HAS ONLY ONE ENTRY POINT:			*
			7184	*	BXPUTX - TRANSLATE PUT STATEMENT			*
			7185	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			7186	*	B BXPUTX			*
			7187	*				*
			7188	*	LINK			*
			7189	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			7190	*	THAT RECORD SEGMENT CONTAINS THE FIRST CHARACTER IN THE			*
			7191	*	LEADING KEYWORD, PUT.			*
			7192	*	TEST CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			7193	*	CHARACTER IN THE LEADING KEYWORD, PUT.			*
			7194	*				*
			7195	*	OUTPUT			*
			7196	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			7197	*	GENERATED BY BXPUTX IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			7198	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			7199	*	SEQUENCES.			*
			7200	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			7201	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			7202	*				*
			7203	*	EXTERNAL REFERENCES			*
			7204	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			7205	*	B\$PUTC - (B\$PCAD, ISPABY, B\$ARSA, - ENTRY TO COMPILER VIRT			*
			7206	*	MEMORY ROUTINE.			*
			7207	*	B\$CSCN - (B\$CSSW) - ENTRY TO BASIC COMPILER CHARACTER SCAN			*
			7208	*	ROUTINE.			*
			7209	*	B\$SCAN - ENTRY TO BASIC COMPILER ARITMETIC EXPRESSION SCAN			*
			7210	*	ROUTINE.			*
			7211	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			7212	*				*
			7213	*	EXITS, NORMAL			*
			7214	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			7215	*				*
			7216	*	EXITS, ERROR			*
			7217	*	N/A			*
			7218	*				*
			7219	*	TABLES/WORK AREAS			*
			7220	*	N/A			*
			7221	*				*
			7222	*	ATTRIBUTES			*
			7223	*	BXPUTX IS NATURALLY RELOCATABLE AND REUSABLE.			*
			7224	*				*

## S/3 BASIC COMPILER -PUT- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 104
				7225	*CHARACTER CODE DEPENDENCY	*
				7226	*	*
				7227	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
				7228	*	*
				7229	*INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
				7230	*	*
				7231	*NOTES	*
				7232	* ERROR PROCEDURES	*
				7233	* N/A	*
				7234	*	*
				7235	* REGISTER USAGE	*
				7236	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				7237	*	*
				7238	* SAVED/RESTORED AREAS	*
				7239	* N/A	*
				7240	*	*
				7241	* MODIFICATION CONSIDERATIONS	*
				7242	* BXPUTX RESIDES ON THE SAME SECTOR WITH BPCLET AND BXGETX. 1-4	*
				7243	* ANY MODIFICATION TO BXPUTX WILL CHANGE THE ENTRY ADDRESSES 1-40	*
				7244	* OF BPCLET AND BXGETX AND MUST CONSIDER THE LIMITATION 1-4.	*
				7245	* OF THE SECTOR BOUNDARY ON SIZE. 1-40	*
				7246	*	*
				7247	* REQUIRED MODULES	*
				7248	* @STSEQ - COMMON SYSTEM EQUATES.	*
				7249	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
				7250	* @CANEQ - COMION CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
				7251	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
				7252	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
				7253	* @ERMEQ - ERROR MESSAGE EQUATES.	*
				7254	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
				7255	* \$B\$EQU - COMPILER FIXED EQUATES.	*
				7256	* \$B@EQU - COMPILER SYSTEM EQUATES.	*
				7257	*	*
				7258	* OTHER	*
				7259	* BXPUTX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
				7260	*****	
1800				7262	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY 1-4
	1800			7263	USING *,@BR	DEFINE BASE AMA FOR CORE PG 1-4
				7264	*	
				7265	* ENTER BXPUTX 'PUT' STATEMENT ROUTINE	
				7266	*	
			1800	7267	BXPUTX EQU *	BXPUTX ENTRY POINT
				7268	*	
				7269	* SET POINTER TO SKIP TO CHARACTER FOLLOWING 'PUT'	
				7270	*	
1800	3C 02 0873			7271	BXP010 MVI B\$NUMC,B@LKPT-1	SET GET RTN TO SKIP KEYWORD
1804	C0 87 0867			7272	B B\$GETC	LINK TO ADVANCE POINTER
1808	C0 87 14B0			7273	B B\$CSCN	LINK TO PROCESS FILE REFERENCE
				7274	*	
				7275	* GENERATE THE 'ADF' PMC IN VIRTUAL MEMORY (IF THE FILENAME IN THE	
				7276	* STMT DID NOT MATCH ONE OF THE TABLE ENTRIES, THE 'ADF' OPERAND WILL	
				7277	* BE ZERO)	
				7278	*	
180C	D2 02 63			7279	BXP100 LA BXPAFC(,@BR),@XR	LOAD CADDR OF 'ADF' INSTR
180F	34 02 0A40			7280	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'ADF'

## S/3 BASIC COMPILER -PUT- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 105

1813	3C	01	0A41	7281	MVI	B\$PNBY,B@LADF-1	SET LNG PARM OF PUT FOR 'ADF'
1817	C0	87	093A	7282	B	B\$PUTC	LINK TO GENERATE 'ADF' PMC
				7283	*		
				7284	*	CALL GET ROUTINE TO GET NEXT CHARACTER	
				7285	*		
181B	3C	00	0873	7286	BXP120	MVI B\$NUMC,B@GETS	DISABLE GET ROUTINE
181F	C0	87	0867	7287	B	B\$GETC	LINK TO GET CHARACTER POINTER
				7288	*		
				7289	*	ATTEMPT TO PROCESS THE VARIABLE AS ARITHMETIC VARIABLE	
				7290	*		
1823	C0	87	1514	7291	BXP140	B B\$SCAN	LINK TO ATTEMPT 4RITH PROCESS
				7292	*		
				7293	*	TEST FOR ANY PMC GENERATION	
				7294	*		
1827	38	01	0A45	7295	BXP150	TBN B\$ARSW,B\$ARMK	IF NO PMC GENERATED
N04 182B	00	00	00	7296	JF	BXPI70	* GO TEST FOR CHAR VARIABLE
				7297	*		
				7298	*	SET 'PUT' OPERAND FOR ARITH VARIABLE AND BRANCH TO GENERATE 'PUT' PMC	
				7299	*		
182E	7C	02	66	7300	BXP160	MVI BXPPTO(,@BR),BXPC02	SET CODE FOR ARITH VARIABLE
1831	D0	87	46	7301	B	BXP210(,@BR)	GO GENERATE 'PUT' PMC
				7302	*		
				7303	*	TEST FOR THIS BEING A CHARACTER VARIABLE	
				7304	*		
1834	38	07	14BC	7305	BXP170	TBN B\$CSSW,B\$CSMK	IF VAR IS CHAR VARIABLE
1838	F2	10	04	7306	JT	BXP190	* JUMP TO PROCESS CHAR VAR
				7307	*		
				7308	*	IF LIST ELEMENT IS A CHAR CONSTANT DISABLE GET ROUTINE SKIP PARAMETER	
				7309	*		
183B	3C	00	0873	7310	BXP180	MVI B\$NUMC,B@GETS	DISABLE GET RTN SKIPPARM
				7311	*		
				7312	*	BRANCH TO CHARACTER SCAN ROUTINE TO PROCESS CHARACTER ELEMENT	
				7313	*		
183F	C0	87	14B0	7314	BXP190	B B\$CSCN	LINK TO PROCESS CHAR ELEMENT
				7315	*		
				7316	*	SET 'PUT' OPERAND FOR A CHARACTER ELEMENT	
				7317	*		
1843	7C	04	66	7318	BXP200	MVI BXPPTO(,@BR),BXPC04	SET CODE FOR CHAR ELEMENT
				7319	*		
				7320	*	GENERATE THE 'PUT' PMC IN VIRTUAL MEMORY	
				7321	*		
1846	D2	02	65	7322	BXP210	LA BXPPTC(,@BR),@XR	LOAD CADOR OF 'PUT' INSTR
1849	34	02	0A40	7323		ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'PUT'
184D	3C	01	0A41	7324		MVI B\$PNBY,B@LPUT-1	SET LNG PARM CF PUT FOR 'PUT'
1851	C0	87	093A	7325	B	B\$PUTC	LINK TO GENERATE 'PUT' PMC
				7326	*		
				7327	*	TEST NEXT TEXT CHAR FOR BEING THE END-OF-STATEMENT	
				7328	*		
1855	35	02	0878	7329	BXP220	L B\$GPTR,@XR	RESTORE TEXT POINTER
1859	BD	1E	00	7330		CLI B@CHAR(,@XR),B@EOST	IF OTHER ELEMENTS EXIST
185C	D0	01	23	7331		BNE BXP140(,@BR)	GO PROCESS NEXT LIST ELEMENT
				7332	*		
				7333	*	TEST NEXT TEXT CHAR BEING THE EOND-OF-STATEMENT	
				7334	*		
185F	C0	87	0700	7335	BXP230	B B\$DIST	RETURN TO DISTRIBUTOR



S/3 BASIC COMPILER -PUT- STATEMENT ROUTINE					
ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 106
			7337	*****	
			7338	* 'PUT' STATEMENT STORAGE AND PARAMETER AREAS	
			7339	*****	
			7340	*	
1863 58		1863	7341	BXPAFC DC AL(B@LCOP)(B@CADF)	'ADF' INSTR OPCODE
1864 01		1864	7342	BXPAFO DC XL1'01'	PUT INDICATOR FOR 'ADF' INSTR
			7343	*	
1865 54		1865	7344	BXPPTC DC AL(B@LCOP)(B@CPUT)	'PUT' INSTR OPCODE
1866		1866	7345	BXPPTO DS CL(B@LCXX)	'PUT' INSTR OPERAND
			7347	*****	
			7348	* 'PUT' STATEMENT CONSTANTS AND EQUATES	
			7349	*****	
			7350	*	
			7351	* CONSTANTS	
			7352	*	
1867 0001		1867	7353	BXPSFA EQU *	
		1868	7354	BXPBN1 DC IL(@CADDR)'1'	BINARY 1
			7355	*	
			7356	* EQUATES	
			7357	*	
		0002	7358	BXPC02 EQU X'02'	ARITH VARIABLE CODE
		0004	7359	BXPC04 EQU X'04'	CHARACTER VAR OR CONSTANT CODE
			7360	*	
			7361	*****	
			7362	*	
			7363	* END OF 'PUT' STATEMENT ROUTINE CODING	
			7364	*	

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 107
			7366		*****			
			7367	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			7368	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			7369	*				*
			7370		*****			
			7371	*	STATUS			*
			7372	*	VERSION 1 MODIFICATION 0			*
			7373	*				*
			7374	*	FUNCTION			*
			7375	*	BPCLET IS EXECUTED TO TRANSLATE CHARACTER ASSIGNMENT AND LET			*
			7376	*	STATEMENTS AS THEY OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE			*
			7377	*	PSEUDOCODE AND TO PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.			*
			7378	*				*
			7379	*	ENTRY POINTS			*
			7380	*	BPCLET HAS TWO ENTRY POINTS:			*
			7381	*	BPCASN - TRANSLATE CHARACTER ASSIGNMENT STATEMENT			*
			7382	*	BPCLET - TRANSLATE CHARACTER LET STATEMENT			*
			7383	*	THE FORMAT OF THE CALLING SEQUENCES IS:			*
			7384	*	B BPCASN			*
			7385	*	B BPCLET			*
			7386	*				*
			7387	*	INPUT			*
			7388	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			7389	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			7390	*	LEADING KEYWORD, LET, OR THE FIRST CHARACTER IN THE ASSIGNMENT			*
			7391	*	LIST IF THE OPTIONAL KEYWORD IS OMITTED.			*
			7392	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST			*
			7393	*	CHARACTER IN THE LEADING KEYWORD, LET, OR IN THE ASSIGNMENT			*
			7394	*	LIST IF THE KEYWORD IS OMITTED.			*
			7395	*				*
			7396	*	OUTPUT			*
			7397	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			7398	*	GENERATED BY BPCLET IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			7399	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			7400	*	SEQUENCES.			*
			7401	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			7402	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			7403	*				*
			7404	*	EXTERNAL REFERENCES			*
			7405	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			7406	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
			7407	*	OUTPUT ROUTINE.			*
			7408	*	B\$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.			*
			7409	*	B\$CSCN - ENTRY TO BASIC COMPILER CHARACTER SCAN ROUTINE.			*
			7410	*	B\$LIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			7411	*				*
			7412	*	EXITS, NORMAL			*
			7413	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR			*
			7414	*				*
			7415	*	EXITS, ERROR			*
			7416	*	N/A			*
			7417	*				*
			7418	*	TABLES/WORK AREAS			*
			7419	*	N/A			*
			7420	*				*
			7421	*	ATTRIBUTES			*

## S/3 BASIC COMPILER CHAR -LET- STATEMENT RTN

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 108
				7422 *	BPCLET IS NATURALLY RELOCATABLE AND REUSABLE.	*
				7423 *		*
				7424 *	CHARACTER CODE DEPENDENCY	*
				7425 *	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR	*
				7426 *	INTERNAL REPRESENTATION OF THE INTERNAL CHARACTER SET.	*
				7427 *		*
				7428 *	NOTES	*
				7429 *	ERROR PROCEDURES	*
				7430 *	N/A	*
				7431 *		*
				7432 *	REGISTER USAGE	*
				7433 *	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				7434 *		*
				7435 *	SAVED/RESTORED AREAS	*
				7436 *	N/A	*
				7437 *		*
				7438 *	MODIFICATION CONSIDERATIONS	*
				7439 *	BPCLET RESIDES ON THE SAME SECTOR WITH BXPUTX AND BXGETX.	1-4*
				7440 *	ANY MODIFICATION TO BPCLET WILL CHANGE THE ENTRY ADDRESS	1-4*
				7441 *	OF BXGETX AND MUST CONSIDER THE LIMITATION OF THE SECTOR	1-4*
				7442 *	BOUNDARY ON SIZE.	1-4*
				7443 *		*
				7444 *	REQUIRED MODULES	*
				7445 *	@SYSEQ - COMMON SYSTEM EQUATES.	*
				7446 *	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
				7447 *	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
				7448 *	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
				7449 *	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
				7450 *	@ERMEQ - ERROR MESSAGE EQUATES.	*
				7451 *	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES	*
				7452 *	\$B\$EQU - COMPILER FIXED EQUATES.	*
				7453 *	\$B@EQU - COMPILER SYSTEM EQUATES.	*
				7454 *		*
				7455 *	OTHER	*
				7456 *	BPCLET IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS	*
				7457 *	*****	
				7459 *		
				7460 *	ENTER BPCLET - CHARACTER 'LET' STATEMENT PROCESSOR	
				7461 *		
			1869	7462	BPCLET EQU * BPCLET ENTRY POINT	
				7463 *		
				7464 *	SKIP PAST 'LET' TO 1ST ASSIGNMENT LIST SYMBOL CHARACTER	
				7465 *		
1869	3C	03	0873	7466	BPC010 MVI B\$NUMC,B@LLET SET GET ROUTINE TO SLIP 'LET'	
186D	C0	87	0867	7467	B B\$GETC LINK TO GET 1ST SYMBOL CHAR	
				7468 *		
				7469 *	ENTER BPCASN - CHARACTER ASSIGNMENT STATEMENT PROCESSOR	
				7470 *		
			1871	7471	BPCASN EQU * BPCASN ENTRY POINT	
				7472 *		
				7473 *	ESTABLISH A COUNT OPERAND FIELD WHICH INDICATES TIME NUMBER OF	
				7474 *	VARIABLES IN THE ASSIGNMENT LIST AND INITIALIZE THE COUNT TO ZERO	
				7475 *		
1871	7C	00	A2	7476	BPC020 MVI BPCUCO(,@BR),@ZERO SET SYMNC. COUNT TO ZERO	
				7477 *		

## S/3 BASIC COMPILER CHAR -LET- STATEMENT RTN

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 109
					7478	*	EVALUATE EACH OF THE CHARACTER SYMBOLS IN THE ASSIGN	
					7479	*		
	1874	C0	87	1853	7480	BPC030 B	B\$LIST LINK TO PROCESS CHAR SYMBOL	
N04	1878	00	00	00 00	7481	ALC	BPCUC0(,@BR),BPCBN1(B@LCNN,@BR) ADD 1 TO LIST COUNT	
					7482	*		
					7483	*	IF DELIMITER IS NOT AN EQUAL SIGN (IE. A COMMA) CONTINUE TO PROCESS	
					7484	*	THE ASSIGNMENT LIST	
					7485	*		
	187C	7D	7E	00	7486	BPC040 CLI	B@CHAR(,@BR),B@EQU	IF DELIMITER IS AN EQUAL SIGN
	187F	F2	81	07	7487	JE	BPC050	* DETERMINE THE ASSGNMNT VALUE
	1882	C0	87	0867	7488	B	B\$GETC	LINK TO GET NEXT SYMBOL CHAR
	1886	D0	87	74	7489	B	BPC030(,@BR)	GO PROCESS NEXT SYMBOL CHAR
					7490	*		
					7491	*	EVALUATE VALUE TO BE ASSIGNED THE CHARACTER SYMBOLS IN THE LIST AND	
					7492	*	SET UP PMC FOR 'USC' BEFORE BRANCHING TO THE KIT ROUTINE	
					7493	*		
	1889	C0	87	14B0	7494	BPC050 B	B\$CSCN	LINK TO CHAR SCAN ROUTINE
	188D	D2	02	A1	7495	LA	BPCUCC(,@BR),@XR	LOAD CADDR OF 'USC' INSTR
	1890	34	02	0A40	7496	ST	B\$PCAD,@XR	SET VADDR PARM FOR PUT RTN
	1894	3C	01	0A41	7497	MVI	B\$PNBY,B@LUSC-1	SET LENGTH PARM FOR PUT RTN
	1898	C0	87	093A	7498	B	B\$PUTC	LINK TO OUTPUT 'USC' INSTR
	189C	C0	87	0700	7499	B	B\$DIST	RETURN TO DISTRIBUTOR
					7501	*****		
					7502	*	CHARACTER 'LET' ROUTINE CONSTANTS	
					7503	*****		
					7504	*		
	18A0	01			18A0 7505	BPCBN1 DC	IL(B@LCNN)'1'	BINARY INTEGER 1
					7507	*****		
					7508	*	CHARACTER 'LET' ROUTINE PMC AND STORAGE PARAMETERS	
					7509	*****		
					7510	*		
	18A1	2C			18A1 7511	BPCUCC DC	AL(B@LCOP)(B@CUSC)	UNSTACK CHAR OPCODE
	18A2				18A2 7512	BPCUCO DS	CL(B@LCNN)	UNSTACK CHAR OPERAND
					7513	*		
					7514	*****		
					7515	*		
					7516	*	END OF CHARACTER 'LET' ROUTINE CODING	
					7517	*		

## S/3 BASIC COMPILER -GET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 110
			7519		*****			
			7520	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			7521	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			7522	*				*
			7523		*****			*
			7524	*	STATUS			*
			7525	*	VERSION 1 MODIFICATION 0			*
			7526	*				*
			7527	*	FUNCTION			*
			7528	*	BXGETX IS EXECUTED TO TRANSLATE GET STATEMENTS AS THEY OCCUR IN			*
			7529	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEODOCODE AND TO PLACE THE			*
			7530	*	PSEODOCODE IN VIRTUAL MEMORY.			*
			7531	*				*
			7532	*	ENTRY POINTS			*
			7533	*	BXGETX HAS ONLY ONE ENTRY POINT:			*
			7534	*	BXGETX - TRANSLATE GET STATEMENT			*
			7535	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			7536	*	B BXGETX			*
			7537	*				*
			7538	*	INPUT			*
			7539	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			7540	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			7541	*	LEADING KEYWORD, GET.			*
			7542	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST			*
			7543	*	CHARACTER IN LEADING KEYWORD, GET.			*
			7544	*				*
			7545	*	OUTPUT			*
			7546	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			7547	*	GENERATED BY BXGETX IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			7548	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			7549	*	SEQUENCES.			*
			7550	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			7551	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			7552	*				*
			7553	*	EXTERNAL REFERENCES			*
			7554	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			7555	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
			7556	*	OUTPUT ROUTINE.			*
			7557	*	B\$LIST - ENTRY TO BASIC COMPILER LIST ADDRESS ROUTINE.			*
			7558	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			7559	*				*
			7560	*	EXITS, NORMAL			*
			7561	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			7562	*				*
			7563	*	EXITS, ERROR			*
			7564	*	N/A			*
			7565	*				*
			7566	*	TABLES/WORE AREAS			*
			7567	*	N/A			*
			7568	*				*
			7569	*	ATTRIBUTES			*
			7570	*	BXGETX IS NATURALLY RELOCATABLE AND REUSABLE.			*
			7571	*				*
			7572	*	CHARACTER CODE DEPENDENCY			*
			7573	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
			7574	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*

## S/3 BASIC COMPILER -GET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 111
				7575	*				*
				7576	*NOTES				*
				7577	* ERROR PROCEDURES				*
				7578	* N/A				*
				7579	*				*
				7580	* REGISTER USAGE				*
				7581	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
				7582	*				*
				7583	* SAVED/RESTORED AREAS				*
				7584	* N/A				*
				7585	*				*
				7586	* MODIFICATION CONSIDERATIONS				*
				7587	* BXGETX RESIDES ON THE SAME SECTOR WITH BXPUTX AND BPCLET.	1-4*			
				7588	* ANY MODIFICATION TO BXGETX MUST CONSIDER THIS CO-RESIDENCY	1-4*			
				7589	* AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.	1-4*			
				7590	*				*
				7591	* REQUIRED MODULES				*
				7592	* @SYSEQ - COMMON SYSTEM EQUATES				*
				7593	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES				*
				7594	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS				*
				7595	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES				*
				7596	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES				*
				7597	* @ERMEQ - ERROR MESSAGE EQUATES				*
				7598	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES				*
				7599	* SB\$EQU - COMPILER FIXED EQUATES				*
				7600	* SB@EQU - COMPILER SYSTEM EQUATES				*
				7601	*				*
				7602	* OTHER				*
				7603	* BXGETX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS				*
				7604	*****				
				7606	*				
				7607	* ENTER BXGETX - 'GET' STATEMENT ROUTINE				
				7608	*				
			18A3	7609	BXGETX EQU *	BXGETX ENTRY POINT			
				7610	*				
				7611	* SET POINTER TO SKIP TO CHARACTER FOLLOWING KEYWORD 'GET'				
				7612	*				
18A3	3C	02	0873	7613	BXG010 MVI B\$NUMC,B@LKGT-1	SET GET RTN TO SKIP KEYWORD			
18A7	C0	87	0867	7614	B B\$GETC	LINK TO ADVANCE POINTER			
18AB	C0	87	14B0	7615	B B\$CSCN	LINK TO PROCESS FILE REFERENCE			
				7616	*				
				7617	* GENERATE THE 'ADF' PMC IN VIRTUAL MEMORY (IF FILE NAME IN THE STMT				
				7618	* DID NOT MATCH ONE OF THE TABLE ENTRIES, THE 'ADF' OPERAND WILL BE				
				7619	* ZERO.				
				7620	*				
18AF	D2	02	EB	7621	BXG100 LA BXGAFC(,@BR),@XR	LOAD CADDR OF 'ADF' INSTR			
18B2	34	02	0A40	7622	ST B\$PCAD,@XR	SET PUT RTN VADDR PARM FOR 'ADF'			
18B6	3C	01	0A41	7623	MVI B\$PNBY,B@LADF-1	SET LNG PARM OF PUT FOR 'ADF'.			
18BA	C0	87	093A	7624	B B\$PUTC	LINK TO GENERATE 'ADF' PMC			
				7625	*				
				7626	* CALL GET RTN TO GET NEXT CHARACTER				
				7627	*				
18BE	C0	87	0867	7628	BXG110 B B\$GETC	LINK TO GET NEXT CHARACTER			
				7629	*				
				7630	* GET NEXT CHARACTER				

## S/3 BASIC COMPILER -GET- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 112
					7631	*		
	18C2	3C	00 0873		7632	MVI	B\$NUMC,B@GETS	DISABLE GET ROUTINE
	18C6	C0	87 0867		7633	BXG120 B	B\$GETC	LINK TO GET CHARACTER POINTER
					7634	*		
					7635	*	CALL LIST ROUTINE TO PROCESS CHARACTER	
					7636	*		
	18CA	C0	87 1853		7637	BXG130 B	B\$LIST	LINK TO PROCESS CHARACTER
					7638	*		
					7639	*	GENERATE 'GET' PMC IN VIRTUAL MEMORY	
					7640	*		
	18CE	D2	02 ED		7641	BXG140 LA	BXGGTC(,@BR),@XR	LOAD CADDR OF 'GET' PMC
	18D1	34	02 0A40		7642	ST	B\$PCAD,@XR	SET PUT RTN VADDR PARM FOR GET
	18D5	3C	02 0A41		7643	MVI	B\$PNBY,B@LGET-1	SET PUT RTN LNG PARM FOR GET
	18D9	C0	87 093A		7644	B	B\$PUTC	LINK TO GENERATE PMC
					7645	*		
					7646	*	TEST FOR END OF STATEMENT	
					7647	*		
	18DD	35	02 0878		7648	BXG150 L	B\$GPTR,@XR	RESTORE TEXT POINTER
	18E1	BD	1E 00		7649	CLI	B@CHAR(,@XR),B@EOST	IF THIS IS NOT TERMINATOR
	18E4	D0	01 C6		7650	BNE	BXG120(,@BR)	* BRANCH TO GET NEXT CHAR
					7651	*		
					7652	*	RETURN CONTROL TO THE COMPLIER DISTRIBUTOR	
					7653	*		
	18E7	C0	87 0700		7654	BXGI60 B	B\$DIST	RETURN TO DISTRIBUTOR
					7656	*****		
					7657	*	'GET' STATEMENT ROUTINE STORAGE AND PARAMETER AREAS	
					7658	*****		
					7659	*		
	18EB	58		18EB	7660	BXGAFC DC	AL(B@LCOP)(B@CADF)	'ADF' INSTR OPCODE
	18EC	00		18EC	7661	BXGAFO DC	XL1'00'	GET INDICATOR FOR 'ADF' INSTR
					7662	*		
	18ED	52		18ED	7663	BXGGTC DC	AL(B@LCOP)(B@CGET)	'GET' INSTR OPCODE
N04	18EE	0000		18EF	7664	BXGGTO DC	AL(B@LCVA)(V\$XSGY)	'GET' INSTR OPERAND
					7666	*****		
					7667	*	'GET' STATEMENT ROUTINE CONSTANTS AND EQUATES	
					7668	*****		
					7669	*		
					7670	*	CONSTANTS	
					7671	*		
				18F0	7672	BXGSFA EQU	*	
	18F0	0001		18F1	7673	BXGBN1 DC	IL(@CADDR)'1'	BINARY 1
					7674	*		
					7675	*****,		
					7676	*		
					7677	*	END OF 'GET' STATEMENT ROUTINE CODING	
					7678	*		



## S/3 BASIC COMPILER -NEXT- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 113
		7680		*****	
		7681	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		7682	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		7683	*		*
		7684		*****	
		7685	*	*STATUS	*
		7686	*	VERSION 1 MODIFICATION 0	*
		7687	*		*
		7688	*	*FUNCTION	*
		7689	*	BKNEXT IS EXECUTED TO TRANSLATE NEXT STATEMENTS AS THEY OCCUR IN	*
		7690	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE	*
		7691	*	PSEUDOCODE IN VIRTUAL MEMORY.	*
		7692	*		*
		7693	*	*ENTRY POINTS	*
		7694	*	BKNEXT HAS ONLY ONE ENTRY POINT:	*
		7695	*	BKNEXT - TRANSLATE NEXT STATEMENT	*
		7696	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		7697	*	B BKNEXT	*
		7698	*		*
		7699	*	*INPUT	*
		7700	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		7701	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE	*
		7702	*	LEADING KEYWORD, NEXT.	*
		7703	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST	*
		7704	*	CHARACTER IN THE LEADING KEYWORD, NEXT.	*
		7705	*	* FOR TABLE - CONTAINS 4-BYTE ENTRIES. EACH CONTAINING THE	*
		7706	*	VIRTUAL ADDRESSES OF A FOR-LOOP CONTROL VARIABLE AND OF THE	*
		7707	*	NXT INSTRUCTION IN THE ASSOCIATED FOR OBJECT CODE SEQUENCE.	*
		7708	*	* B\$FTPT - CONTAINS THE CORE ADDRESS OF THE 1ST BYTE OF THE ENTRY	*
		7709	*	LAST PLACED IN THE FOR TABLE, OR OF THE BOTTOM GUARD ENTRY	*
		7710	*	WHEN THE TABLE IS EMPTY.	*
		7711	*		*
		7712	*	*OUTPUT	*
		7713	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		7714	*	GENERATED BY BKNEXT IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		7715	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		7716	*	SEQUENCES.	*
		7717	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		7718	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		7719	*	* FOR TABLE - WHEN THE CURRENT TABLE ENTRY CONTROL VARIABLE	*
		7720	*	MATCHES THAT SPECIFIED IN THE NEXT STATEMENT, THAT ENTRY IS	*
		7721	*	DELETED FROM THE TABLE. THE TABLE IS NOT AFFECTED WHEN A	*
		7722	*	COMPILER ERROR OCCURS.	*
		7723	*	* B\$FTPT - CONTAINS THE CORE ADDRESS OF THE 1ST BYTE OF THE FOR	*
		7724	*	TABLE ENTRY PRECEDING THAT DELETED FROM THE TABLE. B\$FTPT IS	*
		7725	*	NOT MODIFIED WHEN A COMPILER ERROR OCCURS.	*
		7726	*	* B\$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE	*
		7727	*	OF THE ADDRESS OPERAND FIELD IN THE NXT INSTRUCTION REFERENCED	*
		7728	*	BY THE CURRENT (BEFORE DELETION) FOR TABLE ENTRY.	*
		7729	*	* B\$NXSU - SET TO ON STATUS TO CAUSE RESOLUTION OF THE NXT	*
		7730	*	INSTRUCTION OPERAND BY THE COMPILER DISTRIBUTOR.	*
		7731	*		*
		7732	*	*EXTERNAL REFERENCES	*
		7733	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.	*
		7734	*	B\$PUTC - (B\$PFNC, B\$PCAD, B\$PNBY, B\$PERC) - ENTRY TO COMPILER*	*
		7735	*	VIRTUAL MEMORY OUTPUT ROUTINE.	*

## S/3 BASIC COMPILER -NEXT- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 114
		7736	*	B\$SYMB - (B\$BCKT) - ENTRY TO BASIC SYMBOL TRANSLATION RTN.	*			
		7737	*	B\$BTAB - (B\$BRVA) - ENTRY TO BASIC COMPILER BRANCH TABLE RTN.	*			
		7738	*	B\$FTPT - ENTRY TO FOR TABLE.	*			
		7739	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*			
		7740	*		*			
		7741	*	*EXITS, NORMAL	*			
		7742	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*			
		7743	*		*			
		7744	*	*EXITS, ERROR	*			
		7745	*	N/A	*			
		7746	*		*			
		7747	*	*TABLES/WORK AREAS	*			
		7748	*	FOR TABLE - EXTERNAL TO BKNEXT, THIS PUSH-DONN TABLE CONTAINS	*			
		7749	*	TEN 4-BYTE ENTRY LOCATIONS. THE FIRST ENTRY LOCATION IS	*			
		7750	*	ALWAYS CLEARED TO ZEROS, AND IS USED TO GUARD AGAINST A TABLE	*			
		7751	*	REFERENCE WHEN THE TABLE IS EMPTY. THE FOLLOWING NINE ENTRY	*			
		7752	*	LOCATIONS MAY EACH CONTAIN VIRTUAL ADDRESSES REFERENCING AN	*			
		7753	*	UNFINISHED FOR-LOOP CONTROL VARIABLE AND ITS ASSOCIATED NXT	*			
		7754	*	INSTRUCTION, DEPENDING ON THE CURRENT LOOP NESTING DEPTH IN THE	*			
		7755	*	PROGRAM.	*			
		7756	*		*			
		7757	*	*ATTRIBUTES	*			
		7758	*	BKNEXT IS NATURALLY RELOCATABLE AND REUSABLE.	*			
		7759	*		*			
		7760	*	*CHARACTER CODE DEPENDENCY	*			
		7761	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*			
		7762	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*			
		7763	*		*			
		7764	*	*NOTES	*			
		7765	*	ERROR PROCEDURES	*			
		7766	*	WHEN THE CURRENT NEXT CONTROL VARIABLE DOES NOT MATCH THE	*			
		7767	*	LAST FOR TABLE ENTRY THE ERROR CONDITION CODE FOR	*			
		7768	*	FOR/NEXT NESTED INCORRECTLY IS LOGGED IN VIRTUAL MEMORY.	*			
		7769	*	WHEN NO ACTIVE ENTRY EXISTS IN THE FOR TABLE THE ERROR	*			
		7770	*	CONDITION CODE FOR NEXT STATEMENT OUT OF SEQUENCE IS LOGGED	*			
		7771	*	IN VIRTUAL MEMORY.	*			
		7772	*		*			
		7773	*	REGISTER USAGE	*			
		7774	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*			
		7775	*		*			
		7776	*	SAVED/RESTORED AREAS	*			
		7777	*	N/A	*			
		7778	*		*			
		7779	*	MODIFICATION CONSIDERATIONS	*			
		7780	*	BKNEXT RESIDES ON THE SAME SECTOR WITH BMGETX AND BKGOTO.	1-4*			
		7781	*	ANY MODIFICATION TO BKNEXT WILL CHANGE THE ENTRY ADDRESSES	1-4*			
		7782	*	OF BMGETX AND BKGOTO AND MUST CONSIDER THE LIMITATION	1-4*			
		7783	*	OF THE SECTOR BOUNDARY ON SIZE.	1-4*			
		7784	*		*			
		7785	*	REQUIRED MODULES	*			
		7786	*	@SYSEQ - COMMON SYSTEM EQUATES.	*			
		7787	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*			
		7788	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*			
		7789	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*			
		7790	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*			
		7791	*	@ERMEQ - ERROR MESSAGE EQUATES.	*			

## S/3 BASIC COMPILER -NEXT- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 115
				7792	*	\$V\$EQU - FIXED VIRTUAL ADDRESSES EQUATES.			*
				7793	*	\$B\$EQU - COMPILER FIXED EQUATES.			*
				7794	*	\$B@EQU - COMPILER SYSTEM EQUATES.			*
				7795	*				*
				7796	*	OTHER			*
				7797	*	BKNEXT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			*
				7798	*	*****			*
1900				7800		ORG *,256,0			BEGIN AT CORE PAGE BOUNDARY
			1900	7801		USING *,@BR			DEFINE BASE ADDR FOR CORE PAGE
				7802	*				
				7803	*	ENTER BKNEXT - 'NEXT' STATEMENT ROUTINE			
				7804	*				
			1900	7805	BKNEXT EQU *				BKNEXT ENTRY POINT
				7806	*				
				7807	*	SET INPUT PARAMETER TO SKIP KEYWORD 'NEXT'			
				7808	*				
1900	3C	04 0873		7809	BKN010 MVI	B\$NUMC,B@LNEX			SET GET RTN TO SKIP 'NEXT'
1904	C0	87 0867		7810	B	B\$GETC			LINK TO ADVANCE POINTER
				7811	*				
				7812	*	FIND THE VIRTUAL ADDRESS OF THE 'NEXT' CONTROL VARIABLE			
				7813	*				
1908	C0	87 0DBC		7814	BKN020 B	B\$SYMB			LINK TO FIND CTRL VAR VADDR
				7815	*				
				7816	*	COMPARE 'NEXT' CTRL VAR VADDR WITH 'FOR' TABLE CTRL VAR VADDR			
				7817	*				
190C	35	02 1B0D		7818	BKN030 L	B\$FTPT,@XR			LOAD THE 'FOR' TABLE POINTER
1910	8D	01 01 1590		7819		CLC BKNFTD(,@XR),B\$BCKT(@VADDR)			IF CTRL VARIABLES MATCH
1915	F2	81 1C		7820		JE BKN090			* JUMP TO PROCESS 'BRA' PMC
				7821	*				
				7822	*	SET PUT ROUTINE FOR ERROR OUTPUT			
				7823	*				
1918	3C	33 094E		7824	BKN040 MVI	B\$PFNC,B\$PFAE			SET PUT RTN FOR ADD ERROR COND
				7825	*				
				7826	*	CHECK 'FOR TABLE' CTRL VAR FOR DUMMY ENTRY			
				7827	*				
191C	BD	00 01		7828	BKN050 CLI	BKNFTD(,@XR),BKNDUM			IF 'FOR TABLE' VADDR IS DUMMY
191F	F2	81 07		7829		JE BKN070			* JUMP TO SET PROPER ERROR CODE
				7830	*				
				7831	*	GENERATE ERROR CODE FOR UNBALANCED 'FOR'/'NEXT' CONTROL VARIABLES			
				7832	*				
1922	3C	AC 0A39		7833	BKN060 MVI	B\$PERC,@@E607			GENERATE ERROR CODE
1926	F2	87 04		7834	J	BKN080			JUMP TO LINK TO PUT RTN
				7835	*				
				7836	*	GENERATE ERROR CODE FOR 'NEXT' WITH NON-EXISTENT 'FOR'			
				7837	*				
1929	3C	AB 0A39		7838	BKN070 MVI	B\$PERC,@@E606			GENERATE ERROR CODE
192D	C0	87 093A		7839	BKN080 B	B\$PUTC			LINK TO WRITE ERROR CODE
1931	F2	87 26		7840	J	BKN120			JUMP TO BKNEXT EXIT
				7841	*				
				7842	*	ESTABLISH THE VIRTUAL ADDRESS OF THE 'FOR TABLE' NXT INSTRUCTION			
				7843	*	AS THE OPERAND OF A 'BRA' INSTRUCTION			
				7844	*				
1934	6C	01 64 03		7845	BKN090 MVC	BKNBRO(,@BR),BKNNXT(@VADDR,@XR)			SET 'BRA' OPERAND
1938	D2	02 62		7846	LA	BKNBRC(,@BR),@XR			LOAD CADDR OF 'BRA' INSTR
193B	34	02 0A40		7847	ST	B\$PCAD,@XR			SET PUT RTN FOR VADDR OF 'BRA'

## S/3 BASIC COMPILER -NEXT- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 116
193F	3C 02 0A41		7848		MVI	B\$PNBY,B@LBRA-1	SET PUT RTN FOR LENGTH OF 'BRA'
1943	C0 87 093A		7849		B	B\$PUTC	LINK TO GENERATE PMC
			7850	*			
			7851	*	DECREMENT	FOR TABLE' POINTER TO NEXT OUTER DEPTH LEVEL	
			7852	*			
1947	1F 01 1B0D 5F		7853	BKN100	SLC	B\$FTPT,BKNFEL(@CADDR,@BR)	DECREMENT FOR TABLE' POINTER
			7854	*			
			7855	*	SET PARAMETERS	FOR DISTRIBUTOR BRANCH TABLE UPDATE	
			7856	*			
194C	3A 07 071D		7857	BKN110	SBN	B\$NXSW,B\$NXMK	SET NEXT SWITCH ON
1950	1C 01 19EF 64		7858		MVC	B\$BRVA,BKNBRO(@VADDR,@BR)	MOVE VADDR OF NXT INSTR
1955	1E 01 19EF 61		7859		ALC	B\$BRVA,BKNEX2(@VADDR,@BR)	SET PARAMETER FOR 'NXT' OPND
			7860	*			
			7861	*	RETURN CONTROL	TO THE COMPILER DISTRIBUTOR	
			7862	*			
195A	C0 87 0700		7863	BKN120	B	B\$DIST	RETURN TO DISTRIBUTOR
			7865	*****			
			7866	*	'NEXT' STATEMENT ROUTINE	CONSTANTS AND EQUATES	
			7867	*****			
			7868	*			
			7869	*	EQUATES		
			7870	*			
		0001	7871	BKNFTD	EQU	1	DISP FOR 'FOR TABLE' CTRL VAR
		0000	7872	BKNDUM	EQU	0	DUMMY ENTRY COMPARISON
		0003	7873	BKNNXT	EQU	3	DISP FOR 'FOR TABLE' NXT VADDR
			7874	*			
			7875	*	CONSTANTS		
			7876	*			
195E	0004		195F	7877	BKNFEL	DC	AL(@CADDR)(B@LFRT)
1960	0002		1961	7878	BKNEX2	DC	IL(@CADDR)'2'
							BINARY 2
			7880	*****			
			7881	*	'NEXT' STATEMENT ROUTINE	PMC AND STORAGE PARAMETERS	
			7882	*****			
			7883	*			
1962	46		1962	7884	BKNBRC	DC	AL(B@LCOP)(B@CBRA)
1963			1964	7885	BKNBRO	DS	CL(@VADDR)
							'BRA' INSTR OPERAND
			7886	*			
			7887	*****			
			7888	*			
			7889	*	END OF 'NEXT' STATEMENT ROUTINE	CODING	
			7890	*			

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 117
		7892		*****			
		7893	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		7894	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		7895	*				*
		7896		*****			*
		7897	*	*STATUS			*
		7898	*	VERSION 1 MODIFICATION 0			*
		7899	*				*
		7900	*	*FUNCTION			*
		7901	*	BMGETX IS EXECUTED TO TRANSLATE MAT GET STATEMENTS IF THEY OCCUR			*
		7902	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		7903	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		7904	*				*
		7905	*	*ENTRY POINTS			*
		7906	*	BMGETX HAS ONLY ONE ENTRY POINT:			*
		7907	*	BMGETX - TRANSLATE MAT GET STATEMENT			*
		7908	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		7909	*	B BMGETX			*
		7910	*				*
		7911	*	*INPUT			*
		7912	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		7913	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE			*
		7914	*	LEADING KEYWORD, MAT GET.			*
		7915	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		7916	*	CHARACTER IN THE LEADING KEYWORD, MAT GET.			*
		7917	*				*
		7918	*	*OUTPUT			*
		7919	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		7920	*	GENERATED BY BMGETX IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		7921	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		7922	*	SEQUENCES.			*
		7923	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		7924	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		7925	*				*
		7926	*	*EXTERNAL REFERENCES			*
		7927	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		7928	*	B\$PUTC - (B\$PCAD)(B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
		7929	*	OUTPUT ROUTINE.			*
		7930	*	B\$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.			*
		7931	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		7932	*				*
		7933	*	*EXITS, NORMAL			*
		7934	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		7935	*				*
		7936	*	*EXITS, ERROR			*
		7937	*	N/A			*
		7938	*				*
		7939	*	*TAILS/WORK AREAS			*
		7940	*	N/A			*
		7941	*				*
		7942	*	*ATTRIBUTES			*
		7943	*	BNGETX IS RELOCATABLE AND REUSABLE.			*
		7944	*				*
		7945	*	*CHARACTER CODE DEPENDENCY			*
		7946	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON ANY PARTICULAR			*
		7947	*	INTERNAL REPRESENTATION UP THE EXTERNAL CHARACTER SET.			*

## S/3 BASIC COMPILER -MAT GET- STATEMENT RTN

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 118
			7948	*	*
			7949	*NOTES	*
			7950	* ERROR PROCEDURES	*
			7951	* N/A	*
			7952	*	*
			7953	* REGISTER USAGE	*
			7954	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
			7955	*	*
			7956	* SAVED/RESTORED AREAS	*
			7957	* N/A	*
			7958	*	*
			7959	* MODIFICATION CONSIPERATIPAS	*
			7960	* BMGETX RESIDES ON A SECTOR WITH BKNEXT AND BKGOTO. ANY	1-4*
			7961	* MODIFICATION TO BMGETX WILL CHANGE THE ENTRY ADDRESS OF	1-4*
			7962	* BKCOTO AND MUST CONSIDER THE LIMITATION OF THE SECTOR	1-4*
			7963	* BOUNDARY ON SIZE.	1-4*
			7964	*	*
			7965	* REQUIRED MODULES	*
			7966	* @SYSEQ - COMMON SYSTEM EQUATES.	*
			7967	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
			7968	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.	*
			7969	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
			7970	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
			7971	* @ERMEQ - ERROR MESSAGE EQUATES.	*
			7972	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
			7973	* SB\$EQU - COMPILER FIXED EQUATES.	*
			7974	* SB@EQU - COMPILER SYSTEM EQUATES.	*
			7975	*	*
			7976	* OTHER	*
			7977	* BMGETX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
			7978	*****	
			7980	*	
			7981	* ENTER BMGETX - 'MAT GET' STATEMENT	
			7982	*	
		1965	7983	BMGETX EQU *	BMGETX ENTRY POINT
			7984	*	
			7985	* SET GET ROUTINE TO SKIP TO THE CHARACTER FOLLOWING KEYWORD 'MAT GET'	
			7986	*	
1965	3C 05 0873		7987	BMG010 MVI B\$NUMC,B@LMGT-1	SET GET TO SKIP KEYWORD
1969	C0 87 0867		7988	B B\$GETC	LINK TO ADVANCE POINTER
196D	C0 87 14B0		7989	B B\$CSCN	LINK TO PROCESS FILE REFERENCE
			7990	*	
			7991	* GENERATE THE 'ADF' PMC IN V.M. (IF OPND IS ZERO, THE FILENAME IS	
			7992	* NOT IN THE ENTRY TABLE)	
			7993	*	
1971	D2 02 AC		7994	BMG100 LA BMGAFC(,@BR),@XR	LOAD CADDR OF 'ADF' INSTR
1974	34 02 0A40		7995	ST B\$PCAD,@XR	SET VADIIR PARM OF PUT FOR 'ADF'
1978	3C 01 0A41		7996	MVI B\$PNBY,B@LADF-1	SET LNG PARM, OF PUT FOR 'ADF'
197C	C0 87 093A		7997	B B\$PUTC	LINK TO GENERATE 'ADF' PMC
			7998	*	
			7999	* CALL GET ROUTINE TO REFERENCE THE NEXT VARIABLE	
			8000	*	
N04	1980 00 00 0000		8001	BMG110 MVI B\$NUMC,B\$GETS	DISABLE GET ROUTINE
	1984 C0 87 0867		8002	B B\$GETC	LINK TO GET CHARACTER POINTER
			8003	*	



ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 119

```

      8004 * CALL ROUTINE TO GENERATE DOPE VECTOR STACKING INSTRUCTIONS
      8005 *
1988 C0 87 18F3      8006 BMG120 B      B$MATR      LINK TO GENERATE PMC
198C 74 02 A1      8007      ST      BMG150+@OP1(,@BR),@XR      SAVE TEXT POINTER
      8008 *
      8009 * GENERATE THE 'MF1' INSTRUCTION IN VIRTUAL MEMORY
      8010 *
198F D2 02 AE      8011 BMG140 LA      BMGMFC(,@BR),@XR      LOAD CADDR OF 'MF1' INSTR
1992 34 02 0A40      8012      ST      B$PCAD,@XR      SET VADDR PARM OF PUT FOR 'MF1'
1996 3C 02 0A41      8013      MVI      B$PNBY,B@LMF1-1      SET LNG PARM OF PUT FOR 'MF1'
199A C0 87 093A      8014      B      B$PUTC      LINK TO GENERATE 'MF1' INSTR
      8015 *
      8016 * TEST THE DELIMITER FOR BEING AN END-OF-STATEMENT
      8017 *
199E C2 02 0000      8018 BMG150 LA      *-*,@XR      RESTORE TEXT POINTER
19A2 BD 1E 00      8019      CLI      B@CHAR(,@XR),B@EOST      IF DELIMITER IS AN EOS
19A5 D0 01 88      8020      BNE      BMG120(,@BR)      * BRANCH TO GET NEXT CHAR
      8021 *
      8022 * RETURN CONTROL TO THE COMPILER DISTRIBUTOR
      8023 *
19A8 C0 87 0700      8024 BMG160 B      B$DIST      RETURN TO DISTRIBUTOR

      8026 *****
      8027 * 'MAT GET' STATEMENT STORAGE AND PARAMETER AREA
      8028 *****
      8029 *
19AC 58      19AC 8030 BMGAFC DC      AL(B@LCOP)(B@CADF)      'ADF' INSTR OPCODE
19AD 00      19AD 8031 BMGAFO DC      XL1'00'      'ADF' INSTR OPERAND
      8032 *
N04 19AE 00      19AE 8033 BMGMFC DC      AL(B@LCOP)(B$CMF1)      'MF1' INSTR OPCODE
19AF 3E06      19B0 8034 BMGMFO DC      AL(B@LCVA)(V$XMGT)      'MF1' INSTR OPERAND

      8036 *****
      8037 * 'MAT GET' STATEMENT CONSTANTS AND EQUATES
      8038 *****
      8039 *
      8040 * CONSTANTS
      8041 *
      19B1 8042 BMGSFA EQU      *
19B1 0001      19B2 8043 BMGBN1 DC      IL(@CADDR)'1'      BINARY 1
      8044 *
      8045 *****
      8046 *
      8047 * END OF 'MAT GET' STATEMENT ROUTINE CODING
      8048 *

```



## S/3 BASIC COMPILER -GOTO- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 120
			8050		*****			
			8051	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			8052	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			8053	*				*
			8054		*****			*
			8055	*	*STATUS			*
			8056	*	VERSION 1 MODIFICATION 0			*
			8057	*				*
			8058	*	*FUNCTION			*
			8059	*	BKGOTO IS EXECUTED TO TRANSLATE SIMPLE GOTO STATEMENTS AS THEY			*
			8060	*	OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO			*
			8061	*	PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.			*
			8062	*				*
			8063	*	*ENTRY POINTS			*
			8064	*	BKGOTO HAS ONLY ONE ENTRY POINT:			*
			8065	*	BKGOTO - TRANSLATE GOTO STATEMENT			*
			8066	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			8067	*	B BKGOTO			*
			8068	*				*
			8069	*	*INPUT			*
			8070	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			8071	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			8072	*	LEADING KEYWORD, GOTO.			*
			8073	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			8074	*	CHARACTER IN THE LEADING KEYWORD, GOTO.			*
			8075	*				*
			8076	*	*OUTPUT			*
			8077	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			8078	*	GENERATE BY BKGOTO IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			8079	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			8080	*	SEQUENCES.			*
			8081	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			8082	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			8083	*				*
			8084	*	*EXTERNAL REFERENCES			*
			8085	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
			8086	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER			*
			8087	*	VIRTUAL MEMORY OUTPUT ROUTINE.			*
			8088	*	B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH			*
			8089	*	TABLE ROUTINE.			*
			8090	*	B\$ZDBN - (B\$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL			*
			8091	*	TO BINARY CONVERSION ROUTINE.			*
			8092	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			8093	*				*
			8094	*	*EXITS, NORMAL			*
			8095	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			8096	*				*
			8097	*	*EXITS, ERROR			*
			8098	*	N/A			*
			8099	*				*
			8100	*	*TABLES/WORK AREAS			*
			8101	*	N/A			*
			8102	*				*
			8103	*	*ATTRIBUTES			*
			8104	*	BKGOTO IS NATURALLY RELOCATABLE AND REUSABLE.			*
			8105	*				*

## S/3 BASIC COMPILER -GOTO- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 121

```

8106 *CHARACTER CODE DEPENDENCY *
8107 * THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR *
8108 * INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET. *
8109 * *
8110 *NOTES *
8111 * ERROR PROCEDURES *
8112 * N/A *
8113 * *
8114 * REGISTER USAGE *
8115 * BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION. *
8116 * *
8117 * SAVED/RESTORED AREAS *
8118 * N/A *
8119 * *
8120 * MODIFICATION CONSIDERATIONS *
8121 * BKGOTO RESIDES ON A SECTOR WITH BKNEXT AND BMGETX. 1-4*
8122 * ANY MODIFICATION TO BKGOTO MUST CONSIDER THIS CO-RESIDENCY 1-4*
8123 * AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE. 1-4*
8124 * *
8125 * REQUIRED MODULES *
8126 * @SYSEQ - COMMON SYSTEM EQUATES *
8127 * @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES *
8128 * @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS *
8129 * @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES *
8130 * @SPFEQ - SYSTEM PROGRAM FILE EQUATES *
8131 * @ERMEQ - ERROR MESSAGE EQUATES *
8132 * $VSEQU - FIXED VIRTUAL ADDRESS *
8133 * $B$EQU - COMPILER FIXED EQUATES *
8134 * $B@EQU - COMPILER SYSTEM EQUATES *
8135 * *
8136 * OTHER *
8137 * BKGOTO IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS *
8138 * *****

8140 *
8141 * ENTER BKGOTO - 'GOTO' STATEMENT ROUTINE
19B3 8142 *
19B3 8143 BKGOTO EQU * BKGOTO ENTRY POINT
8144 *
8145 * SET INPUT PARAMETER TO SKIP KEYWORD 'GOTO'
8146 *
19B3 3C 04 0873 8147 BKG010 MVI B$NUMC,B@LGTO SET GET RTN TO SKIP 'GOTO'
19B7 C0 87 0867 8148 B B$GETC LINK TO ADVANCE POINTER
8149 *
8150 * CONVERT THE 'GOTO' LINE NUMBER TO BINARY FROM ITS DECIMAL FORM
8151 *
19BB C0 87 19F2 8152 BKG020 B B$ZDBN LINK TO CONVERT LINE NO. TO BIN
8153 *
8154 * GENERATE A 'BRA' PMC IMAGE IN VIRTUAL MEMORY
8155 *
19BF D2 02 E7 8156 BKG030 LA BKGBRC(,@BR),@XR LOAD CADOR OF 'BRA' INSTR
19C2 34 02 0A40 8157 ST B$PCAD,@XR SET VADDR PARM FOR PUT RTN
19C6 3C 02 0A41 8158 MVI B$PNBY,B@LBRA-1 SET LENGTH PARM FOR PUT RTN
19CA C0 87 093A 8159 B B$PUTC LINK TO GENERATE PMC
8160 *
8161 * UPDATE UNRESOLVED BRANCH TABLE

```

## S/3 BASIC COMPILER -GOTO- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 122
					8162	*				
		19CE	0C 01 19F1 1A6A		8163	BKG040 MVC	B\$BRLN,B\$BINO(@VADDR)		SET BRANCH TABLE LINE NUMBER	
		19D4	0C 01 19EF 0A43		8164	MVC	B\$BRVA,B\$PVAD(@VADDR)		SET BRANCH TABLE VADDR	
N04	19DA	00 00 0000 00			8165	SLC	B\$BRVA,BKGBN1(@VADDR,@BR)		ADJUST VADDR FOR 'BRA' OPERAND	
					8166	*				
					8167	*	ESTABLISH RESOLUTION OF LINE NUMBER AND VIRTUAL ADDR IN BRANCH TABLE			
					8168	*				
	19DF	C0 87 1996			8169	BKG050 B	B\$BTAB		LINK TO WRITE BRANCH TBL ENTRY	
					8170	*				
					8171	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR			
					8172	*				
	19E3	C0 87 0700			8173	BKG060 B	B\$DIST		RETURN TO DISTRIBUTOR	
					8175	*****				
					8176	*	'GOTO' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS			
					8177	*****				
					8178	*				
	19E7	46		19E7	8179	BKGBRC DC	AL(B@LCOP)(B@CBRA)		'BRA' INSTR OPCODE	
	19E8	0000		19E9	8180	BKGBRO DC	XL(B@LCVA)'00'		'BRA' INSTR OPERAND IMAGE	
					8182	*****				
					8183	*	'GOTO' STATEMENT CONSTANTS			
					8184	*****				
					8185	*				
	19EA	0001		19EB	8186	BKGIN1 DC	IL(@VADDR)'1'		BINARY '1'	
					8187	*				
					8188	*****				
					8189	*				
					8190	*	END OF 'GOTO' STATEMENT ROUTINE CODING			
					8191	*				

## S/3 BASIC COMPILER -IF- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 123
		8193		*****	
		8194	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		8195	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		8196	*		*
		8197		*****	
		8198	*	*STATUS	*
		8199	*	VERSION 1 MODIFICATION 0	*
		8200	*		*
		8201	*	*FUNCTION	*
		8202	*	BKARIF IS EXECUTED TO TRANSLATE ARITHMETIC IF STATEMENTS AS THEY	*
		8203	*	OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO	*
		8204	*	PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.	*
		8205	*		*
		8206	*	*ENTRY POINTS	*
		8207	*	BKARIF HAS ONLY ONE ENTRY POINT:	*
		8208	*	BKARIF - TRANSLATE ARITHMETIC IF STATEMENT	*
		8209	*	THE FORMAT FOR THE CALLING SEQUENCE IS:	*
		8210	*	B BKARIF	*
		8211	*		*
		8212	*	*INPUT	*
		8213	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		8214	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE	*
		8215	*	LEADING KEYWORD, IF.	*
		8216	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST.	*
		8217	*	CHARACTER IN THE LEADING KEYWORD, IF.	*
		8218	*		*
		8219	*	*OUTPUT	*
		8220	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		8221	*	GENERATED BY BKARIF IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		8222	*	MEMORY LOCATION. FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		8223	*	SEQUENCES.	*
		8224	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		8225	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		8226	*		*
		8227	*	*EXTERNAL REFERENCES	*
		8228	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.	*
		8229	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRT	*
		8230	*	MEMORY OUTPUT ROUTINE.	*
		8231	*	B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH	*
		8232	*	TABLE ROUTINE.	*
		8233	*	B\$ZOBN - (B\$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL	*
		8234	*	TO BINARY CONVERSION ROUTINE.	*
		8235	*	B\$SCAN - ENTRY TO BASIC COMPILER ARITHMETIC EXPRESSION SCAN	*
		8236	*	ROUTINE.	*
		8237	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		8238	*		*
		8239	*	*EXITS, NORMAL	*
		8240	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		8241	*		*
		8242	*	*EXITS, ERROR	*
		8243	*	N/A	*
		8244	*		*
		8245	*	*TABLES/WORK AREAS	*
		8246	*	* RELATIONAL OPERATOR TABLE - INTERNAL TO OKARIF, THIS TABLE	*
		8247	*	CONTAINS BRC INSTRUCTION CONDITION CODES ASSOCIATED WITH EVERY	*
		8248	*	SIMPLE OR COMPOUND RELATIONAL OPERATOR. OPERATOR ENTRIES IN	*

## S/3 BASIC COMPILER -IF- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 124
				8249 *	THE TABLE CONSIST OF THE EBCDIC CHARACTER CODE FOR SIMPLE	*
				8250 *	OPERATORS AND THE SUM OF EBCDIC CHARACTER CODES FOR COMPOUND	*
				8251 *	OPERATORS.	*
				8252 *	* RELATIONAL OPERATOR BUCKET - INTERNAL TO BKARIF, THIS 1-BYTE	*
				8253 *	FIELD IS USED TO STORE SIMPLE AND COMPOUND RELATIONAL OPERATOR	*
				8254 *	CHARACTERS FOR ASSOCIATION WITH A RELATIONAL OPERATOR TABLE	*
				8255 *	ENTRY.	*
				8256 *		*
				8257 *	*ATTRIBUTES	*
				8258 *	BKARIF IS NATURALLY RELOCATABLE AND REUSABLE.	*
				8259 *		*
				8260 *	*CHARACTER CODE DEPENDENCY	*
				8261 *	THE OPERATION OF THIS MODULE DEPENDS UPON AN INTERNAL REPRRESEN-	*
				8262 *	TATION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE	*
				8263 *	ONE USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT	*
				8264 *	REDEFINITION OF CHARACTER CONSTANTS. BY REASSEMBLY, WILL RESULT	*
				8265 *	IN A CORRECT MODULE FOR THE NEW DEFINITIONS.	*
				8266 *		*
				8267 *	*NOTES	*
				8268 *	ERROR PROCEDURES	*
				8269 *	N/A	*
				8270 *		*
				8271 *	REGISTER USAGE	*
				8272 *	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				8273 *		*
				8274 *	SAVED/RESTORED AREAS	*
				8275 *	N/A	*
				8276 *		*
				8277 *	MODIFICATION CONSIDERATIONS	*
				8278 *	BKARIF RESIDES ON A SECTOR WITH BMDPRT. ANY MODIFICATION	1-4*
				8279 *	TO BKARIF WILL CHANGE THE ENTRY ADDRESS OF BMDPRT AND	1-4*
				8280 *	MUST TAKE INTO CONSIDERATION THE LIMITATION OF THE SECTOR	1-4*
				8281 *	BOUNDARY ON SIZE.	1-4*
				8282 *		*
				8283 *	REQUIRED MODULES	*
				8284 *	@SYSEQ - COMMON SYSTEM EQUATES	*
				8285 *	@FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUES EQUATES	*
				8286 *	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES	*
				8287 *	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
				8288 *	@SPFEQ - SYSTEM PROGRAM FILE EQUATES	*
				8289 *	@ERNEQ - ERROR MESSAGE EQUATES	*
				8290 *	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES	*
				8291 *	\$B\$EQU - COMPILER FIXED EQUATES	*
				8292 *		*
				8293 *	OTHER	*
				8294 *	BKARIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
				8295 *	*****	*
				8296 *		*
1A00				8297	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
	1A00			8298	USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
				8299 *		
				8300 *	ENTER BKARIF - ARITHMETIC IF STATEMENT ROUTINE	
				8301 *		
	1A00			8302	BKARIF EQU *	BKARIF ENTRY POINT
				8303 *		
				8304 *	SET INPUT PARAMETER TO SKIP 'I' IN KEYWORD 'IF' TO REFERENCE THE	

## S/3 BASIC COMPILER -IF- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 125

```

      8305 * CHARACTER PRECEDING THE FIRST ARITHMETIC EXPRESSION
      8306 *
1A00 3C 01 0873      8307 BKA010 MVI   B$NUMC,B@LKIF-1      SET GET RTN TO SKIP 'I' IN IF.
1A04 C0 87 0867      8308         B    B$GETC              LINK TO ADVANCE POINTER
      8309 *
      8310 * BRANCH TO SCAN ROUTINE TO GENERATE 'STF' INSTR
      8311 *
1A08 C0 87 1514      8312 BKA020 B      B$SCAN              LINK TO GENERATE 'STF' PMC
      8313 *
      8314 * STORE THE FIRST RELATIONAL OPERATOR IN THE OPERAND OF A CLI INSTR.
      8315 *
1A0C 6C 00 32 00      8316 BKA030 MVC   BKA090+@Q(,@BR),B@CHAR(1,@XR)  STORE 1ST RELATIONAL OPTR
      8317 *
      8318 * GET NEXT CHARACTER TO CHECK IF COMPOUND OPERATOR IS INDICATED
      8319 *
N04 1A10 00 00 0000      8320 BKA040 B      B$GFC              LINK TO GET NEXT CHARACTER
1A14 BD 7E 00          8321         CLI   B@CHAR(,@XR),B@EQL      IF CHAR IS '='
1A17 F2 81 0D          8322         JE    BKA060              * GO COMPUTE OPERATOR
1A1A BD 6E 00          8323         CLI   B@CHAR(,@XR),B@GRTR     IF CHAR IS '>'
1A1D F2 81 07          8324         JE    BKA060              * GO COMPUTE OPERATOR
      8325 *
      8326 * IF NO SECOND RELATIONAL OPERATOR DISABLE BAGETC TO KEEP THE TEXT
      8327 * POINTER IN PLACE
      8328 *
1A20 3C 00 0873      8329 BKA050 MVI   B$NUMC,B@GETS      DISABLE GET ROUTINE
1A24 F2 87 04          8330         J     BKA070              GO SEARCH OPERATOR TABLE
      8331 *
      8332 * IF RELATIONAL OPERATOR IS COMPOUND ADD CURRENTLY REFERENCED CHARACTER
      8333 * TO THE CONTENTS OF THE OPERATOR OPERAND TO DEKIVE A CHARACTER CODE
      8334 *
1A27 6E 00 32 00      8335 BKA060 ALC   BKA090+@Q(,@BR),B@CHAR(1,@XR)  ADD TO GET CHAR CODE
      8336 *
      8337 * SEARCH RELATIONAL OPERATOR TABLE FOR THE CONDITION CODE THAT MATCHES
      8338 * THE CHARACTER CODE IN THE OPERATOR BUUKET-EITHER SIMPLE OR COMPOUND
      8339 *
N04 1A2B 00 00 00      8340 BKA070 LA     BKA0TB(,@BR),@XR      LOAD TABLE BASE ADM IN XR
1A2E E2 02 02          8341 BKA080 LA     BKALTH(,@XR),@XR      ADD LENGTH TO ADDR IN XR
N04 1A31 00 00 00      8342 BKA090 CLI   BKAODI(,@XR),*-*      IF TEXT OPERATOR  TABLE ENTRY
1A34 D0 01 2E          8343         BNE   BKA080(,@BR)        * FALL THROUGH
      8344 *
      8345 * STORE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTRUCTION IMAGE
      8346 *
N04 1A37 00 00 00 00      8347 BKA100 MVC   BKAB02(,@BR),BKA0D2(,@XR)  SET 'BRC' COND CODE OPERAND
      8348 *
      8349 * GO TO ARITHMETIC SCAN ROUTINE TO GENERATE PMC FOR THE SECOND
      8350 * ARITHMETIC EXPRESSION
      8351 *
1A3B 35 02 0878      8352 BKA110 L      B$GPTR,@XR      RESTORE TEXT POINTER
1A3F C0 87 1514      8353         B      B$SCAN              LINK TO GENERATE PMC
      8354 *
      8355 * SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN' TO ADVANCE
      8356 * THE TEXT POINTER TO THE LINE NUMBER
      8357 *
N04 1A43 00 00 0000      8358 BKA120 MVI   B$NUNC,B@LTHN-1      SET GET RTN TO SKIP KEYWORD
1A47 C0 87 0867      8359         B      B$GETC              LINK TO ADVANCE POINTER
      8360 *

```



## S/3 BASIC COMPILER -IF- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 126
					8361	*	CONVERT THE 'GOTO' LINE NUMBER TO BINARY FROM DECIMAL			
					8362	*				
	1A4B	C0	87	19F2	8363	BKA130 B	B\$ZDBN			LINK TO CONVERT LINE NUMBER
					8364	*				
					8365	*	GENERATE A COMPARE FLOATING POINT VALUE PMC IN VIRTUAL MEMORY			
					8366	*				
	1A4F	D2	02	86	8367	BKA140 LA	BKACMC(,@BR),@XR			LOAD CADDR OF 'CMF' INSTR
N04	1A52	00	00	0000	8368		ST B\$PCAP,@XR			SET PUT RTN FOR VADDR OF 'CMF'
	1A56	3C	00	0A41	8369		MVI B\$PNBY,B@LCMF-1			SET PUT RTN FOR LENGTH OF 'CMF'
	1A5A	C0	87	093A	8370		B B\$PUTC			LINK TO GENERATE 'CMF' INSTK
					8371	*				
					8372	*	GENERATE BRANCH ON CONDITION INSTR IN VIRTUAL MEMORY			
					8373	*				
	1A5E	D2	02	87	8374	BKA150 LA	BKABRC(,@BR),@XR			LOAD CADDR OF 'BRC' INSTR
	1A61	34	02	0A40	8375		ST B\$PCAD,@XR			SET PUT RTN FOR VADDR OF 'BRC'
	1A65	3C	03	0A41	8376		MVI B\$PNBY,B@LBRC-1			SET PUT RTN FOR LENGTH OF 'BRC'
	1A69	C0	87	093A	8377		B B\$PUTC			UNK TO GENERATE 'BRC' INSTR
					8378	*				
					8379	*	ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR BRANCH TABLE			
					8380	*	RESOLUTION ROUTINE			
					8381	*				
	1A6D	0C	01	19EF 0A43	8382	BKA160 MVC	B\$BRVA,B\$PVAD(@VADDR)			SET ADDR PARAMETER
	1A73	1F	01	19EF 8C	8383		SLC B\$BRVA,BKALNG(@VADDR,@BR)			* TO ADDRESS BRANCH VADDR
	1A78	0C	01	19F1 1A6A	8384		MVC B\$BRLN,B\$BINO(B@LCLN)			SET LINE NO PARAMETER
	1A7E	C0	87	1996	8385		B B\$BTAB			LINK TO WRITE BRANCH TAT ENTRY
					8386	*				
					8387	*	RETURN CONTROL TO THE DISTRIBUTOR			
					8388	*				
	1A82	C0	87	0700	8389	BKA170 B	B\$DIST			RETURN TO DISTRIBUTOR



## S/3 BASIC COMPILER -IF- STATEMENT ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 127

```

8391 *****
8392 * ARITHMETIC 'IF' ROUTINE PMC AND STORAGE PARAMETERS
8393 *****
8394 *
1A86 40      1A86 8395 BKACMC DC    AL(B@LCOP)(B@CCMF)      COMPARE FLOATING VALUES OPCODE
1A87 44      1A87 8396 BKABRC DC    AL(B@LCOP)(B@CBRC)      BRANCH ON CONDITION OPCODE
1A88 0000    1A89 8397 BKAB01 DC    XL(B@LCVA)'00'          BRANCH ON CONDITION VABOR OPND
1A8A        1A8A 8398 BKAB02 DS    CL(B@LCCC)              BRANCH ON COND COMO CODE OPND

8400 *****
8401 * ARITHMETIC 'IF' ROUTINE CONSTANTS
8402 *****
8403 *
1A8B 0002    1A8C 8404 BKALNG DC    AL(@VADDR)(B@LCCC+1)    LENGTH OF CONDITION CODE + 1

8406 *****
8407 * RELATIONAL OPERATOR - CONDITION CODE TABLE
8408 *****
8409 *
1A8D        1A8D 8410 BKATAB EQU    *                      START OF CODE TABLE
0000        0000 8411 BKAOD1 EQU    0                      DISP FOR TABLE OPERATOR
0001        0001 8412 BKAOD2 EQU    1                      DISP FOR TABLE COND CODE
0002        0002 8413 BKALTH EQU    2                      LENGTH OF TABLE ENTRY
1A8B        1A8B 8414 BKAOT1 EQU    BKATAB-BKALTH          CODE TABLE BASE ADDRESS
8415 *
1A8D 7E      1A8D 8416          DC    AL1(B@EQL)            RELATIONAL OPERATOR - '='
1A8E 84      1A8E 8417          DC    AL1(B@BREQ)          BRANCH CONDITION - EQUAL
8418 *
1A8F 6E      1A8F 8419          DC    AL1(B@GRTR)          RELATIONAL OPERATOR - '>'
1A90 88      1A90 8420          DC    AL1(B@BRHI)          BRANCH CONDITION - HIGH
8421 *
1A91 4C      1A91 8422          DC    AL1(B@LESS)          RELATIONAL OPERATOR - '<'
1A92 82      1A92 8423          DC    AL1(B@BRLO)          BRANCH CONDITION - LOW
8424 *
1A93 BA      1A93 8425          DC    AL1(B@LESS+B@GRTR)    RELATIONAL OPERATOR - '><'
1A94 94      1A94 8426          DC    AL1(B@BRNE)          BRANCH CONDITION - NOT EQUAL
8427 *
N04 1A95 00  1A95 8428          DC    AL1(B@LESS+B@EQL)    RELATIONAL OPERATOR - '<='
1A96 98      1A96 8429          DC    AL1(B@BRNH)          BRANCH CONDITION - NOT HIGH
8430 *
N04 1A97 00  1A97 8431          DC    AL1(B@GRTR+B@EQL)    RELATIONAL OPERATOR - '>='
1A98 92      1A98 8432          DC    AL1(B@BRNL)          BRANCH CONDITION - NOT LOW
8433 *
1A99 5F      1A99 8434          DC    AL1(B@NEQL)          RELATIONAL OPERATOR - ''
1A9A 94      1A9A 8435          DC    AL1(B@BRNE)          BRANCH CONDITION - NOT EQUAL
8436 *
8437 *****
8438 *
8439 * END OF ARITHMETIC IF ROUTINE CODING
8440 *

```

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 128
		8442		*****			*
		8443	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		8444	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		8445	*				*
		8446		*****			*
		8447	*	*STATUS			*
		8448	*	VERSION 1 MODIFICATION 0			*
		8449	*				*
		8450	*	*FUNCTION			*
		8451	*	BMDPRT IS EXECUTED TO TRANSLATE MAT PRINT STATEMENTS AS THEY OCCUR			*
		8452	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		8453	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		8454	*				*
		8455	*	*ENTRY POINTS			*
		8456	*	BMDPRT HAS ONLY ONE ENTRY POINT:			*
		8457	*	BMDPRT - TRANSLATE MAT PRINT STATEMENT			*
		8458	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		8459	*	B BMDPRT			*
		8460	*				*
		8461	*	*INPUT			*
		8462	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		8463	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		8464	*	LEADING KEYWORD, MAT PRINT.			*
		8465	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		8466	*	CHARACTER IN THE LEADING KEYWORD, MAT PRINT.			*
		8467	*				*
		8468	*	*OUTPUT			*
		8469	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		8470	*	GENERATED BY BMDPRT IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		8471	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		8472	*	SEQUENCES.			*
		8473	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		8474	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		8475	*				*
		8476	*	*EXTERNAL REFERENCES			*
		8477	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL RTN.			*
		8478	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRT TO COMPILER VIRTUAL MEMORY			*
		8479	*	OUTPUT ROUTINE.			*
		8480	*	B\$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.			*
		8481	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8482	*				*
		8483	*	*EXITS, NORMAL			*
		8484	*	B@DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8485	*				*
		8486	*	*EXITS, ERROR			*
		8487	*	N/A			*
		8488	*				*
		8489	*	*TABLES/WORK AREAS			*
		8490	*	N/A			*
		8491	*				*
		8492	*	*ATTRIBUTES			*
		8493	*	BMDPRT IS NATURALLY RELOCATABLE AND REUSABLE.			*
		8494	*				*
		8495	*	*CHARACTER CODE DEPENDENCY			*
		8496	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
		8497	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 129
					8498	*				*
					8499	*NOTES				*
					8500	* ERROR PROCEDURES				*
					8501	* N/A				*
					8502	*				*
					8503	* REGISTER USAGE				*
					8504	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
					8505	*				*
					8506	* SAVED/RESTORED AREAS				*
					8507	* N/A				*
					8508	*				*
					8509	* MODIFICATION CONSIDERATIONS				*
					8510	* BADPRT RESIDES ON A SECTOR WITH BKARIF. ANY MODIFICATION				1-4*
					8511	* TO RMDPRT MUST TAKE INTO CONSIDERATION THIS CO-RESIDENCY				1-4*
					8512	* AND THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.				1-4*
					8513	*				*
					8514	* REQUIRED MODULES				*
					8515	* @SYSEQ - COMMON JESTER EQUATES.				*
					8516	* @FXDEQ - SYSTEM NUCLEUS ADDRESS AND INDICATOR VALUES EQUATES.				*
					8517	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.				*
					8518	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
					8519	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
					8520	* @ERMEQ - ERROR MESSAGE EQUATES.				*
					8521	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.				*
					8522	* \$B\$EQU - COMPILER FIXED EQUATES.				*
					8523	* \$B@EQU - COMPILER SYSTEM EQUATES.				*
					8524	*				*
					8525	* OTHER				*
					8526	* BMDPRT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS				*
					8527	*****				*
					8529	*				*
					8530	* ENTER BMDPRT - MAT PRINT STATEMENT ROUTINE				*
					8531	*				*
		1A9B			8532	BMDPRT EQU *	BMDPRT ENTRY POINT			*
					8533	*				*
					8534	* SET GET ROUTINE TO SKIP TO CHAR FOLLOWING KEYWORDS 'MAT PRINT'				*
					8535	*				*
		1A9B	3C	08	0873	8536	BMD010 MVI B\$NUMC,B@LMPR	SET GET TO SKIP 'MAT PRINT'		*
		1A9F	C0	87	0867	8537	B B\$GETC	LINK TO ADVANCE POINTER		*
					8538	*				*
					8539	* DISABLE GET RTN BEFORE CALLING THE MATRIX REFERENCE PROCESSOR				*
					8540	*				*
		1AA3	3C	00	0873	8541	BMD020 MVI B\$NUMC,B@GETS	DISABLE GET RTN NOT TO GET CHAR		*
		1AA7	C0	87	18F3	8542	B B\$MATR	LINK TO PROCESS MAT-REFERENCE		*
					8543	*				*
					8544	* TEST DELIMITER FOR BEING A SEMI-COLON (INDICATING SHORT FORM)				*
					8545	*				*
		1AAB	BD	5E	00	8546	BMD030 CLI B@CHAR(,@XR),B@SCLN	IF CHAR IS NOT SEMI-COLON		*
		1AAE	F2	01	12	8547	JNE BMD050	* GO GENERATE 'MF1' FOR LONG FORM		*
					8548	*				*
					8549	* GENERATE AN 'MF1' INSTR FOR SHORT FORM				*
					8550	*				*
		1AB1	D2	02	EA	8551	BMD040 LA BMDM1C(,@BR),@XR	LOAD CADDR OF 'MF1' INSTR		*
		1AB4	34	02	0A40	8552	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR 'MF1'		*
N04	1AB8	00	00	0000	8553	MVI B\$PNBY,BELMF1-1	SET LNG PARM OF PUT FOR 'MF1'			*

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 130
	1ABC	C0	87	093A	8554	B	B\$PUTC			LINK TO GENERATE 'MF1' INSTR
	1AC0	F2	87	19	8555	J	BMD060			GO GET NEXT CHARACTER
					8556	*				
					8557	*	GENERATE AN 'MF1' INSTR FOR LONG FORM			
					8558	*				
	1AC3	D2	02	ED	8559	BMD050 LA	BMDM2C(,@BR),@XR			LOAD CADDR OF 'MF1' INSTR
	1AC6	34	02	0A40	8560	ST	B\$PCAD,@XR			SET VADDR PARM OF PUT FOR 'MF1'
	1ACA	3C	02	0A41	8561	MVI	B\$PNBY,B@LMF1-1			SET LNG PARM OF PUT FOR 'MF1'
	1ACE	C0	87	093A	8562	B	B\$PUTC			LINK TO GENERATE 'MF1' INSTR
					8563	*				
					8564	*	TEST DELIMITER FOR BEING A STATEMENT TERMINATOR			
					8565	*				
	1AD2	35	02	0878	8566	BMD055 L	B\$GPTR,@XR			RESTORE TEXT POINTER
	1AD6	BD	1E	00	8567	CLI	B@CHAR(,@XR),B@EOST			IF DELIMITER IS AN EOS
	1AD9	D0	81	E6	8568	BE	BMD080(,@BR)			* RETURN CONTROL TO DIST
					8569	*				
					8570	*	CALL GET ROUTINE TO GET NEXT CHARACTER			
					8571	*				
	1ADC	C0	87	0867	8572	BMD060 B	B\$GETC			LINK TO GET NEXT CHAR
					8573	*				
					8574	*	TEST DELIMITER FOR BEING A STATEMENT TERMINATOR			
					8575	*				
	1AE0	BD	1E	00	8576	BMD070 CLI	B@CHAR(,@XR),B@EOST			IF DELIMITER IS NOT AN EOS
	1AE3	D0	01	A3	8577	BNE	BMD020(,@BR)			* GO PROCESS NEXT LIST ELEMENT
					8578	*				
					8579	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR			
					8580	*				
	1AE6	C0	87	0700	8581	BMD080 B	B\$DIST			RETURN TO DISTRIBUTOR
					8583	*****				
					8584	*	MAT PRINT STATEMENT ROUTINE STORAGE AND PARAMETER AREA			
					8585	*****				
					8586	*				
	1AEA	18			1AEA 8587	BMDM1C DC	AL(B@LCOP)(B@CMF1)			'MF1' INSTR OPCODE
	1AEB	3F00			1AEC 8588	BMDM10 DC	AL(B@LCVA)(V\$XMPS)			'MF1' INSTR OPND - SHORT FORM
					8589	*				
	1AED	18			1AED 8590	BMDM2C DC	AL(B@LCOP)(B@CMF1)			'MF1' INSTR OPCODE
	1AEE	3F06			1AEF 8591	BMDM20 DC	AL(B@LCVA)(V\$XMPL)			'MF1' INSTR OPND - LONG FORM
					8592	*				
					8593	*****				
					8594	*				
					8595	*	END OF 'MAT PRINT' STATEMENT ROUTINE CODING			
					8596	*				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 131
		8598		*****			
		8599	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		8600	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		8601	*				*
		8602		*****			*
		8603	*	STATUS			*
		8604	*	VERSION 1 MODIFICATION 0			*
		8605	*				*
		8606	*	FUNCTION			*
		8607	*	BKCRIF IS EXECUTED TO TRANSLATE CHARACTER IF STATEMENTS AS THEY			*
		8608	*	OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO			*
		8609	*	PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		8610	*				*
		8611	*	ENTRY POINTS			*
		8612	*	BKCRIF HAS ONLY ONE ENTRY POINT			*
		8613	*	BKCRIF - TRANSLATE CHARACTER IF STATEMENT			*
		8614	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		8615	*	B BKCRIF			*
		8616	*				*
		8617	*	INPUT			*
		8618	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		8619	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		8620	*	LEADING KEYWORD, IF.			*
		8621	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		8622	*	CHARACTER IN THE LEADING KEYWORD, IF.			*
		8623	*				*
		8624	*	OUTPUT			*
		8625	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		8626	*	GENERATED BY BKCRIF IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		8627	*	MEMORY LOCATION, FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		8628	*	SEQUENCES.			*
		8629	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		8630	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		8631	*				*
		8632	*	EXTERNAL REFERENCES			*
		8633	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE			*
		8634	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRT			*
		8635	*	MEMORY OUTPUT ROUTINE.			*
		8636	*	B\$BTAB - (B\$BRVA, B\$BRIN) - ENTRY TO BASIC COMPILER BRANCH			*
		8637	*	TABLE ROUTINE.			*
		8638	*	B\$ZDBN - (B\$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO			*
		8639	*	BINARY CONVERSION ROUTINE.			*
		8640	*	B\$CSCN - ENTRY TO BASIC COMPILER CHARACTER SCAN ROUTINE			*
		8641	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8642	*				*
		8643	*	EXITS, NORMAL			*
		8644	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8645	*				*
		8646	*	EXITS, ERROR			*
		8647	*	N/A			*
		8648	*				*
		8649	*	TABLES/WORK AREAS			*
		8650	*	* RELATIONAL OPERATOR TABLE - INTERNAL TO BKCRIF, THIS TABLE			*
		8651	*	CONTAINS 'BRC' INSTRUCTION CONDITION CODES ASSOCIATED WITH			*
		8652	*	EVERY SIMPLE OR COMPOUND RELATIONAL OPERATOR. OPERATOR ENTRIES			*
		8653	*	IN THE TABLE CONSIST OF THE EBCDIC CHARACTER CODE FOR SIMPLE			*

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 132
		8654	*	OPERATORS AND THE SUM OF EBCDIC CHARACTER CODES FOR COMPOUND	*
		8655	*	OPERATORS.	*
		8656	*	* RELATIONAL OPERATOR BUCKET - INTERNAL TO BKCRIF, THIS 1-BYTE	*
		8657	*	FIELD IS USED TO STORE SIMPLE AND COMPOUND RELATIONAL OPERATOR	*
		8658	*	CHARACTERS FOR ASSOCIATION WITH A RELATIONAL OPERATOR TABLE	*
		8659	*	ENTRY.	*
		8660	*		*
		8661	*	*ATTRIBUTES	*
		8662	*	BKCRIF IS NATURALLY RELOCATABLE AND REUSABLE.	*
		8663	*		*
		8664	*	*CHARACTER CODE DEPENDENCY	*
		8665	*	THE OPERATION OF THIS MODULE DEPENDS UPON AS INTERNAL REPRESENTA-	*
		8666	*	TION OF THE EXTERNAL CHARACTER SET WHICH IS EQUIVALENT TO THE	*
		8667	*	ONE USED AT ASSEMBLY TIME. THE CODING HAS BEEN ARRANGED SO THAT	*
		8668	*	REDEFINITION OF CHARACTER CONSTANTS, BY REASSEMBLY, WILL RESULT	*
		8669	*	IN A CORRECT MODULE FOR THE NEW DEFINITIONS.	*
		8670	*		*
		8671	*	*NOTES	*
		8672	*	ERROR PROCEDURES	*
		8673	*	N/A	*
		8674	*		*
		8675	*	REGISTER USAGE	*
		8676	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
		8677	*		*
		8678	*	SAVED/RESTORED AREAS	*
		8679	*	N/A	*
		8680	*		*
		8681	*	MODIFICATION CONSIDERATIONS	*
		8682	*	BKCRIF RESIDES ON A SECTOR WITH BMPUTX. ANY MODIFICATION	1-4*
		8683	*	TO BKCRIF SHOULD CONSIDER THIS CO-RESIDENCY SINCE IT WILL	1-4*
		8684	*	CHANGE THE ENTRY ADDRESS OF BMPUTX. THE SIZE LIMITATION	1-4*
		8685	*	OF THE SECTOR BOUNDARY MUST ALSO BE CONSIDERED.	*
		8686	*		*
		8687	*	REQUIRED MODULES	*
		8688	*	@SYSEQ - COMMON SYSTEM EQUATES.	*
		8689	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
		8690	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.	*
		8691	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
		8692	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
		8693	*	@ERMEQ - ERROR MESSAGE EQUATES.	*
		8694	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
		8695	*	\$B\$EQU - COMPILER FIXED EQUATES.	*
		8696	*	\$B@EQU - COMPILER SYSTEM EQUATES.	*
		8697	*		*
		8698	*	OTHER	*
		8699	*	BKCRIF IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
		8700	*	*****	*
		8701	*		*
1B00		8702		ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
	1B00	8703		USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
		8704	*		
		8705	*	ENTER BKCRIF - CHARACTER 'IF' STATEMENT PROCESSOR	
		8706	*		
	1B00	8707		BKCRIF EQU *	BKCRIF ENTRY POINT
		8708	*		
		8709	*	SKIP PAST 'I' IN KEYWORD 'IF' TO REFERENCE CHARACTER PRECEDING THE	



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 133
					8710	*	FIRST EXPRESSION CHARACTER			
					8711	*				
	1B00	3C	01	0873	8712	BKC010	MVI B\$NUMC,B@LKIF-1			SET PARAMETER TO SKIP 'I' IN IF
	1B04	C0	87	0867	8713		B B\$GETC			LINK TO ADVANCE POINTER
					8714	*				
					8715	*	GENERATE PNC FOR FIRST CHARACTER EXPRESSION			
					8716	*				
	1B08	C0	87	14B0	8717	BKC020	B B\$CSCN			LINK TO GENERATE PMC
					8718	*				
					8719	*	STORE FIRST RELATIONAL OPERATOR CHARACTER IN OPERAND OF CLI INSTR.			
					8720	*				
	1B0C	6C	00	32 00	8721	BKC030	MVC BKC090+@Q(, @BR), B@CHAR(1, @XR)			STORE 1ST RELATIONAL OPTR
					8722	*				
					8723	*	GET NEXT CHARACTER TO CHECK IF COMPOLND OPERATOR IS INDICATED			
					8724	*				
	1B10	C0	87	0867	8725	BKC040	B B\$GETC			LINK TO GET NEXT CHARACTER
	1B14	BD	7E	00	8726		CLI B@CHAR(, @XR), B@EQL			IF CHAR IS '='
	1B17	F2	81	0D	8727		JE BKC060			* GO COMPUTE OPERATOR
N04	1B1A	00	00	00	8728		CLI B@CHAR(, @XR), B\$GRTR			IF CHAR IS '>'
	1B1D	F2	81	07	8729		JE BKC060			* GO COMPUTE OPERATOR
					8730	*				
					8731	*	IF RELATIONAL CPERATOR IS NOT COMPOUND DISABLE BAGETC TO KEEP TEXT			
					8732	*	POINTER STATIONARY			
					8733	*				
	1B20	3C	00	0873	8734	BKC050	MVI B\$NUMC,B@GETS			DISABLE GET RTN FOR NEXT CHAR
	1B24	F2	87	04	8735		J BKC070			GO SEARCH OPERATOR TABLE
					8736	*				
					8737	*	IF RELATIONAL OPERATOR IS COMPOUND ADD CURRENTLY REFERENCED CHARACTER			
					8738	*	TO THE CONTENTS OF THE OPERATOR BUCKET TO DERIVE A CHARACTER CODE			
					8739	*				
	1B27	6E	00	32 00	8740	BKC060	ALC BKC090+@Q(, @BR), B@CHAR(1, @XR)			ADD TO GET CHAR CODE
					8741	*				
					8742	*	SEARCH THE RELATIONAL OPERATOR TABLE FOR THE CONDITION CODE THAT			
					8743	*	MATCHES THE CHARACTER CODE IN THE OPERATOR BUCKET-EITHER SIMPLE OR			
					8744	*	COMPOUND			
					8745	*				
	1B2B	D2	02	8B	8746	BKC070	LA BKCOTB(, @BR), @XR			LOAD TABLE BASE ADOR IN OR
	1B2E	E2	02	02	8747	BKC080	LA BKCLTH(, @XR), @XR			ADD LENGTH TO ADDR IN XR
	1B31	BD	00	00	8748	BKC090	CLI BKC0D1(, @XR), *-*			IF TEXT OPERATOR = TABLE ENTRY
	1B34	D0	01	2E	8749		BNE BKC080(, @BR)			* FALL THROUGH
					8750	*				
					8751	*	STORE CONDITION CODE IN OPERAND FIELD OF 'BRC' INSTRUCTION IMAGE			
					8752	*				
	1B37	6C	00	8A 01	8753	BKC100	MVC BKCB02(, @BR), BKCCD2(, @XR)			SET 'BRC' CORD CODE OPERAND
					8754	*				
					8755	*	GOTO CHARACTER SCAN ROUTINE TO GENERATE PMC FOR THE SECOND CHARACTER			
					8756	*	EXPRESSION			
					8757	*				
	1B3B	35	02	0878	8758	BKC110	L B\$GPTR, @XR			RESTORE TEXT POINTER
	1B3F	C0	87	14B0	8759		B B\$CSCN			LINK TO GENERATE PMC
					8760	*				
					8761	*	SET PARAMETER TO SKIP EMBEDDED KEYWORD 'GOTO' OR 'THEN' TO ADVANCE			
					8762	*	THE TEXT POINTER TO THE LINE NUMBER			
					8763	*				
	1B43	3C	04	0873	8764	BKC120	MVI B\$NUMC,B@LTHN			SET GET RTN TO SKIP KEYWORD
	1B47	C0	87	0867	8765		B B\$GETC			LINK TO ADVANCE POINTER



ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 134
					8766	*		
					8767	*	CONVERT THE 'GOTO' LINE NUMBER TO BINARY RION DECIMAL	
					8768	*		
N04	1B4B	00	00	0000	8769	BKC130 B	B\$ZDON LINK TO CONVERT LINE NUMBER	
					8770	*		
					8771	*	GENERATE A COMPARE CHARACTER PMC IN VIRTUAL MEMORY	
					8772	*		
	1B4F	D2	02	86	8773	BKC140 LA	BKCCMC(,@BR),@XR LOAD CADDR OF 'CMC' INSTR	
	1B52	34	02	0A40	8774	ST	B\$PCAD,@XR SET PUT RTN FOR VADDR OF 'CMC'	
	1B56	3C	00	0A41	8775	MVI	B\$PNBY,B@LCMC-1 SET PUT RTN FOR LENGTH OF 'CMC'	
	1B5A	C0	87	093A	8776	B	B\$PUTC LINK TO GENERATE PMC	
					8777	*		
					8778	*	GENERATE BRANCH ON CONDITION INSTRUCTION IMAGE IN VIRTUAL MEMORY	
					8779	*		
	1B5E	D2	02	87	8780	BKC150 LA	BKCBRC(,@BR),@XR LOAD CADDR OF 'BRC' INSTR	
	1B61	34	02	0A40	8781	ST	B\$PCAD,@XR SET PUT RTN FOR VADDR OF 'BRC'	
	1B65	3C	03	0A41	8782	MVI	B\$PNBY,B@LBRC-1 SET PUT RTN FOR LENGTH OF 'BRC'	
	1B69	C0	87	093A	8783	B	B\$PUTC LINK TO GENERATE 'BRC' INSTR	
					8784	*		
					8785	*	ESTABLISH ADDRESS AND LINE NUMBER PARAMETERS FOR BRANCH TABLE	
					8786	*	RESOLUTION ROUTINE	
					8787	*		
	1B6D	0C	01	19EF 0A43	8788	BKC160 MVC	B\$BRVA,B\$PVAD(@VADDR) SET ADDR PARAMETER	
	1B73	1F	01	19EF 8C	8789	SLC	B\$BRVA,BKCLNG(@VADDR,@BR) SET PARAMETER FOR VADDR OF BRC	
N04	1B78	00	00	0000 0000	8790	MVC	B\$BRLN,B\$BINO(B@LCIN) SET LINE NO PARAMETER	
	1B7E	C0	87	1996	8791	B	B\$BTAB LINK TO SET RESOLUTION COND	
					8792	*		
					8793	*	RETURN CONTROL TO THE DISTRIBUTOR	
					8794	*		
	1B82	C0	87	0700	8795	B	B\$DIST RETURN TO DISTRIBUTOR	
					8797	*****		
					8798	*	CHARACTER IF ROUTINE PMC AND STORAGE PARAMETERS	
					8799	*****		
					8800	*		
	1B86	42			1B86	8801 BKCCMC DC	AL(B@LCOP)(B@CCMC) COMPARE CHAR OPCODE	
					8802	*		
	1B87	44			1B87	8803 BKCBRC DC	AL(B@LCOP)(B@CBRC) BRANCH ON CONDITION OPCODE	
	1B88	0000			1B89	8804 BKCB01 DC	XL(B@LCVA)'00' BRANCH ON CORD VADDR OPERAND	
	1B8A				1B8A	8805 BKCB02 DS	CL(B@LCCC) BRANCH ON COND COND CODE OPND	
					8807	*****		
					8808	*	CHARACTER IF ROUTINE CONSTANTS	
					8809	*****		
					8810	*		
	1B8B	0002			1B8C	8811 BKCLNG DC	AL(@VADDR)(B@LCCC+1) LENGTH OF CONDITION CODE + 1	
					8813	*****		
					8814	*	RELATIONAL OPERATOR - CONDITION CODE TABLE	
					8815	*****		
					8816	*		
					1B8D	8817 BKCTAB EQU	* START OF CODE TABLE	
	0000				8818	BKCOD1 EQU	0 DISP FOR TABLE OPERATOR	
	0001				8819	BKCCD2 EQU	1 DISP FOR TABLE COND CODE	
	0002				8820	BKCLTH EQU	2 LENGTH OF TABLE ENTRY	
	1B8B				8821	BKCOTB EQU	BKCTAB-BKCLTH CODE TABLE BASE ADDRESS	

ERR LOC		OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20		PAGE 135
				8822	*				
	1B8D	7E	1B8D	8823	DC	AL1(B@EQL)			RELATIONAL OPERATOR '='
	1B8E	84	1B8E	8824	DC	AL1(B@BRQ)			BRANCH CONDITION - EQUAL
				8825	*				
	1B8F	6E	1B8F	8826	DC	AL1(B@GRTR)			RELATIONAL OPERATOR '>'
	1B90	88	1B90	8827	DC	AL1(B@BRHI)			BRANCH CONDITION - HI
				8828	*				
	1B91	4C	1B91	8829	DC	AL1(B@LESS)			RELATIONAL OPERATOR '<'
	1B92	82	1B92	8830	DC	AL1(B@BRLO)			BRANCH CONDITION - LOW
				8831	*				
N04	1B93	00	1B93	8832	DC	AL1(B@LESS+B@GRIR)			RELATIONAL OPERATOR '<>'
	1B94	94	1B94	8833	DC	AL1(B@BRNE)			BRANCH CONDITION - NOT EQUAL
				8834	*				
	1B95	CA	1B95	8835	DC	AL1(B@LESS+B@EQL)			RELATIONAL OPERATOR '<='
	1B96	98	1B96	8836	DC	AL1(B@BRNH)			BRANCH CONDITION - NOT HIGH
				8837	*				
	1B97	EC	1B97	8838	DC	AL1(B@GRTR+B@EQL)			RELATIONAL OPERATOR '>='
	1B98	92	1B98	8839	DC	AL1(B@BRNL)			BRANCH CONDITION - NOT LOW
				8840	*				
	1B99	5F	1B99	8841	DC	AL1(B@NEQL)			RELATIONAL OPERATOR ''
	1B9A	94	1B9A	8842	DC	AL1(B@BRNE)			BRANCH CONDITION - NOT EQUAL
				8843	*				
				8844	*****				
				8845	*				
				8846	*	END OF 'CHAR IF' ROUTINE CODING			
				8847	*				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 136
		8849		*****			
		8850	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		8851	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		8852	*				*
		8853		*****			*
		8854	*	STATUS			*
		8855	*	VERSION 1 MODIFICATION 0			*
		8856	*				*
		8857	*	FUNCTION			*
		8858	*	BMPUTX IS EXECUTED TO TRANSLATE MAT PUT STATEMENTS AS THEY OCCUR			*
		8859	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		8860	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		8861	*				*
		8862	*	ENTRY POINTS			*
		8863	*	BMPUTX HAS ONLY ONE ENTRY POINT:			*
		8864	*	BMPUTX - TRANSLATE MAT PUT STATEMENT			*
		8865	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
		8866	*	B BMPUTX			*
		8867	*				*
		8868	*	INPUT			*
		8869	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		8870	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
		8871	*	LEADING KEYWORD. MAT PUT.			*
		8872	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		8873	*	CHARACTER IN THE LEADING KEYWORD. MAT PUT.			*
		8874	*				*
		8875	*	OUTPUT			*
		8876	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		8877	*	GENERATED BY BMPUTX IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		8878	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		8879	*	SEQUENCES.			*
		8880	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		8881	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		8882	*				*
		8883	*	EXTERNAL REFERENCES			*
		8884	*	B\$GETU - (B\$NUNC) - ENTRY TO BASIC RETRIEVAL ROUTINE.			*
		8885	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
		8886	*	ROUTINE.			*
		8887	*	B\$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE			*
		8888	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8889	*				*
		8890	*	EXITS, NORMAL			*
		8891	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
		8892	*				*
		8893	*	EXITS, ERROR			*
		8894	*	N/A			*
		8895	*				*
		8896	*	TABLES/WORK AREAS			*
		8897	*	N/A			*
		8898	*				*
		8899	*	ATTRIBUTES			*
		8900	*	BMPUTX IS NATURALLY RELOCATABLE AND REUSABLE.			*
		8901	*				*
		8902	*	CHARACTER CODE DEPENDENCY			*
		8903	*	THE OPERATION OF THIS NODULE DOES NOT DEPEND ON A PARTICULAR			*
		8904	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 137
					8905	*				*
					8906	*NOTES				*
					8907	* ERROR PROCEDURES				*
					8908	* N/A				*
					8909	*				*
					8910	* REGISTER USAGE				*
					8911	* BOTH THE INNS AND BASE REGISTERS ARE USED DURING EXECUTION.				*
					8912	*				*
					8913	* SAVED/RESTORED AREAS				*
					8914	* N/A				*
					8915	*				*
					8916	* MODIFICATION CONSIDERATIONS				*
					8917	* BMPUTX RESIDES ON A SECTOR WITH IKCRIF. ANY MODIFICATION	1-4*			
					8918	* TO BMPUTX SHOULD CONSIDER THIS CO-RESIDENCY AND TAKE INTO	1-4*			
					8919	* CONSIDERATION THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE.	1-4*			
					8920	*				*
					8921	* REQUIRED MODULES				*
					8922	* @SYSEQ - COMMON SYSTEM EQUATES.				*
					8923	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.				*
					8924	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
					8925	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
					8926	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
					8927	* @ERMEQ - ERROR MESSAGE EQUATES.				*
					8928	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.				*
					8929	* \$B\$EQU - COMPILER FIXED EQUATES.				*
					8930	* \$B@EQU - COMPILER SYSTEM EQUATES.				*
					8931	*				*
					8932	* OTHER				*
					8933	* BMPUTX IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
					8934	*****				
					8936	*				
					8937	* ENTER BMPUTX - MAT PUT STATEMENT ROUTINE				
					8938	*				
				1B9B	8939	BMPUTX EQU *	BMPUTX ENTRY POINT			
					8940	*				
					8941	* SET GET ROUTINE TO SKIP TO THE CHARACTER FOLLOWING KEYWORDS .MAT PUT				
					8942	*				
1B9B	3C	05	0873		8943	BMP010 MVI B\$NUMC,B@LMPT-1	SET GET TO SKIP KEYWORD			
1B9F	C0	87	0867		8944	B B\$GETC	LINK TO ADVANCE POINTER			
1BA3	C0	87	14B0		8945	B B\$CSCN	LINK TO PROCESS FILE REFERENCE			
					8946	*				
					8947	* GENERATE THE 'ADF' PMC IN VIRT. MEM. (IF OPERAND IS ZERO, THE FILE				
					8948	* IS NOT IN ENTRY TABLE)				
					8949	*				
1BA7	D2	02	E2		8950	BMP100 LA BMPAFC(,@BR),@XR	LOAD CADDR OF 'ADF' INSTR			
1BAA	34	02	0A40		8951	ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR AVE			
1BAE	3C	01	0A41		8952	MVI B\$PNBY,B@LADF-1	SET LNG PARM OF PUT FOR 'ADF'			
1BB2	C0	87	093A		8953	B B\$PUTC	LINK TO GENERATE 'ADF' INSTR			
					8954	*				
					8955	* CALL GET ROUTINE TO GET NEXT CHAR				
					8956	*				
1BB6	3C	00	0873		8957	BMP110 MVI B\$NUMC,B@GETS	DISABLE GET ROUTINE			
1BBA	C0	87	0867		8958	B B\$GETC	LINK TO GET CHARACTER POINTER			
					8959	*				
					8960	* CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING				

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 138
					8961	*	INSTRUCTIONS			
					8962	*				
	1BBE	C0	87 18F3		8963	BMP120 B	B\$MATR		LINK TO PROCESS MAT-REFERENCE	
	1BC2	74	02 D7		8964	ST	BMP140+@OP1(,@BR),@XR		SAVE TEXT POINTER	
					8965	*				
					8966	*	GENERATE THE 'MF1' INSTR IN VIRTAL MEMORY.			
					8967	*				
	1BC5	D2	02 E4		8968	BMP130 LA	BMPMFC(,@BR),@XR		LOAD CADDR OF 'MF1' INSTR	
	1BC8	34	02 0A40		8969	ST	B\$PCAD,@XR		SET VADDR PARM OF PUT FOR 'MF1'	
	1BCC	3C	02 0A41		8970	MVI	B\$PNBY,B@LMF1-1		SET LNG PARM OF PUT FOR 'MF1'	
	1BD0	C0	87 093A		8971	B	B\$PUTC		LINK TO GENERATE 'MF1' INSTR	
					8972	*				
					8973	*	TEST THE DELIMITER FOR BEING A STATEMENT TERMINATOR			
					8974	*				
	1BD4	C2	02 0000		8975	BMP140 LA	*-*,@XR		RESTORE TEXT POINTER	
	1BD8	BD	1E 00		8976	CLI	B@CHAR(,@XR),B@EOST		IF DELIMITER IS NOT EOS	
	1BDB	D0	01 BE		8977	BNE	BMP120(,@BR)		* GO PROCESS NEXT MAT-REFERENCE	
					8978	*				
					8979	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR			
					8980	*				
	1BDE	C0	87 0700		8981	BMP150 B	B\$DIST		RETURN TO DISTRIBUTER	
					8983	*****				
					8984	*	MAT PUT STATEMENT ROUTINE PARAMETER AND STORAGE AREAS			
					8985	*****				
					8986	*				
	1BE2	58		1BE2	8987	BMPAFC DC	AL(B@LCOP)(B@CADF)		'ADF' INSTR OPCODE	
	1BE3	01		1BE3	8988	BMPAFO DC	XL1'01'		'ADF' INSTR OPERAND	
					8989	*				
	1BE4	18		1BE4	8990	BMPMFC DC	AL(B@LCOP)(B@CMF1)		'MF1' INSTR OPCODE	
	1BE5	3E0C		1BE6	8991	BMPMFO DC	AL(B@LCVA)(V\$XMPT)		'MF1' INSTR OPND - PUT	
					8993	*****				
					8994	*	MAT PUT STATEMENT CONSTANTS AND EQUATES			
					8995	*****				
					8996	*				
				1BE7	8997	BMPSFA EQU	*			
					8998	*				
	1BE7	0001		1BE8	8999	BMPBN1 DC	IL(@CADDR)'1'		BINARY 1	
					9000	*				
					9001	*****				
					9002	*				
					9003	*	END OF 'MAT PUT' STATEMENT ROUTINE CODING			
					9004	*				

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 139
		9006		*****	
		9007	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		9008	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		9009	*		*
		9010		*****	
		9011	*	*STATUS	*
		9012	*	VERSION 1 MODIFICATION 0	*
		9013	*		*
		9014	*	*FUNCTION	*
		9015	*	BKMGTO IS EXECUTED TO TRANSLATE MULTIPLE GOTO STATEMENTS AS THEY	*
		9016	*	OCCUR IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO	*
		9017	*	PLACE THE PSEUDOCODE INTO VIRTUAL MEMORY.	*
		9018	*		*
		9019	*	*ENTRY POINTS	*
		9020	*	BKMGTO HAS ONLY ONE ENTRY POINT:	*
		9021	*	BKMGTO - TRANSLATE MULTIPLE GOTO STATEMENT	*
		9022	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		9023	*	B BKMGTO	*
		9024	*		*
		9025	*	*INPUT	*
		9026	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		9027	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE	*
		9028	*	LEADING KEYWORD, GOTO.	*
		9029	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST	*
		9030	*	CHARACTER IN THE LEADING KEYWORD, GOTO.	*
		9031	*		*
		9032	*	*OUTPUT	*
		9033	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		9034	*	GENERATED BY BKMGTO IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		9035	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		9036	*	SEQUENCES.	*
		9037	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		9038	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		9039	*	* B\$BRVA - CONTAINS THE VIRTUAL ADDRESS OF THE RIGHT BYTE OF	*
		9040	*	THE ADDRESS OPERAND FIELD IN THE EXCEPTION BYPASS ADDRESS	*
		9041	*	STACKING INSTRUCTION.	*
		9042	*	* B\$NXSW - SET TO ON STATUS TO CAUSE RESOLUTION OF THE EXCEPTION	*
		9043	*	BYPASS ADDRESS STACKING INSTRUCTION OPERAND.	*
		9044	*		*
		9045	*	*EXTERNAL REFERENCES	*
		9046	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTRY TO BASIC RETRIEVAL ROUTINE.	*
		9047	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRT	*
		9048	*	MEMORY OUTPUT ROUTINE.	*
		9049	*	B\$SCAN - ENTRY TO BASIC ARITHMETIC EXPRESSION SCAN ROUTINE.	*
		9050	*	B\$BTAB - (B\$BRVA, B\$BRLN) - ENTRY TO BASIC COMPILER BRANCH	*
		9051	*	TABLE ROUTINE.	*
		9052	*	B\$ZDBN - (B\$BINO) - ENTRY TO BASIC COMPILER ZONED DECIMAL TO	*
		9053	*	BINARY CONVERSION ROUTINE.	*
		9054	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR	*
		9055	*		*
		9056	*	*EXITS, NORMAL	*
		9057	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR	*
		9058	*		*
		9059	*	*EXITS, ERROR	*
		9060	*	N/A	*
		9061	*		*



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 140
			9062	*TABLES/WORK AREAS	*
			9063	* N/A	*
			9064	*	*
			9065	*ATTRIBUTES	*
			9066	* BKMGT0 IS NATURALLY RELOCATABLE AND REUSABLE	*
			9067	*	*
			9068	*CHARACTER CODE DEPENDENCY	*
			9069	* THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
			9070	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
			9071	*	*
			9072	*NOTES	*
			9073	* ERROR PROCEDURES	*
			9074	* N/A	*
			9075	*	*
			9076	* REGISTER USAGE	*
			9077	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
			9078	*	*
			9079	* SAVED/RESTORED AREAS	*
			9080	* N/A	*
			9081	*	*
			9082	* MODIFICATION CONSIDERATIONS	*
			9083	* BKMGT0 RESIDES ON THE SAME SECTOR WITH BXRSET AND BTPAUS.	1-4*
			9084	* AND MODIFICATION TO BKMGT0 SHOULD TAKE INTO CONSIDERATION	1-4*
			9085	* THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESSES	1-4*
			9086	* OF BXRSET AND BTPAUS AND MUST TAKE INTO CONSIDERATION THE	1-4*
			9087	* LIMITATION OF THE SECTOR BOUNDARY ON SIZE.	1-4*
			9088	*	*
			9089	* REQUIRED MODULES	*
			9090	* @SYSEQ - COMMON SYSTEM EQUATES	*
			9091	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES	*
			9092	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS	*
			9093	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES	*
			9094	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES	*
			9095	* @ERMEQ - ERROR MESSAGE EQUATES	*
			9096	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES	*
			9097	* \$B\$EQU - COMPILER FIXED EQUATES	*
			9098	* \$B@EQU - COMPILER SYSTEM EQUATES	*
			9099	*	*
			9100	* OTHER	*
			9101	* BKMGT0 IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
			9102	*****	
1C00			9103	ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY 1-4
		1C00	9104	USING *,@BR	DEFINE BASE ADDR FOR CORE PG 1-4
			9105	*	
			9106	* ENTER BKMGT0 - MULTIPLE 'GOTO' STATEMENT ROUTINE	
			9107	*	
		1C00	9108	BKMGT0 EQU *	BKMGT0 ENTRY POINT
			9109	*	
			9110	* SET INPUT PARAMETER TO SKIP KEYWORD 'GOTO'.	
			9111	*	
1C00 3C 04 0873			9112	BKM010 MVI B\$NUMC,B@LGTO	SET GET RTN TO SKIP 'GOTO'
1C04 C0 87 0867			9113	B B\$GETC	LINK TO ADVANCE POINTER
			9114	*	
			9115	* GENERATE AN 'STA' INSTRUCTION IMAGE PMC IN VIRTUAL MEMORY	
			9116	*	
N04 1C08 00 00 00			9117	BKM020 LA BKMSTC(,@BR),@XR	LOAD CADDR OF 'STA' INSTR



## S/3 BASIC COMPILER -MULT GOTO- STATEMENT RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 141

1C0B	34	02	0A40	9118	ST	B\$PCAD,@XR	SET PUT RTN FOR VADDR OF 'STA'
1C0F	3C	02	0A41	9119	MVI	B\$PNBY,B@LSTA-1	SET PUT RTN FOR LENGTH OF 'STA'
1C13	C0	87	093A	9120	B	B\$PUTC	LINK TO GENERATE PMC
				9121	*		
				9122	*	SAVE THE VADDS FOLLOWING THE OPERAND OF THE 'STA' PMC	
				9123	*		
1C17	4C	01	A5 0A43	9124	BKM030 MVC	BKMVAD(,@BR),B\$PVAD(@VADDR)	SAVE VADDR TO RESOLVE 'STA'
				9125	*		
				9126	*	CONVERT A LIST LINE NUMBER TO BINARY FROM DECIMAL	
				9127	*		
1C1C	35	02	0878	9128	BKM035 L	B\$GPTR,@XR	RESTORE TEXT POINTER
1C20	7C	00	A1	9129	MVI	BKMCSO(,@BR),@ZERO	INITLZ LINE NO. COUNT TO ZERO
1C23	C0	87	19F2	9130	BKM040 B	B\$ZDBN	CONVERT LIST LN NO TO BINARY
				9131	*		
				9132	*	GENERATE AN 'STA' INSTRUCTION PMC IN VIRTUAL MEMORY	
				9133	*		
N04 1C27	00	00	00	9134	BKM050 LA	BKMSTC(,@BR),@XR	LOAD CADDR OF 'STA' INSTR
1C2A	34	02	0A40	9135	ST	B\$PCAD,@XR	SET PUT RTN FOR VADDR OF 'STA'
1C2E	3C	02	0A41	9136	MVI	B\$PNBY,B@LSTA-1	SET PUT RTN FOR LENGTH OF 'STA'
1C32	C0	87	093A	9137	B	B\$PUTC	LINK TO GENERATE 'STA' PMC
				9138	*		
				9139	*	ESTABLISH THE CURRENT 'STA' OPERAND FOR ADDRESS RESOLUTION	
				9140	*		
1C36	0C	01	19EF 0A43	9141	BKM060 MVC	B\$BRVA,B\$PVAD(@VADDR)	SET VADDR PARAMETER FOR BR TBL
1C3C	1F	01	19EF A3	9142	SLC	B\$BRVA,BKMBN1(@VADDR,@BR)	ADJUST VADDR TO 'STA' OPND
				9143	*		
				9144	*	ESTABLISH THE LIST LINE NUMBER AS THE RESOLUTION LINE NUMBER	
				9145	*		
1C41	0C	01	19F1 1A6A	9146	BKM070 MVC	B\$BRLN,B\$BINO(@VADDR)	SET LN NO PARAMETER FOR BR TBL
1C47	C0	87	1996	9147	B	B\$BTAB	LINK TO RESOLVE *STA' OPND
				9148	*		
				9149	*	INCREMENT CURRENT LIST LINE NUMBER COUNT BY ONE	
				9150	*		
1C4B	5E	01	A1 A3	9151	BKM080 ALC	BKMCSO(,@BR),BKMBN1(@VADDR,@BR)	INCREMENT LK NO COUNT
				9152	*		
				9153	*	CHECK FOR THE END OF THE LINE NUMBER LIST	
				9154	*		
1C4F	35	02	0878	9155	BKM090 L	B\$GPTR,@XR	RESTORE TEXT POINTER
1C53	BD	6B	00	9156	CLI	B@CHAR(,@XR),B@CMMA	IF LINE NUMBER LIST AT END
1C56	F2	01	07	9157	JNE	BKM100	* JUMP TO PROCESS ARITH EXPR
1C59	C0	87	0867	9158	B	B\$GETC	LINK TO GET NEXT CHAR
1C5D	D0	87	60	9159	B	BKM100(,@BR)	BRANCH TO PROCESS NEXT LN NO
				9160	*		
				9161	*	SET INPUT PARAMETER TO SKIP TO 'N' IN KEYWORD 'ON'	
				9162	*		
1C60	3C	01	0873	9163	BKM100 MVI	B\$NUMC,B@LKON-1	SET GET RTN TO SKIP 'O' IN 'ON'
1C64	C0	87	0867	9164	B	B\$GETC	LINK TO ADVANCE POINTER
				9165	*		
				9166	*	CALL ARITH SCAN RTN TO GENERATE PMC FOR ARITH EXPRESSION	
				9167	*		
1C68	C0	87	1514	9168	BKM110 B	B\$SCAN	LINK TO SCAN ARITH EXPRESSION
				9169	*		
				9170	*	GENERATE A 'CSA' INSTRUCTION WITH LIST LINE NO COUNT AS OPERAND	
				9171	*		
1C6C	D2	02	A0	9172	RKM120 LA	BKMCSC(,@BR),@XR	LOAD CADDR OF 'CSA' INSTR
1C6F	34	02	0A40	9173	ST	B\$PCAD,@XR	SET PUT RTN FOR VADDR OF 'CSA'

## S/3 BASIC COMPILER -MULT GOTO- STATEMENT RTN

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 142

N04 1C73 00 00 0000 9174 MVI DOPNBY,ISLCSA-1 SET PUT RTN FOR LENGTH OF 'CSA'  
1C77 C0 87 093A 9175 B B\$PUTC LINK TO GENERATE 'CSA' PMC  
9176 \*  
9177 \* GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY  
9178 \*  
N04 1C7B 00 00 00 9179 BKM125 LA BKMBRC(,@BR),@XR LOAD CADDR OF 'BRS' INSTR  
1C7E 34 02 0A40 9180 ST B\$PCAD,@XR SET VADDR PARM OF PUT FOR BRS  
1C82 3C 00 0A41 9181 MVI B\$PNBY,B@LBRS-1 SET LNG PARM OF PUT FOR 'BRS'  
1C86 C0 87 093A 9182 B B\$PUTC LINK TO GENERATE 'BRS' INSTR  
9183 \*  
9184 \* ESTABLISH THE VADDR OF THE FIRST 'STA' INSTR AS THE BRANCH ADDRESS  
9185 \* TABLE RESOLUTION ADDRESS  
9186 \*  
1C8A 1C 01 19EF A5 9187 BKM130 MVC B\$BRVA,BKMBAD(@VADDR,@BR) SET VADDR PARAMETER FOR BR TBL  
1C8F 1F 01 19EF A3 9188 SLC B\$BRVA,BKMBN1(@VADDR,@BR) ADJUST VADOR FOR 'STA' OPERAND  
9189 \*  
9190 \* SET 'NEXT' SW FOR RESOLUTION OF 'STA' OPERAND WITH NEXT IN NO  
9191 \*  
1C94 3A 07 071D 9192 BKM140 SBN B\$NXSW,B\$NXMK SET 'NEXT' SW TO RESOLVE LN NO  
9193 \*  
9194 \* RETURN CONTROL TO THE COMPILER DISTRIBUTOR  
9195 \*  
1C98 C0 87 0700 9196 BKM150 B B\$DIST RETURN TO DISTRIBUTOR  
9198 \*\*\*\*\*  
9199 \* MULTIPLE 'GOTO' STATEMENT ROUTINE PMC STORAGE AND PARAMETERS  
9200 \*\*\*\*\*  
9201 \*  
1C9C 34 1C9C 9202 BKKSTC DC AL(B@LCOP)(B@CSTA) 'STA' INSTR IMAGE OPCODE  
1C9D 0000 1C9E 9203 BKMSTO DC XL(B@LCVA)'00' 'STA' INSTR OPERAND IMAGE  
9204 \*  
1C9F 4C 1C9F 9205 BKKBRC DC AL(B@LCOP)(B@CBRS) 'BRS' INSTR OPCODE  
9206 \*  
1CA0 3E 1CA0 9207 BKMCSO DC AL(B@LCOP)(B@CCSA) 'CSA' INSTR OPCODE  
1CA1 1CA1 9208 BKMCSO DS CL(B@LCNN) 'CSA' OPND - LIST LN NO COUNT  
9210 \*\*\*\*\*  
9211 \* MULTIPLE 'GOTO' STATEMENT ROUTINE CONSTANTS  
9212 \*\*\*\*\*  
9213 \*  
1CA2 0001 1CA3 9214 BKMBN1 DC IL(B@LCVA)'1' BINARY 1  
1CA4 1CA5 9215 BKMVAD DS CL(@VADDR) VADDR FOLLOWING 'STA' OPERAND  
9217 \*\*\*\*\*  
9218 \*  
9219 \* END OF MULTIPLE 'GOTO' STATEMENT ROUTINE CODING  
9220 \*

## S/3 BASIC COMPILER -RESET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 143
			9222		*****			
			9223	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			9224	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			9225	*				*
			9226		*****			*
			9227	*	STATUS			*
			9228	*	VERSION 1 MODIFICATION 0			*
			9229	*				*
			9230	*	FUNCTION			*
			9231	*	BXRSET IS EXECUTED TO TRANSLATE RESET STATEMENTS AS THEY OCCUR			*
			9232	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
			9233	*	THE PSEUDOCODE INTO VIRTUAL MEMORY.			*
			9234	*				*
			9235	*	ENTRY POINTS			*
			9236	*	BXRSET HAS ONLY ONE ENTRY POINT:			*
			9237	*	BXRSET - TRANSLATE RESET STATEMENT			*
			9238	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			9239	*	B BXRSET			*
			9240	*				*
			9241	*	INPUT			*
			9242	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			9243	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			9244	*	LEADING KEYWORD, RESET.			*
			9245	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			9246	*	CHARACTER IN THE LEADING KEYWORD. RESET.			*
			9247	*				*
			9248	*	OUTPUT			*
			9249	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			9250	*	GENERATED BY BXRSET IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			9251	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			9252	*	SEQUENCES.			*
			9253	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			9254	*	* CHARACTER WHICH TERMINATES THE STATEMENT.			*
			9255	*				*
			9256	*	EXTERNAL REFERENCES			*
			9257	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC TEXT RETRIEVAL ROUTINE.			*
			9258	*	B\$PUTC - (B\$PCAD) - B\$PNBY) - ENTRY TO COMPILER VIRT MEMORY			*
			9259	*	OUTPUT ROUTINE.			*
			9260	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			9261	*				*
			9262	*	EXITS, NORMAL			*
			9263	*	B\$DIST - ENTRY TO THE BASIC COMPILER DISTRIBUTOR			*
			9264	*				*
			9265	*	EXITS, ERROR			*
			9266	*	N/A			*
			9267	*				*
			9268	*	TABLES/WORK AREAS			*
			9269	*	N/A			*
			9270	*				*
			9271	*	ATTRIBUTES			*
			9272	*	* BXRSET IS NATURALLY RELOCATABLE AND REUSABLE.			*
			9273	*				*
			9274	*	CHARACTER CODE DEPENDENCY			*
			9275	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
			9276	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
			9277	*				*

## S/3 BASIC COMPILER -RESET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 144
				9278	*	NOTES			*
				9279	*	ERROR PROCEDURES			*
				9280	*	N/A			*
				9281	*				*
				9282	*	REGISTER USAGE			*
				9283	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.			*
				9284	*				*
				9285	*	SAVED/RESTORED AREAS			*
				9286	*	N/A			*
				9287	*				*
				9288	*	MODIFICATION CONSIDERATIONS			*
				9289	*	BXRSET RESIDES ON THE SAME SECTOR WITH BKMGT0 AND BTPAUS.	1-4	*	
				9290	*	ANY MODIFICATION TO BXRSET MUST CONSTER THIS CO-RESIDENCY	1-4	*	
				9291	*	SINCE WILL CHANGE THE ENTRY ADDRESS OF BTPAUS. THE	1-4	*	
				9292	*	LIMITATION OF THE SECTOR BOUNDARY ON SIZE MUST ALSO BE	1-4	*	
				9293	*	CONSIDERID.	1-4	*	
				9294	*				*
				9295	*	REQUIRED MODULES			*
				9296	*	@SYSEQ - COMMON SYSTEM EQUATES			*
				9297	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES			*
				9298	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS			*
				9299	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES			*
				9300	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES			*
				9301	*	@ERMEQ - ERROR MESSAGE EQUATES			*
				9302	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES			*
				9303	*	\$B\$EQU - COMPILER FIXED EQUATES			*
				9304	*	\$B@EQU - COMPILER SYSTEM EQUATES			*
				9305	*				*
				9306	*	NOTES			*
				9307	*	BXRSET IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			*
				9308	*	*****			*
				9310	*				*
				9311	*	ENTER BXRSET - 'RESET' STATEMENT ROUTINE			*
				9312	*				*
			1CA6	9313	BXRSET EQU	*	BXRSET ENTRY POINT		*
				9314	*				*
				9315	*	SET POINTER TO SKIP TO 'T' IN KEYWORD 'RESET'			*
				9316	*				*
1CA6	3C	04	0873	9317	BXR010	MVI B\$NUMC,B@LKRT-1	SET GET RTN TO SKIP TO 'T'		*
1CAA	C0	87	0867	9318		B B\$GETC	LINK TO ADVANCE POINTER		*
1CAE	C0	87	14B0	9319	BXR020	B B\$CSCN	LINK TO PROCESS FILE REFERENCE		*
				9320	*				*
				9321	*	GENERATE THE 'ADF' PMC IN V.M. IF OPERAND IS NOT ZERO			*
				9322	*				*
1CB2	D2	02	E2	9323	BXR110	LA BXRAFC(,@BR),@XR	LOAD CADDR OF 'ADF' INSTR		*
1CB5	34	02	0A40	9324		ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR ADF		*
1CB9	3C	01	0A41	9325		MVI B\$PNBY,B@LADF-1	SET LNG PARM OF PUT FOR 'ADP'		*
1CBD	C0	87	093A	9326		B B\$PUTC	LINK TO GENERATE 'ADF' PMC		*
				9327	*				*
				9328	*	GENERATE THE 'RST' PMC IN V.M.			*
				9329	*				*
1CC1	D2	02	E4	9330	BXR120	LA BXRRTC(,@BR),@XR	LOAD CADDR OF 'RST' INSTR		*
1CC4	34	02	0A40	9331		ST B\$PCAD,@XR	SET VADDR PARM OF PUT FOR RST		*
1CC8	3C	00	0A41	9332		MVI B\$PNBY,B@LRST-1	SET LNG PARM OF PUT FOR 'RST'		*
1CCC	C0	87	093A	9333		B B\$PUTC	LINK TO GENERATE 'RST' PMC		*

## S/3 BASIC COMPILER -RESET- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00 20/07/20 PAGE 145
				9334	*		
				9335	*	TEST NEXT LIST CHARACTER FOR BEING AN END-OF-STATEMENT	
				9336	*		
1CD0	3C	00 0873		9337	BXR130 MVI	B\$NUMC,B@GETS	DISABLE GET ROUTINE
1CD4	C0	87 0867		9338	B	B\$GETC	LINK TO GET CHARACTER POINTER
1CD8	BD	1E 00		9339	CLI	B@CHAR(,@XR),B@EOST	IF CHAR IS EOS
1CDB	D0	01 AE		9340	BNE	BXR020(,@BR)	* BRANCH TO PROCESS FILENAME
				9341	*		
				9342	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR	
				9343	*		
1CDE	C0	87 0700		9344	BXR140 B	B\$DIST	RETURN TO DISTRIBUTOR
				9345	*		
				9346	*****		
				9347	*	'RESET' STATEMENT PARAMETER AND STORAGE AREAS	
				9348	*****		
				9349	*		
1CE2	58		1CE2	9350	BXRAFC DC	AL(B@LCOP)(B@CADF)	'ADF' INSTR OPCODE
1CE3	00		1CE3	9351	BXRAFO DC	XL1'00'	'ADF' INSTR OPERAND
				9352	*		
1CE4	5C		1CE4	9353	BXRRTC DC	AL(B@LCOP)(B@CRST)	'RST' INSTR OPCODE
				9355	*****		
				9356	*	'RESET' STATEMENT CONSTANTS AND EQUATES	
				9357	*****		
				9358	*		
				9359	*	CONSTANTS	
				9360	*		
			1CE5	9361	BXRSFA EQU	*	
				9362	*		
1CE5	0001		1CE6	9363	BXRBNI DC	IL(@CADDR)'1'	BINARY +1
				9364	*		
				9365	*****		
				9366	*		
				9367	*	END OF 'RESET' STATEMENT ROUTINE CODING	
				9368	*		

## S/3 BASIC COMPILER -PAUSE- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 146
		9370		*****	
		9371	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		9372	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		9373	*		*
		9374		*****	
		9375	*	*STATUS	*
		9376	*	VERSION 1 MODIFICATION 0	*
		9377	*		*
		9378	*	*FUNCTION	*
		9379	*	BTPAUS IS EXECUTED TO TRANSLATE PAUSE STATEMENTS AS THEY OCCUR IN	*
		9380	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE	*
		9381	*	PSEUDOCODE IN VIRTUAL MEMORY.	*
		9382	*		*
		9383	*	*ENTRY POINTS	*
		9384	*	BTPAUS HAS ONLY ONE ENTRY POINT:	*
		9385	*	BTPAUS - TRANSLATE PAUSE STATEMENT	*
		9386	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		9387	*	B BTPAUS	*
		9388	*		*
		9389	*	*INPUT	*
		9390	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		9391	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE	*
		9392	*	LEADING KEYWORD, PAUSE.	*
		9393	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST	*
		9394	*	CHARACTER IN THE LEADING KEYWORD, PAUSE.	*
		9395	*		*
		9396	*	*OUTPUT	*
		9397	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		9398	*	GENERATED BY BTPAUS IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		9399	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		9400	*	SEQUENCES.	*
		9401	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		9402	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		9403	*		*
		9404	*	*EXTERNAL REFERENCES	*
		9405	*	B\$PUTC(B\$PCAD.B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY	*
		9406	*	OUTPUT.	*
		9407	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.	*
		9408	*		*
		9409	*	*EXITS, NORMAL	*
		9410	*	BMW - ENTRY TO BASIC COMPILER REMARK ROUTINE.	*
		9411	*		*
		9412	*	*EXITS, ERROR	*
		9413	*	N/A	*
		9414	*		*
		9415	*	*TABLES/WORK AREAS	*
		9416	*	N/A	*
		9417	*		*
		9418	*	*ATTRIBUTES	*
		9419	*	BTPAUS IS NATURALLY RELOCATABLE AND REUASBLE.	*
		9420	*		*
		9421	*	*CHARACTER CODE DEPENDENCY	*
		9422	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		9423	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		9424	*		*
		9425	*	*NOTES	*



## S/3 BASIC COMPILER -PAUSE- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 147
				9426	*	ERROR PROCEDURES			*
				9427	*	N/A			*
				9428	*				*
				9429	*	REGISTER USAGE			*
				9430	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.			*
				9431	*				*
				9432	*	SAVED/RESTORED AREAS			*
				9433	*	N/A			*
				9434	*				*
				9435	*	MODIFICATION CONSIDERATIONS			*
				9436	*	BTPAUS RESIDES ON THE SAME SECTOR WITH BKMGT0 AND BXRSET.			1-4*
				9437	*	ANY MODIFICATION OF BTPAUS MUST TAKE INTO CONSIDERATION			1-4*
				9438	*	THIS CO-RESIDENCY AND THE LIMITATION OF THE SECTOR BOUNDARY			1-4*
				9439	*	ON SIZE.			1-4*
				9440	*				*
				9441	*	REQUIRED MODULES			*
				9442	*	@SYSEQ - COMMON SYSTEM EQUATES			*
				9443	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES			*
				9444	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS			*
				9445	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES			*
				9446	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES			*
				9447	*	@ERMEQ - ERROR MESSAGE EQUATES			*
				9448	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES			*
				9449	*	\$B\$EQU - COMPILER FIXED EQUATES			*
				9450	*	\$B@EQU - COMPILER SYSTEM EQUATES			*
				9451	*				*
				9452	*	OTHER			*
				9453	*	BTPAUS IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			*
				9454	*	*****			*
				9456	*				*
				9457	*	ENTER BTPAUS - 'PAUSE' STATEMENT ROUTINE			*
				9458	*				*
			1CE7	9459	BTPAUS EQU *	BTPAUS ENTRY POINT			*
				9460	*				*
				9461	*	GENERATE A HALT INSTRUCTION IN VIRTUAL MEMORY			*
				9462	*				*
N04	1CE7	00 00 00		9463	BTP010 LA	BTPHTC(,@BR),@XR			LOAD CADDR OF 'HLT' INSIR
	1CEA	34 02 0A40		9464	ST	B\$PCAD,@XR			SET PUT RTN FOR VADDR OF 'HLT'
	1CEE	3C 00 0A41		9465	MVI	B\$PNBY,B@LHLT-1			SET PUT RTN FOR LENGTH OF 'HLT'
	1CF2	C0 87 093A		9466	B	B\$PUTC			LINK TO GENERATE PMC
				9467	*				*
				9468	*	RETURN CONTROL TO THE REMARK STATEMENT ROUTINE			*
				9469	*				*
N04	1CF6	00 00 0000		9470	BTP020 B	B@RMNK			RETURN CONTROL TO REM STNNT RTN
				9471	*				*
				9472	*	*****			*
				9473	*	'PAUSE' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS			*
				9474	*	*****			*
				9475	*				*
1CFA	04		1CFA	9476	ITPHTC DC	AL(B@LCOP)(B@CHLT)			'HLT' INSTRUCTION OPCODE
				9477	*				*
				9478	*	*****			*
				9479	*				*
				9480	*	END OF 'PAUSE' STATEMENT ROUTINE CODING			*
				9481	*				*



ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 148
		9483		*****	*
		9484	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		9485	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		9486	*		*
		9487		*****	*
		9488	*	*STATUS	*
		9489	*	VERSION 1 MODIFICATION 0	*
		9490	*		*
		9491	*	*FUNCTION	*
		9492	*	BMUPRT IS EXECUTED TO TRANSLATE MAT PRINT USING STATEMENTS AS THEY	*
		9493	*	OCCUR IN A B' IC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO	*
		9494	*	PLACE THE PSEUDOCODE IN VIRTUAL MEMORY.	*
		9495	*		*
		9496	*	*ENTRY POINTS	*
		9497	*	BMUPRT HAS ONLY ONE ENTRY POINT:	*
		9498	*	BMUPRT - TRANSLATE MAT PRINT USING STATEMENT	*
		9499	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		9500	*	B BMUPRT	*
		9501	*		*
		9502	*	*INPUT	*
		9503	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		9504	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE	*
		9505	*	LEADING KEYWORD, MAT PRINT USING.	*
		9506	*	* TEXT CHARACTER POINTER - CONTAINS THE CCM€ ADDRESS OF THE FIRST	*
		9507	*	CHARACTER IN THE LEADING KEYWORD, MAT ERINT USING.	*
		9508	*		*
		9509	*	*OUTPUT	*
		9510	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		9511	*	* GENERATED BY BRUFRT IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		9512	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		9513	*	SEQUENCES.	*
		9514	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		9515	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		9516	*		*
		9517	*	*EXTERNAL REFERENCES	*
		9518	*	B\$GETC - (B\$NUMC, B\$GPTR) - ENTR, TO BASIC RETRIEVAL ROUTINE.	*
		9519	*	B\$PUTC - (B\$PCAD, B\$PNBY, B\$PVAD) - ENTRY TO COMPILER VIRTUAL	*
		9520	*	MEMORY ROUTINE.	*
		9521	*	B\$BTAW - B\$BRVA, B\$BRIN) - BASIC COMPILER BRANCH TABLE ROUTINE.	*
		9522	*	B\$ZDBN - (B\$BINO) - ENTRY TO COMPILER ZONED DECIMAL TO BINARY	*
		9523	*	ROUTINE.	*
		9524	*	B\$MATR - ENTRY TO BASIC COMPILER MATRIX REFERENCE ROUTINE.	*
		9525	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		9526	*		*
		9527	*	*EXITS, NORMAL	*
		9528	*	B\$DIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.	*
		9529	*		*
		9530	*	*EXITS, ERROR	*
		9531	*	N/A	*
		9532	*		*
		9533	*	*TABLES/WORK AREAS	*
		9534	*	N/A	*
		9535	*		*
		9536	*	*ATTRIBUTES	*
		9537	*	BRUPRT IS NATURALLY RELOCATABLE AND REUSABLE.	*
		9538	*		*

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 149
				9539	*CHARACTER CODE DEPENDENCY	*
				9540	* THE OPERATION OF THIS MULE DOES NOT DEPEND UPON A PARTICULAR	*
				9541	* INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SEI.	*
				9542	*	*
				9543	*NOTES	*
				9544	* ERROR PROCEDURES	*
				9545	* N/A	*
				9546	*	*
				9547	* REGISTER USAGE	*
				9548	* BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
				9549	*	*
				9550	* SAVED/RESTORED AREAS	*
				9551	* N/A	*
				9552	*	*
				9553	* MODIFICATION CONSIDERATIONS	*
				9554	* BMUPRT RESIDES ON THE SAME SECTOR WITH BXCLOS AND BTSTOP.	1-4*
				9555	* ANY MODIFICATION TO BMUPRT MUST TAKE INTO CONSIDERATION	1-4*
				9556	* THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESSES	1-4*
				9557	* OF BXCLOS AND BTSTOP. THE LIMITATION OF THE SECTOR	1-4*
				9558	* BOUNDARY ON SIZE MUST ALSO BE CONSIDERED.	1-4*
				9559	*	*
				9560	* REQUIRED MODULES	*
				9561	* @SYSEQ - COMMON SYSTEM EQUATES	*
				9562	* @FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES	*
				9563	* @CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS	*
				9564	* @VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES	*
				9565	* @SPFEQ - SYSTEM PROGRAM FILE EQUATES	*
				9566	* @ERMEQ - ERROR MESSAGE EQUATES	*
				9567	* \$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES	*
				9568	* \$B\$EQU - COMPILER FIXED EQUATES	*
				9569	* \$B@EQU - COMPILER SYSTEM EQUATES	*
				9570	*	*
				9571	* OTHER	*
				9572	* BMUPRT IS ASSEMBLED WITH ALL THE STATEMENT PROCESSORS.	*
				9573	*****	
1D00				9574	ORG *,256,0 BEGIN AT CORE PAGE BOUNDARY	1-4
	1D00			9575	USING *,@BR DEFINE BASE ADDR FOR CORE PS	1-4
				9576	*	
				9577	* ENTER BMUPRT - MAT PRINT USING STATEMENT ROUTINE	
				9578	*	
	1D00			9579	BMUPRT EQU * BMUPRT ENTRY POINT	
				9580	*	
				9581	* SET GET ROUTINE TO SKIP TO CHAR FOLLOWING 'MAT PRINT USING'	
				9582	*	
N04 1D00 00 00 0000				9583	BMU010 MVI B@NUMC,BEILMPU SET GET TO SKIP KEYWORDS	
1D04 C0 87 0867				9584	B B\$GETC LINK TO ADVANCE POINTER	
				9585	*	
				9586	* GENERATE 'STA' INSTRUCTION 'MACE IN V.M.	
				9587	*	
1D08 D2 02 88				9588	BMU020 LA BMUSTC(,@BR),@XR LOAD CADDR OF 'STA' INSTR	
1D0B 34 02 0A40				9589	ST B\$PCAD,@XR SET VADDR PARAN OF PUT FOR STA	
1D0F 3C 02 0A41				9590	MVI B\$PNBY,B@LSTA-1 SET LNG PARAN OF PUT FOR 'STA'	
1D13 C0 87 093A				9591	B B\$PUTC LINK TO GENERATE 'STA' INSTR	
				9592	*	
				9593	* ESTABLISH 'STA' OPERAND FOR BRANCH TABLE ADDRESS RESOLUTION	
				9594	*	

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	
N04	1D17	00	00	0000	0000	9595	BMU030 MVC	B\$DRVA,B\$PVAD(@VADDR) SET VADDR FOR BR TBL RESOLUTION
N04	1D1D	00	00	0000	00	9596	SLC	B\$BRVA,BMURN1(@VADDR,@BR) ADJUST TO 'STA' OPND
						9597	*	
						9598	* GENERATE A 'BMX' INSTRUCTION IMAGE IN V.M.	
						9599	*	
	1D22	D2	02	8B		9600	BMU040 LA	BMUBNC(,@BR),@XR LOAD CADDR OF 'BMX' INSTR
	1D25	34	02	0A40		9601	ST	B\$PCAD,@XR SET VADDR PARM OF PUT FOR ICI
	1D29	3C	02	0A41		9602	MVI	B\$PNBY,B@LBNX-1 SET LNG PARM OF PUT FOR WU
	1D2D	C0	87	093A		9603	B	B\$PUTC LINK TO GENERATE 'BMX' INSTR
	1D31	35	02	0878		9604	L	B\$GPTR,@XR RESTORE TEXT POINTER
						9605	*	
						9606	* ESTABLISH NEXT AVAILABLE ADDR IN V.M. FOR BR TBL RESOLUTION (I.E.	
						9607	* THE VADDR OF 1ST INSTR IN DATA OUTPUT SEQUENCE)	
						9608	*	
	1D35	0C	01	19F1	0A43	9609	BMU050 MVC	B\$BRLN,B\$PVAD(@VADDR) SET VADDR FOR BR TBL RESOLUTION
	1D3B	C0	87	1996		9610	B	B\$BTAB LINK TO RESOLVE BR TBL ADDRS
						9611	*	
						9612	* ESTABLISH 'BNX' INSTR OPND FOR ADDRESS RESOLUTION	
						9613	*	
	1D3F	0C	01	19EF	0A43	9614	BMU060 MVC	B\$BRVA,B\$PVAD(@VADDR) SET VADDR FOR BR TBL RESOLUTION
	1D45	1F	01	19EF	94	9615	SLC	B\$BRVA,BMUBN1(@VADDR,@BR) ADJUST TO 'BNX' OPND
						9616	*	
						9617	* CONVERT THE LINE NUMBER OF THE IMAGE STATEMENT TO BINARY	
						9618	*	
	1D4A	C0	87	19F2		9619	BMU070 B	B\$ZDBN LINK TO CONVERT LINE NO TO BINARY
						9620	*	
						9621	* ESTABLISH IMAGE LN NO AS RESOLUTION LN NG	
						9622	*	
	1D4E	0C	01	19F1	1A6A	9623	BMU080 MVC	B\$BRLN,B\$BINO(@VADDR) SET RESOLUTION LINE NO
	1D54	C0	87	1996		9624	B	B\$BTAB LINK TO RESOLVE BR TBL ADDRS
						9625	*	
						9626	* CALL MATRIX REFERENCE PROCESSOR TO GENERATE DOPE VECTOR STACKING	
						9627	* INSTRUCTIONS IN VIRTUAL MEMORY	
						9628	*	
	1D58	C0	87	18F3		9629	BMU090 B	B\$MATR LINK TO PROCESS MAT-REFERENCE
						9630	*	
						9631	* GENERATE 'MF1' INSTRUCTION IN V.M. TO INDICATE MAT PRINT USING	
						9632	*	
	1D5C	D2	02	8E		9633	BMU100 LA	BMUMFC(,@BR),@XR LOAD CADDR OF 'MF1' INSTR
	1D5F	34	02	0A40		9634	ST	B\$PCAD,@XR SET VADDR PARM OF PUT FOR 'MF1'
	1D63	3C	02	0A41		9635	MVI	B\$PNBY,B@LMF1-1 SET LNG PARM OF PUT FOR 'MF1'
	1D67	C0	87	093A		9636	B	B\$PUTC LINK TO GENERATE 'MF1' PMC
						9637	*	
						9638	* TEST LIST DELIMITER FOR BEING A STATEMENT TERMINATOR	
						9639	*	
	1D6B	35	02	0878		9640	BMU110 L	B\$GPTR,@XR RESTORE TEXT POINTER
	1D6F	BD	1E	00		9641	CLI	B@CHAR(,@XR),B@EOST IF DELIMITER IS NOT EOS
	1D72	D0	01	58		9642	BNE	BMU090(,@BR) * GO PROCESS NEXT MAT REFERENCE
						9643	*	
						9644	* GENERATE 'PRU' INSTRUCTION WITH OPCOEE TO INDICATE IMAGE RELEASE	
						9645	*	
	1D75	D2	02	91		9646	BMU120 LA	BMUPRC(,@BR),@XR LOAD CADDR OF 'PRU' INSTR
	1D78	34	02	0A40		9647	ST	B\$PCAD,@XR SET VADDR PARM OF PUT FOR 'PRU'
	1D7C	3C	01	0A41		9648	MVI	B\$PNBY,B@LPRU-1 SET LNG PARM OF PUT FOR 'PRU'
	1D80	C0	87	093A		9649	B	B\$PUTC LINK TO GENERATE 'PRU' INSTR
						9650	*	

		9651	*	RETURN CONTROL TO COMPILER DISTRIBUTOR	
		9652	*		
1D84	C0 87 0700	9653	BMU130 B	B\$DIST	RETURN TO DISTRIBUTOR
		9655	*****		
		9656	*	MAT PRINT USING STATEMENT RTN STORAGE AND PARAMETER AREAS	
		9657	*****		
		9658	*		
1D88	34	1D88	9659	BMUSTC DC	AL(B@LCOP)(B@CSTA) 'STA' INSTR OPCODE
1D89	0000	1D8A	9660	BMUSTO DC	XL(B@LCVA)'00' 'STA' INSTR OPND IMAGE
		9661	*		
1D8B	4A	1D8B	9662	BMUBNC DC	AL(B@LCOP)(B@CBNX) 'BNX' INSTR OPCODE
1D8C	0000	1D8D	9663	BMURNO DC	XL(B@LCVA)'00' 'BNX' INSTR OPND IMAGE
		9664	*		
1D8E	18	1D8E	9665	BMUMFC DC	AL(B@LCOP)(B@CMF1) 'MF1' INSTR OPCODE
1D8F	3F13	1D90	9666	BMUMFO DC	AL(B@LCVA)(V\$XMPU) 'MF1' INSTR OPERAND
		9667	*		
1D91	62	1D91	9668	BMUPRC DC	AL(B@LCOP)(B@CPRU) 'PRU' INSTR OPCODE
1D92	10	1D92	9669	BMUPRO DC	AL(B@LCXX)(B@PUTM) 'PRU' INSTR OPND
		9670	*		
		9671	*	CONSTANTS	
		9672	*		
1D93	0001	1D94	9673	BMUBN1 DC	IL(@CADDR)'1' BINARY 1
		9674	*		
		9675	*****		
		9676	*		
		9677	*	END OF MAT PRINT USING STATEMENT ROUTINE CODING	
		9678	*		

## S/3 BASIC COMPILER -CLOSE- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 152
			9680		*****			
			9681	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
			9682	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
			9683	*				*
			9684		*****			*
			9685	*	*STATUS			*
			9686	*	VERSION 1 MODIFICATION 0			*
			9687	*				*
			9688	*	*FUNCTION			*
			9689	*	BXCLOS IS EXECUTED TO TRANSLATE CLOSE STATEMENTS AS THEY OCCUR			*
			9690	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
			9691	*	THE PSEUDOCODE INTO VIRTUAL MEMORY.			*
			9692	*				*
			9693	*	*ENTRY POINTS			*
			9694	*	BXCLOS HAS ONLY ONE ENTRY POINT:			*
			9695	*	BXCLOS - TRANSLATE CLOSE STATEMENT			*
			9696	*	THE FORMAT OF THE CALLING SEQUENCE IS:			*
			9697	*	B BXCLOS			*
			9698	*				*
			9699	*	*INPUT			*
			9700	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
			9701	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER IN THE			*
			9702	*	LEADING KEYWORD. CLOSE.			*
			9703	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
			9704	*	CHARACTER IN THE LEADING KEYWORD. CLOSE.			*
			9705	*				*
			9706	*	*OUTPUT			*
			9707	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
			9708	*	GENERATED BY BXCLOS IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
			9709	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
			9710	*	SEQUENCES.			*
			9711	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
			9712	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
			9713	*				*
			9714	*	*EXTERNAL REFERENCES			*
			9715	*	B\$GETC - (B\$NUMC) - ENTRY TO BASIC TEXT RETRIEVAL ROUTINE.			*
			9716	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
			9717	*	OUTPUT ROUTINE.			*
			9718	*	BSDIST - ENTRY TO BASIC COMPILER DISTRIBUTOR.			*
			9719	*				*
			9720	*	*EXITS, NORMAL			*
			9721	*	BSDIST - ENTRY TO THE BASIC COMPILER DISTRIBUTOR			*
			9722	*				*
			9723	*	*EXITS, ERROR			*
			9724	*	N/A			*
			9725	*				*
			9726	*	*TABLES/WORK AREAS			*
			9727	*	N/A			*
			9728	*				*
			9729	*	*ATTRIBUTES			*
			9730	*	BXCLOS IS NATURALLY RELOCATABLE AND REUSABLE.			*
			9731	*				*
			9732	*	*CHARACTER CODE DEPENDENCY			*
			9733	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
			9734	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
			9735	*				*

## S/3 BASIC COMPILER -CLOSE- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 153	
					9736	*	NOTES			*	
					9737	*	ERROR PROCEDURES			*	
					9738	*	N/A			*	
					9739	*				*	
					9740	*	REGISTER USAGE			*	
					9741	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.			*	
					9742	*				*	
					9743	*	SAVED/RESTORED AREAS			*	
					9744	*	N/A			*	
					9745	*				*	
					9746	*	MODIFICATION CONSIDERATIONS			*	
					9747	*	BXCLOS RESIDES ON THE SAME SECTOR WITH BMUPRT AND BTSTOP.	1-4	*		
					9748	*	ANY MODIFICATION TO BXCLOS MUST TAKE INTO CONSIDERATION	1-4	*		
					9749	*	THIS CO-RESIDENCY SINCE IT WILL CHANGE THE ENTRY ADDRESS	1-4	*		
					9750	*	OF BTSTOP. THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE	1-4	*		
					9751	*	MUST ALSO BE CONSIDERED.	1-4	*		
					9752	*				*	
					9753	*	REQUIRED MODULES			*	
					9754	*	@SYSEQ - COMMON SYSTEM EQUATES			*	
					9755	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES			*	
					9756	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS			*	
					9757	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES			*	
					9758	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES			*	
					9759	*	@ERMEQ - ERROR MESSAGE EQUATES			*	
					9760	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES			*	
					9761	*	\$B\$EQU - COMPILER FIXED EQUATES			*	
					9762	*	\$B@EQU - COMPILER SYSTEM EQUATES			*	
					9763	*				*	
					9764	*	OTHER			*	
					9765	*	BXCLOS IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			*	
					9766	*	*****			*	
					9768	*				*	
					9769	*	ENTER BXCLOS - 'CLOSE' STATEMENT ROUTINE			*	
					9770	*				*	
				1D95	9771	BXCLOS EQU	*	BXCLOS ENTRY POINT		*	
					9772	*				*	
					9773	*	SET GET ROUTINE TO SKIP TO 'E' IN KEYWORD 'CLOSE'			*	
					9774	*				*	
				1D95	3C	04	0873	9775	BXC010 MVI	B\$NUMC,B@LKCL-1	SET GET TO SKIP TO 'E'
				1D99	C0	87	0867	9776	B	B\$GETC	LINK TO ADVANCE POINTER
				1D9D	C0	87	14B0	9777	BXC020 B	B\$CSCN	LINK TO PROCESS FILE REFERENCE
					9778	*				*	
					9779	*	GENERATE THE 'ADF' PMC IN V.M. IF OPND IS NOT ZERO			*	
					9780	*				*	
				1DA1	D2	02	D1	9781	BXC120 LA	BXCAFC(,@BR),@XR	LOAD CADDR OF 'ADE' INSTR
				1DA4	34	02	0A40	9782	ST	B\$PCAD,@XR	SET VADDR PARAM OF PUT FOR 'ADE'
				1DA8	3C	01	0A41	9783	MVI	B\$PNBY,B@LADF-1	SET LNG PARAM OF PUT FOR 'ADE'
				1DAC	C0	87	093A	9784	B	B\$PUTC	LINK TO GENERATE 'ADE' PMC
					9785	*				*	
					9786	*	GENERATE THE 'CLS' PMC IN V.M.			*	
					9787	*				*	
				1DB0	D2	02	D3	9788	BXC130 LA	BXCCLC(,@BR),@XR	LOAD CADOR OF 'CLS. INSTR
N04				1DB3	00	00	0000	9789	ST	B\$PCAD,@XR	SET VADOR PARAM OF PUT FOR CL:
N04				1DB7	00	00	0000	9790	MVI	B\$PNBY,B@LCLS-1	SET LNG PARAM OF PUT FOR 'CLS'
				1DBB	C0	87	093A	9791	B	B\$PUTC	LINK TO GENERATE 'CLS' PMC



## S/3 BASIC COMPILER -CLOSE- STATEMENT ROUTINE

ERR	LOC	OBJECT	CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 154
					9792	*				
					9793	*	TEST NEXT LIST CHARACTER FOR BEING AN END-OF-STATEMENT			
					9794	*				
		1DBF	3C 00 0873		9795	BXC140 MVI	B\$NUMC,B@GETS		DISABLE GET ROUTINE	
		1DC3	C0 87 0867		9796	B	B\$GETC		LINK TO GET CHARACTER POINTER	
		1DC7	BD 1E 00		9797	CLI	B@CHAR(,@XR),B@EOST		IF CHAR IS EOS	
N04		1DCA	00 00 00		9798	BNE	BXC020(,@BR)		* BRANCH TO PROCESS FILENAME	
					9799	*				
					9800	*	RETURN CONTROL TO THE COMPILER DISTRIBUTOR			
					9801	*				
		1DCD	C0 87 0700		9802	BXC150 B	B\$DIST		RETURN TO DISTRIBUTOR	
					9804	*****	*****AA*****			
					9805	*	'CLOSE' STATEMENT PARAMETER AND STORAGE AREAS			
					9806	*****	*****			
					9807	*				
		1DD1	58	1DD1	9808	BXCAFC DC	AL(B@LCOP)(B@CADF)		'ADF' INSTR OPCODE	
		1DD2	00	1DD2	9809	BXCAFO DC	XL1'00'		'ADF' INSTR OPERAND	
					9810	*				
		1DD3	5E	1DD3	9811	BXCCLC DC	AL(B@LCOP)(B@CCLS)		'CLS' INSTR OPCODE	
					9813	*****	*****			
					9814	*	'CLOSE' STATEMENT CONSTANTS AND EQUATES			
					9815	*****	*****			
					9816	*				
					9817	*	CONSTANTS			
					9818	*				
				1DD4	9819	BXCSFA EQU	*			
					9820	*				
		1DD4	0001	1DD5	9821	BXCBN1 DC	IL(@CADDR)'1'		BINARY '1'	
					9822	*				
					9823	*	END OF 'CLOSE' STATEMENT ROUTINE CODING			
					9824	*				



## S/3 BASIC COMPILER -STOP- STATEMENT ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 155
		9826		*****			
		9827	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		9828	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		9829	*				*
		9830		*****			
		9831	*	STATUS			*
		9832	*	VERSION 1 MODIFICATION 0			*
		9833	*				*
		9834	*	FUNCTION			
		9835	*	BTSTOP IS EXECUTED TO TRANSLATE STOP STATEMENTS AS THEY OCCUR IN			
		9836	*	A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE THE			
		9837	*	PSEUDOCODE IN VIRTUAL MEMORY.			
		9838	*				
		9839	*	ENTRY POINTS			
		9840	*	BTSTOP HAS ONLY ONE ENTRY POINT:			
		9841	*	BTSTOP - TRANSLATE STOP STATEMENT			
		9842	*	THE FORMAT OF THE CALLING SEQUENCE IS:			
		9843	*	B BTSTOP			
		9844	*				
		9845	*	INPUT			
		9846	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			
		9847	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE			
		9848	*	LEADING KEYWORD, STOP.			
		9849	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE 1ST			
		9850	*	CHARACTER IN THE LEADING KEYWORD, STOP.			
		9851	*				
		9852	*	OUTPUT			
		9853	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			
		9854	*	GENERATED BY BTSTOP IS STORED IN THE NEXT AVAILABLE VIRTUAL			
		9855	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			
		9856	*	SEQUENCES.			
		9857	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			
		9858	*	CHARACTER WHICH TERMINATES THE STATEMENT.			
		9859	*				
		9860	*	EXTERNAL REFERENCES			
		9861	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			
		9862	*	OUTPUT ROUTINE.			
		9863	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.			
		9864	*				
		9865	*	EXITS, NORMAL			
		9866	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.			
		9867	*				
		9868	*	EXITS, ERROR			
		9869	*	N/A			
		9870	*				
		9871	*	TABLES/WORK AREAS			
		9872	*	N/A			
		9873	*				
		9874	*	ATTRIBUTES			
		9875	*	BTSTOP IS NATURALLY RELOCATABLE AND REUSABLE.			
		9876	*				
		9877	*	CHARACTER CODE DEPENDENCY			
		9878	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			
		9879	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			
		9880	*				
		9881	*	NOTES			

## S/3 BASIC COMPILER -STOP- STATEMENT ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 156
				9882	*	ERROR PROCEDURES			
				9883	*	N/A			
				9884	*				
				9885	*	REGISTER USAGE			
				9886	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.			
				9887	*				
				9888	*	SAVED/RESTORED AREAS			
				9889	*	N/A			
				9890	*				
				9891	*	MODIFICATION CONSIDERATIONS			
				9892	*	BTSTOP RESIDES ON THE SAME SECTOR WITH BMUPRT AND BXCLOS.			1-4
				9893	*	ANY MODIFICATION TO BTSTOP MUST TAKE INTO CONSIDERATION			1-4
				9894	*	THIS CO-RESIDENCY AND ALSO THE LIMITATION OF THE SECTOR			1-4
				9895	*	BOUNDARY ON SIZE.			1-4
				9896	*				
				9897	*	REQUIRED MODULES			
				9898	*	@SYSEQ - COMMON SYSTEM EQUATES			
				9899	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES			
				9900	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS			
				9901	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES			
				9902	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES			
				9903	*	@ERMEQ - ERROR MESSAGE EQUATES			
				9904	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES			
				9905	*	\$B\$EQU - COMPILER FIXED EQUATES			
				9906	*	\$B@EQU - COMPILER SYSTEM EQUATES			
				9907	*				
				9908	*	OTHER			
				9909	*	BTSTOP IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.			
				9910	*	*****			
				9912	*				
				9913	*	ENTER BTSTOP - 'STOP' STATEMENT ROUTINE			
				9914	*				
			1DD6	9915	BTSTOP EQU *	BTSTOP ENTRY POINT			
				9916	*				
				9917	*	GENERATE AN 'SVC' INSTRUCTION IN VIRTUAL MEMORY			
				9918	*				
1DD6	D2	02	E9	9919	BTS010	LA BTSSVC(,@BR),@XR			LOAD CADDR OF 'SVC' INSTR
1DD9	34	02	0A40	9920		ST B\$PCAD,@XR			SET PUT RTN FOR VADDR OF 'SVC'
1DDD	3C	00	0A41	9921		MVI B\$PNBY,B@LSVC-1			SET PUT RTN FOR LENGTH OF 'SVC'
1DE1	C0	87	093A	9922		B B\$PUTC			LINK TO GENERATE PMC
				9923	*				
				9924	*	RETURN CONTROL TO THE REMARK STATEMENT ROUTINE			
				9925	*				
1DE5	C0	87	1AE6	9926	BTS020	B B\$RMRK			RETURN TO REMARK VINT RTN
				9928	*	*****			
				9929	*	'STOP' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS			
				9930	*	*****			
				9931	*				
1DE9	02			1DE9	9932	BTSSVC DC AL(B@LCOP)(B@CSVC)			'SVC' INSTR OPCODE
				9933	*				
				9934	*	*****			
				9935	*				
				9936	*	END OF 'STOP' STATEMENT ROUTINE CODING			
				9937	*				

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 157
9939				*****			
9940	*			5703-XM1 COPYRIGHT IBM CORP. 1970			*
9941	*			REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
9942	*						*
9943				*****			*
9944	*			*STATUS			*
9945	*			VERSION 1 MODIFICATION 0			*
9946	*						*
9947	*			*FUNCTION			*
9948	*			* BTRMNT IS EXECUTED TO TRANSLATE THE FIRST END STATEMENT OR			*
9949	*			END-OF-FILE RECORD ENCOUNTERED IN THE SOURCE PROGRAM TEXT INTO			*
9950	*			THE APPROPRIATE PSEUDOCODE AND TO PLACE THE PSEUDOCODE IN			*
9951	*			VIRTUAL MEMORY.			*
9952	*			* BTRMNT ALSO PERFORMS THE FOLLOWING FUNCTIONS:			*
9953	*			* BASIC PROGRAM PROCESSING IS ABORTED IN THE PRESENCE OF ANY			*
9954	*			LOGGED OR CURRENTLY ENCOUNTERED COMPILER ERROR CONDITION.			*
9955	*			RISIDUAL CORE-RESIDENT PMC AND PROGRAM GENERATED CONSTANTS ARE			*
9956	*			WRITTEN TO DISK VIRTUAL MEMORY, PMC GENERATION IS CLOSED.			*
9957	*			* RISIDUAL STATEMENT ADDRESS TABLE AND BRANCH ADDRESS TABLE			*
9958	*			ENTRIES ARE WRITTEN TO THE RESPECTIVE DISK FILES, ADDRESS TABLE			*
9959	*			FILES ARE CLOSED.			*
9960	*			* CRITICAL VIRTUAL ADDRESSES ARE ESTABLISHED IN A HIGH CORE			*
9961	*			PARAMETER REGION FOR TRANSFER TO THE NEXT PROCESSOR PHASE.			*
9962	*			* SCALAR VARIABLE SYMBOL TABLES ARE ORGANIZED AND ESTABLISHED			*
9963	*			IN THE #LOADR PARAMETER TRANSFER AREA.			*
9964	*			* FUNCTION AND ARRAY SYMBOL TABLES ARE EXTRACTED FROM THE COMPILE			*
9965	*			TIME SYMBOL TABLE/ATTRIBUTE CONGLOMERATES AND ESTABLISHED IN			*
9966	*			THE #LOADR PARAMETER TRANSFER AREA.			*
9967	*			* THE RUN-TIME FUNCTION AND ARRAY TABLE IS CONSTRUCTED IN THE			*
9968	*			#LOADR PARAMETER TRANSFER AREA FROM DATA EXTRACTED FROM THE			*
9969	*			COMPILE-TIME SYMBOL TABLE/ATTRIBUTE CONGLOMERATES; THIS TABLE			*
9970	*			IS CONSTRUCTED AS IT WILL EVENTUALLY APPEAR IN VIRTUAL MEMORY.			*
9971	*			* THE NEXT PROCESSOR PHASE (#LOADR) IS CORE-LOADED AND EXECUTED			*
9972	*			USING SYSTEM ENTRY POINT #RLOAD.			*
9973	*						*
9974	*			*ENTRY POINTS			*
9975	*			BTRMNT HAS ONLY ONE ENTRY POINT:			*
9976	*			BTRMNT - TERMINATE COMPILATION			*
9977	*			THE FORMAT OF THE CALLING SEQUEICE IS:			*
9978	*			B BTRMNT			*
9979	*						*
9980	*			*INPUT			*
9981	*			* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
9982	*			THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE			*
9983	*			LEADING KEYWORD, END. IF THE END IS IMPLICIT THE RECORD			*
9984	*			SEGMENT CONTAINS THE END-OF-STATEMENT CHARACTER.			*
9985	*			* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
9986	*			CHARACTER IN THE LEADING KEYWORD, END. IF THE END IS IMPLICIT,			*
9987	*			THE CORE ADDRESS IS OF THE END-OF-STATEMENT CHARACTER.			*
9988	*			* B\$ERSN - SET TO ON STATUS WHEN COMPILE-TIME ERRORS HAVE BEEN			*
9989	*			ENCOUNTERED AND LOGGED IN VIRTUAL MEMORY PRIOR TO BTRMNT			*
9990	*			EXECUTION.			*
9991	*			* LOGGED ERRORS - WHEN B\$ERSW IS FOUND ON, THE FIRST 3 VIRTUAL			*
9992	*			MEMORY PAGES NORMALLY USED FOR PMC STORAGE ARE EXPECTED TO			*
9993	*			CONTAIN FROM 1 TO 255 3-BYTE ERROR CODE RECORDS.			*
9994	*			* DIPECT - WHEN MERU IS ON, THIS IS EXPECTED TO CONTAIN A COUNT			*

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 158
		9995	*	OF THE NUMBER OF ERROR CODE RECORDS LOGGED IN VIRTUAL MEMORY;	*		
		9996	*	THIS IS NEVER PERMITTED TO EXCEED A VALUE OF 255.	*		
		9997	*	* B\$FTPT - CONTAINS THE CORE ADDRESS OF THE FIRST BYTE IN THE	*		
		9998	*	TOP FOR TABLE ENTRY. WHEN THIS IS NOT IDENTICAL WITH THE	*		
		9999	*	ADDRESS OF THE TABLE ITSELF, AN INCOMPLETE FOR LOOP IS	*		
			*	INDICATED.	*		
		1	*	* B\$PVAD - CONTAINS THE VIRTUAL ADDRESS OF THE NEXT AVAILABLE PMC	*		
		2	*	BYTE, AND IS USED TO ESTABLISH THE LAST PAGE OCCUPIED BY	*		
		3	*	PMC FOR VM REGION 1 DEFINITION.	*		
		4	*	* B\$PCPG - CONTAINS THE VIRTUAL PAGE NUMBER OF THE PAGE CURRENTLY,	*		
		5	*	BEING FILLED WITH PROGRAM GENERATED CONSTANTS, AND USED TO	*		
		6	*	DEFINE THE UPPER BOUNDARY ADDRESS OF VM REGION 19	*		
		7	*	* B\$CVPD - CONTAINS THE DISPLACEMENT VALUE USED AS A CONSTANT	*		
		8	*	OUTPUT BUFFER POINTER WHEN THIS VALUE IS LESS THAN X'FF',	*		
		9	*	RISIDUAL BUFFER CONSTANTS ARE INDICATED.	*		
		10	*	* B\$BSDA - CONTAINS THE LOGICAL SECTOR ADDRESS OF THE SECTOR	*		
		11	*	CURRENTLY BEING FILLED WITH BRANCH TABLE ENTRIES.	*		
		12	*	* B\$SVPB - CONTAINS THE VIRTUAL ADDRESS OF THE NEXT BYTE	*		
		13	*	AVAILABLE FOR PROGRAM VARIABLE ALLOCATION.	*		
		14	*	* B\$SFAB - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE IN THE	*		
		15	*	LAST ARRAY DOPE VECTOR OR USER FUNCTION ADDRESS DEFINED IN THE	*		
		16	*	PROGRAM.	*		
		17	*	* B\$FAIS - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE	*		
		18	*	ALLOCATED FOR INTERNAL CONSTANTS IN THE PROGRAM.	*		
		19	*	* B\$FAIW - CONTAINS THE VIRTUAL ADDRESS OF THE FIRST BYTE	*		
		20	*	ALLOCATED FOR INTERNAL VARIABLES IN THE PROGRAM.	*		
		21	*	* \$EXFTR - CONTAINS A COUNT OF THE NUMBER OF CORE PAGES AVAILABLE	*		
		22	*	BEYCND 8K FOR GENERAL PROGRAM UTILIZATION.	*		
		23	*	* B\$SLVT - THE 58-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES	*		
		24	*	FOR EACH LETTER VARIABLE DEFINED IN THE PROGRAM.	*		
		25	*	* B\$SLDT - THE 580-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES	*		
		26	*	FOR EACH LETTER-DIGIT VARIABLE DEFINED IN THE PROGRAM.	*		
		27	*	* B\$SCVT - THE 58-BYTE SYMBOL TABLE CONTAINING VIRTUAL ADDRESSES	*		
		28	*	FOR EACH CHARACTER VARIABLE DEFINED IN THE PROGRAM.	*		
		29	*	* B\$SNAT - THE 174-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL	*		
		30	*	ADDRESSES AND DOPE VECTOR INFORMATION FOR EACH ARITHMETIC ARRAY	*		
		31	*	DEFINED IN THE PROGRAM.	*		
		32	*	* B\$SCAT - THE 116-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL	*		
		33	*	ADDRESSES AND DOPE VECTOR INFORMATION FOR EACH CHARACTER ARRAY	*		
		34	*	DEFINED IN THE PROGRAM.	*		
		35	*	* B\$SFNT - THE 116-BYTE SYMBOL/ATTRIBUTE TABLE CONTAINING VIRTUAL.	*		
		36	*	ADDRESSES AND RUN-TIME ENTRY POINTS FOR EACH USER FUNCTION	*		
		37	*	DEFINED IN THE PROGRAM.	*		
		38	*		*		
		39	*	*OUTPUT	*		
		40	*	* VIRTUAL MEMORY - IN THE ABSENCE OF ANY ERROR CONDITION, THE PMC	*		
		41	*	SEQUENCE GENERATED UNDER CONTROL OF BTRMNT IS STORED IN THE	*		
		42	*	NEXT AVAILABLE VIRTUAL MEMORY LOCATION FOLLOWING PREVIOUSLY	*		
		43	*	STORED INSTRUCTION SEQUENCES, VIRTUAL MEMORY IS THEN CLOSED	*		
		44	*	FOR BOTH PMC AND PROGRAM GENERATED CONSTANTS.	*		
		45	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*		
		46	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*		
		47	*	* \$CAERR - WHEN ERROR 2 OR ERROR 3 (SEE ERROR PROCEDURES UNDER	*		
		48	*	NOTES) IS IN EFFECT, THIS IS SET TO CONTAIN A CODE DEFINING	*		
		49	*	THE APPROPRIATE ERROR MESSAGE FOR #ERRPG.	*		
		50	*	* #ERRPG - WHEN ERROR 1 IS IN EFFECT, THIS IS SET TO CODE \$ERSTK	*		

## S/3 BASIC COMPILER TERMINATION ROUTINE

```

ERR LOC  OBJECT CODE      ADDR STMT SOURCE STATEMENT                                VER 15, MOD 00  20/07/20  PAGE 159

51 *      TO INDICATE MULTIPLE ERROR MESSAGE DISPLAY.  WHEN ERROR 2 OR *
52 *      ERROR 3 IS IN EFFECT, THIS IS SET TO CODE $$$NLN TO INDICATE *
53 *      THE SUPPRESSION OF LINE NUMBER DISPLAY. *
54 *      * $ERRCT - WHEN ERROR 1 IS IN EFFECT, THIS IS SET TO CONTAIN THE *
55 *      VALUE IN ERROR RECORD COUNT B$PECT. *
56 *      * ERROR RECORD STACK - WHEN ERROR 1 IS IN EFFECT, CORE REGION *
57 *      X'1C00' THROUGH X'1EFF' IS LOADED WITH THE ERROR RECORDS *
58 *      LOGGED AT COMPILE TIME. *
59 *      * $XIND1 - WHEN ERROR 1 IS IN EFFECT, THIS SYSTEM INDICATOR IS *
60 *      CLEARED TO SPECIFY VIRTUAL MEMORY AS UNDEFINED. *
61 *      * STATEMENT ADDRESS TABLE FILE - A FINAL ENTRY (X'FFFF', X'FFFF') *
62 *      IS STORED IN THE LAST ENTRY POSITION OF THE STATEMENT ADDRESS *
63 *      TABLE BUFFER, AND THE BUFFER IS OUTPUT TO CLOSE THE STATEMENT *
64 *      ADDRESS TABLE FILE. *
65 *      * BRANCH ADDRESS TABLE FILE - WHEN ERROR 3 IS NOT IN EFFECT, THE *
66 *      BRANCH ADDRESS TABLE BUFFER IS OUTPUT TO CLOSE THE FILE. *
67 *      * #LOADR PARAMETER TRANSFER AREA - A COMMON AREA FOR TRANSFER OF *
68 *      INFORMATION BETWEEN THE COMPILER AND LOADER PHASES. *
69 * *
70 *EXTERNAL REFERENCEACES *
71 *      B$PUTC - (B$PFNC, B$PCAD, B$PNBY, B$PVAD, B$PCPG, B$ERSW) - *
72 *      ENTRY TO COMPILER VIRTUAL MEMORY OUTPUT ROUTINE. *
73 *      B$FCON - (B$CVPD) - ENTRY TO BASIC COMPILER CONSTANT ROUTINE. *
74 *      B$SYMB - (B$SLVT, B$SLDT, B$SCVT, B$SNAT, B$SCAT, B$SFNT, *
75 *      B$SVBB, B$SFAB) - ENTRY TO BASIC COMPILER SYMBOL *
76 *      TRANSLATION ROUTINE. *
77 *      B$SCAN - (B$FAIS, B$FAIW) - ENTRY TO BASIC COMPILER ARITHMETIC *
78 *      EXPRESSION SCAN ROUTINE. *
79 *      B$BTAB - (B$BSDA, B$BDPL) - ENTRY TO BASIC COMPILER BRANCH *
80 *      TABLE ROUTINE. *
81 *      B$DIST - (B$DST2, B$SDPL) - ENTRY TO BASIC COMPILER DISTRIBUTOR *
82 *      BVDL4T. *
83 *      COMMOM - (B$FORT, B$FTPT, B$LDRP, B$CSBF, B$CSXA) - ENTRY TO *
84 *      COMMON CORE LOCATIONS OUTSIDE NUCLEUS. *
85 *      NUCLEUS - ($XIND1, $ERRPG, $ERRCT, $CAERR, $CAERK, $DISKN, *
86 *      $WAITF, $EXFTR, $RLOAD) - ENTRY TO INDICATORS AND *
87 *      ADDRESSES IN NUCLEUS. *
88 * *
89 *EXITS, NORMAL *
90 *      IN THE ABSENCE OF COMPILER ERRORS, CONTROL IS ALWAYS PASSED TO *
91 *      SYSTEM LOADER *
92 *      $RLOAD *
93 * *
94 *EXITS, ERROR *
95 *      THE FIRST ERROR CONDITION TO BE DISCOVERED CAUSES AN EXIT *
96 *      TO SYSTEM ERROR MESSAGE ROUTINE *
97 *      #ERRPG VIA *
98 *      $CAERK WITH APPROPRIATE ERROR CODE IN *
99 *      $CAERR *
100 * *
101 *TABLES/WORK AREAS *
102 *      * SEE INPUT AND OUTPUT SECTIONS ABOVE. *
103 *      * BTREPL - THE DISK PARAMETER LIST USED TO CORELOAD ERROR RECORDS *
104 *      LOGGED IN VIRTUAL MEMORY WHEN B$ERSW IS ON. *
105 *      * BTRDPL - THE DISK PARAMETER LIST USED AS ARGUMENT FOR $RLOAD *
106 *      DEFINING #LOADR DISK AND CORELOAD PARAMETERS. *

```



## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 160
		107	*		*
		108	*	*ATTRIBUTES	*
		109	*	BTRMNT IS NATURALLY RELOCATABLE AND REUSABLE.	*
		110	*		*
		111	*	**CHARACTER CODE DEPENDENCY	*
		112	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		113	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		114	*		*
		115	*	*NOTES	*
		116	*	ERROR PROCEDURES	*
		117	*	ERROR 1 - SWITCH B\$ERSW IS FOUND ON, INDICATING THAT AT LEAST	*
		118	*	ONE COMPILE-TIME ERROR HAS BEEN GENERATED IN VIRTUAL MEMORY,	*
		119	*	VIRTUAL MEMORY IS SET UNDEFINED AND THE FIRST 3 PMC VIRTUAL	*
		120	*	PAGES ARE READ INTO CORE.	*
		121	*	ERROR 2 - THE FOR TABLE IS FOUND TO CONTAIN AT LEAST ONE ENTRY	*
		122	*	WHICH HAS NOT BEEN PAIRED WITH A MATCHING NEXT STATEMENT.	*
		123	*	AN ERROR CODE IS ESTABLISHED FOR 'FOR/NEXT LOOP INCOMPLETE'.	*
		124	*	ERROR 3 - THE BRANCH ADDRESS TABLE FILE IS FILLED TO CAPACITY	*
		125	*	AND MORE TABLE ENTRIES REMAIN TO BE OUTPUT. AN ERROR CODE	*
		126	*	IS ESTABLISHED FOR 'TOO MANY LINE NUMBER REFERENCES'.	*
		127	*		*
		128	*	REGISTER USAGE	*
		129	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.	*
		130	*		*
		131	*	SAVED/RESTORED AREAS	*
		132	*	N/A	*
		133	*		*
		134	*	MODIFICATION CONSIDERATIONS	*
		135	*	BTRMNT RESIDES ON TWO SECTORS, CO-RESIDENT ON THE SECOND	1-4*
		136	*	SECTOR WITH BKRTRN AND BPXRSR. ANY MODIFICATION TO BTRMNT	1-4*
		137	*	MUST MAINTAIN THE LINKAGE BETWEEN THE TWO SECTORS AND ALSO	1-4*
		138	*	TAKE INTO CONSIDERATION THE CO-RESIDENCY SINCE A CHANGE	1-4*
		139	*	TO BTRMNT CAN CHANGE THE ENTRY ADDRESSES OF BKRTRN AND	1-4*
		140	*	BPXRSR. THE LIMITATION OF THE SECTOR BOUNDARY ON SIZE	1-4*
		141	*	MUST ALSO BE CONSIDERED.	1-4*
		142	*		*
		143	*	REQUIRED MODULE	*
		144	*	@\$YSEQ - COMMON SYSTEM EQUATES.	*
		145	*	@FXDEQ - SYSTEM NUCLEUS ADDR AND INDICATOR VALUES EQUATES.	*
		146	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS.	*
		147	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.	*
		148	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.	*
		149	*	@ERMEQ - ERROR MESSAGE EQUATES.	*
		150	*	\$V\$EQU - FIXED VIRTUAL ADDRESS EQUATES.	*
		151	*	\$B\$EQU - COMPILER FIXED EQUATES.	*
		152	*	\$B@EQU - COMPILER SYSTEM EQUATES.	*
		153	*		*
		154	*	OTHER	*
		155	*	BTRMNT IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.	*
		156	*	*****	*
1E00		158		ORG *,256,0	BEGIN AT CORE PAGE BOUNDARY
	1E00	159		USING *,@BR	DEFINE BASE ADDR FOR CORE PAGE
		160	*		
		161	*	ENTER BTRMNT - COMPILER TERMINATOR	
		162	*		

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 161

1E00 74 01 FB	1E00	163	BTRMNT EQU *	BTRMNT ENTRY POINT
		164	ST BTRCA2(, @BR), @BR	SAVE BTRMNT BASE ADDRESS
		165	*	
		166	* TEST FOR COMPILER-GENERATED ERRORS	
		167	*	
1E03 38 07 0993		168	BTR010 TBN B\$ERSW, B\$ERMK	TEST THE COMPILER ERROR SWITCH
1E07 F2 90 21		169	JF BTR040	BRANCH IF NO COMPILER ERRORS
		170	*	
		171	* COMPILER ERRORS - CORELOAD ERROR CODES FROM VIRTUAL MEMORY	
		172	*	
1E0A 3C 9D 094E		173	BTR020 MVI B\$PFNC, B\$PFCL	SET PUT ROUTINE 'CLOSE' FUNC
1E0E C0 87 093A		174	B B\$PUTC	LINK TO CLOSE THE ERROR FILE
		175	*	
1E12 D2 02 F2		176	LA BTREPL(, @BR), @XR	LOAD COMPILER ERROR DPL CADDR
1E15 C0 87 1A6B		177	B B\$DL4T	LINK TO READ ERRORS FROM VM
		178	*	
		179	* ERROR EXIT 1 - PRINT COMPILER-GENERATED STACKED ERROR MESSAGES	
		180	*	
1E19 3C 00 03D0		181	BTR030 MVI \$XIND1, @ZERO	DELETE VM DEFINITION INDICATOR
1E1D 3C 30 03CE		182	MVI \$ERRPG, \$ERSTK	SET ERROR RTN FOR STACKED CODE
1E21 0C 00 03CF 0A44		183	MVC \$ERRCT, B\$PECT(1)	SET ERROR RTN MESSAGE COUNT
1E27 C0 87 0469		184	B \$CAERK	EXIT TO SYSTEM ERROR ROUTINE
		185	*	
		186	* TEST FOR AN INCOMPLETE 'FOR' LOOP IN THE PROGRAM	
		187	*	
1E2B 1D 01 1B0D ED		188	BTR040 CLC B\$FTPT, BTRFTA(@CADDR, @BR)	TEST FOR AN EMPTY 'FOR' TABLE
1E30 F2 81 0C		189	JE BTR060	BRANCH IF NO ACTIVE 'FOR' ENTRY
		190	*	
		191	* ERROR EXIT 2 - PRINT 'INCOMPLETE 'FOR' LOOP' ERROR MESSAGE	
		192	*	
1E33 3C A0 03CE		193	BTR050 MVI \$ERRPG, \$\$\$NLN	SET FOR NO LINE NO. PRINTOUT
1E37 3C AE 03CD		194	MVI \$CAERR, @@E609	SET THE ERROR MESSAGE CODE
1E3B C0 87 0469		195	B \$CAERK	EXIT TO SYSTEM ERROR ROUTINE
		196	*	
		197	* GENERATE THE FINAL PROGRAM PSEUDO INSTRUCTION SEQUENCE - AN ERROR	
		198	* CONDITION (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT	
		199	*	
1E3F D2 02 F8		200	BTR060 LA BTRPCA(, @BR), @XR	LOAD FINAL PMC SEQUENCE CADDR
1E42 34 02 0A40		201	ST B\$PCAD, @XR	SET PUT RTN CORE ADDR PARAMETER
1E46 3C 01 0A41		202	MVI B\$PNBY, B@LSVC+B@LEOF-1	SET PUT RTN LENGTH PARAMETER
1E4A C0 87 093A		203	B B\$PUTC	LINK TO OUTPUT THE FINAL PMC
		204	*	
		205	* CLOSE OUTPUT OF PSEUDO INSTRUCTIONS TO VIRTUAL MEMORY - AN ERROR	
		206	* CONDITION (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT	
		207	*	
1E4E 3C 9D 094E		208	BTR070 MVI B\$PFNC, B\$PFCL	SET PUT ROUTINE 'CLOSE' FUNC
1E52 C0 87 093A		209	B B\$PUTC	LINK TO CLOSE THE PMC FILE
		210	*	
		211	* TEST FOR ANY CONSTANTS REMAINING TO BE OUTPUT	
		212	*	
1E56 3D FF 0C5D		213	BTR080 CLI B\$CVPD, BTRBND	TEST FOR AN EMPTY CONSTANT BFR
1E5A F2 81 08		214	JE BTR100	BRANCH WHEN BUFFER IS EMPTY
		215	*	
		216	* OUTPUT THE FINAL PAGE OF PROGRAM CONSTANTS - AN ERROR CONDITION	
		217	* (PROGRAM TOO LARGE) IS POSSIBLE AT THIS POINT	
		218	*	



## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 162

```

1E5D 3C 15 094E      219 BTR090 MVI   B$PFNC,B$PFWP      SET PUT RTN TO WRITE A PAGE
1E61 C0 87 093A      220          B    B$PUTC      LINK TO OUTPUT CONSTANT BUFFER
221 *
222 * TEST FOR POSSIBLE OVERFLOW OF THE BRANCH ADDRESS TABLE FILE
223 *
1E65 C2 02 19E8      224 BTR100 LA     B$BDPL,@XR      LOAD BRANCH TABLE DPL CADDR
1E69 3D 60 19EA      225          CLI   B$BDSA,B@DTB1+B@DTBN  IF BRANCH ADDR FILE NOT FULL
1E6D F2 82 0C        226          JL    BTR120      * GO OUTPUT THE FINAL FILE BFR
227 *
228 * ERROR EXIT 3 - PRINT 'TOO MANY LINE NO. REFERENCES' ERROR MESSAGE
229 *
1E70 3C A0 03CE      230 BTR110 MVI   $ERRPG,$$$NLN      SET FOR NO LINE NO. PRINTOUT
1E74 3C B1 03CD      231          MVI   $CAERR,@E612     SET THE ERROR MESSAGE CODE
1E78 C0 87 0469      232          B    $CAERK      EXIT TO SYSTEM ERROR ROUTINE
233 *
234 * OUTPUT THE FINAL BRANCH ADDRESS TABLE BUFFER TO DISK
235 *
1E7C C0 87 1A6B      236 BTR120 B     B$DL4T      LINK TO WRITE BRANCH TABLE BFR
237 *
238 * OUTPUT THE FINAL STATEMENT ADDRESS TABLE BUFFER TO DISK
239 *
N04 1E80 00 00 0000 00 240 BTR130 MVC   BTRSHA,BTRSHE(BTRSEL,@BR) SET STMT TABLE MAXIMUM ENTRY
241 *
1E85 C2 02 07DA      242          LA     B$SDPL,@XR      LOAD STATEMENT TABLE DPL CADDR
1E89 C0 87 1A6B      243          B     B$DL4T      LINK TO WRITE STMT TABLE BUFF
244 *
1E8D C0 87 0025      245          B     $DISKN      LINK TO WAIT OUTPUT COMPLETED
1E91 057F          1E92 246          DC    AL(@CADDR)($WAITF) CADDR OF DISK IOCR 'WAIT' DPL
248 *****
249 * ESTABLISH CRITICAL COMPILER-GENERATED VIRTUAL ADDRESSES FOR LOADER
250 *****
251 *
252 * CLEAR THE VIRTUAL MEMORY REGION INDICATOR AREAS
253 *
1E93 0F 07 1A07 1A07 254 BTR150 SLC   B$LDRP+B@DL04,B$LDRP+B@DL04(4*@VADDR) CLEAR REGION ADDRS
255 *
256 * ESTABLISH VIRTUAL MEMORY REGION-1 BEGINNING ADDRESS
257 *
1E99 0C 00 1A00 0A42 258 BTR160 MVC   B$LDRP+B@DL01-1,B$PVAD-1(@VADDR-1) SET UP PAGE AFTER PMC
259 *
260 * ESTABLISH VIRTUAL MEMORY REGION-1 ENDING ADDRESS
261 *
1E9F 0C 00 1A02 0A35 262 BTR170 MVC   B$LDRP+B@DL02-1,B$PCPG(@VADDR-1) SET UP LOW CONSTANT PAGE
263 *
264 * ESTABLISH VIRTUAL MEMORY REGION-2 BEGINNING ADDRESS
265 *
1EA5 1E 01 0E46 E9   266 BTR180 ALC   B$SVRB,BTRVBA(@VADDR,@BR) ADJUST VARIABLE BASE VADDR
267 *
1EAA 0C 00 1A04 0E45 268          MVC   B$LDRP+B@DL03-1,B$SVRB-1(@VADDR-1) SET UP PAGE AFTER VARS
269 *
270 * ESTABLISH VIRTUAL MEMORY REGION-2 ENDING ADDRESS
271 *
1EB0 0C 00 1A06 0E47 272 BTR190 MVC   B$LDRP+B@DL04-1,B$SFAB-1(@VADDR-1) SET UP LOW NAT PAGE
273 *
274 * ESTABLISH VIRTUAL ADDRESSES FOR SYSTEM INTERNAL ELEMENTS

```

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 163

```

275 *
1EB6 0C 01 1A09 15AC 276 BTR200 MVC B$LDRP+B@DL05,B$FAIS(@VADDR) SET UP 1ST CONSTANT VADDR
1EBC 0C 01 1A0B 15A0 277 MVC B$LDRP+B@DL06,B$FAIW(@VADDR) SET UP 1ST VARIABLE VADDR

279 *****
280 * TERMINATOR 2ND SEGMENT CALLING SEQUENCE ROUTINE
281 *****
282 *
283 * TEST WHETHER CURRENT SEGMENT WAS DISK OR CORE RESIDENT
284 *
N04 1EC2 00 00 00 00 285 BTR250 CLC BTRCA2(,@BR),BTRPBA(@CADDR,@BR) IF CURR SEG CAME FR DISK
1EC6 F2 81 10 286 JE BTR280 * GO LOAD & EXEC 2ND SEG
287 *
288 * CURRENT SEGMENT WAS CORE RESIDENT - TEST WHETHER 2ND SEGMENT HAS
289 * ALSO BEEN LOADED INTO CORE
290 *
1EC9 4E 00 FD 043B 291 BTR260 ALC BTRFCP-1(,@BR),$EXFTR(1) CALC MAX PROCESSOR CORE PAGE
1ECE 5D 01 FB FE 292 CLC BTRCA2(,@BR),BTRFCP(@CADDR,@BR) IF 2ND SEGMENT IN CORE
1ED2 F2 82 0B 293 JL BTR290 * GO SET TO EXEC 2ND SEG
294 *
295 * 2ND SEGMENT IS DISK RESIDENT - ESTABLISH DISTRIBUTOR PARAMETERS FOR
296 * CORELOADING AND EXECUTING THE 2ND SEGMENT
297 *
N04 1ED5 00 00 00 00 298 BTR270 MVC BTRCA2(,@BR),BTRPBA(@CADDR,@BR) SET UP DISKLOAD CADDR
299 *
300 * EXIT TO DISTRIBUTOR FOR 2ND SEGMENT CORELOAD AND EXECUTION
301 *
1ED9 D2 02 FA 302 BTR280 LA BTRAD2(,@BR),@XR LOAD DISTRIBUTOR PARM CADDR
1EDC C0 87 073A 303 B B$DST2 GO LOAD & EXECUTE 2ND SEGMENT
304 *
305 * 2ND SEGMENT IS CORE RESIDENT - BRANCH TO NEXT CONSECUTIVE CORE PAGE
306 * AND CONTINUE TERMINATOR EXECUTION
307 *
1EE0 76 01 E7 308 BTR290 A BTRBLS(,@BR),@BR SET 2ND SEGMENT BASE CORE ADDR
1EE3 D0 87 00 309 B BTRSG2(,@BR) GO EXECUTE THE 2ND SEGMENT

311 *****
312 * COMPILER TERMINATOR SEGMENT-1 CONSTANTS
313 *****
314 *
1EE6 0100 1EE7 315 BTRBLS DC AL(@CADDR)(B@BLSZ) LENGTH OF CORE BLOCK OR PAGE
1EE8 00FF 1EE9 316 BTRVBA DC AL(@VADDR)(B@BLSZ-1) REGION-2 VIRTUAL ADDR ADJUSTER
1EEA 0600 1EEB 317 BTRPRA DC AL(@CADDR)(B$CSBF) PROCESSOR DISK BUFFER CADDR
318 *
1EEC 1B0E 1EED 319 BTRFTA DC AL(@CADDR)(B$FORT) CADDR OF 1ST 'FOR' TABLE ENTRY
320 *
1CFF 321 BTRSHA EQU B$SABF+B@BLSZ-1 CADDR OF STMT TBL BFR RH BYTE
0004 322 BTRSEL EQU @VADDR+B@LSNO LENGTH OF A STATEMENT TBL ENTRY
1EEE FFFFFFFF 1EF1 323 BIRSHE DC XL(BTRSEL)'FFFFFFFF' MAXIMUM ENTRY FOR STMT TABLE

325 *****
326 * COMPILER TERMINATOR SEGMENT-1 DISK PARAMETER LIST
327 *****
328 *
1EF2 01 1EF2 329 BTREPL EQU * ERROR STACK CORELOAD DPL ADDR
1EF2 330 BTREFN DC AL1(@DGET) DISK IOCR 'READ' FUNCTION

```

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 164

	1EF3 07	1EF3	331	BTRECY DC	AL1(B@DVCY)	ERROR STACK BASE CYLINDER ADDR
	1EF4 56	1EF4	332	BTRESA DC	AL1(B@DVC1)	ERROR STACK 1ST LOGICAL SECTOR
	1EF5 03	1EF5	333	BTRESC DC	IL1 '3'	SECTOR COUNT FOR THE ERR STACK
N04	1EF6 0000	1EF7	334	BTRECA DC	AL(@CADDR)(\$\$ERSK)	ERROR STACK CORELOAD ADDRESS
		336	*****			
		337	* COMPILER TERMINATOR PSEUDO INSTRUCTION SEQUENCE			
		338	*****			
		339	*			
		1EF8	340	BTRPCA EQU *		CADDR OF ENDING PMC SEQUENCE
			341	*		
	1EF8 02	1EF8	342	BTRSVC DC	AL(B@LCOP)(B@CSVC)	'SUPERVISOR CALL' PSEUDO OPCODE
	1EF9 70	1EF9	343	BTREOF DC	AL(B@LCOP)(B@CEOF)	'END-OF-FILE' PSEUDO OPCODE
		345	*****			
		346	* COMPILER TERMINATOR SEGMENT-1 MORK AREAS			
		347	*****			
		348	*			
		1EFA	349	BTRAD2 EQU *		DISTR PARMS FOR SEG-2 EXEC
1EFA		1EFB	350	BTRCA2 DS	CL(@CADDR)	TERMINATOR SEGMENT CORE ADDRESS
1EFC 5C		1EFC	351	BTRSA2 DC	AL1(B@DEND+BTRPSI)	BTRMNT SEG-2 PHYS SECTOR ADDR
			352	*		
	1EFD	1EFE	353	BTRFCP DS	CL(@CADDR)	FINAL AVAILABLE CORE PAGE ADDR
	1EFD		354	ORG	*-@CADDR	INITIALIZE CORE PAGE ADDR TO
	1EFD 1F00	1EFE	355	DC	AL(@CADDR)(B\$CSXA-B@BLSZ)	* FINAL PAGE BEFORE EXTENSION
		357	*****			
		358	* COMPILER TERMINATOR SECOND SEGMENT			
		359	*****			
		360	*			
		361	* ESTABLISH TERMINATOR SEGMENT-2 ADDRESSABILITY			
		362	*			
1F00			363	ORG	BTRMNT+B@BLSZ	BEGIN SEGMENT-2 AT PAGE BOUND
		1F00	364	USING *	@BR	DEFINE SEGMENT-2 BASE ADDRESS
			365	*		
			366	* ESTABLISH LETTER VARIABLE SYMBOL TABLE FOR THE LOADER		
			367	*		
	1F00 0C 39 1A45 109B		368	BTR300 MVC	B\$LDRP+B@DL07,B\$SLVT+B@LL07-1(B@LL07)	SET UP LTR VAR TBL
			369	*		
			370	* ESTABLISH LETTER-DIGIT VARIABLE SYMBOL TABLE FOR THE LOADER		
			371	*		
N04	1F06 00 00 0000 0000		372	BTR310 MVC	B\$IDRP+B@DL08,B\$SLDT+B@LL08-1(B@LL08)	SET UP LTR-
N04	1F0C 00 00 0000 0000		373	MVC	B\$LDRP+B@DL09,B\$SLDT+B@LL08+B@LL09-1(B@LL09)	* DIGIT TFIL
	1F12 0C 43 1C89 12DF		374	MVC	B\$LDRP+B@DL10,B\$SLDT+B@LL08+B@LL09+B@LL10-1(B@LL10)	
			375	*		
			376	* ESTABLISH CHARACTER VARIABLE SYMBOL TABLE FOR THE LOADER		
			377	*		
	1F18 0C 39 1CC3 1319		378	BTR320 MVC	B\$LDRP+B@DL11,B\$SCVT+B@LL11-1(B@LL11)	SET UP CHAR VAR TBL
			379	*		
			380	* CLEAR THE FUNCTION AND ARRAY TABLE AREA FOR THE LOADER		
			381	*		
	1F1E 0F FF 1E71 1E71		382	BTR330 SLC	B\$LDRP+B@DL15,B\$LDRP+B@DL15(B@LL15)	INITLZ THE FUNC AND
	1F24 0F 95 1F07 1F07		383	SLC	B\$LDRP+B@DL16,B\$LDRP+B@DL16(B@LL16)	* ARRAY AREA TO ZEROS
		385	*****			
		386	* ESTABLISH ARITHMETIC ARRAY SYMBOL TABLE AND DOPE VECTORS FOR LOADER *			

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 165
				387	*****	
				388	*	
				389	* GET AN ENTRY FROM THE COMPILE-TIME ARITHMETIC (NUMERIC) ARRAY TABLE	
				390	*	
	1F2A	75 02 CA		391	BTR350 L BTRCNP(, @BR), @XR LOAD COMPILE-TIME NAT POINTER	
	1F2D	6C 05 C8 05		392	MVC BTRCNE(, @BR), @VADDR+B@ACD2(B@LCNA, @XR) SAVE THE NAT ENTRY	
				393	*	
				394	* ESTABLISH A LOADER-TIME NUMERIC ARRAY TABLE VIRTUAL ADDRESS ENTRY	
				395	*	
	1F31	C2 02 1CC3		396	BTR360 LA B\$LDRP+B@DL11, @XR LOAD LOADER-TIME NAT BASE ADDR	
				397	*	
	1F35	9C 01 00 C4		398	BTR370 MVC *-*(, @XR), BTRVAD(@VADDR, @BR) HOVE THE ARRAY VADDR INTO	
	1F37			399	ORG BTR370+@D1 * LOADER-TIME NAT ENTRY	
	1F37	3A	1F37	400	DC AL1(B@LL12) INITIALIZE LOADER-TIME NAT	
	1F39			401	ORG BTR370+@INST4 * POINTER TO RIGHTMOST ENTRY	
				402	*	
				403	* TEST WHETHER CURRENT ENTRY ARRAY WAS REFERENCED IN PROGRAM	
				404	*	
	1F39	7D 56 C3		405	BTR380 CLI BTRVAD-1(, @BR), B@DVC1 IF ARRAY WAS NOT REFERENCED	
	1F3C	F2 82 0A		406	JL BTR400 * SKIP PAST FAT PROCESSING	
				407	*	
				408	* ESTABLISH A FUNCTION AND ARRAY TABLE DOPE VECTOR FOR CURRENT ENTRY	
				409	*	
	1F3F	75 02 C4		410	BTR390 L BTRVAD(, @BR), @XR LOAD THE ARRAY VIRTUAL ADDRESS	
	1F42	76 02 B5		411	A BTRFAC(, @BR), @XR CONVERT THE VADDR TO A CADDR	
	1F45	9C 03 03 C8		412	MVC B@ACD2(, @XR), BTRCND(2*B@LDMN, @BR) SET DOPE VECTOR DIMENS	
				413	*	
				414	* DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS	
				415	*	
	1F49	5F 01 CA B7		416	BTR400 SLC BTRCNP(, @BR), BTRCNL(@CADDR, @BR) DECR COMPILE-TIME NAT PT	
	1F4D	5F 00 37 BC		417	SLC BTRNTP(, @BR), BTRSTL(1, @BR) DECR LOADER-TIME NAT PT	
	1F51	D0 84 2A		418	BH BTR350(, @BR) IF MORE NAT ENTRIES, GO PROCESS	
				420	*****	
				421	* ESTABLISH CHARACTER ARRAY SYMBOL TABLE AND DOPE VECTORS FOR LOADER	
				422	*****	
				423	*	
				424	* GET AN ENTRY FROM THE COMPILE-TIME CHARACTER ARRAY TABLE	
				425	*	
	1F54	75 02 CC		426	BTR410 L BTRCCP(, @BR), @XR LOAD COMPILE-TIME CAT POINTER	
N04	1F57	00 00 00 00		427	MVC BTRCCE(, @BR), @VADDR+B@CDMN(B@LCCA, @XR) SAVE THE CAT ENTRY	
				428	*	
				429	* ESTABLISH A LOADER-TIME CHARACTER ARRAY TABLE VIRTUAL ADDRESS ENTRY	
				430	*	
N04	1F5B	00 00 0000		431	BTR420 LA B\$LORP+B@DL12, @XR LOAD LOADER-TIME CAT BASE ADDR	
				432	*	
	1F5F	9C 01 00 C4		433	BTR430 MVC *-*(, @XR), BTRVAD(@VADDR, @BR) MOVE THE ARRAY VADDR INTO	
	1F61			434	ORG BTR430+@D1 * LOADER-TIME CAT ENTRY	
N04	1F61	00	1F61	435	DC AL1(B@LL1X) INITIALIZE LOADER-TIME CAT	
	1F63			436	ORG BTR430+@INST4 CHECK OBJ * POINTER TO RIGHTMOST ENTRY	
				437	*	
				438	* TEST WHETHER CURRENT ENTRY ARRAY WAS REFERENCED IN PROGRAM	
				439	*	
	1F63	7D 56 C3		440	BTR440 CLI BTRVAD-1(, @BR), B@DVC1 IF ARRAY WAS NOT REFERENCED	
	1F66	F2 82 0A		441	JL BTR460 * SKIP PAST FAT PROCESSING	
				442	*	

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 166

```

443 * ESTABLISH A FUNCTION AND ARRAY TABLE DOPE VECTOR FOR CURRENT ENTRY
444 *
1F69 75 02 C4 445 BTR450 L BTRVAD(, @BR), @XR LOAD THE ARRAY VIRTUAL ADDRESS
1F6C 76 02 B5 446 A BTRFAC(, @BR), @XR CONVERT THE VADDR TO A CADDR
N04 1F6F 00 00 00 00 447 MVC B@CDMN(, @XR), BTRCCD(B@LDMN, @BR) SET DOPE VECTOR DIMENSION
448 *
449 * DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS
450 *
1F73 5F 01 CC B9 451 BTR460 SLC BTRCCP(, @BR), BTRCCL(@CADDR, @BR) DECR COMPILE-TIME CAT PT
1F77 5F 00 61 BC 452 SLC BTRCTP(, @BR), BTRSTL(1, @BR) DECR LOADER-TIME CAT PT
1F7B D0 84 54 453 BH BTR410(, @BR) IF MORE CAT ENTRIES, GO PROCESS

455 *****
456 * ESTABLISH USER FUNCTION SYMBOL TABLE AND ADDRESSES FOR LOADER
457 *****
458 *
459 * GET AN ENTRY FROM THE COMPILE-TIME USER FUNCTION TABLE
460 *
1F7E 75 02 CE 461 BTR470 L BTRCFP(, @BR), @XR LOAD COMPILE-TIME FNT POINTER
1F81 6C 03 C6 03 462 MVC BTRCFE(, @BR), @VADDR+B@FVAD(B@LCFN, @XR) SAVE THE FNT ENTRY
463 *
464 * ESTABLISH A LOADER-TIME USER FUNCTION TABLE VIRTUAL ADDRESS ENTRY
465 *
1F85 C2 02 1D37 466 BTR480 LA B$LDRP+B@DL13, @XR LOAD LOADER-TIME FNT BASE ADDR
467 *
1F89 9C 01 00 C4 468 BTR490 MVC *-*(, @XR), BTRVAD(@VADDR, @BR) MOVE THE FUNCTION VADDR
1F8B 469 ORG BTR490+@D1 * INTO LOADER-TIME FNT ENTRY
1F8B 3A 1F8B 470 DC AL1(B@LL14) INITIALIZE LOADER-TIME FNT
1F8D 471 ORG BTR490+@INST4 * POINTER TO RIGHTMOST ENTRY
472 *
473 * TEST WHETHER CURRENT ENTRY FUNCTION WAS REFERENCED IN PROGRAM
474 *
1F8D 7D 56 C3 475 BTR500 CLI BTRVAD-1(, @BR), B@DVC1 IF FUNCTION WAS NOT REFERENCED
1F90 F2 82 0A 476 JL BTR520 * SKIP PAST FAT PROCESSING
477 *
478 * ESTABLISH A FUNCTION AND ARRAY TABLE ADDRESS FOR CURRENT ENTRY
479 *
1F93 75 02 C4 480 BTR510 L BTRVAD(, @BR), @XR LOAD THE FUNCTION VIRTUAL ADDR
1F96 76 02 B5 481 A BTRFAC(, @BR), @XR CONVERT THE VADDR TO A CADDR
N04 1F99 00 00 00 00 482 MVC BAFVAD(, @XR), BTRCFA(@VADDR, @BR) SET FUNCTION VIRTUAL ADDR
483 *
484 * DECREMENT TABLE POINTERS AND TEST FOR MORE ENTRIES TO PROCESS
485 *
1F9D 5F 01 CE BB 486 BTR520 SLC BTRCFP(, @BR), BTRCFL(@CADDR, @BR) DECR COMPILE-TIME FNT PT
1FA1 5F 00 8B BC 487 SLC BTRFTP(, @BR), BTRSTL(1, @BR) DECR LOADER-TIME FNT PT
1FA5 D0 84 7E 488 BH BTR470(, @BR) IF MORE FNT ENTRIES, GO PROCESS

```



## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR LOC OBJECT CODE ADDR STMT SOURCE STATEMENT VER 15, MOD 00 20/07/20 PAGE 167

```

490 *****
491 * NORMAL COMPILER EXIT ROUTINE
492 *****
493 *
494 * LOAD AND TRANSFER CONTROL TO THE BASIC LOADER
495 *
N04 1FA8 D2 02 BD 496 BTR600 LA BTRDPL(,@BR),@XR STORE LOADER CORELOAD DPL ADDR
1FAB 00 00 00 497 ST BIRDPA(,@BR),@XR * FOR SYSTEM LOADER PARAMETER
1FAE C0 87 051E 498 B $RLOAD EXIT THE COMPILER
1FB2 1FB3 499 BTRDPA DS CL(@CADDR) LOADER CORELOAD DPL ADDRESS

501 *****
502 * COMPILER TERMINATOR SEGMENT-2 CONSTANTS
503 *****
504 *
1FB4 1F08 1FB5 505 BTRFAC DC AL(@CADDR)(B$LDRP+B@DL16+1) FUNC & ARRAY ADDR CONVERTER
506 *
1FB6 0006 1FB7 507 BTRCNL DC AL(@CADDR)(B@LCNA) COMPILE-TIME NAT ENTRY LENGTH
1FB8 0004 1FB9 508 BTRCCL DC AL(@CADDR)(B@LCCA) COMPILE-TIME CAT ENTRY LENGTH
1FBA 0004 1FBB 509 BTRCFL DC AL(@CADDR)(B@LCFN) COMPILE-TIME FNT ENTRY LENGTH
510 *
1FBC 02 1FBC 511 BTRSTL DC AL1(@VADDR) LOADER-TIME SYM TBL ENTRY LNG

513 *****
514 * COMPILER TERMINATOR SEGMENT-2 DISK PARAMETER LIST
515 *****
516 *
N04 1FBD 00 1FBD 517 *TRDPL $DPL FUNC-DGET,DADDR-#$LOAD,CNT-#$@LOA,CADDR-#$SLOA
1FBD 518+BTRDPL EQU * DISK PARAMETER LIST
N04 1FBE 0000 1FBD 519+ DC AL1(DGET) REQUESTED FUNCTION
N04 1FC0 00 1FBD 520+ DC AL2($LOAD) DISK ADDRESS
N04 1FC1 0000 1FC0 521+ DC AL1($@LOA) SECTOR COUNT
1FC2 522+ DC AL2($SLOA) BUFFER ADDRESS
523+*** END OF EXPANSION ***

525 *****
526 * COMPILER TERMINATOR SEGMENT-2 WORK AREAS
527 *****
528 *
1FC3 1FC3 529 BTRTEN EQU * COMPILE-TIME FUNCTION & ARRAY
1FC8 530 DS CL(B@LCNA) * SYMBOL TABLES ENTRY SAVE AREA
531 *
1FC9 1FCA 532 BTRCNP DS CL(@CADDR) COMPILE-TIME NAT POINTER -
1FC9 533 ORG *-@CADDR * INITLZ TO THE
1FC9 13C2 1FCA 534 DC AL(@CADDR)(B$SNAT+B@NAAR*B@LCNA-B@LCNA) * RIGHTMOST ENTRY
535 *
1FCB 1FCC 536 BTRCCP DS CL(@CADDR) COMPILE-TIME CAT POINTER -
1FCB 537 ORG *-@CADDR * INITLZ TO THE
1FCB 1438 1FCC 538 DC AL(@CADDR)(B$SCAT+B@NCAR*B@LCCA-B@LCCA) * RIGHTMOST ENTRY
539 *
1FCD 1FCE 540 BTRCFP DS CL(@CADDR) COMPILE-TIME FNT POINTER -
1FCD 541 ORG *-@CADDR * INITLZ TO THE
1FCD 14AC 1FCE 542 DC AL(@CADDR)(B$SFNT+B@NUFN*B@LCFN-B@LCFN) * RIGHTMOST ENTRY

544 *****
545 * COMPILER TERMINATOR EQUATES REFERENCING CONSTANTS

```

## S/3 BASIC COMPILER TERMINATION ROUTINE

ERR	LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 168
			546	*****		
			547	*		
		0000	548	BTRSG2 EQU 0	DISP FOR BTRMNT SEG-2 ENTRY PT	
		0004	549	BTRPSI EQU X'04'	PHYSICAL SECTOR ADDR INCREMENT	
		00FF	550	BTRBND EQU B@BLSZ-1	DISP INDICATING EMPTY CON BFR	
			552	*****		
			553	* COMPILER TERMINATOR EQUATES REFERENCING PROGRAM LABELS		
			554	*****		
			555	*		
		1FC4	556	BTRVAD EQU BTRTEN+@VADDR-1	COMPILE-TIME FIA SYMBOL VADDR	
		1FC8	557	BTRCNE EQU BTRTEN+@VADDR+B@ACD2	COMPILE-TIME NAT ENTRY ADDR	
		1FC8	558	BTRCND EQU BTRCNE	COMPILE-TIME NAT ENTRY DINERS	
N04			559	BTRCCE EQU BTRTEN+@VADDR+B\$CDMN	COMPILE-TIME CAT ENTRY ADDR	
N04			560	BTRCCD EQU BTRCCE	COMPILE-TIME CAT ENTRY DIMEN	
		1FC6	561	BTRCFE EQU BTRTEN+@VADDR+B@FVAD	COMPILE-TIME FNT ENTRY ADDR	
		1FC6	562	BTRCFA EQU BTRCFE	COMPILE-TIME FNT ENTRY VADDR	
			563	*		
		1F37	564	BTRNTP EQU BTR370+@D1	LOADER-TIME NAT POINTER DISP	
		1F61	565	BTRCTP EQU BTR430+@D1	LOADER-TIME CAT POINTER DISP	
		1F8B	566	BTRFTP EQU BTR490+@D1	LOADER-TIME FNT POINTER DISP	
			567	*		
			568	*****		
			569	*		
			570	* END OF COMPILER TERMINATOR CODING		
			571	*		



## S/3 BASIC COMPILER -RETURN- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00	20/07/20	PAGE 169
		573		*****			
		574	*	5703-XM1 COPYRIGHT IBM CORP. 1970			*
		575	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083			*
		576	*				*
		577		*****			*
		578	*	*STATUS			*
		579	*	VERSION 1 MODIFICATION 0			*
		580	*				*
		581	*	*FUNCTION			*
		582	*	BKRTRN IS EXECUTED TO TRANSLATE RETURN STATEMENTS AS THEY OCCUR			*
		583	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE			*
		584	*	THE PSEUDOCODE IN VIRTUAL MEMORY.			*
		585	*				*
		586	*	*ENTRY POINTS			*
		587	*	BKRTRN HAS OILY ONE ENTRY POINT:			*
		588	*	BKRTRN - TRANSLATE RETURN STATEMENT			*
		589	*	THE FORMAT OF THE CALLING SEQUENCE:			*
		590	*	B BKRTRN			*
		591	*				*
		592	*	* INPUT			*
		593	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING			*
		594	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE			*
		595	*	LEADING KEYWORD, RETURN.			*
		596	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST			*
		597	*	CHARACTER IN THE LEADING KEYWORD, RETURN.			*
		598	*				*
		599	*	*OUTPUT			*
		600	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE			*
		601	*	GENERATED BY BKRTRN IS STORED IN THE NEXT AVAILABLE VIRTUAL			*
		602	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION			*
		603	*	SEQUENCES.			*
		604	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE			*
		605	*	CHARACTER WHICH TERMINATES THE STATEMENT.			*
		606	*				*
		607	*	*EXTERNAL REFERENCES			*
		608	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY			*
		609	*	OUTPUT ROUTINE.			*
		610	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.			*
		611	*				*
		612	*	*EXITS, NORMAL			*
		613	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.			*
		614	*				*
		615	*	*EXITS, ERROR			*
		616	*	N/A			*
		617	*				*
		618	*	*TABLES/WORK AREAS			*
		619	*	N/A			*
		620	*				*
		621	*	*ATTRIBUTES			*
		622	*	BKRTRN IS NATURALLY RELOCATABLE AND REUSABLE.			*
		623	*				*
		624	*	*CHARACTER CODE DEPENDENCY			*
		625	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR			*
		626	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.			*
		627	*				*
		628	*	*NOTES			*

## S/3 BASIC COMPILER -RETURN- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 170
		629	*	ERROR PROCEDURES				*
		630	*	N/A				*
		631	*					*
		632	*	REGISTER USAGE				*
		633	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
		634	*					*
		635	*	SAVED/RESTORED AREAS				*
		636	*	N/A				*
		637	*					*
		638	*	MODIFICATION CONSIDERATIONS				*
		639	*	BKRTRN RESIDES ON THE SAME SECTOR WITH BTRMNT AND BPXRSR.	1-4*			
		640	*	ANY MODIFICATION TO BKRTRN MUST CONSIDER THIS CO-RESIDENCY	1-4*			
		641	*	SINCE IT WILL CHANGE THE ENTRY ADDRESS OF BPXRSR. THE	1-4*			
		642	*	LIMITATION OF THE SECTOR BOUNDARY ON SIZE MUST ALSO BE	1-4*			
		643	*	CONSIDERED.	1-4*			
		644	*					*
		645	*	REQUIRED MODULES				*
		646	*	@NYSEQ - COMMON SYSTEM EQUATES.				*
		647	*	@FXDEQ - SYSTEM NUCLEUS AND INDICATOR EQUATES.				*
		648	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
		649	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
		650	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
		651	*	@ERMEQ - ERROR MESSAGE EQUATES.				*
		652	*	\$VSEQU - FIXED VIRTUAL ADDRESS EQUATES.				*
		653	*	\$B\$EQU - COMPILER FIXED EQUATES.				*
		654	*	\$B@EQU - COMPILER SYSTEM EQUATES.				*
		655	*					*
		656	*	OTHER				*
		657	*	BKRTRN IS ASSEMBLED WITH ALL OF THE STATEMENT PROCESSORS.				*
		658	*	*****				
		660	*					
		661	*	ENTER BKRTRN - 'RETURN' STATEMENT ROUTINE				
		662	*					
	1FCF	663	BKRTRN EQU *		BKRTRN ENTRY POINT			
		664	*					
		665	*	GENERATE A 'BRS' INSTRUCTION IN VIRTUAL MEMORY				
		666	*					
1FCF D2 02 E2		667	BKR010 LA	BKRBRC(,@BR),@XR	LOAD CADDR OF 'BRS' INSTR			
1FD2 34 02 0A40		668	ST	B\$PCAD,@XR	SET PUT RTN FOR VADDR OF 'BRS'			
1FD6 3C 00 0A41		669	MVI	B\$PNBY,B@LBRS-1	SET PUT RTN FOR LENGTH OF 'BRS'			
1FDA C0 87 093A		670	B	B\$PUTC	LINK TO GENERATE PMC			
		671	*					
		672	*	RETURN CONTROL TO THE REM STATEMENT ROUTINE				
		673	*					
1FDE C0 87 1AE6		674	BKR020 B	B\$RMRK	RETURN TO REMARK STMT RTN			
		675	*					
		676	*	*****				
		677	*	'RETURN' STATEMENT ROUTINE PMC AND STORAGE PARAMETERS				
		678	*	*****				
		679	*					
N04 1FE2 00	1FE2	680	BKRBRC DC	AL(B@LCOP)(B\$CBRS)	'BRS' INSTR OPCODE			
		681	*					
		682	*	*****				
		683	*					
		684	*	END OF 'RETURN' STATEMENT ROUTINE CODING				



## S/3 BASIC COMPILER -RESTORE- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE STATEMENT	VER 15, MOD 00 20/07/20 PAGE 172
		687		*****	
		688	*	5703-XM1 COPYRIGHT IBM CORP. 1970	*
		689	*	REFER TO INSTRUCTIONS ON COPY RIGHT NOTICE, 120-2083	*
		690	*		*
		691		*****	
		692	*	STATUS	*
		693	*	VERSION 1 MODIFICATION 0	*
		694	*		*
		695	*	FUNCTION	*
		696	*	BPXRSR IS EXECUTED TO TRANSLATE RESTORE STATEMENTS AS THEY OCCUR	*
		697	*	IN A BASIC PROGRAM INTO THE APPROPRIATE PSEUDOCODE AND TO PLACE	*
		698	*	THE PSEUDOCODE IN VIRTUAL MEMORY.	*
		699	*		*
		700	*	ENTRY POINTS	*
		701	*	BPXRSR HAS ONLY ONE ENTRY POINT:	*
		702	*	BPXRSR - TRANSLATE RESTORE STATEMENT	*
		703	*	THE FORMAT OF THE CALLING SEQUENCE IS:	*
		704	*	B BPXRSR	*
		705	*		*
		706	*	INPUT	*
		707	*	* COMPILER INPUT BUFFER - CONTAINS SOURCE PROGRAM TEXT INCLUDING	*
		708	*	THAT RECORD SEGMENT CONTAINING THE FIRST CHARACTER OF THE	*
		709	*	LEADING KEYWORD, RESTORE.	*
		710	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE FIRST	*
		711	*	CHARACTER IN THE LEADING KEYWORD, RESTORE.	*
		712	*		*
		713	*	OUTPUT	*
		714	*	* VIRTUAL MEMORY - THE PSEUDO MACHINE INSTRUCTION SEQUENCE	*
		715	*	GENERATED BY BPXRSR IS STORED IN THE NEXT AVAILABLE VIRTUAL	*
		716	*	MEMORY LOCATION FOLLOWING PREVIOUSLY STORED INSTRUCTION	*
		717	*	SEQUENCES.	*
		718	*	* TEXT CHARACTER POINTER - CONTAINS THE CORE ADDRESS OF THE	*
		719	*	CHARACTER WHICH TERMINATES THE STATEMENT.	*
		720	*		*
		721	*	EXTERNAL REFERENCES	*
		722	*	B\$PUTC - (B\$PCAD, B\$PNBY) - ENTRY TO COMPILER VIRTUAL MEMORY	*
		723	*	OUTPUT ROUTINE.	*
		724	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.	*
		725	*		*
		726	*	EXITS, NORMAL	*
		727	*	B\$RMRK - ENTRY TO BASIC COMPILER REMARK ROUTINE.	*
		728	*		*
		729	*	EXITS, ERROR	*
		730	*	N/A	*
		731	*		*
		732	*	TABLES/WORK AREAS	*
		733	*	N/A	*
		734	*		*
		735	*	ATTRIBUTES	*
		736	*	BPXRSR IS NATURALLY RELOCATABLE AND REUSABLE.	*
		737	*		*
		738	*	CHARACIER CODE DEPENDENCY	*
		739	*	THE OPERATION OF THIS MODULE DOES NOT DEPEND UPON A PARTICULAR	*
		740	*	INTERNAL REPRESENTATION OF THE EXTERNAL CHARACTER SET.	*
		741	*		*
		742	*	NOTES	*

## S/3 BASIC COMPILER -RESTORE- ROUTINE

ERR LOC	OBJECT CODE	ADDR	STMT	SOURCE	STATEMENT	VER 15, MOD 00	20/07/20	PAGE 173
		743	*	ERROR PROCEDURES				*
		744	*	N/A				*
		745	*					*
		746	*	REGISTER USAGE				*
		747	*	BOTH THE INDEX AND BASE REGISTERS ARE USED DURING EXECUTION.				*
		748	*					*
		749	*	SAVED/RESTORED AREAS				*
		750	*	N/A				*
		751	*					*
		752	*	MODIFICATION CONSIDERATIONS				*
		753	*	BPXRSR RESIDES ON THE SAME SECTOR WITH BTRMNT AND BKRTN.				*
		754	*	ANY MODIFICATION TO BPXRSR MUST TAKE INTO CONSIDERATION				*
		755	*	THIS CO RESIDENCY ANY ALSO THE LIMITATION OF THE SECTOR				*
		756	*	BOUNDARY ON SIZE.				*
		757	*					*
		758	*	REQUIRED MODULES				*
		759	*	@NYSEQ - COMMON SYSTEM EQUATES.				*
		760	*	@FXDEQ - SYSTEM NUCLEUS AND INDICATOR EQUATES.				*
		761	*	@CANEQ - COMMON CORE LOCATIONS OUTSIDE NUCLEUS EQUATES.				*
		762	*	@VMDEQ - VIRTUAL MEMORY DIRECTORY EQUATES.				*
		763	*	@SPFEQ - SYSTEM PROGRAM FILE EQUATES.				*
		764	*	@ERMEQ - ERROR MESSAGE EQUATES.				*
		765	*	\$VSEQU - FIXED VIRTUAL ADDRESS EQUATES.				*
		766	*	\$B\$EQU - COMPILER FIXED EQUATES.				*
		767	*	\$B@EQU - COMPILER SYSTEM EQUATES.				*
		768	*					*
		769	*	OTHER				*
		770	*	BPXRSR IS ASSEMBLED WITH ALL THE STATEMENT PROCESSORS.				*
		771	*	*****				*
		773	*					*
		774	*	ENTER BPXRSR	'RESTORE' STMT ROUTINE			*
		775	*					*
		1FE3	776	BPXRSR EQU	*			BPXRSR ENTRY POINT
			777					*
			778		GENERATE AN 'RSR' INSTRUCTION PMC IN VIRTUAL MEMORY			*
			779					*
1FE3 D2 02 F6			780	BPX010 LA	BPXRSC(,@BR),@XR			LOAD CADDR OF 'RSR' INSTR
1FE6 34 02 0A40			781	ST	B\$PCAD,@XR			SET PUT RTN VADDR FOR 'RSR'
1FEA 3C 00 0A41			782	MVI	B\$PNBY,B@LRSR-1			SET PUT RTN LNG CODE FOR 'RSR'
1FEE C0 87 093A			783	B	B\$PUTC			LINK TO GENERATE 'RSR' PMC
			784	*				*
			785	*	RETURN CONTROL TO THE REMARK ROUTINE			*
			786	*				*
1FF2 C0 87 1AE6			787	BPX020 B	B\$RMRK			*
			788	*				*
			789	*	*****			*
			790	*	'RESTORE' STATEMENT ROUTINE PARAMETER AND STORAGE AREA			*
			791	*	*****			*
			792	*				*
1FF6 5A		1FF6	793	BPXRSC DC	AL(B@LCOP)(B@CRSR)			'RSR' INSTR OPCODE
			794	*				*
			795	*	*****			*
			796	*				*
			797	*	END OF 'RESTORE' STATEMENT ROUTINE CODING			*
			798	*				*





## DIAGNOSTICS

VER 15, MOD 00 20/07/20 PAGE 175

STMT	ERROR CODE	MESSAGE
2781	N04	REFERENCE TO UNDEFINED SYMBOL
2847	N04	REFERENCE TO UNDEFINED SYMBOL
2847	P10	INVALID CONSTANT
2854	N04	REFERENCE TO UNDEFINED SYMBOL
2854	P10	INVALID CONSTANT
2911	N04	REFERENCE TO UNDEFINED SYMBOL
2917	N04	REFERENCE TO UNDEFINED SYMBOL
3082	N04	REFERENCE TO UNDEFINED SYMBOL
3118	N04	REFERENCE TO UNDEFINED SYMBOL
3132	N04	REFERENCE TO UNDEFINED SYMBOL
3330	N04	REFERENCE TO UNDEFINED SYMBOL
3381	N04	REFERENCE TO UNDEFINED SYMBOL
3491	N04	REFERENCE TO UNDEFINED SYMBOL
3493	N04	REFERENCE TO UNDEFINED SYMBOL
3534	N04	REFERENCE TO UNDEFINED SYMBOL
3550	N04	REFERENCE TO UNDEFINED SYMBOL
3550	P10	INVALID CONSTANT
3926	N04	REFERENCE TO UNDEFINED SYMBOL
3961	N04	REFERENCE TO UNDEFINED SYMBOL
3972	N04	REFERENCE TO UNDEFINED SYMBOL
4182	N04	REFERENCE TO UNDEFINED SYMBOL
4182	P10	INVALID CONSTANT
4202	N04	REFERENCE TO UNDEFINED SYMBOL
4232	N04	REFERENCE TO UNDEFINED SYMBOL
4264	N04	REFERENCE TO UNDEFINED SYMBOL
4314	N04	REFERENCE TO UNDEFINED SYMBOL
4314	P10	INVALID CONSTANT
4477	N04	REFERENCE TO UNDEFINED SYMBOL
4492	N04	REFERENCE TO UNDEFINED SYMBOL
4594	N04	REFERENCE TO UNDEFINED SYMBOL
4594	P10	INVALID CONSTANT
4620	N04	REFERENCE TO UNDEFINED SYMBOL
4673	N04	REFERENCE TO UNDEFINED SYMBOL
4853	N04	REFERENCE TO UNDEFINED SYMBOL
5054	N04	REFERENCE TO UNDEFINED SYMBOL
5327	N04	REFERENCE TO UNDEFINED SYMBOL
5333	N04	REFERENCE TO UNDEFINED SYMBOL
5397	N04	REFERENCE TO UNDEFINED SYMBOL
5398	N04	REFERENCE TO UNDEFINED SYMBOL
5399	N04	REFERENCE TO UNDEFINED SYMBOL
5571	N04	REFERENCE TO UNDEFINED SYMBOL
5604	N04	REFERENCE TO UNDEFINED SYMBOL
5623	N04	REFERENCE TO UNDEFINED SYMBOL
5644	N04	REFERENCE TO UNDEFINED SYMBOL
5653	N04	REFERENCE TO UNDEFINED SYMBOL
5728	N04	REFERENCE TO UNDEFINED SYMBOL
5728	P10	INVALID CONSTANT
5732	N04	REFERENCE TO UNDEFINED SYMBOL
5732	P10	INVALID CONSTANT
5745	N04	REFERENCE TO UNDEFINED SYMBOL
5745	P10	INVALID CONSTANT
5746	N04	REFERENCE TO UNDEFINED SYMBOL
5746	P10	INVALID CONSTANT
5750	N04	REFERENCE TO UNDEFINED SYMBOL
5750	P10	INVALID CONSTANT
5753	N04	REFERENCE TO UNDEFINED SYMBOL

## DIAGNOSTICS

VER 15, MOD 00 20/07/20 PAGE 176

STMT	ERROR CODE	MESSAGE
5753	P10	INVALID CONSTANT
5941	N04	REFERENCE TO UNDEFINED SYMBOL
5952	N04	REFERENCE TO UNDEFINED SYMBOL
6354	N04	REFERENCE TO UNDEFINED SYMBOL
6371	N04	REFERENCE TO UNDEFINED SYMBOL
6371	P10	INVALID CONSTANT
6516	N04	REFERENCE TO UNDEFINED SYMBOL
6536	N04	REFERENCE TO UNDEFINED SYMBOL
6540	N04	REFERENCE TO UNDEFINED SYMBOL
6551	N04	REFERENCE TO UNDEFINED SYMBOL
6583	N04	REFERENCE TO UNDEFINED SYMBOL
6584	N04	REFERENCE TO UNDEFINED SYMBOL
6595	N04	REFERENCE TO UNDEFINED SYMBOL
6620	N04	REFERENCE TO UNDEFINED SYMBOL
6621	N04	REFERENCE TO UNDEFINED SYMBOL
6647	N04	REFERENCE TO UNDEFINED SYMBOL
6652	N04	REFERENCE TO UNDEFINED SYMBOL
6652	P10	INVALID CONSTANT
6760	N04	REFERENCE TO UNDEFINED SYMBOL
6900	N04	REFERENCE TO UNDEFINED SYMBOL
6908	N04	REFERENCE TO UNDEFINED SYMBOL
7138	N04	REFERENCE TO UNDEFINED SYMBOL
7296	N04	REFERENCE TO UNDEFINED SYMBOL
7481	N04	REFERENCE TO UNDEFINED SYMBOL
7664	N04	REFERENCE TO UNDEFINED SYMBOL
7664	P10	INVALID CONSTANT
8001	N04	REFERENCE TO UNDEFINED SYMBOL
8033	N04	REFERENCE TO UNDEFINED SYMBOL
8033	P10	INVALID CONSTANT
8165	N04	REFERENCE TO UNDEFINED SYMBOL
8320	N04	REFERENCE TO UNDEFINED SYMBOL
8340	N04	REFERENCE TO UNDEFINED SYMBOL
8342	N04	REFERENCE TO UNDEFINED SYMBOL
8347	N04	REFERENCE TO UNDEFINED SYMBOL
8358	N04	REFERENCE TO UNDEFINED SYMBOL
8368	N04	REFERENCE TO UNDEFINED SYMBOL
8428	N04	REFERENCE TO UNDEFINED SYMBOL
8428	P10	INVALID CONSTANT
8431	N04	REFERENCE TO UNDEFINED SYMBOL
8431	P10	INVALID CONSTANT
8553	N04	REFERENCE TO UNDEFINED SYMBOL
8728	N04	REFERENCE TO UNDEFINED SYMBOL
8769	N04	REFERENCE TO UNDEFINED SYMBOL
8790	N04	REFERENCE TO UNDEFINED SYMBOL
8832	N04	REFERENCE TO UNDEFINED SYMBOL
8832	P10	INVALID CONSTANT
9117	N04	REFERENCE TO UNDEFINED SYMBOL
9134	N04	REFERENCE TO UNDEFINED SYMBOL
9174	N04	REFERENCE TO UNDEFINED SYMBOL
9179	N04	REFERENCE TO UNDEFINED SYMBOL
9463	N04	REFERENCE TO UNDEFINED SYMBOL
9470	N04	REFERENCE TO UNDEFINED SYMBOL
9583	N04	REFERENCE TO UNDEFINED SYMBOL
9583	P17	INVALID SYMBOL
9595	N04	REFERENCE TO UNDEFINED SYMBOL
9596	N04	REFERENCE TO UNDEFINED SYMBOL

DIAGNOSTICS

STMT      ERROR CODE      MESSAGE      VER 15, MOD 00    20/07/20    PAGE 177

9789	N04	REFERENCE TO UNDEFINED SYMBOL
9790	N04	REFERENCE TO UNDEFINED SYMBOL
9798	N04	REFERENCE TO UNDEFINED SYMBOL
240	N04	REFERENCE TO UNDEFINED SYMBOL
285	N04	REFERENCE TO UNDEFINED SYMBOL
298	N04	REFERENCE TO UNDEFINED SYMBOL
334	N04	REFERENCE TO UNDEFINED SYMBOL
334	P10	INVALID CONSTANT
372	N04	REFERENCE TO UNDEFINED SYMBOL
373	N04	REFERENCE TO UNDEFINED SYMBOL
427	N04	REFERENCE TO UNDEFINED SYMBOL
431	N04	REFERENCE TO UNDEFINED SYMBOL
435	N04	REFERENCE TO UNDEFINED SYMBOL
435	P10	INVALID CONSTANT
447	N04	REFERENCE TO UNDEFINED SYMBOL
482	N04	REFERENCE TO UNDEFINED SYMBOL
497	N04	REFERENCE TO UNDEFINED SYMBOL
519	N04	REFERENCE TO UNDEFINED SYMBOL
519	P10	INVALID CONSTANT
520	N04	REFERENCE TO UNDEFINED SYMBOL
520	P10	INVALID CONSTANT
521	N04	REFERENCE TO UNDEFINED SYMBOL
521	P10	INVALID CONSTANT
522	N04	REFERENCE TO UNDEFINED SYMBOL
522	P10	INVALID CONSTANT
559	N04	REFERENCE TO UNDEFINED SYMBOL
560	N04	REFERENCE TO UNDEFINED SYMBOL
680	N04	REFERENCE TO UNDEFINED SYMBOL
680	P10	INVALID CONSTANT

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY =    114

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 178

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$\$\$NLN	001	00A0	2564	0193 0230
\$\$ERSK	UNDEFINED	SYMBOL		0334
\$\$ZERO	001	0000	0221	0222 0224 0225 0226 0230
\$ABORT	001	0010	0333	
\$BASIC	001	0080	0391	
\$BIGCD	001	0080	0467	
\$BLDPL	001	0579	0600	0602
\$BLNOE	001	0569	0590	
\$BLOAD	001	0522	0581	0583 0586 0599 0600
\$BLRTN	001	0550	0589	0590
\$BRSAV	001	03C5	0278	0279
\$BSADR	001	0587	0605	0607
\$BUFPT	001	03E3	0486	0487
\$CABLD	001	04B4	0559	0560
\$CAERK	001	0469	0536	0539 0184 0195 0232
\$CAERR	001	03CD	0284	0286 0194* 0231*
\$CAIPL	001	049D	0555	0557
\$CALLI	001	0008	0476	
\$CARDI	001	0001	0247	
\$CARPL	001	04A1	0557	0559
\$CIENT	001	0483	0546	0547
\$CIEXT	001	0480	0545	0546
\$CIMSK	001	0476	0542	0545
\$CISUS	001	0496	0550	0555
\$CLBFR	001	0010	0434	
\$CMDKY	001	0008	0346	
\$CMODE	001	0002	0396	
\$CONFIG	001	03DD	0459	0469
\$CRPOS	001	03E2	0485	0486
\$CRTAD	001	044D	0524	0525
\$CRTAV	001	0002	0340	
\$CRTDN	001	0002	0364	
\$CRTIN	001	03D3	0361	0368
\$CRTNO	001	0004	0343	
\$CRTPU	001	0004	0365	
\$CRTSP	001	0008	0366	
\$CRTUP	001	0001	0363	
\$CRUSH	001	0080	0472	
\$CSDPL	001	050E	0571	0572
\$C0001	001	0464	0528	0534
\$DATE	001	043A	0509	0510
\$DBGUF	001	03E0	0471	0480
\$DBLOK	001	0001	0421	
\$DFDET	001	03E8	0492	0493
\$DISKN	001	0025	0224	0245
\$DKERR	001	0008	0402	
\$DKSIZ	001	03D7	0446	0454 0495
\$DK100	001	0001	0448	
\$DK200	001	0002	0449	
\$DK400	001	0004	0450	
\$DK600	001	0008	0451	
\$DK800	001	0010	0452	
\$DPLSV	001	0449	0520	0522
\$DTNMB	001	0040	0267	
\$DTRDR	001	0040	0355	
\$ENDNU	001	0600	0614	0625

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 179

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$ERDPL	001	046F	0539	0541
\$ERFIL	001	0040	0294	
\$ERHRD	001	0004	0426	
\$ERKEY	001	0080	0298	
\$ERLOG	001	0345	0229	
\$ERMAD	001	0472	0541	0542
\$ERPND	001	0004	0399	
\$ERRCT	001	03CF	0300	0183*
\$ERRPG	001	03CE	0288	0182* 0193* 0230*
\$ERSFL	001	0035	0293	
\$ERSTK	001	0030	0291	0182
\$ER050	001	0363	0230	
\$ER1N2	001	0050	0296	
\$EXADR	001	0517	0574	0576
\$EXCMD	001	0001	0328	
\$EXFTR	001	043B	0510	0515 2805 3405 3936 4550 0291
\$FCIND	001	0010	0406	
\$FDIND	001	0040	0413	
\$FEARR	001	0004	0222	
\$FEMAP	001	0588	0607	0608
\$FILIB	001	03DA	0457	0458
\$FITIN	001	0010	0382	
\$FUIND	001	0020	0411	
\$GUFIO	001	0583	0604	0605
\$GUFIR	001	0008	0256	
\$HISTE	001	042E	0507	0508
\$HIST1	001	0435	0508	0509
\$HRDER	001	0020	0352	
\$INDR1	001	03D4	0368	0394
\$INDR2	001	03D5	0394	0419
\$INDR3	001	03D6	0419	0446
\$INLNO	001	03CF	0286	0288 0300 0307 5036 5042*
\$INRPT	001	0020	0264	
\$IOIND	001	03D2	0335	0361
\$IOPGS	001	0010	0475	
\$IOYES	001	0002	0250	
\$IPLDV	001	05FF	0611	0614
\$IRKEY	001	0020	0474	
\$KEYBD	001	03E1	0480	0485
\$KEYCD	001	03C3	0244	0278
\$KEYDT	001	0040	0388	
\$KE090	001	00DE	0225	
\$KE130	001	01D5	0226	
\$KYBSY	001	0010	0261	
\$LDRTN	001	0571	0599	
\$LEVEL	001	03DF	0469	0471
\$LIST	001	0002	0423	
\$LMRGN	001	03C1	0239	0241
\$LNPTR	001	0080	0358	
\$LOADB	001	054A	0583	
\$LOADR	001	051A	0576	0579
\$LPRIO	001	03E9	0493	
\$LPROS	001	03E5	0488	0490
\$LPRP3	001	03E4	0487	0488
\$MOUNT	001	0020	0437	
\$MPDWN	001	0001	0337	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 180

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$NEXTB	001	03E6	0490	0491
\$NEXTL	001	03E7	0491	0492
\$NOENB	001	0008	0429	
\$NOLST	001	0004	0253	
\$NUCBS	001	03C0	0236	0237
\$NWRKF	001	0080	0442	
\$NWRKR	001	0040	0439	
\$PASWD	001	042D	0506	0507
\$PAUSD	001	04BA	0560	0562
\$PAUSE	001	0002	0330	
\$PGMDT	001	0020	0385	
\$PGMST	001	0010	0349	
\$PKERT	001	0419	0504	0506
\$PLST1	001	0454	0525	0526
\$PLST2	001	045B	0526	0527
\$PLST3	001	0462	0527	0528
\$PRDEV	001	044B	0522	0524
\$PRESN	001	0002	0373	
\$PROCI	001	0001	0370	
\$PRPOS	001	03C2	0241	0244
\$PSDBR	001	04FA	0565	
\$PSDXR	001	04F2	0564	0565
\$PSTEP	001	0004	0331	
\$PSTMT	001	0008	0332	
\$PTCH1	001	03F5	0495	0499
\$READY	001	0080	0415	
\$REORD	001	0040	0473	
\$RLOAD	001	051E	0579	0581 0498
\$RMGRN	001	03C0	0237	0239
\$RSTR	001	04D6	0562	0564 0566 0571
\$RUNIT	001	0001	0309	
\$SFAID	001	050D	0567	
\$SPRNT	001	0465	0534	0536
\$SRTRN	001	04FE	0566	0567
\$STEPT	001	0002	0310	
\$SWPCR	001	0511	0572	0574
\$TABLN	001	03CB	0281	0284
\$TFLOW	001	0008	0316	
\$TRACE	001	0004	0311	
\$TRALL	001	0010	0317	
\$TROVR	001	054E	0586	0589
\$TRUNK	001	0080	0269	
\$TRVAR	001	0020	0318	
\$UNMSK	001	048D	0547	0550
\$USRDR	001	03DC	0458	0459
\$VMDEF	001	0080	0322	
\$VOLF1	001	03FE	0501	0502
\$VOLF2	001	040E	0503	
\$VOLID	001	03F6	0499	0500 0504
\$VOLR1	001	03F6	0500	0501
\$VOLR2	001	0406	0502	0503
\$WAITF	001	057F	0602	0604 0246
\$WFDEF	001	0040	0516	
\$WFLOK	001	0008	0379	
\$WFNME	001	0443	0515	0520
\$WSIND	001	0004	0376	



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 181

SYMBOL	LEN	VALUE	DEFN	REFERENCES
\$XIND1	001	03D0	0307	0326 5366 6281 0181*
\$XIND2	001	03D1	0326	0335
\$XIND3	001	03D8	0454	0457
\$XPREC	001	0040	0319	5366 6281
\$XRSAB	001	03C7	0279	0281
\$ZTRAD	001	05A2	0608	
\$12K	001	0004	0463	
\$16CKY	001	0008	0465	
\$16K	001	0002	0462	
\$22IMP	001	0001	0460	
#\$BOV	001	0800	2565	2566
#\$LOA	UNDEFINED		SYMBOL	0522
#\$LOA	UNDEFINED		SYMBOL	0521
#\$LOA	UNDEFINED		SYMBOL	0520
#BOVLY	001	0000	0001	
@@E001	001	0000	2052	2054
@@E003	001	0001	2054	2056
@@E004	001	0002	2056	2058
@@E005	001	0003	2058	2060
@@E006	001	0004	2060	2062
@@E007	001	0005	2062	2064
@@E008	001	0006	2064	2066
@@E009	001	0007	2066	2068
@@E010	001	0008	2068	2070
@@E011	001	0009	2070	2072
@@E012	001	000A	2072	2074
@@E013	001	000B	2074	2076
@@E014	001	000C	2076	2078
@@E015	001	000D	2078	2080
@@E016	001	000E	2080	2082
@@E017	001	000F	2082	2084
@@E018	001	0010	2084	2086
@@E019	001	0011	2086	2088
@@E020	001	0012	2088	2090
@@E021	001	0013	2090	2092
@@E023	001	0014	2092	2094
@@E024	001	0015	2094	2096
@@E025	001	0016	2096	2098
@@E026	001	0017	2098	2100
@@E027	001	0018	2100	2102
@@E028	001	0019	2102	2104
@@E029	001	001A	2104	2106
@@E030	001	001B	2106	2108
@@E031	001	001C	2108	2110
@@E032	001	001D	2110	2112
@@E035	001	001E	2112	2114
@@E036	001	001F	2114	2116
@@E037	001	0020	2116	2118
@@E038	001	0021	2118	2120
@@E039	001	0022	2120	2122
@@E040	001	0023	2122	2124
@@E041	001	0024	2124	2126
@@E042	001	0025	2126	2128
@@E043	001	0026	2128	2130
@@E044	001	0027	2130	2132
@@E045	001	0028	2132	2134

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 182

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E046	001	0029	2134	2136
@@E060	001	002A	2136	2138
@@E080	001	002B	2138	
@@E100	001	0000	1524	1526
@@E101	001	0001	1526	1528
@@E102	001	0002	1528	1530
@@E103	001	0003	1530	1532
@@E110	001	0004	1532	1534
@@E112	001	0005	1534	1536
@@E113	001	0006	1536	1538
@@E114	001	0007	1538	1540
@@E115	001	0008	1540	1542
@@E116	001	0009	1542	1544
@@E117	001	000A	1544	1546
@@E120	001	000B	1546	1548
@@E122	001	000C	1548	1550
@@E123	001	000D	1550	1552
@@E124	001	000E	1552	1554
@@E129	001	000F	1554	1556
@@E130	001	0010	1556	1558
@@E131	001	0011	1558	1560
@@E133	001	0012	1560	1562
@@E134	001	0013	1562	1564
@@E135	001	0014	1564	1566
@@E136	001	0015	1566	1568
@@E137	001	0016	1568	1570
@@E138	001	0017	1570	1572
@@E139	001	0018	1572	1574
@@E142	001	0019	1574	1576
@@E143	001	001A	1576	1578
@@E150	001	001B	1578	1580
@@E151	001	001C	1580	1582
@@E160	001	001D	1582	1584
@@E162	001	001E	1584	1586
@@E163	001	001F	1586	1588
@@E164	001	0020	1588	1590
@@E200	001	0021	1590	1592
@@E205	001	0022	1592	1594
@@E210	001	0023	1594	1596
@@E211	001	0024	1596	1598
@@E212	001	0025	1598	1600
@@E213	001	0026	1600	1602
@@E215	001	0027	1602	1604
@@E216	001	0028	1604	1606
@@E217	001	0029	1606	1608
@@E220	001	002A	1608	1610
@@E221	001	002B	1610	1612
@@E222	001	002C	1612	1614
@@E223	001	002D	1614	1616
@@E225	001	002E	1616	1618
@@E226	001	002F	1618	1620
@@E227	001	0030	1620	1622
@@E228	001	0031	1622	1624
@@E229	001	0032	1624	1626
@@E230	001	0033	1626	1628
@@E232	001	0034	1628	1630

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 183

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E234	001	0035	1630	1632
@@E237	001	0036	1632	1634
@@E240	001	0037	1634	1636
@@E241	001	0038	1636	1638 2527
@@E242	001	0039	1638	1640
@@E248	001	003A	1640	1642
@@E249	001	003B	1642	1644
@@E250	001	003C	1644	1646
@@E251	001	003D	1646	1648
@@E252	001	003E	1648	1650
@@E253	001	003F	1650	1652
@@E254	001	0040	1652	1654
@@E255	001	0041	1654	1656
@@E256	001	0042	1656	1658
@@E300	001	0043	1658	1660
@@E301	001	0044	1660	1662
@@E302	001	0045	1662	1664
@@E303	001	0046	1664	1666
@@E304	001	0047	1666	1668
@@E305	001	0048	1668	1670
@@E308	001	0049	1670	1672
@@E310	001	004A	1672	1674
@@E315	001	004B	1674	1676
@@E316	001	004C	1676	1678
@@E320	001	004D	1678	1680
@@E325	001	004E	1680	1682
@@E330	001	004F	1682	1684
@@E335	001	0050	1684	1686
@@E338	001	0051	1686	1688
@@E340	001	0052	1688	1690
@@E350	001	0053	1690	1692
@@E351	001	0054	1692	1694
@@E352	001	0055	1694	1696
@@E360	001	0056	1696	1698
@@E361	001	0057	1698	1700
@@E362	001	0058	1700	1702
@@E371	001	0059	1702	1704
@@E380	001	005A	1704	1706
@@E390	001	005B	1706	1708
@@E400	001	005C	1708	1710
@@E410	001	005D	1710	1712
@@E415	001	005E	1712	1714
@@E417	001	005F	1714	1716
@@E420	001	0060	1716	1718
@@E430	001	0061	1718	1720
@@E432	001	0062	1720	1722
@@E433	001	0063	1722	1724
@@E450	001	0064	1724	1726
@@E451	001	0065	1726	1728
@@E460	001	0066	1728	1730
@@E461	001	0067	1730	1732
@@E464	001	0068	1732	1734
@@E465	001	0069	1734	1736
@@E466	001	006A	1736	1738
@@E467	001	006B	1738	1740
@@E469	001	006C	1740	1742

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 184

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E470	001	006D	1742	1744
@@E471	001	006E	1744	1746
@@E473	001	006F	1746	1748
@@E474	001	0070	1748	1750
@@E475	001	0071	1750	1752
@@E476	001	0072	1752	1754
@@E477	001	0073	1754	1756
@@E478	001	0074	1756	1758
@@E479	001	0075	1758	1760
@@E480	001	0076	1760	1762
@@E481	001	0077	1762	1764
@@E482	001	0078	1764	1766
@@E483	001	0079	1766	1768
@@E484	001	007A	1768	1770
@@E485	001	007B	1770	1772
@@E486	001	007C	1772	1774
@@E487	001	007D	1774	1776
@@E488	001	007E	1776	1778
@@E489	001	007F	1778	1780
@@E490	001	0080	1780	1782
@@E491	001	0081	1782	1784
@@E492	001	0082	1784	1786
@@E493	001	0083	1786	1788
@@E494	001	0084	1788	1790
@@E495	001	0085	1790	1792
@@E496	001	0086	1792	1794
@@E497	001	0087	1794	1796
@@E498	001	0088	1796	1798
@@E500	001	0089	1798	1800
@@E501	001	008A	1800	1802
@@E530	001	008B	1802	1804
@@E531	001	008C	1804	1806
@@E535	001	008D	1806	1808
@@E540	001	008E	1808	1810
@@E541	001	008F	1810	1812
@@E542	001	0090	1812	1814
@@E543	001	0091	1814	1816
@@E544	001	0092	1816	1818
@@E545	001	0093	1818	1820
@@E546	001	0094	1820	1822
@@E547	001	0095	1822	1824
@@E548	001	FFFF	2028	
@@E549	001	0096	1824	1826
@@E550	001	0097	1826	1828
@@E551	001	0098	1828	1830
@@E552	001	0099	1830	1832
@@E553	001	009A	1832	1834
@@E554	001	009B	1834	1836
@@E555	001	009C	1836	1838
@@E556	001	009D	1838	1840
@@E558	001	009E	1840	1842
@@E570	001	009F	1842	1844
@@E571	001	00A0	1844	1846
@@E572	001	00A1	1846	1848
@@E573	001	00A2	1848	1850
@@E574	001	00A3	1850	1852

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 185

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E575	001	FFFF	2030	
@@E578	001	00A4	1852	1854
@@E579	001	FFFF	2032	
@@E580	001	FFFF	2034	
@@E585	001	00A5	1854	1856
@@E595	001	FFFF	2036	
@@E597	001	FFFF	2038	
@@E598	001	FFFF	2040	
@@E600	001	00A6	1856	1858 3104
@@E601	001	00A7	1858	1860
@@E602	001	00A8	1860	1862
@@E603	001	00A9	1862	1864
@@E604	001	00AA	1864	1866 6276
@@E606	001	00AB	1866	1868 7838
@@E607	001	00AC	1868	1870 7833
@@E608	001	00AD	1870	1872 5390
@@E609	001	00AE	1872	1874 0194
@@E610	001	00AF	1874	1876 1884
@@E611	001	00B0	1876	1878
@@E612	001	00B1	1878	1880 0231
@@E613	001	00B2	1880	1882
@@E614	001	00B3	1882	
@@E700	001	00B0	1884	1886
@@E701	001	00B1	1886	1888
@@E710	001	00B2	1888	1890
@@E712	001	00B3	1890	1892
@@E713	001	00B4	1892	1894
@@E714	001	00B5	1894	1896
@@E715	001	00B6	1896	1898
@@E716	001	00B7	1898	1900
@@E717	001	00B8	1900	1902
@@E718	001	00B9	1902	1904
@@E720	001	00BA	1904	1906
@@E721	001	00BB	1906	1908
@@E723	001	00BC	1908	1910
@@E724	001	00BD	1910	1912
@@E725	001	00BE	1912	1914
@@E726	001	00BF	1914	1916
@@E727	001	00C0	1916	1918
@@E728	001	00C1	1918	1920
@@E729	001	00C2	1920	1922
@@E730	001	00C3	1922	1924
@@E732	001	00C4	1924	1926
@@E752	001	00C5	1926	1928
@@E753	001	00C6	1928	1930
@@E754	001	00C7	1930	1932
@@E755	001	00C8	1932	1934
@@E756	001	00C9	1934	1936
@@E757	001	00CA	1936	1938
@@E758	001	00CB	1938	1940
@@E759	001	00CC	1940	1942
@@E760	001	00CD	1942	1944
@@E761	001	00CE	1944	1946
@@E762	001	00CF	1946	1948
@@E763	001	00D0	1948	1950
@@E764	001	00D1	1950	1952

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 186

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@@E765	001	00D2	1952	1954
@@E766	001	00D3	1954	1956
@@E767	001	00D4	1956	1958
@@E768	001	00D5	1958	1960
@@E769	001	00D6	1960	1962
@@E770	001	00D7	1962	1964
@@E771	001	00D8	1964	1966
@@E772	001	00D9	1966	1968
@@E773	001	00DA	1968	1970
@@E774	001	00DB	1970	1972
@@E775	001	00DC	1972	1974
@@E776	001	00DD	1974	1976
@@E777	001	00DE	1976	1978
@@E778	001	00DF	1978	1980
@@E779	001	00E0	1980	1982
@@E780	001	00E1	1982	1984
@@E781	001	00E2	1984	1986
@@E782	001	00E3	1986	1988
@@E783	001	00E4	1988	1990
@@E784	001	00E5	1990	1992
@@E785	001	00E6	1992	1994
@@E786	001	00E7	1994	1996
@@E790	001	00E8	1996	1998
@@E791	001	00E9	1998	2000
@@E792	001	00EA	2000	2002
@@E793	001	00EB	2002	2004
@@E794	001	00EC	2004	2006
@@E795	001	00ED	2006	2008
@@E796	001	00EE	2008	2010
@@E797	001	00EF	2010	2012
@@E798	001	00F0	2012	2014
@@E800	001	FFFF	2042	
@@E801	001	FFFF	2044	
@@E802	001	FFFF	2046	
@@E803	001	FFFF	2048	
@@E804	001	FFFF	2050	
@@E900	001	00F1	2014	2016 2523
@@E901	001	00F2	2016	2018 2525
@@E902	001	00F3	2018	2020 2524
@@E903	001	00F4	2020	2022 2526
@@E905	001	00F5	2022	2024
@@E906	001	00F6	2024	2026
@@E910	001	00F7	2026	2522
@ARR	001	0008	0016	3987 4112 4128 4290 5704 6094
@ASIGN	001	007C	0071	
@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	
@BM	001	0082	0054	
@BNE	001	0001	0046	
@BNH	001	0004	0044	
@BNL	001	0002	0045	



## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 20/07/20 PAGE 187

@BNM	001	0002	0057												
@BNOL	001	0020	0050												
@BNOZ	001	0008	0049												
@BNP	001	0004	0056												
@BNZ	001	0001	0058												
@BOL	001	00A0	0048												
@BOZ	001	0088	0047												
@BP	001	0084	0053												
@BR	001	0001	0013	2672	2677	2677	2686	2694	2698	2714	2719	2723	2751	2751	2763
				2763	2769	2770	2774	2783	2787	2798	2798	2804	2804	2805	2806
				2806	2812	2812	2816	2822	2822*	2823	2823*	2824	2876	2886	2890
				2894	2899*	2906	2910	2917	2917	2921	2921	2925	2929	2936	3093
				3128	3134	3135	3141	3145	3160	3169	3175	3296	3301	3301	3325
				3326	3345	3346	3351	3352	3389	3390	3398	3398	3404	3404	3405
				3407	3407	3413	3413	3417	3423	3423*	3424	3463	3491	3491*	3492
				3494	3520	3540	3544	3712	3721	3842	3848	3848	3851	3867	3867
				3875	3876	3896	3901	3902	3902	3908	3910	3911*	3918	3919	3920
				3926	3927	3929	3929	3935	3936	3937	3937	3944	3944	3945	3951
				3951*	3952	3952*	3953	3961	3962	3963	3967	3972	3973	3973	3974
				3974	3975	3976	3987	3991	4027	4040	4042	4047	4048	4049	4049
				4050	4051	4054	4056	4057	4058	4059	4065	4066	4067	4068	4069
				4070	4075	4077	4083	4089	4090*	4097	4097	4099	4112	4116	4128
				4129	4130	4131	4132	4134	4135	4136	4137	4137	4138	4139	4140
				4140	4141	4142	4143	4144	4145	4146	4188	4195	4196	4202	4210
				4211	4212	4223	4228	4230	4231*	4241	4242	4245	4246	4247	4248
				4249	4251	4252	4253	4254	4255	4264	4265	4266	4271	4290	4294
				4432	4440	4441	4448	4449	4449	4469	4483	4484	4488	4490	4495
				4504	4506	4508*	4515	4515	4516	4517	4524	4525	4533	4543	4543
				4549	4549	4550	4551	4551	4557	4557	4561	4562	4562	4568	4568*
				4569	4569*	4570	4601	4613	4619	4621	4626	4631	4640	4655*	4656
				4661	4671	4678	4679	4688	4824	4828	4832	4842	4847	4848	5009
				5021	5054	5070	5077	5078	5080	5081	5084	5091	5094	5098	5108
				5114	5118	5125	5130	5319	5333	5355	5358	5359	5368	5369	5373
				5374	5388	5396	5399	5549	5562	5566	5567	5588	5593	5594	5598
				5600	5604	5605	5606	5610	5614	5615	5619	5624	5628	5638	5640
				5644	5645	5646	5648	5649	5653	5657	5659	5663	5664	5668	5672
				5672	5673	5684	5699	5704	5916	5929	5931	5936	5940	5942	5957
				5974	5979	5983	5987	5995	6014	6025	6026	6040	6041	6045	6049
				6053	6055	6059	6060	6064	6069	6073	6073	6077	6089	6094	6239
				6252	6283	6284	6288	6290	6291	6297	6330	6331	6346	6354	6508
				6527	6536	6540	6546	6551	6555	6562	6569	6575	6583	6584	6592
				6592	6593	6602	6611	6620	6760	6769	6886	6898	6899	6900	6901
				6913	6921	6925	6941	6945	6952	6968	6973	6977	6984	6993	6997
				6997	7001	7002	7006	7143	7152	7263	7279	7300	7301	7318	7322
				7331	7476	7481	7481	7486	7489	7495	7621	7641	7650	7801	7845
				7846	7853	7858	7859	7994	8007	8011	8020	8156	8165	8298	8316
				8335	8340	8343	8347	8367	8374	8383	8551	8559	8568	8577	8703
				8721	8740	8746	8749	8753	8773	8780	8789	8950	8964	8968	8977
				9104	9117	9124	9129	9134	9142	9151	9151	9159	9172	9179	9187
				9188	9323	9330	9340	9463	9575	9588	9596	9600	9615	9633	9642
				9646	9781	9788	9798	9919	0159	0164	0164	0176	0188	0200	0240
				0266	0285	0285	0291	0292	0292	0298	0298	0302	0308	0308*	0309
				0364	0391	0392	0398	0405	0410	0411	0412	0416	0416	0417	0417
				0418	0426	0427	0433	0440	0445	0446	0447	0451	0451	0452	0452
				0453	0461	0462	0468	0475	0480	0481	0482	0486	0486	0487	0487
				0488	0496	0497	0667	0780							

CROSS REFERENCE																		
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	20/07/20	PAGE 188
@BT	001	0010	0051															
@BZ	001	0081	0055															
@B1	001	0001	0063	2752	2764	3883	3919	3919	3919*	3935*	3936	4066	4066	4066*	4211			
				4211	4211*	4219	4440	4465	4515	4550	4562	4613	4631	4644	4649			
				4678	5340													
@CADDR	001	0002	0141	1373	1374	1375	2476	2503	2798	2804	2806	2812	2844	2846	2847			
				2853	2854	3398	3404	3407	3413	3430	3433	3434	3435	3441	3444			
				3445	3851	3902	3926	3929	3937	3944	3973	3974	3975	4011	4012			
				4016	4019	4020	4083	4089	4097	4180	4182	4516	4543	4549	4551			
				4557	4578	4580	4581	4582	4583	4705	5171	5176	5181	5186	5191			
				5196	5383	5388	5433	7354	7673	7853	7877	7878	8043	8999	9363			
				9673	9821	0188	0246	0285	0292	0298	0315	0317	0319	0334	0350			
				0353	0354	0355	0416	0451	0486	0499	0505	0507	0508	0509	0532			
				0533	0534	0536	0537	0538	0540	0541	0542							
@CARDL	001	0060	0087															
@CHARA	001	00C1	0072															
@CHARF	001	00C6	0073															
@CHARR	001	00D9	0074															
@CHARZ	001	00E9	0075															
@CLOFF	001	0010	0094															
@CLON	001	0011	0093															
@COMMA	001	006B	0066															
@CPLUS	001	004E	0079															
@DADDR	001	0002	0139															
@DBFR1	001	0004	0128															
@DBFR2	001	0005	0129															
@DCALK	001	0001	0081															
@DCBCY	001	0009	0114	1202														
@DCBT1	001	0050	0116	1205														
@DCNT	001	0003	0127															
@DCST1	001	0040	0115	1203														
@DCTRL	001	0000	0124															
@DCYL	001	0001	0125															
@DD2	001	0003	0030															
@DGET	001	0001	0133	0330														
@DOLAR	001	005B	0068															
@DOP2	001	0004	0028															
@DPLNG	001	0006	0131															
@DPOS	001	0000	0132															
@DPUT	001	0002	0134															
@DSAD	001	0002	0126															
@DSBCY	001	0004	0105	1140														
@DSCS1	001	0000	0106	1141														
@DSIVF	001	0003	0137															
@DSPIN	001	0002	0130															
@DTRSZ	001	0018	0085															
@DVBCY	001	0007	0107	1199														
@DVRFY	001	0031	0135															
@DWAIT	001	00FF	0136															
@DWBCY	001	0005	0102	1196														
@DWSIZ	001	00C0	0104															
@DWTB1	001	0003	0103	1197														
@DZERO	001	00F0	0064															
@D1	001	0002	0026	2719*	2723*	2751*	2763*	2770*	2894*	2921*								

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 189

SYMBOL	LEN	VALUE	DEFN	REFERENCES
@EOF	001	001C	0077	
@EOFTC	001	0075	0160	
@EOS	001	001E	0076	1212 5343
@FDDBC	001	0000	0193	
@FDE1	001	000C	0198	
@FDFNA	001	000B	0196	
@FDHLN	001	0002	0206	
@FDLNC	001	0002	0191	
@FDNSC	001	0003	0208	
@FDSD	001	0000	0204	
@FLACE	001	0009	0195	
@FLDBC	001	0001	0194	
@FLENT	001	0004	0199	
@FLFNA	001	0002	0197	
@FLHLN	001	0002	0207	
@FLLNC	001	0002	0192	
@FLNSC	001	0001	0209	
@FLSD	001	0001	0205	
@HDRLN	001	0007	0092	
@IAR	001	0010	0017	
@INDEX	001	0001	0154	0155
@INST3	001	0003	0032	2719 2770 5680
@INST4	001	0004	0033	0401 0436 0471
@INST5	001	0005	0034	
@INST6	001	0006	0035	
@I1IAR	001	00C0	0020	
@LINSZ	001	00F4	0084	
@MAPEN	001	0005	0089	
@MINCR	001	2000	0083	
@MINUS	001	0060	0080	
@NOP	001	0080	0040	5679 5813
@NUMBR	001	007B	0070	
@OPD2	001	0004	0029	
@OP1	001	0003	0027	3093* 3135 3145 3160 3851* 3987* 4089* 4112* 4128* 4290* 5704* 6094* 8007* 8964*
@OP2	001	0005	0031	
@PCTRL	001	0000	0147	
@PDATA	001	0003	0149	
@PGCSZ	001	0020	0082	0083
@PPLNG	001	0004	0146	
@PRCNT	001	0001	0148	
@PRETR	001	00C0	0152	
@PRINT	001	0040	0150	0152
@PSR	001	0004	0015	
@PWAIT	001	00FF	0156	
@P1IAR	001	0020	0018	
@P2IAR	001	0040	0019	
@Q	001	0001	0024	3962* 3991* 4058* 4116* 4134* 4247* 4254* 4294* 4613* 4631* 5593* 5678 5812 6546* 8316* 8335* 8721* 8740*
@REGL	001	0002	0012	
@RETRN	001	0080	0151	0152
@RLDWN	001	004F	0157	
@RTRNC	001	0080	0159	
@SBLNL	001	0002	0182	
@SCTS	001	0100	0099	
@SDFLN	001	0007	0090	

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 20/07/20 PAGE 190

@SDF0	001	0000	0164												
@SDF1	001	0001	0165												
@SDF2	001	0002	0166												
@SDF3	001	0003	0167												
@SDLN	001	0005	0168												
@SECCY	001	0030	0086												
@SIST	001	0001	0179												
@SLASH	001	0061	0067												
@SLAST	001	0002	0181												
@SMIDL	001	0003	0180												
@SNULL	001	0080	0171												
@SONLY	001	0000	0178												
@STEXT	001	0007	0170												
@STYPE	001	0006	0169												
@SYLVL	001	0005	2558												
@TBCNT	001	0000	0158												
@TBLEF	001	0010	0153	0155											
@TBLIX	001	0011	0155												
@UCB	001	0087	0039	5813											
@UPARW	001	005A	0078	2541											
@VADDR	001	0002	0140	0934	1369	1381	1382	1383	1383	1397	1400	1402	1426	1427	1428
				1466	1469	1472	1475	1478	1481	1484	1493	1496	1499	1502	1505
				2477	2503	2693	2694	2705	2713	2714	2855	2880	2885	2886	2965
				3182	3877	4000	4040	4049	4129	4130	4137	4140	4201	4202	4203
				4264	4270	4271	4301	4303	4315	4584	4687	4688	4701	4828	4840
				4841	4842	4847	4848	4873	4879	5032	5042	5048	5084	5108	5124
				5125	5129	5130	5203	5333	5358	5396	5397	5398	5399	5413	5434
				5436	5653	5672	5796	5806	5935	5936	5947	5956	5957	5965	6045
				6073	6120	6122	6259	6290	6291	6297	6330	6331	6354	6392	6536
				6540	6583	6584	6585	6592	6619	6620	6632	6920	6921	6968	6997
				7022	7029	7032	7819	7845	7858	7859	7885	8163	8164	8165	8186
				8382	8383	8404	8788	8789	8811	9124	9141	9142	9146	9151	9187
				9188	9215	9595	9596	9609	9614	9615	9623	0254	0258	0262	0266
				0268	0272	0276	0277	0316	0322	0392	0398	0427	0433	0462	0468
				0482	0511	0556	0557	0559	0561						
@VENTA	001	0056	0112	1200	1455										
@VMDDV	001	00FE	0113												
@VMFD1	001	0000	0108												
@VMFD2	001	0001	0109												
@VMRS3	001	0002	0111												
@VMTRL	001	0001	0110												
@VOLID	001	0006	0091												
@VQ	001	0001	0025												
@WSFIT	001	0500	0100												
@WSTBL	001	0503	0101												
@XR	001	0002	0014	2686*	2687	2698*	2699	2735*	2736	2736*	2745	2752	2752*	2757	2764
				2764*	2765	2769	2774*	2775	2781*	2782	2787*	2788	2816*	2898*	2899
				2901	2906	2910*	2929*	2930	2936*	2937	3092*	3093	3097	3119*	3123
				3127	3132*	3133	3135*	3140	3145*	3150	3160*	3165	3174	3316	3321
				3323	3328	3330	3335*	3336	3338	3340	3356*	3357	3417*	3473*	3480*
				3492*	3493	3498	3498*	3499	3520*	3521	3534*	3540*	3541	3712*	3713
				3719*	3720	3875*	3878*	3908*	3909	3912*	3918*	3927*	3928	3945*	3961*
				3964	3988	3992*	4041	4047*	4050*	4053	4057*	4065*	4068*	4099*	4113
				4131*	4135*	4138*	4141*	4143*	4145*	4195*	4210*	4214	4228*	4229	4232*
				4241*	4244	4246*	4251*	4253*	4265*	4291	4457	4482	4484*	4485	4490*
				4491	4504*	4505	4509*	4561*	4609*	4613	4618	4620	4631	4637*	4638

CROSS REFERENCE															
SYMBOL	LEN	VALUE	DEFN	REFERENCES											
				VER 15, MOD 00 20/07/20 PAGE 191											
				4638*	4639	4644	4661*	4671*	4672	4679*	4680	4824*	4825	4832*	4833
				5021*	5022	5054*	5055	5063	5071	5077	5078*	5079	5079*	5080	5082
				5083	5083*	5084	5091	5093	5098*	5113	5118*	5119	5343	5355*	5359*
				5360	5374*	5375	5380*	5381	5381*	5382	5396	5397	5398	5399	5593
				5594*	5596	5596*	5597	5599	5604	5605	5633*	5647	5657*	5699*	5705
				5707*	5929*	5940*	6023*	6024	6053*	6089*	6095	6097*	6252*	6253	6264*
				6269*	6270	6296*	6297	6306	6311	6316	6346*	6347	6527*	6528	6544*
				6546	6551*	6552	6555*	6556	6575*	6576	6593*	6594	6602*	6603	6611*
				6612	6760*	6761	6767*	6768	6901*	6902	6913*	6914	6925*	6945*	6946
				6952*	6953	6977*	6978	6984*	6985	7143*	7144	7150*	7151	7279*	7280
				7322*	7323	7329*	7330	7495*	7496	7621*	7622	7641*	7642	7648*	7649
				7818*	7819	7828	7845	7846*	7847	7994*	7995	8007	8011*	8012	8018*
				8019	8156*	8157	8316	8321	8323	8335	8340*	8341	8341*	8342	8347
				8352*	8367*	8368	8374*	8375	8546	8551*	8552	8559*	8560	8566*	8567
				8576	8721	8726	8728	8740	8746*	8747	8747*	8748	8753	8758*	8773*
				8774	8780*	8781	8950*	8951	8964	8968*	8969	8975*	8976	9117*	9118
				9128*	9134*	9135	9155*	9156	9172*	9173	9179*	9180	9323*	9324	9330*
				9331	9339	9463*	9464	9588*	9589	9600*	9601	9604*	9633*	9634	9640*
				9641	9646*	9647	9781*	9782	9788*	9789	9797	9919*	9920	0176*	0200*
				0201	0224*	0242*	0302*	0391*	0392	0396*	0398	0410*	0411*	0412	0426*
				0427	0431*	0433	0445*	0446*	0447	0461*	0462	0466*	0468	0480*	0481*
@ZERO	001	0000	0062	0482	0496*	0497	0667*	0668	0780*	0781					
				2723	2890	2894	2901	3309	3378	3849	3885	3895	4000	4020	4221
				4222	4301	4448	4467	4468	4480	4482	4525	4656	5342	5639	6035
				6935	7476	9129	0181								
B\$ADMK	001	0001	0838												
B\$ADSW	001	159D	0837												
B\$ARMK	001	0001	0823	5583	6018	7295									
B\$ARSW	001	0A45	0822	5583	6018	7295									
B\$BABF	001	1D00	0628												
B\$BCKT	001	1590	0750	4040	4129	5108	5333	5358	6045	6968	7819				
B\$BDPL	001	19E8	0702	0224											
B\$BDSA	001	19EA	0703	0225											
B\$BINO	001	1A6A	0766	3123	3150	3169	4689	4							

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 192

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$CDIM	001	0673	0641	
B\$CDUM	001	0000	0677	
B\$CDWA	UNDEFINED	SYMBOL	6371	
B\$CEND	001	0600	0675	0676
B\$CEOF	001	0600	0676	
B\$CFOR	001	0600	0648	
B\$CGET	001	06A3	0656	
B\$CGSB	001	0690	0654	
B\$CGTO	001	06B3	0652	
B\$CIFA	001	0600	0650	
B\$CIFC	001	0600	0651	
B\$CIMG	001	0600	0665	
B\$CINP	001	0600	0660	
B\$CLTA	001	0000	0642	
B\$CLTC	001	0669	0646	
B\$CLTM	001	0600	0644	
B\$CMAT	001	0600	0666	
B\$CMF1	UNDEFINED	SYMBOL	8033	
B\$CMGT	001	0665	0667	
B\$CMIN	001	06D3	0668	
B\$CMMA	UNDEFINED	SYMBOL	5745	
B\$CMPR	001	069B	0671	
B\$CMPT	001	069B	0670	
B\$CMPU	001	0600	0672	
B\$CMRD	001	06D0	0669	
B\$CNXT	001	0600	0649	
B\$CPCT	001	0CA8	0731	5639 5663* 6035 6059* 6935 7001*
B\$CPRT	001	0600	0663	
B\$CPRU	001	0600	0664	
B\$CPSE	001	06E7	0673	
B\$CPUT	001	0600	0657	
B\$CPWA	001	0CA6	0802	
B\$CRAD	001	150D	0772	4229* 4505*
B\$CRBS	001	1509	0774	4230* 4506*
B\$CREA	001	06CF	0661	
B\$CREM	001	0000	0638	
B\$CRMK	001	0001	0850	2740 3117
B\$CRSR	001	06E3	0662	
B\$CRST	001	06A6	0658	
B\$CRSW	001	0E42	0849	3117 3895 4222 4468
B\$CRTN	001	06CF	0655	
B\$CSBF	001	0600	0625	0639 0640 0641 0644 0645 0646 0647 0648 0649 0650 0651 0652 0653 0654 0655 0656 0657 0658 0659 0660 0661 0662 0663 0664 0665 0666 0667 0668 0669 0670 0671 0672 0673 0674 0675 0678 0679 0680 0681 0682 3435 4012 4583 4705 0317
B\$CSCN	001	14B0	0747	4217 4463 4479 5579 6008 7273 7314 7494 7615 7989 8717 8759 8945 9319 9777
B\$CSIF	UNDEFINED	SYMBOL	4182	
B\$CSMK	001	0007	0853	5575 6003 7305
B\$CSSW	001	14BC	0852	5575 6003 7305
B\$CSTP	001	06D6	0674	
B\$CSTR	001	14CC	0771	4233 4510
B\$CSXA	001	2000	0631	3445 4581 0355
B\$CTYP	001	0A5F	0725	5065* 5632* 6030* 6929*
B\$CVPD	001	0C5D	0730	0213
B\$CVPG	001	0CA5	0729	



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 193

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$CWRK	001	F500	0799	4005 4163 4306
B\$DIST	001	0700	0691	2947 3176 3527 3725 4278 4694 4857 5135 5403 5688 5991 6362
				6626 6773 6963 7156 7335 7499 7654 7863 8024 8173 8389 8581
				8795 8981 9196 9344 9653 9802
B\$DLNK	001	1B37	0797	5048 5124* 5125*
B\$DL4T	001	1A6B	0768	0177 0236 0243
B\$DPWA	001	0E46	0803	
B\$DRVA	UNDEFINED	SYMBOL		9595*
B\$DST2	001	073A	0692	2817 3418 3946 4100 4563 4662 0303
B\$EOST	UNDEFINED	SYMBOL		5732
B\$EQUL	UNDEFINED	SYMBOL		8431
B\$ERMK	001	0007	0826	6896 0168
B\$ERSW	001	0993	0825	6896 0168
B\$FACA	001	0E93	0734	3092 6269 6296
B\$FAIS	001	15AC	0751	0276
B\$FAIW	001	15A0	0752	0277
B\$FCON	001	0A46	0724	5103 5357 5634 6031 6930
B\$FORT	001	1B0E	0793	0319
B\$FPWA	001	15AC	0804	
B\$FRMK	001	0007	0844	
B\$FRSW	001	16CC	0843	
B\$FSC1	001	0E4C	0735	6306*
B\$FSC2	001	0E4D	0736	6316* 6324*
B\$FSMK	001	0007	0835	6336 6341
B\$FSSW	001	0E5C	0834	6336* 6341*
B\$FSVA	001	0E4F	0737	6330* 6331*
B\$FTND	001	1B0B	0795	5383
B\$FTPT	001	1B0D	0794	5380 5382* 5383 5388* 7818 7853* 0188
B\$FVME	001	15A2	0756	5186
B\$FVMP	001	15A4	0757	5191
B\$FVMS	001	15A6	0758	5196
B\$FVPE	001	15A8	0753	5171
B\$FVPP	001	15AA	0754	5176
B\$FVPS	001	15AC	0755	5181
B\$GBSW	001	08AF	0828	
B\$GBWK	001	0001	0829	
B\$GETC	001	0867	0705	2682 2727 3083 3087 3109 3154 3173 3307 3311 3322 3329 3331
				3373 3515 3699 3703 3859 3966 4034 4076 4098 4213 4239 4277
				4450 4478 4494 4532 4617 4815 5059 5076 5086 5092 5328 5339
				5349 5558 5925 6248 6301 6307 6318 6325 6517 6568 6751 6909
				7134 7272 7287 7467 7488 7614 7628 7633 7810 7988 8002 8148
				8308 8359 8537 8572 8713 8725 8765 8944 8958 9113 9158 9164
				9318 9338 9584 9776 9796
B\$GETS	UNDEFINED	SYMBOL		5623 8001
B\$GFIC	UNDEFINED	SYMBOL		8320
B\$GPIR	UNDEFINED	SYMBOL		4232
B\$GPTF	UNDEFINED	SYMBOL		2781
B\$GPTR	001	0878	0707	3356 3719 3878 3912 3992 4509 4609 5633 5707 6023 6097 6264
				6544 6767 7150 7329 7648 8352 8566 8758 9128 9155 9604 9640
B\$GRTR	UNDEFINED	SYMBOL		3132 8728
B\$GTBF	001	1E00	0629	
B\$IDRP	UNDEFINED	SYMBOL		0372*
B\$IFMK	001	0007	0847	
B\$IFSW	001	16E5	0846	
B\$INVT	001	1B38	0787	2735 2866 2898
B\$KWMK	001	0001	0841	



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 194

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B\$KWSW	001	159E	0840	3885* 4221* 4467* 4480*
B\$LBAS	001	185E	0778	3911
B\$LBSV	001	18E7	0776	3910*
B\$LCMC	UNDEFINED	SYMBOL		4673
B\$LDRP	001	1A00	0626	0254 0254* 0258* 0262* 0268* 0272* 0276* 0277* 0368* 0373* 0374* 0378* 0382 0382* 0383 0383* 0396 0466 0505
B\$LINE	001	07D0	0693	6900
B\$LIST	001	1853	0760	2731 3707 6545 7480 7637
B\$LORP	UNDEFINED	SYMBOL		0431
B\$LRTN	001	18EB	0777	3909*
B\$LSTA	UNDEFINED	SYMBOL		6595
B\$LSTR	001	1862	0775	3913
B\$LTYP	001	18F2	0761	2740
B\$MATR	001	18F3	0763	3310 3372 3382 3475 3481 3516 3535 6756 8006 8542 8963 9629
B\$MBMK	001	0007	0862	3383
B\$MBSW	001	1903	0861	3381* 3383*
B\$MFBK	001	1B8F	0789	3321* 3328* 3335 3364 3366 3473 3480 3486* 3487* 3493 3534
B\$MFCK	UNDEFINED	SYMBOL		3330*
B\$MGMK	001	0007	0859	3380 3384 3474 3482 3533 3536
B\$MGSW	001	18FF	0858	3380* 3384* 3474* 3482* 3533* 3536*
B\$MPMK	001	0007	0865	3308 3312
B\$MPSW	001	1981	0864	3308* 3312*
B\$MRMK	001	0007	0856	
B\$MRSW	001	0DDE	0855	
B\$NUMC	001	0873	0706	2681* 3082* 3306* 3472 3479* 3532* 3698* 3858* 4033* 4216* 4238* 4447* 4462* 4477* 4531* 4625* 4814* 5017* 5082* 5099* 5338* 5348* 5356* 5557* 5623* 5924* 6247* 6317* 6750* 6908* 6931* 7133* 7271* 7286* 7310* 7466* 7613* 7632* 7809* 7987* 8001* 8147* 8307* 8329* 8536* 8541* 8712* 8734* 8764* 8943* 8957* 9112* 9163* 9317* 9337* 9775* 9795*
B\$NUNC	UNDEFINED	SYMBOL		8358*
B\$NXMK	001	0007	0832	2943 4272 4853 5131 6358 6621 6959 7857 9192
B\$NXSM	UNDEFINED	SYMBOL		4853*
B\$NXSN	UNDEFINED	SYMBOL		6621*
B\$NXSW	001	071D	0831	2943* 4272* 5131* 6358* 6959* 7857* 9192*
B\$PARP	001	0A41	0714	
B\$PBNL	001	0A01	0720	6898*
B\$PCAD	001	0A40	0715	2687* 2699* 2775* 2788* 2911* 2930* 2937* 3499* 3521* 3541* 3713* 3988* 4113* 4291* 4485* 4491* 4672* 4680* 4825* 4833* 5022* 5055* 5119* 5360* 5375* 5705* 6095* 6253* 6347* 6528* 6552* 6556* 6576* 6594* 6603* 6612* 6761* 6902* 6914* 6946* 6953* 6978* 6985* 7144* 7280* 7323* 7496* 7622* 7642* 7847* 7995* 8012* 8157* 8375* 8552* 8560* 8774* 8781* 8951* 8969* 9118* 9135* 9173* 9180* 9324* 9331* 9464* 9589* 9601* 9634* 9647* 9782* 9920* 0201* 0668* 0781*
B\$PCAP	UNDEFINED	SYMBOL		8368*
B\$PCDL	001	09D3	0719	6291
B\$PCPG	001	0A35	0718	0262
B\$PECT	001	0A44	0722	0183
B\$PERC	001	0A39	0721	3104* 3309* 3378 5390* 6276* 7833* 7838*
B\$PFAE	001	0033	0712	3103 5389 6275 7824
B\$PFCL	001	009D	0713	0173 0208
B\$PFNC	001	094E	0710	3103* 5389* 6275* 7824* 0173* 0208* 0219*
B\$PFWP	001	0015	0711	0219
B\$PNBY	001	0A41	0716	2688* 2700* 2776* 2789* 2912* 2931* 2938* 3500* 3522* 3542* 3714* 3989* 4114* 4292* 4486* 4492* 4673* 4681* 4826* 4834* 5027* 5361* 5373* 5658* 5700* 5930* 5941* 6054* 6090* 6288* 6348* 6529* 6553* 6557* 6577* 6595* 6604* 6613* 6762* 6903* 6915* 6947* 6954* 6979* 6986* 7145* 7281* 7324*

## CROSS REFERENCE

SYMBOL LEN VALUE DEFN REFERENCES VER 15, MOD 00 20/07/20 PAGE 195

7497\* 7623\* 7643\* 7848\* 7996\* 8013\* 8158\* 8369\* 8376\* 8553\* 8561\* 8775\*  
8782\* 8952\* 8970\* 9119\* 9136\* 9181\* 9325\* 9332\* 9465\* 9590\* 9602\* 9635\*  
9648\* 9783\* 9921\* 0202\* 0669\* 0782\*

B\$PPWA 001 0A35 0801  
B\$PRM1 001 1AF3 0805  
B\$PRSL UNDEFINED SYMBOL  
B\$PRSS UNDEFINED SYMBOL  
B\$PTBF 001 1F00 0630  
B\$PUTC 001 093A 0709

4644\* 4678  
5746  
5750  
2689 2701 2777 2790 2913 2932 2939 3105 3359 3371 3468 3501  
3510 3523 3531 3543 3715 3990 4115 4293 4487 4493 4674 4682  
4827 4835 5028 5085 5109 5120 5362 5376 5391 5706 6096 6254  
6277 6289 6349 6530 6554 6558 6578 6596 6605 6614 6763 6904  
6916 6948 6955 6980 6987 7146 7282 7325 7498 7624 7644 7839  
7849 7997 8014 8159 8370 8377 8554 8562 8776 8783 8953 8971  
9120 9137 9175 9182 9326 9333 9466 9591 9603 9636 9649 9784

B\$PVAD 001 0A43 0717

9791 9922 0174 0203 0209 0220 0670 0783  
2693 2705 2713 2880 2885 3877 4203 4687 4828 4841 5032 5042  
5124 5935 5947 5956 6259 6290 6536 6585 6899\* 6920 8164 8382  
8788 9124 9141 9595 9609 9614 0258

B\$RMRK 001 1AE6 0770  
B\$RTRN 001 1AF5 0806  
B\$SABF 001 1C00 0627  
B\$SCAN 001 1514 0749

3379 9926 0674 0787  
3849 3851 3928\* 3975\* 4089  
0321  
3514 4052 4055 4133 4243 4250 4481 4489 5337 5341 5350 5999  
6340 6601 7291 8312 8353 9168

B\$SCAT 001 13C8 0744  
B\$SCLN UNDEFINED SYMBOL  
B\$SSCON 001 001B 0727

0538  
5728  
5632 6030 6929

B\$SCVT 001 12E0 0742  
B\$SDPL 001 07DA 0695  
B\$SFAB 001 0E48 0739  
B\$SFNT 001 143C 0745  
B\$SLDT 001 109C 0741  
B\$SLVT 001 1062 0740  
B\$SNAT 001 131A 0743

0378  
0242  
0272  
0542  
0372 0373 0374  
0368  
0534

B\$SPAT 001 07E0 0696  
B\$SSTA 001 1BAC 0791  
B\$STAS 001 061B 0680  
B\$STIF 001 0606 0682  
B\$STMA 001 061B 0681  
B\$STML 001 0600 0679  
B\$STRL 001 0600 0678  
B\$SVRB 001 0E46 0738

3883\* 4219\* 4465\* 5340\* 5342\*  
0266\* 0268

B\$SXA UNDEFINED SYMBOL  
B\$SYMB 001 0DBC 0733  
B\$TCD2 001 0001 0811  
B\$TLTH 001 0002 0812  
B\$TOD1 001 0000 0810  
B\$TOTB 001 1AF8 0813  
B\$TTAB 001 1AFA 0809  
B\$TYPE 001 0739 0694  
B\$WORK 001 15A0 0798  
B\$ZDBN 001 19F2 0765  
B\$ZDON UNDEFINED SYMBOL  
B@ABAS 001 0007 1397

2847  
3091 3884 4035 4220 4240 4466 5332 6265 7814  
4644  
0813 4638  
4639  
4637  
0813

B@ACD1 001 0001 1394  
B@ACD2 001 0003 1395

3919 4066 4130 4211  
3113 3158 4537 4819 5961 8152 8363 9130 9619  
8769  
1395 3150\*  
1396 3169\* 0392 0412\* 0557

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 196

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@AFLG	001	0000	1389	3097 3127* 3140* 3165*
B@ALLA	001	005C	1214	
B@AMAX	001	0005	1396	1397
B@BLNK	001	0040	1223	3486 3487 6324
B@BLSZ	001	0100	1348	1487 1490 1493 1508 1511 2847 2853 2875 3430 3445 3462 4011 4019 4026 4187 4581 4582 4600 0315 0316 0321 0355 0363 0550
B@BREQ	001	0084	1004	8417 8824
B@BRHI	001	0088	1005	8420 8827
B@BRLO	001	0082	1003	8423 8830
B@BRNE	001	0094	1007	8426 8435 8833 8842
B@BRNH	001	0098	1008	8429 8836
B@BRNL	001	0092	1006	8432 8839
B@CADD	001	0006	0873	
B@CADF	001	0058	0914	7341 7660 8030 8987 9350 9808
B@CBAS	001	0003	1400	
B@CBNX	001	004A	0907	6107 9662
B@CBRA	001	0046	0905	2833 3999 4300 4311 4866 5141 6368 6638 6641 7015 7884 8179
B@CBRC	001	0044	0904	4698 8396 8803
B@CBRD	001	0048	0906	6382
B@CBRS	001	004C	0908	2959 7024 9205
B@CCLS	001	005E	0917	9811
B@CCMC	001	0042	0903	4702 8801
B@CCMF	001	0040	0902	8395
B@CCNT	001	001F	1326	
B@CCSA	001	003E	0901	9207
B@CDCA	001	006A	0923	5144
B@CDDL	001	006C	0924	5147
B@CDIV	001	000C	0876	
B@CDMN	001	0001	1399	1400 3123* 0427 0447*
B@CDWA	001	006E	0925	5415
B@CEOF	001	0070	0926	0343
B@CEOP	001	0068	0922	
B@CFCI	001	0016	0881	
B@CFNO	UNDEFINED	SYMBOL		4314 4594
B@CFN0	001	0012	0879	4165
B@CFN1	001	0014	0880	
B@CFOR	001	004E	0909	5409
B@CGET	001	0052	0911	2836 3731 7663
B@CHAR	001	0000	1339	2782 3133 3174 3316 3321 3323 3328 3330 3336 3338 3340 3357 3720 3964 4041 4053 4214 4244 4457 4613 4618 4620 4631 5063 5071 5077 5091 5093 5113 5343 5593 5647 5974 6024 6306 6311 6316 6546 6768 7151 7330 7486 7649 8019 8316 8321 8323 8335 8546 8567 8576 8721 8726 8728 8740 8976 9156 9339 9641 9797
B@CHLT	001	0004	0872	9476
B@CIEX	001	00C5	1299	5170 5185
B@CIMH	001	0066	0921	7012
B@CINI	001	0056	0913	2953
B@CIPI	001	00D7	1302	5175 5190
B@CIS2	001	00E2	1305	5180 5195
B@CMF1	001	0018	0882	3587 3591 3595 6779 7161 8587 8590 8990 9665
B@CMF2	001	001A	0883	3553 3579 3583
B@CMF3	001	001C	0884	3567 3571 3575
B@CMA	001	006B	1234	5647 5724 5766 9156
B@CMPY	001	000A	0875	
B@CMSM	001	001E	0885	
B@CMSN	UNDEFINED	SYMBOL		3550

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 197

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@CNEG	001	0010	0878	
B@CNXT	001	0050	0910	5412
B@COLN	001	007A	1236	
B@CPMK	001	00FF	1144	1148 1152 1153 1187
B@CPRS	001	0060	0918	5802
B@CPRU	001	0062	0919	6110 7018 9668
B@CPUT	001	0054	0912	7344
B@CPWR	001	000E	0877	
B@CRSR	001	005A	0915	0793
B@CRST	001	005C	0916	9353
B@CSA1	001	0036	0897	
B@CSA2	001	0038	0898	
B@CSB1	001	003A	0899	4171
B@CSC1	001	002A	0891	4168
B@CSD0	001	002E	0893	
B@CSD1	001	0030	0894	
B@CSD2	001	0032	0895	
B@CSF1	001	0022	0887	
B@CSF2	001	0024	0888	
B@CSTA	001	0034	0896	2830 4153 4305 4863 6104 6644 9202 9659
B@CSTC	001	0028	0890	4004 4156 4162 5805 6113 7021
B@CSTF	001	0020	0886	4174 5424 6647
B@CSTH	001	0064	0920	
B@CSTX	001	003C	0900	2956 4159 4308 4591
B@CSUB	001	0008	0874	
B@CSVC	001	0002	0871	9932 0342
B@CTYP	001	0020	1324	
B@CUSC	001	002C	0892	4002 4317 7511
B@CUSF	001	0026	0889	4177 6650
B@CVAR	001	005B	1213	
B@DAMK	001	0080	1392	3097 3127
B@DASA	001	00FF	1153	
B@DASC	001	0040	1157	
B@DASM	001	0038	1155	
B@DCGT	001	0050	1163	
B@DCLS	001	0054	1169	
B@DDAT	001	0024	1149	
B@DDEF	001	0034	1150	
B@DDIM	001	0004	1151	
B@DDUM	001	00FF	1187	
B@DEC0	001	00F0	1282	
B@DEC1	001	00F1	1283	
B@DEC2	001	00F2	1284	
B@DEC3	001	00F3	1285	
B@DEC4	001	00F4	1286	
B@DEC5	001	00F5	1287	
B@DEC6	001	00F6	1288	
B@DEC7	001	00F7	1289	
B@DEC8	001	00F8	1290	
B@DEC9	001	00F9	1291	
B@DEND	001	0058	1185	1186 0351
B@DEOF	001	0058	1186	
B@DFOR	001	0028	1158	
B@DGET	001	0040	1166	
B@DGSB	001	0020	1164	
B@DGTO	001	0044	1162	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 198

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@DIFA	001	0048	1160	
B@DIFC	001	004C	1161	
B@DIGS	001	007B	1216	
B@DIMG	001	003C	1175	
B@DINP	001	0000	1170	2845
B@DIVD	001	0061	1233	
B@DLTA	001	00FF	1152	
B@DLTC	001	0040	1156	
B@DLTM	001	0038	1154	
B@DL01	001	0001	1467	1470 0258*
B@DL02	001	0003	1470	1473 0262*
B@DL03	001	0005	1473	1476 0268*
B@DL04	001	0007	1476	1479 0254 0254* 0272*
B@DL05	001	0009	1479	1482 0276*
B@DL06	001	000B	1482	1485 0277*
B@DL07	001	0045	1485	1488 0368*
B@DL08	001	0145	1488	1491 0372*
B@DL09	001	0245	1491	1494 0373*
B@DL10	001	0289	1494	1497 0374*
B@DL11	001	02C3	1497	1500 0378* 0396
B@DL12	001	02FD	1500	1503 0431
B@DL13	001	0337	1503	1506 0466
B@DL14	001	0371	1506	1509
B@DL15	001	0471	1509	1512 0382 0382*
B@DL16	001	0507	1512	0383 0383* 0505
B@DMAT	001	0008	1176	3442
B@DMGT	001	0044	1177	
B@DMIN	001	0038	1178	
B@DMPR	001	0048	1181	
B@DMPT	001	004C	1180	
B@DMPU	001	0054	1182	
B@DMRD	001	003C	1179	
B@DNXT	001	0044	1159	
B@DPNT	001	004B	1224	
B@DPRT	001	002C	1173	
B@DPRU	001	0030	1174	
B@DPSE	001	0050	1183	
B@DPUT	001	0040	1167	
B@DREA	001	000C	1171	
B@DREM	001	00FF	1148	
B@DRSR	001	005C	1172	
B@DRST	001	0050	1168	
B@DRTN	001	005C	1165	
B@DSCY	001	0004	1140	
B@DSIF	001	001C	1189	4579 4706
B@DSLT	001	0010	1188	
B@DSML	001	0010	1190	4008 4181
B@DSNS	001	0018	1142	
B@DSS1	001	0000	1141	
B@DSTP	001	0054	1184	
B@DTBN	001	0010	1206	0225
B@DTB1	001	0050	1205	0225
B@DTCY	001	0009	1202	
B@DTSN	001	0010	1204	
B@DTS1	001	0040	1203	
B@DTYP	001	0040	1318	

VER 15, MOD 00 20/07/20 PAGE 199

SYMBOL	LEN	VALUE	DEFN	REFERENCES						VER 15, MOD 00				20/07/20		PAGE 199	
B@DURE	001	0020	1037														
B@DVCY	001	0007	1199	0331													
B@DVC1	001	0056	1200	5036	6270	0332	0405	0440	0475								
B@DWCY	001	0005	1196														
B@DWT1	001	0003	1197														
B@D1MK	001	0080	1390	3140													
B@D2MK	001	00C0	1391	3165													
B@EOST	001	001E	1212	2782	3174	3323	3357	3720	5113	5774	5974	6024	6768	7031	7151		
				7330	7649	8019	8567	8576	8976	9339	9641	9797					
B@EQUL	001	007E	1238	3964	4618	7486	8321	8416	8726	8823	8835	8838					
B@EXPC	001	00C5	1215														
B@FOFL	001	005C	1217														
B@FVAD	001	0001	1402	6270	6297*	0462	0561										
B@GETC	001	0001	1341														
B@GETE	001	00FF	1342														
B@GETS	001	0000	1340	3472	3479	3532	4216	4462	4625	5099	5356	6931	7286	7310	7632		
				8329	8541	8734	8957	9337	9795								
B@GRIR	UNDEFINED SYMBOL			8832													
B@GRTR	001	006E	1235	8323	8419	8425	8431	8826	8838								
B@ICON	001	006C	1297	5071	5093												
B@LADD	001	0001	0942														
B@LADF	001	0002	0983	7281	7623	7996	8952	9325	9783								
B@LADV	001	0008	1426	1447													
B@LBIN	001	0002	1351	1352	1358												
B@LBNX	001	0003	0976	9602													
B@LBRA	001	0003	0974	2700	2789	4834	6380	6391	6398	6529	6577	6613	6915	7848	8158		
B@LBRC	001	0004	0973	4681	8376	8782											
B@LBRD	001	0003	0975	6348													
B@LBRS	001	0001	0977	2938	6954	9181	0669										
B@LCCA	001	0004	1382	0427	0508	0538	0538										
B@LCCC	001	0001	0935	0973	4700	4701	8398	8404	8805	8811							
B@LCDV	001	0004	1427	1448													
B@LCER	001	0001	0933	0997													
B@LCFN	001	0004	1383	0462	0509	0542	0542										
B@LCIN	UNDEFINED SYMBOL			8790													
B@LCLN	001	0002	0938	0989	0990	0997	4689	6900	7013	8384							
B@LCLS	001	0001	0986	9790													
B@LCMC	001	0001	0972	8775													
B@LCMF	001	0001	0971	8369													
B@LCNA	001	0006	1381	0392	0507	0530	0534	0534									
B@LCNN	001	0001	0936	0961	0970	0982	0994	2917	2954	5416	5418	5795	6372	6374	7481		
				7505	7512	9208											
B@LCOP	001	0001	0932	0940	0941	0942	0943	0944	0945	0946	0947	0948	0949	0950	0951		
				0952	0953	0954	0955	0956	0957	0958	0959	0960	0961	0962	0963		
				0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975		
				0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987		
				0988	0989	0990	0991	0992	0993	0994	0995	2830	2833	2836	2953		
				2956	2959	3550	3553	3567	3571	3575	3579	3583	3587	3591	3595		
				3731	3999	4002	4004	4153	4156	4159	4162	4165	4168	4171	4174		
				4177	4300	4305	4308	4311	4314	4317	4591	4594	4698	4702	4863		
				4866	5027	5141	5144	5147	5409	5412	5415	5424	5802	5805	6104		
				6107	6110	6113	6368	6371	6382	6638	6641	6644	6650	6779	7012		
				7015	7018	7021	7024	7161	7341	7344	7511	7660	7663	7884	8030		
				8033	8179	8395	8396	8587	8590	8801	8803	8987	8990	9202	9205		
				9207	9350	9353	9476	9659	9662	9665	9668	9808	9811	9932	0342		
				0343	0680	0793											



CROSS REFERENCE																			
SYMBOL	LEN	VALUE	DEFN	REFERENCES												VER 15, MOD 00	20/07/20	PAGE 200	
B@LCRV	001	0013	1425	1445	5796	6122	7032												
B@LCSA	001	0002	0970																
B@LCVA	001	0002	0934	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0959	0960				
				0962	0963	0964	0965	0966	0967	0968	0973	0974	0975	0976	0978				
				0979	0980	0992	0993	2831	2834	2837	3551	3554	3568	3572	3576				
				3580	3584	3588	3592	3596	3732	4699	4864	4867	5027	5142	5145				
				5148	5410	5425	6105	6108	6114	6369	6383	6639	6642	6645	6648				
				6780	7016	7162	7664	8034	8180	8397	8588	8591	8804	8991	9203				
				9214	9660	9663	9666												
B@LCXX	001	0001	0937	0969	0981	0983	0987	0988	2957	5803	6111	7019	7345	9669					
B@LDAT	001	0004	1096	5017															
B@LDCA	001	0003	0992																
B@LDDL	001	0003	0993																
B@LDDM	001	0004	1355																
B@LDEF	001	0003	1097	6247															
B@LDIM	UNDEFINED SYMBOL			3082															
B@LDIN	001	0004	1354	1355	1356														
B@LDIV	001	0001	0945																
B@LDMN	001	0002	1352	1381	1382	1394	1395	1396	1399	1426	1427	3123	3150	3169	0412				
				0447															
B@LDSN	001	0004	1356																
B@LDWA	001	0002	0994	5422	5440	6380	6391	6398											
B@LELP	001	0010	1424	5369	5419	5440													
B@LEND	001	0003	1124																
B@LEOF	001	0001	0995	0202															
B@LEOP	001	0001	0991																
B@LERC	001	0003	0997																
B@LESP	001	0008	1423	5418	5422														
B@LESS	001	004C	1225	8422	8425	8428	8829	8832	8835										
B@LET\$	001	005B	1245																
B@LET#	001	007B	1246																
B@LET@	001	007C	1247																
B@LETA	001	00C1	1249																
B@LETB	001	00C2	1251																
B@LETC	001	00C3	1252																
B@LETD	001	00C4	1253																
B@LETE	001	00C5	1254																
B@LETF	001	00C6	1255																
B@LETG	001	00C7	1256																
B@LETH	001	00C8	1257																
B@LETI	001	00C9	1258																
B@LETJ	001	00D1	1259																
B@LETK	001	00D2	1260																
B@LETL	001	00D3	1261																
B@LETM	001	00D4	1262																
B@LETN	001	00D5	1263																
B@LETO	001	00D6	1264																
B@LETP	001	00D7	1265																
B@LETQ	001	00D8	1266																
B@LETR	001	00D9	1267																
B@LETS	001	00E2	1268																
B@LETT	001	00E3	1269																
B@LETU	001	00E4	1270																
B@LETV	001	00E5	1271																



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 201

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LETY	001	00E8	1274	
B@LETZ	001	00E9	1275	
B@LEXP	001	0008	1314	
B@LFCI	001	0003	0950	
B@LFNA	001	0002	1428	1449
B@LFNO	UNDEFINED	SYMBOL		4492
B@LFN0	001	0003	0948	
B@LFN1	001	0003	0949	
B@LFOR	001	0003	0978	5422 5436 5440
B@LFRT	001	0004	1369	1370 5381 5433 7877
B@LGET	001	0003	0980	2776 3714 7643
B@LGSB	001	0005	1103	4814
B@LGTO	001	0004	1102	8147 9112
B@LHLT	001	0001	0941	9465
B@LIET	UNDEFINED	SYMBOL		4477
B@LIEX	001	0002	1300	5172 5187
B@LIFN	001	0003	1363	3493 3498 3566 3570 3574 3578 3582 3586 3590 3594
B@LILP	001	0009	1422	1440 1441 1442 6284 6376 6398
B@LIMG	001	0001	1114	
B@LIMH	001	0003	0990	6903
B@LINI	001	0002	0982	2931
B@LINP	001	0005	1109	2681
B@LINX	UNDEFINED	SYMBOL		5941
B@LIP1	001	0003	1303	5177 5192
B@LISP	001	0005	1421	1429 1435 1436 1437 6374 6380
B@LIS2	001	0005	1306	5182 5197
B@LIVT	001	0001	1379	
B@LKCL	001	0005	1108	9775
B@LKFR	001	0003	1099	5327
B@LKGT	001	0003	1105	7613
B@LKIF	001	0002	1101	4447 8307 8712
B@LKON	001	0002	1134	9163
B@LKPT	001	0003	1106	7271
B@LKPU	001	000A	1113	5924
B@LKRR	001	0007	1111	
B@LKRT	001	0005	1107	9317
B@LKTO	001	0002	1128	5338
B@LLET	001	0003	1098	3858 3989 3991 4033 4114 4116 4238 4292 4294 7466
B@LLO8	UNDEFINED	SYMBOL		0372
B@LL01	001	0002	1466	1467
B@LL02	001	0002	1469	1470
B@LL03	001	0002	1472	1473
B@LL04	001	0002	1475	1476
B@LL05	001	0002	1478	1479
B@LL06	001	0002	1481	1482
B@LL07	001	003A	1484	1485 0368 0368
B@LL08	001	0100	1487	1488 0372 0373 0374
B@LL09	001	0100	1490	1491 0374
B@LL1X	UNDEFINED	SYMBOL		0435
B@LL10	001	0044	1493	1494 0374 0374
B@LL11	001	003A	1496	1497 0378 0378
B@LL12	001	003A	1499	1500 0400
B@LL13	001	003A	1502	1503
B@LL14	001	003A	1505	1506 0470
B@LL15	001	0100	1508	1509 0382
B@LL16	001	0096	1511	1512 0383

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 202

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@LMAT	001	0003	1115	3306
B@LMF1	001	0003	0951	3500 6762 7145 8013 8561 8970 9635
B@LMF2	001	0003	0952	3542
B@LMF3	001	0003	0953	
B@LMGT	001	0006	1116	7987
B@LMIN	001	0008	1117	6750
B@LMPR	001	0008	1120	8536
B@LMPT	001	0006	1119	8943
B@LMPU	001	000D	1121	
B@LMPY	001	0001	0944	
B@LMRD	001	0007	1118	7133
B@LMSM	001	0003	0954	3522
B@LNEG	001	0001	0947	
B@LNEX	001	0004	1100	7809
B@LNXT	001	0003	0979	5422 5440
B@LPAR	001	004D	1226	3316 4041
B@LPRS	001	0002	0987	5700
B@LPRT	001	0005	1112	5557
B@LPRU	001	0002	0988	6090 6947 6986 9648
B@LPSE	001	0005	1122	
B@LPUT	001	0002	0981	7324
B@LPWR	001	0001	0946	
B@LREA	001	0004	1110	3698
B@LREM	001	0003	1095	
B@LRSR	001	0001	0984	0782
B@LRST	001	0001	0985	9332
B@LRTN	001	0006	1104	
B@LSA1	001	0003	0966	
B@LSA2	001	0003	0967	
B@LSB1	001	0003	0968	
B@LSC1	001	0003	0960	
B@LSDF	001	0004	1349	
B@LSD0	001	0003	0962	
B@LSD1	001	0003	0963	
B@LSD2	001	0003	0964	
B@LSF1	001	0003	0956	
B@LSF2	001	0003	0957	
B@LSKW	001	0002	1365	
B@LSNO	001	0002	1358	0322
B@LSPT	001	0003	1373	1376
B@LSTA	001	0003	0965	2688 4826 5930 9119 9136 9590
B@LSTC	001	0003	0959	5658 6054 6979
B@LSTE	001	0004	1129	
B@LSTF	001	0003	0955	5361 6553
B@LSTH	001	0003	0989	7030
B@LSTP	001	0004	1123	
B@LSTX	001	0002	0969	2912 4058 4247 4486
B@LSUB	001	0001	0943	
B@LSVC	001	0001	0940	9921 0202
B@LTHN	001	0004	1130	4531 8358 8764
B@LTYP	001	0001	1359	
B@LUFN	001	0002	1366	
B@LUSC	001	0002	0961	3962 4254 7497
B@LUSF	001	0001	0958	4134 6557 6604
B@LVPG	001	0100	1453	1456
B@MATR	UNDEFINED	SYMBOL		7138

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 203

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@MINS	001	0060	1232	3338 5184 5189 5194
B@MULT	001	005C	1229	3340
B@NAAR	001	001D	1417	1447 1499 0534
B@NCAR	001	001D	1418	1448 1502 0538
B@NCRV	001	001D	1416	1445 1496
B@NDGT	001	000A	1409	1415
B@NEQL	001	005F	1239	8434 8841
B@NFRT	001	000A	1368	1370
B@NICN	001	0006	1411	1413
B@NIEL	001	0007	1413	1429 1435 1440
B@NIFN	001	0018	1362	
B@NIVR	001	0001	1412	1413
B@NIVT	001	0057	1378	2718 2866
B@NLDV	001	0122	1415	1437 1442 1493
B@NLRV	001	001D	1414	1436 1441 1484
B@NLTR	001	001D	1408	1414 1415 1416 1417 1418 1419
B@NSKW	001	0004	1364	
B@NSPT	001	0028	1372	
B@NUFN	001	001D	1419	1449 1505 0542
B@NUMC	UNDEFINED	SYMBOL		9583*
B@NVPG	001	0100	1452	1456
B@NXLO	001	001E	1332	
B@NXMI	001	00E3	1333	
B@NXZR	001	0080	1331	1332 1333
B@PLUS	001	004E	1227	3336 5070 5169 5174 5179
B@POWR	001	005A	1228	
B@PREC	001	0020	1320	
B@PROD	001	0023	1429	
B@PRPL	001	0002	1017	5725
B@PRPN	001	0001	1016	5649 5737 5758 5771 5779
B@PRPR	001	0004	1019	5733
B@PRPS	001	0003	1018	5729
B@PRRC	001	0007	1022	5754 5775
B@PRRL	001	0008	1023	5646
B@PRSL	001	0005	1020	5767
B@PRSS	001	0006	1021	
B@PTAB	001	0000	1374	
B@PTAD	001	0001	1375	
B@PTSA	001	0002	1376	
B@PUD1	001	0006	1033	6014 6049
B@PUD2	001	0007	1034	6069
B@PUI0	001	0001	1027	6941
B@PUI1	001	0004	1028	6973
B@PUI2	001	0005	1029	6993
B@PUNL	001	0002	1031	5979
B@PUNS	001	0003	1032	6040
B@PUTM	001	0010	1036	5983 9669
B@RMNK	UNDEFINED	SYMBOL		9470
B@RPAR	001	005D	1230	3133 4053 4244 4482 6311
B@SADV	001	00E8	1447	1450
B@SAVL	001	0B76	1443	1460
B@SAVS	001	065E	1438	1459
B@SCAN	UNDEFINED	SYMBOL		5571
B@SCDV	001	0074	1448	1450
B@SCLN	001	005E	1231	5749 5770 8546
B@SCRV	001	0227	1445	1459 1460

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 204

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@SDMK	001	0080	1360	
B@SEXP	001	0004	1313	
B@SFAT	001	0196	1450	1459 1460 1511
B@SFNA	001	003A	1449	1450
B@SFRT	001	0028	1370	
B@SIEL	001	003F	1440	1443
B@SIES	001	0023	1435	1438
B@SIGN	001	0010	1322	
B@SLDL	001	0A32	1442	1443
B@SLDS	001	05AA	1437	1438
B@SLVL	001	0105	1441	1443
B@SLVS	001	0091	1436	1438
B@SQUO	001	007D	1237	4214 4457 5063
B@STAT	001	0000	1312	
B@TASA	001	0012	1048	
B@TASC	001	001E	1054	
B@TASM	001	0018	1050	
B@TASS	001	007B	1055	
B@TCGT	001	0030	1063	
B@TCLS	001	0042	1069	
B@TDAT	001	0006	1044	
B@TDEF	001	0009	1045	
B@TDIM	001	000C	1046	
B@TDUM	001	0078	1087	
B@TEND	001	0072	1085	
B@TEOF	001	0075	1086	
B@TFOR	001	0021	1057	
B@TGET	001	0039	1066	
B@TGSB	001	0033	1064	
B@TGTO	001	002D	1062	
B@TIFA	001	0027	1059	
B@TIFC	001	002A	1060	
B@TIFS	001	007D	1061	
B@TIMG	001	0054	1075	
B@TINP	001	0045	1070	
B@TLTA	001	000F	1047	
B@TLTC	001	001B	1051	
B@TLTM	001	0015	1049	
B@TLTS	001	0079	1052	
B@TMAS	001	007C	1056	
B@TMAT	001	0057	1076	
B@TMGT	001	005A	1077	
B@TMIN	001	005D	1078	
B@TMLS	001	007A	1053	
B@TMPR	001	0066	1081	
B@TMPT	001	0063	1080	
B@TMPU	001	0069	1082	
B@TMRD	001	0060	1079	
B@TNXT	001	0024	1058	
B@TPRT	001	004E	1073	
B@TPRU	001	0051	1074	
B@TPSE	001	006C	1083	
B@TPUT	001	003C	1067	
B@TRAC	001	0080	1316	
B@TREA	001	0048	1071	
B@TREM	001	0003	1043	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 205

SYMBOL	LEN	VALUE	DEFN	REFERENCES
B@TRSR	001	004B	1072	
B@TRST	001	003F	1068	
B@TRTN	001	0036	1065	
B@TSTP	001	006F	1084	
B@VMC1	001	0056	1455	
B@VMLB	001	F0CD	1460	
B@VMSB	001	F5E5	1459	
B@VMSZ	001	0000	1456	1458 1459 1460
B@VMTB	001	0000	1458	
B@WORK	UNDEFINED	SYMBOL	6540	
B@ZNEG	001	00D0	1329	
B@ZPOS	001	00F0	1328	
BAFVAD	UNDEFINED	SYMBOL	0482*	
BAGRTR	UNDEFINED	SYMBOL	4620	
BDBTAB	UNDEFINED	SYMBOL	5952	
BELL09	UNDEFINED	SYMBOL	0373	0373
BELMF1	UNDEFINED	SYMBOL	8553	
BILLET	UNDEFINED	SYMBOL	6516	
BINUMC	UNDEFINED	SYMBOL	5327*	
BIRDPA	UNDEFINED	SYMBOL	0497*	
BIRSHE	004	1EF1	0323	
BITAD2	001	0FE7	4577	4561
BITBLS	002	0FEF	4582	4569
BITBN1	002	0FF3	4584	
BITBRC	001	1086	4698	4679
BITB01	002	1088	4699	
BITB02	001	1089	4700	4678*
BITCA2	002	0FE8	4578	4449* 4543 4551 4557* 4562* 4568
BITCMC	001	108C	4702	4671
BITEN2	001	0006	4703	4649
BITERM	001	104A	4670	
BITFCP	002	0FEB	4580	4549* 4550* 4551
BITFNO	001	0FF8	4594	4490
BITFPE	002	0FED	4581	4549
BITLNG	002	108B	4701	4688
BITLSW	001	0FF4	4588	4440* 4448* 4515* 4516
BITOOP	002	0FFA	4595	
BITPBA	002	0FF1	4583	4543 4557
BITREL	001	1000	4608	
BITRE1	001	0F06	4446	
BITSG2	001	0000	4575	4570
BITSTX	001	0FF6	4591	4484
BITTRM	001	004A	4576	4533
BIT001	001	0FF5	4589	4515
BIT100	003	0F0D	4449	4441
BIT110	004	0F25	4465	4458
BIT120	004	0F64	4489	4483
BIT140	003	0F68	4490	4488
BIT150	004	0F73	4493	
BIT160	003	0F7E	4504	4469
BIT200	004	0F95	4515	4464 4495 4504
BIT240	004	101F	4631	4619 4621
BIT260	004	1023	4637	4626
BIT270	003	1027	4638	4640
BIT280	003	102A	4639	4613* 4631*
BIT290	003	1043	4661	4650

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 206

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BIT300	004	0FA9	4531	4517
BIT340	004	0FB8	4543	4526
BIT350	004	0FBF	4549	
BIT360	004	0FCF	4557	4544
BIT370	003	0FD3	4561	
BIT380	003	0FDE	4568	4552
BIT390	003	0FE4	4570	4525* 4533* 4562
BKABRC	001	1A87	8396	8374
BKAB01	002	1A89	8397	
BKAB02	001	1A8A	8398	8347*
BKACMC	001	1A86	8395	8367
BKALNG	002	1A8C	8404	8383
BKALTH	001	0002	8413	8341 8414
BKAODI	UNDEFINED	SYMBOL		8342
BKAOD1	001	0000	8411	
BKAOD2	001	0001	8412	
BKAOT1	001	1A8B	8414	
BKARIF	001	1A00	8302	
BKATAB	001	1A8D	8410	8414
BKA0D2	UNDEFINED	SYMBOL		8347
BKA0TB	UNDEFINED	SYMBOL		8340
BKA010	004	1A00	8307	
BKA020	004	1A08	8312	
BKA030	004	1A0C	8316	
BKA040	004	1A10	8320	
BKA050	004	1A20	8329	
BKA060	004	1A27	8335	8322 8324
BKA070	003	1A2B	8340	8330
BKA080	003	1A2E	8341	8343
BKA090	003	1A31	8342	8316* 8335*
BKA100	004	1A37	8347	
BKA110	004	1A3B	8352	
BKA120	004	1A43	8358	
BKA130	004	1A4B	8363	
BKA140	003	1A4F	8367	
BKA150	003	1A5E	8374	
BKA160	006	1A6D	8382	
BKA170	004	1A82	8389	
BKCBO1	002	1B89	8804	
BKCBO2	001	1B8A	8805	8753*
BKCBRC	001	1B87	8803	8780
BKCCD2	001	0001	8819	8753
BKCCMC	001	1B86	8801	8773
BKCLNG	002	1B8C	8811	8789
BKCLTH	001	0002	8820	8747 8821
BKCOD1	001	0000	8818	8748
BKCOTB	001	1B8B	8821	8746
BKCRIF	001	1B00	8707	
BKCTAB	001	1B8D	8817	8821
BKC010	004	1B00	8712	
BKC020	004	1B08	8717	
BKC030	004	1B0C	8721	
BKC040	004	1B10	8725	
BKC050	004	1B20	8734	
BKC060	004	1B27	8740	8727 8729
BKC070	003	1B2B	8746	8735

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 207

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BKC080	003	1B2E	8747	8749
BKC090	003	1B31	8748	8721* 8740*
BKC100	004	1B37	8753	
BKC110	004	1B3B	8758	
BKC120	004	1B43	8764	
BKC130	004	1B4B	8769	
BKC140	003	1B4F	8773	
BKC150	003	1B5E	8780	
BKC160	006	1B6D	8788	
BKFBN2	002	12E7	5434	
BKFDAC	001	12BE	5415	
BKFDAN	001	12BF	5416	5369* 5417
BKFLLP	001	0027	5440	5368
BKFLSP	001	0001	5441	5348
BKFOCV	001	0001	5442	5396*
BKFOC1	001	12E8	5435	5355
BKFOFA	001	12E0	5420	5368* 5373 5421
BKFOFC	001	12B8	5409	5374
BKFOFO	002	12BA	5410	5396
BKFOF0	UNDEFINED	SYMBOL		5333*
BKFONC	001	12BB	5412	
BKFOND	UNDEFINED	SYMBOL		5397* 5399*
BKFONI	UNDEFINED	SYMBOL		5398*
BKFONL	001	0003	5443	
BKFONO	002	12BD	5413	
BKFOPR	032	12DF	5419	
BKFORX	001	1200	5323	
BKFOSC	001	12E1	5424	5359
BKFOSO	002	12E3	5425	5358*
BKFOTL	002	12E5	5433	5388
BKFOX3	002	12EA	5436	5399
BKF010	004	1200	5327	
BKF020	004	1208	5332	
BKF030	004	1211	5337	
BKF040	004	122F	5348	
BKF050	003	123E	5355	5344
BKF060	004	125D	5366	5351
BKF070	005	126A	5373	5367
BKF080	004	127A	5380	
BKF090	005	128E	5388	
BKF100	004	12A2	5396	5384
BKF120	004	12B4	5403	5392
BKGBN1	UNDEFINED	SYMBOL		8165
BKGBRC	001	19E7	8179	8156
BKGBRO	002	19E9	8180	
BKGIN1	002	19EB	8186	
BKGOTO	001	19B3	8143	
BKG010	004	19B3	8147	
BKG020	004	19BB	8152	
BKG030	003	19BF	8156	
BKG040	006	19CE	8163	
BKG050	004	19DF	8169	
BKG060	004	19E3	8173	
BKKBRC	001	1C9F	9205	
BKKSTC	001	1C9C	9202	
BKMBN1	002	1CA3	9214	9142 9151 9188



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 208

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BKMBRC		UNDEFINED	SYMBOL	9179
BKMCSC	001	1CA0	9207	9172
BKMCSO	001	1CA1	9208	9129* 9151*
BKMGTO	001	1C00	9108	
BKMSTC		UNDEFINED	SYMBOL	9117 9134
BKMSTO	002	1C9E	9203	
BKMOVAD	002	1CA5	9215	9124* 9187
BKM010	004	1C00	9112	
BKM020	003	1C08	9117	
BKM030	005	1C17	9124	
BKM035	004	1C1C	9128	
BKM040	004	1C23	9130	
BKM050	003	1C27	9134	
BKM060	006	1C36	9141	
BKM070	006	1C41	9146	
BKM080	004	1C4B	9151	
BKM090	004	1C4F	9155	
BKM100	004	1C60	9163	9157 9159
BKM110	004	1C68	9168	
BKM125	003	1C7B	9179	
BKM130	005	1C8A	9187	
BKM140	004	1C94	9192	
BKM150	004	1C98	9196	
BKNBRC	001	1962	7884	7846
BKNBRO	002	1964	7885	7845* 7858
BKNDUM	001	0000	7872	7828
BKNEXT	001	1900	7805	
BKNEX2	002	1961	7878	7859
BKNFEL	002	195F	7877	7853
BKNFTD	001	0001	7871	7819 7828
BKNNXT	001	0003	7873	7845
BKN010	004	1900	7809	
BKN020	004	1908	7814	
BKN030	004	190C	7818	
BKN040	004	1918	7824	
BKN050	003	191C	7828	
BKN060	004	1922	7833	
BKN070	004	1929	7838	7829
BKN080	004	192D	7839	7834
BKN090	004	1934	7845	7820
BKN100	005	1947	7853	
BKN110	004	194C	7857	
BKN120	004	195A	7863	7840
BKRBRRC	001	1FE2	0680	0667
BKRTRN	001	1FCF	0663	
BKR010	003	1FCF	0667	
BKR020	004	1FDE	0674	
BKSBN1	002	10ED	4873	4842 4848
BKSBRC	001	10E9	4866	4832
BKSBRO	002	10EB	4867	
BKSTAC	001	10E6	4863	4824
BKSTAO	002	10E8	4864	
BKSUBG	001	1090	4810	
BKSVAS	002	10EF	4879	4828* 4847
BKS010	004	1090	4814	
BKS020	004	1098	4819	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 209

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BKS030	003	109C	4824	
BKS040	003	10B0	4832	
BKS050	006	10BF	4840	
BKS060	005	10D4	4847	
BKS070	004	10DE	4853	
BKS080	004	10E2	4857	
BMDM1C	001	1AEA	8587	8551
BMDM10	002	1AEC	8588	
BMDM2C	001	1AED	8590	8559
BMDM20	002	1AEF	8591	
BMDPRT	001	1A9B	8532	
BMD010	004	1A9B	8536	
BMD020	004	1AA3	8541	8577
BMD030	003	1AAB	8546	
BMD040	003	1AB1	8551	
BMD050	003	1AC3	8559	8547
BMD055	004	1AD2	8566	
BMD060	004	1ADC	8572	8555
BMD070	003	1AE0	8576	
BMD080	004	1AE6	8581	8568
BMGAFC	001	19AC	8030	7994
BMGAFO	001	19AD	8031	
BMGBN1	002	19B2	8043	
BMGETX	001	1965	7983	
BMGMFC	001	19AE	8033	8011
BMGMFO	002	19B0	8034	
BMGSFA	001	19B1	8042	
BMG010	004	1965	7987	
BMG100	003	1971	7994	
BMG110	004	1980	8001	
BMG120	004	1988	8006	8020
BMG140	003	198F	8011	
BMG150	004	199E	8018	8007*
BMG160	004	19A8	8024	
BMIMFC	001	16FB	6779	
BMIMFO	002	16FD	6780	
BMINPT	001	16D2	6746	
BMI010	004	16D2	6750	
BMI020	004	16DA	6756	6769
BMI030	003	16DE	6760	
BMI040	004	16ED	6767	
BMI050	004	16F7	6773	
BMKKBK0	UNDEFINED	SYMBOL		3534
BMKKBK2	UNDEFINED	SYMBOL		3493
BMMAD2	001	0AF3	3440	3417
BMMATA	001	0A00	3300	3462
BMMAT2	001	0B00	3464	3325 3326 3345 3346 3351 3352 3389 3390
BMMBK0	001	0000	3449	3321* 3473 3486*
BMMBK1	001	0001	3450	3328* 3335 3364 3366
BMMBK2	001	0002	3451	3330* 3480 3487*
BMMBLS	002	0AF0	3430	3423
BMMCA2	002	0AF4	3441	3301* 3398 3407 3413*
BMMFCP	002	0AF7	3444	3404* 3405* 3407
BMMFND	001	0002	3562	3493
BMMFPE	002	0AF9	3445	3404
BMMIA2	001	0AF5	3442	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 210

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BMMINV	001	00D5	3437	3364
BMMMSC	001	0B99	3550	3520
BMMMSO	002	0B9B	3551	
BMMM2C	001	0B9C	3553	3540
BMMM2O	002	0B9E	3554	
BMPBA	002	0AF2	3433	3325* 3345* 3351* 3389* 3398 3413
BMPID	001	0003	3561	
BMPPI	001	0004	3454	3442
BMPSG2	001	0000	3453	
BMMTAB	001	0B9F	3564	3598
BMMTBS	001	0B99	3598	
BMMTB5	UNDEFINED	SYMBOL		3491
BMMTEL	001	0006	3560	3492 3598
BMMTRN	001	00D9	3438	3366
BMM005	005	0A3D	3328	3324
BMM010	003	0A65	3345	3337 3339
BMM020	003	0A6E	3351	3341
BMM030	004	0A85	3364	3358
BMM040	004	0A93	3371	3365
BMM050	004	0AA2	3378	3367
BMM060	003	0AC1	3389	3317
BMM070	004	0AC7	3398	3327 3347 3360 3374 3385
BMM080	004	0ADE	3413	3399
BMM090	003	0AE9	3423	3408
BMM095	003	0AEC	3424	3326* 3346* 3352* 3390*
BMM100	004	0B00	3468	3345 3346
BMM110	003	0B2C	3491	3351 3352
BMM120	003	0B2F	3492	3494
BMM130	003	0B3A	3498	
BMM140	004	0B4C	3510	3389 3390
BMM150	004	0B6B	3527	3505 3544
BMM160	004	0B6F	3531	3325 3326
BMPAFC	001	1BE2	8987	8950
BMPAFO	001	1BE3	8988	
BMPBN1	002	1BE8	8999	
BMPMFC	001	1BE4	8990	8968
BMPMFO	002	1BE6	8991	
BMPSFA	001	1BE7	8997	
BMPUTX	001	1B9B	8939	
BMP010	004	1B9B	8943	
BMP100	003	1BA7	8950	
BMP110	004	1BB6	8957	
BMP120	004	1BBE	8963	8977
BMP130	003	1BC5	8968	
BMP140	004	1BD4	8975	8964*
BMP150	004	1BDE	8981	
BMREAD	001	17D0	7129	
BMRMFC	001	17F9	7161	7143
BMRMFO	002	17FB	7162	
BMR010	004	17D0	7133	
BMR020	004	17D8	7138	7152
BMR030	003	17DC	7143	
BMR040	004	17EB	7150	
BMR050	004	17F5	7156	
BMUBNC	001	1D8B	9662	9600
BMUBN1	002	1D94	9673	9615

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 211

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BMUMFC	001	1D8E	9665	9633
BMUMFO	002	1D90	9666	
BMUPRC	001	1D91	9668	9646
BMUPRO	001	1D92	9669	
BMUPRT	001	1D00	9579	
BMURNO	002	1D8D	9663	
BMURN1	UNDEFINED	SYMBOL		9596
BMUSTC	001	1D88	9659	9588
BMUSTO	002	1D8A	9660	
BMU010	004	1D00	9583	
BMU020	003	1D08	9588	
BMU030	006	1D17	9595	
BMU040	003	1D22	9600	
BMU050	006	1D35	9609	
BMU060	006	1D3F	9614	
BMU070	004	1D4A	9619	
BMU080	006	1D4E	9623	
BMU090	004	1D58	9629	9642
BMU100	003	1D5C	9633	
BMU110	004	1D6B	9640	
BMU120	003	1D75	9646	
BMU130	004	1D84	9653	
BNABNI	002	09F7	3182	
BNADIN	001	0973	3078	
BNAL20	UNDEFINED	SYMBOL		3118
BNA010	004	0973	3082	
BNA020	004	097B	3087	3175
BNA030	004	097F	3091	
BNA040	003	098A	3097	
BNA060	004	099C	3109	3099
BNA070	004	09A0	3113	
BNA080	004	09A4	3117	
BNA090	004	09AB	3119	3093* 3135 3145 3160
BNA100	005	09AF	3123	
BNA110	003	09B4	3127	
BNA120	004	09BA	3132	
BNA130	003	09CD	3145	3134
BNA140	005	09D0	3150	
BNA150	004	09D5	3154	
BNA160	004	09D9	3158	
BNA170	003	09E0	3165	
BNA180	005	09E3	3169	3141
BNA190	004	09E8	3173	3128
BNDATA	001	1100	5013	
BNDBKL	001	0002	5163	5080 5166
BNDBKT	001	11DA	5165	5070* 5077* 5080 5091* 5098
BNDBK0	001	0000	5154	5070* 5091* 5098
BNDBK1	001	0001	5155	5077* 5080
BNDBN1	001	11FA	5203	5125 5130
BNDBRC	001	11D1	5141	5021
BNDBRO	002	11D3	5142	
BNDDAC	001	11D4	5144	
BNDDAO	002	11D6	5145	5084* 5108*
BNDDLCL	001	11D7	5147	5118
BNDDLLO	002	11D9	5148	
BNDICA	001	0000	5162	5084

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 212

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BNDTAB	001	11DC	5168	5078
BNDTB1	001	0001	5158	5080
BNDTB3	001	0003	5159	5083
BNDTB4	001	0004	5160	5082
BNDTEL	001	0005	5157	5078 5079
BND010	004	1100	5017	
BND020	003	1104	5021	
BND030	006	1113	5032	
BND040	004	1119	5036	
BND050	006	1120	5042	
BND060	006	1129	5048	5037
BND070	003	1133	5054	5043
BND080	004	113A	5059	5114
BND090	003	113E	5063	
BND100	003	114B	5070	5064
BND110	004	1154	5076	5094
BND120	003	115F	5079	5081
BND130	004	1180	5091	5072
BND170	004	1195	5103	5066
BND180	005	1199	5108	
BND190	003	11A2	5113	5087
BND200	003	11A8	5118	
BND210	006	11B3	5124	
BND220	006	11BE	5129	
BND230	004	11CD	5135	
BNFBDC	001	15CB	6382	6346
BNFBDO	002	15CD	6383	6290* 6291* 6297 6330
BNFBNI	001	15CF	6392	
BNFBN1	UNDEFINED	SYMBOL		6354
BNFBRC	001	15BC	6368	6252
BNFBRO	002	15BE	6369	
BNFDAC	001	15BF	6371	
BNFDAN	001	15C0	6372	6284* 6373
BNFDEF	001	1500	6243	
BNFLIP	001	000D	6398	6283
BNFLTH	001	15CE	6391	6331
BNFSKP	001	0002	6396	6317
BNFSPA	001	15CA	6378	6283* 6288 6379
BNFWKA	009	15C9	6376	
BNF010	004	1500	6247	
BNF020	003	1508	6252	
BNF030	006	1513	6259	
BNF040	004	1519	6264	
BNF050	004	1521	6269	
BNF060	004	152B	6275	
BNF070	004	1537	6281	6271
BNF080	005	1544	6288	6282
BNF090	004	1557	6296	
BNF100	004	155F	6301	
BNF110	005	1563	6306	
BNF120	003	156C	6311	
BNF130	005	1572	6316	
BNF140	004	1582	6324	6312
BNF150	005	158A	6330	6319
BNF160	004	1594	6336	
BNF170	004	1598	6340	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 213

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BNF180	003	15A0	6346	
BNF190	005	15AF	6354	
BNF200	004	15B4	6358	
BNF210	004	15B8	6362	
BNIBN1	002	17CB	7029	6921 7001
BNIBRC	001	17C1	7015	6913
BNIBRO	002	17C3	7016	
BNIBSC	001	17C9	7024	6952
BNIEOS	001	17CD	7031	6925
BNIIHE	UNDEFINED	SYMBOL		6900*
BNIIHO	002	17C0	7013	
BNIIMH	001	17BE	7012	6901
BNIMAG	001	1700	6890	
BNIPRC	001	17C4	7018	6945 6984
BNIPRO	001	17C5	7019	6941* 6973* 6993*
BNISHL	001	17CC	7030	6898 6899
BNISTC	001	17C6	7021	6977
BNISTO	002	17C8	7022	6968* 6997*
BNISUB	002	17CF	7032	6997
BNI005	004	1725	6908	6897
BNI010	003	172D	6913	
BNI020	006	173C	6920	
BNI030	003	1747	6925	
BNI040	004	174A	6929	
BNI050	004	1756	6935	
BNI060	003	175D	6941	
BNI070	003	1760	6945	
BNI080	003	176F	6952	7006
BNI090	004	177E	6959	
BNI100	004	1782	6963	
BNI110	005	1786	6968	6936
BNI120	003	178B	6973	
BNI130	003	178E	6977	7002
BNI140	003	179D	6984	
BNI150	003	17AC	6993	
BNI160	004	17AF	6997	
BNI170	005	17B3	7001	
BNI180	003	17BB	7006	
BNODAC	UNDEFINED	SYMBOL		5054
BOLIMG	UNDEFINED	SYMBOL		6908
BPCASN	001	1871	7471	
BPCBN1	001	18A0	7505	7481
BPCLET	001	1869	7462	
BPCUCC	001	18A1	7511	7495
BPCUCO	001	18A2	7512	7476*
BPCUC0	UNDEFINED	SYMBOL		7481*
BPC010	004	1869	7466	
BPC020	003	1871	7476	
BPC030	004	1874	7480	7489
BPC040	003	187C	7486	
BPC050	004	1889	7494	7487
BPMASN	001	1608	6521	
BPMBIC	001	16C5	6638	6527 6575
BPMBIO	002	16C7	6639	
BPMBN1	UNDEFINED	SYMBOL		6584 6620
BPMBRC	001	16C8	6641	6611

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 214

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BPMBRO	UNDEFINED	SYMBOL	6536*	6583
BPMIND	001	16D1	6652	6562
BPMINI	002	16C4	6632	
BPMLET	001	1600	6512	
BPMSAC	001	16CB	6644	6593
BPMSAO	002	16CD	6645	6592*
BPMSFC	UNDEFINED	SYMBOL	6551	
BPMSFO	002	16CF	6648	6592
BPMSF0	UNDEFINED	SYMBOL	6540*	
BPMUFC	001	16D0	6650	6555 6602
BPM010	004	1600	6516	
BPM020	003	1608	6527	
BPM030	005	1617	6536	
BPM040	005	161C	6540	
BPM045	004	1621	6544	
BPM050	004	1625	6545	6569
BPM060	003	162D	6551	
BPM070	003	164B	6562	6546*
BPM080	004	1651	6568	
BPM090	003	1658	6575	6563
BPM100	005	1667	6583	
BPM110	004	167B	6592	
BPM120	004	168E	6601	
BPM130	003	16A1	6611	
BPM140	006	16B0	6619	
BPM150	004	16BF	6626	
BPREAD	001	0BCF	3694	
BPRGTC	001	0BFC	3731	3712
BPRGTO	002	0BFE	3732	
BPR010	004	0BCF	3698	
BPR020	004	0BD7	3703	3721
BPR030	004	0BDB	3707	
BPR040	003	0BDF	3712	
BPR050	004	0BEE	3719	
BPR060	004	0BF8	3725	
BPXRSC	001	1FF6	0793	0780
BPXRSR	001	1FE3	0776	
BPX010	003	1FE3	0780	
BPX020	004	1FF2	0787	
BRA050	004	0990	3103	
BRIMFC	UNDEFINED	SYMBOL	6760	
BSBCKT	UNDEFINED	SYMBOL	5653	
BSCSBF	UNDEFINED	SYMBOL	2854	
BSMBMK	UNDEFINED	SYMBOL	3381	
BSPCAD	UNDEFINED	SYMBOL	9789*	
BSPCDL	UNDEFINED	SYMBOL	5398	
BSPNBY	UNDEFINED	SYMBOL	9790*	
BSRVAD	UNDEFINED	SYMBOL	5397	
BSTRAS	001	0C1B	3866	
BSTRIF	001	0F00	4433	4600
BSTRLT	001	0C00	3847	4026 4187
BST010	004	0C0F	3852	3851*
BST020	004	0C13	3858	3850
BST080	003	0C1E	3875	
BST100	004	0C2E	3883	3967
BST120	004	0C3A	3895	



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 215

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BST130	003	0C4B	3908	3896
BST131	003	0C62	3918	3908
BST132	005	0C70	3926	3903
BST134	004	0C7C	3929	3976
BST136	004	0C92	3944	3930
BST138	003	0C9D	3951	3938
BST140	003	0CA6	3961	3921 3927
BST145	003	0CBC	3972	3965
BST150	003	0CCF	3987	3876 3920 3963
BST160	004	0CD6	3989	3962* 3991*
BST170	004	0CE5	3993	3987*
BST200	004	0D00	4033	
BST210	004	0D27	4052	4077
BST220	003	0D38	4057	4054
BST230	003	0D41	4065	4056
BST240	003	0D55	4075	4042
BST250	005	0D5F	4083	4070
BST260	004	0D70	4091	4089*
BST270	004	0D74	4097	4084
BST300	003	0D83	4112	4048 4051 4059 4067 4069 4132 4136 4139 4142 4144 4146
BST310	004	0D8A	4114	4058* 4116* 4134*
BST320	004	0D95	4117	4112*
BST340	003	0D99	4128	4075
BST360	004	0DD9	4147	4128*
BST400	003	0E00	4195	
BST410	004	0E3B	4219	4215
BST440	003	0E4E	4228	
BST460	004	0E51	4229	4242
BST500	004	0E65	4238	4223
BST540	004	0E77	4243	4241
BST545	004	0E8D	4250	4245
BST547	003	0E91	4251	4249
BST550	003	0EC2	4290	4196 4212 4248 4252 4255 4266
BST560	004	0EC9	4292	4247* 4254* 4294*
BST570	004	0ED4	4295	4290*
BST600	003	0E97	4253	4218 4228
BTPAUS	001	1CE7	9459	
BTPHTC	UNDEFINED	SYMBOL		9463
BTP010	003	1CE7	9463	
BTP020	004	1CF6	9470	
BTRAD2	001	1EFA	0349	0302
BTRBLS	002	1EE7	0315	0308
BTRBND	001	00FF	0550	0213
BTRCA2	002	1EFB	0350	0164* 0285 0292 0298*
BTRCCD	UNDEFINED	SYMBOL		0447
BTRCCE	UNDEFINED	SYMBOL		0427*
BTRCCL	002	1FB9	0508	0451
BTRCCP	002	1FCC	0536	0426 0451*
BTRCFA	001	1FC6	0562	0482
BTRCFE	001	1FC6	0561	0462* 0562
BTRCFL	002	1FBB	0509	0486
BTRCFP	002	1FCE	0540	0461 0486*
BTRCND	001	1FC8	0558	0412
BTRCNE	001	1FC8	0557	0392* 0558
BTRCNL	002	1FB7	0507	0416
BTRCNP	002	1FCA	0532	0391 0416*

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 216

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BTRCTP	004	1F61	0565	0452*
BTRDPA	002	1FB3	0499	
BTRDPL	001	1FBD	0518	0496
BTRECA	002	1EF7	0334	
BTRECY	001	1EF3	0331	
BTREFN	001	1EF2	0330	
BTREOF	001	1EF9	0343	
BTREPL	001	1EF2	0329	0176
BTRESA	001	1EF4	0332	
BTRESC	001	1EF5	0333	
BTRFAC	002	1FB5	0505	0411 0446 0481
BTRFCP	002	1EFE	0353	0291* 0292
BTRFTA	002	1EED	0319	0188
BTRFTP	004	1F8B	0566	0487*
BTRMNT	001	1E00	0163	0363
BTRNTP	004	1F37	0564	0417*
BTRPBA	UNDEFINED	SYMBOL		0285 0298
BTRPCA	001	1EF8	0340	0200
BTRPRA	002	1EEB	0317	
BTRPSI	001	0004	0549	0351
BTRSA2	001	1EFC	0351	
BTRSEL	001	0004	0322	0240 0323
BTRSG2	001	0000	0548	0309
BTRSHA	001	1CFF	0321	0240*
BTRSHE	UNDEFINED	SYMBOL		0240
BTRSTL	001	1FBC	0511	0417 0452 0487
BTRSVL	001	1EF8	0342	
BTRTEN	001	1FC3	0529	0556 0557 0559 0561
BTRVAD	001	1FC4	0556	0398 0405 0410 0433 0440 0445 0468 0475 0480
BTRVBA	002	1EE9	0316	0266
BTR010	004	1E03	0168	
BTR020	004	1E0A	0173	
BTR030	004	1E19	0181	
BTR040	005	1E2B	0188	0169
BTR050	004	1E33	0193	
BTR060	003	1E3F	0200	0189
BTR070	004	1E4E	0208	
BTR080	004	1E56	0213	
BTR090	004	1E5D	0219	
BTR100	004	1E65	0224	0214
BTR110	004	1E70	0230	
BTR120	004	1E7C	0236	0226
BTR130	005	1E80	0240	
BTR150	006	1E93	0254	
BTR160	006	1E99	0258	
BTR170	006	1E9F	0262	
BTR180	005	1EA5	0266	
BTR190	006	1EB0	0272	
BTR200	006	1EB6	0276	
BTR250	004	1EC2	0285	
BTR260	005	1EC9	0291	
BTR270	004	1ED5	0298	
BTR280	003	1ED9	0302	0286
BTR290	003	1EE0	0308	0293
BTR300	006	1F00	0368	
BTR310	006	1F06	0372	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 217

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BTR320	006	1F18	0378	
BTR330	006	1F1E	0382	
BTR350	003	1F2A	0391	0418
BTR360	004	1F31	0396	
BTR370	004	1F35	0398	0399 0401 0564
BTR380	003	1F39	0405	
BTR390	003	1F3F	0410	
BTR400	004	1F49	0416	0406
BTR410	003	1F54	0426	0453
BTR420	004	1F5B	0431	
BTR430	004	1F5F	0433	0434 0436 0565
BTR440	003	1F63	0440	
BTR450	003	1F69	0445	
BTR460	004	1F73	0451	0441
BTR470	003	1F7E	0461	0488
BTR480	004	1F85	0466	
BTR490	004	1F89	0468	0469 0471 0566
BTR500	003	1F8D	0475	
BTR510	003	1F93	0480	
BTR520	004	1F9D	0486	0476
BTR600	003	1FA8	0496	
BTSSVC	001	1DE9	9932	9919
BTSTOP	001	1DD6	9915	
BTS010	003	1DD6	9919	
BTS020	004	1DE5	9926	
BWOUL	UNDEFINED	SYMBOL		6652
BXCAFC	001	1DD1	9808	9781
BXCAFO	001	1DD2	9809	
BXCBN1	002	1DD5	9821	
BXCCLC	001	1DD3	9811	9788
BXCLOS	001	1D95	9771	
BXCO20	004	1D9D	9777	
BXCSFA	001	1DD4	9819	
BXC010	004	1D95	9775	
BXC020	UNDEFINED	SYMBOL		9798
BXC120	003	1DA1	9781	
BXC130	003	1DB0	9788	
BXC140	004	1DBF	9795	
BXC150	004	1DCD	9802	
BXDBN1	001	13EF	5795	5663
BXDDMY	001	0009	5791	
BXDDP0	001	0000	5788	5597 5599
BXDDP1	001	0001	5789	
BXDDP2	001	0002	5790	5605
BXDDUM	001	0000	5717	5599 5736 5757 5778
BXDLTH	001	0003	5714	5566 5588 5596 5638 5644 5791
BXDMD1	001	13CB	5723	5566
BXDMD2	001	13D7	5744	5588
BXDMD3	001	13E3	5765	5638
BXDMP1	UNDEFINED	SYMBOL		5644
BXDM14	001	13D6	5738	5567* 5645*
BXDOP1	UNDEFINED	SYMBOL		5604
BXDPRC	001	13F2	5802	5699
BXDPRO	001	13F3	5803	5604* 5646* 5649*
BXDPRT	001	1300	5553	5566 5567 5588 5638 5644 5645 5726 5730 5734 5747 5751 5755 5759 5768 5772 5776 5780

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 218

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BXDRM1	001	0007	5813	5562 5614
BXDROM	001	0004	5715	
BXDRS1	003	13A8	5812	5562* 5614*
BXDSTC	001	13F4	5805	5657
BXDSTO	002	13F6	5806	5653* 5672*
BXDSUB	002	13F1	5796	5672
BXD010	004	1300	5557	
BXD020	003	1308	5562	
BXD030	003	130B	5566	5615 5624 5772
BXD040	004	1311	5571	
BXD050	004	1315	5575	
BXD060	004	1319	5579	
BXD065	004	131D	5583	
BXD070	003	1324	5588	
BXD080	004	1327	5593	5584 5640 5664
BXD090	003	132B	5594	5566* 5588* 5638* 5644*
BXD095	003	132E	5596	5600
BXD100	003	1331	5597	5593*
BXD110	004	133D	5604	5598
BXD120	003	1345	5606	5605*
BXD140	003	1348	5610	5726 5730 5747 5751 5768
BXD150	003	134B	5614	
BXD160	003	1351	5619	5645
BXD170	004	1354	5623	5780
BXD180	003	135B	5628	5567
BXD190	004	135E	5632	5759
BXD200	003	136A	5638	
BXD210	003	1374	5644	
BXD220	005	1386	5653	
BXD230	003	138B	5657	5673
BXD240	005	1395	5663	
BXD250	003	139D	5668	
BXD260	004	13A0	5672	
BXD270	003	13A7	5677	5678 5680 5755 5812
BXD280	003	13AA	5684	5734 5776
BXD290	004	13AD	5688	5677
BXD300	003	13B1	5699	5610 5619 5628 5648 5668 5684
BXD310	003	13B8	5704	5659
BXD320	004	13C7	5708	5704*
BXGAFC	001	18EB	7660	7621
BXGAFO	001	18EC	7661	
BXGBN1	002	18F1	7673	
BXGETX	001	18A3	7609	
BXGGTC	001	18ED	7663	7641
BXGGTO	002	18EF	7664	
BXGI60	004	18E7	7654	
BXGSFA	001	18F0	7672	
BXG010	004	18A3	7613	
BXG100	003	18AF	7621	
BXG110	004	18BE	7628	
BXG120	004	18C6	7633	7650
BXG130	004	18CA	7637	
BXG140	003	18CE	7641	
BXG150	004	18DD	7648	
BXIAD2	001	08EE	2843	2816
BXIBLS	002	08F6	2853	2823

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 219

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BXIBN1	002	08FA	2855	2694 2714 2751 2763 2769
BXIBRC	001	08E8	2833	2698 2787
BXIBRO	002	08EA	2834	
BXIBSC	001	0970	2959	2936
BXICA2	002	08EF	2844	2677* 2798 2806 2812* 2822
BXICMK	001	0080	2867	2745 2757 2765
BXIFCP	002	08F2	2846	2804* 2805* 2806
BXIFPE	002	08F4	2847	2804
BXIGTC	001	08EB	2836	2774
BXIGTO	002	08ED	2837	
BXIINC	001	096C	2953	2929
BXIINO	001	096D	2954	2890*
BXIINO	UNDEFINED SYMBOL			2917*
BXILTE	001	0001	2864	
BXINPT	001	0800	2676	2875
BXIONE	002	0972	2965	2886 2917 2921
BXIPBA	002	08F8	2854	2798 2812
BXIPSI	001	0004	2861	2845
BXISG2	001	0000	2862	2824
BXISTC	001	08E5	2830	2686
BXISTO	002	08E7	2831	
BXISXC	001	096E	2956	2910
BXISXO	001	096F	2957	2906*
BXITB1	001	1B8E	2866	2718 2718*
BXIVTE	001	0000	2863	2745 2757 2765* 2769* 2901 2906
BXI010	004	0803	2681	
BXI020	003	080B	2686	
BXI030	006	081A	2693	
BXI040	003	0825	2698	
BXI050	006	0834	2705	
BXI060	004	083A	2709	
BXI070	006	083E	2713	
BXI080	006	0849	2718	
BXI090	003	0852	2723	
BXI100	004	0855	2727	2783
BXI110	004	0859	2731	
BXI120	004	085D	2735	
BXI130	003	0861	2736	2723* 2751* 2763*
BXI140	004	0864	2740	
BXI145	003	0868	2741	2719 2719* 2770 2770*
BXI150	003	086B	2745	
BXI160	004	0871	2751	
BXI170	003	087B	2757	2770
BXI180	004	0881	2763	
BXI185	003	0888	2765	2719
BXI190	004	088B	2769	2746 2753 2758
BXI210	003	0892	2774	
BXI220	004	08A1	2781	
BXI230	003	08AB	2787	
BXI240	004	08BA	2798	
BXI250	004	08C1	2804	
BXI260	004	08D1	2812	2799
BXI270	003	08D5	2816	
BXI280	003	08DC	2822	2807
BXI290	006	0900	2880	
BXI300	006	090A	2885	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 220

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BXI310	003	0915	2890	
BXI320	003	0918	2894	
BXI330	004	091B	2898	2925
BXI340	003	091F	2899	2894* 2921*
BXI350	003	0922	2901	
BXI360	004	0928	2906	
BXI370	003	092C	2910	
BXI380	004	093B	2917	
BXI390	004	093F	2921	
BXI400	003	0943	2925	
BXI410	003	0946	2929	2902
BXI420	003	0955	2936	
BXI430	004	0964	2943	
BXI440	004	0968	2947	
BXPAFC	001	1863	7341	7279
BXPAFO	001	1864	7342	
BXPBN1	002	1868	7354	
BXPC02	001	0002	7358	7300
BXPC04	001	0004	7359	7318
BXPI70	UNDEFINED SYMBOL			7296
BXPPTC	001	1865	7344	7322
BXPPTO	001	1866	7345	7300* 7318*
BXPSFA	001	1867	7353	
BXPUTX	001	1800	7267	
BXP010	004	1800	7271	
BXP100	003	180C	7279	
BXP120	004	181B	7286	
BXP140	004	1823	7291	7331
BXP150	004	1827	7295	
BXP160	003	182E	7300	
BXP170	004	1834	7305	
BXP180	004	183B	7310	
BXP190	004	183F	7314	7306
BXP200	003	1843	7318	
BXP210	003	1846	7322	7301
BXP220	004	1855	7329	
BXP230	004	185F	7335	
BXRAFC	001	1CE2	9350	9323
BXRAFO	001	1CE3	9351	
BXRBNI	002	1CE6	9363	
BXRRTC	001	1CE4	9353	9330
BXRSET	001	1CA6	9313	
BXRSFA	001	1CE5	9361	
BXR010	004	1CA6	9317	
BXR020	004	1CAE	9319	9340
BXR110	003	1CB2	9323	
BXR120	003	1CC1	9330	
BXR130	004	1CD0	9337	
BXR140	004	1CDE	9344	
BXUBNC	001	14DF	6107	5940
BXUBNO	002	14E1	6108	
BXUBN1	002	14E8	6120	5936 5957 6059
BXUPRC	001	14E2	6110	6049* 6089
BXUPRO	001	14E3	6111	5979* 5983* 6014* 6040* 6069*
BXUPRT	001	1400	5920	
BXUSCC	001	14E4	6113	6053

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 221

SYMBOL	LEN	VALUE	DEFN	REFERENCES
BXUSCO	002	14E6	6114	6045* 6073*
BXUSTC	001	14DC	6104	5929
BXUSTO	002	14DE	6105	
BXUSUB	002	14EA	6122	6073
BXU010	004	1400	5924	
BXU020	003	1408	5929	
BXU025	006	1412	5935	
BXU030	003	141D	5940	
BXU040	006	1427	5947	
BXU050	004	142D	5952	
BXU060	006	1431	5956	
BXU070	004	143C	5961	
BXU080	006	1440	5965	
BXU090	004	1446	5970	
BXU100	003	144A	5974	
BXU110	003	1450	5979	
BXU120	003	1453	5983	6026
BXU130	003	1456	5987	
BXU140	004	1459	5991	
BXU150	003	145D	5995	6025
BXU170	004	1460	5999	5975
BXU180	004	1464	6003	
BXU190	004	146B	6008	
BXU200	003	146F	6014	6004
BXU210	004	1472	6018	
BXU220	004	1479	6023	6041 6060
BXU230	004	1486	6030	6019
BXU240	004	148E	6035	
BXU250	003	1495	6040	
BXU260	005	149B	6045	6036
BXU270	003	14A0	6049	
BXU280	003	14A3	6053	6077
BXU290	005	14AD	6059	
BXU300	003	14B5	6064	
BXU310	003	14B8	6069	
BXU320	004	14BB	6073	
BXU340	003	14BF	6077	
BXU350	003	14C2	6089	5987 5995 6064
BXU360	003	14C9	6094	5931 5942 6055
BXU370	004	14D8	6098	6094*
B0EQUL	UNDEFINED	SYMBOL		8428
B2EOST	UNDEFINED	SYMBOL		5753
CNICA2	UNDEFINED	SYMBOL		3926
CNITRM	UNDEFINED	SYMBOL		3972
CNTAD2	001	0CF5	4015	3945
CNTBLS	002	0CF2	4011	3952
CNTBL1	002	0CFB	4019	3974
CNTBOP	002	0CEB	4000	3975
CNTBRA	001	0CE9	3999	3875
CNTCA2	002	0CF6	4016	3848* 3867* 3902 3929 3944* 3973
CNTCWR	001	0CEE	4004	3918 3919*
CNTENT	001	0000	4007	3953
CNTFCP	002	0CFD	4020	3935* 3936* 3937
CNTFPE	001	001F	4021	3935
CNTPBA	002	0CF4	4012	3929 3944
CNTPSI	001	0004	4006	4008 4009



## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 222

SYMBOL	LEN	VALUE	DEFN	REFERENCES
CNTSAD	001	0CF7	4017	3901* 3972*
CNTSTR	001	0014	4008	3901 4009
CNTTRM	001	0018	4009	
CNTUSC	UNDEFINED	SYMBOL		3961
CNTWRK	002	0CF9	4018	3902* 3937 3951 3973* 3974*
CWTU##	001	0CEC	4002	
DGET	UNDEFINED	SYMBOL		0519
DOPNBY	UNDEFINED	SYMBOL		9174*
IPMBRO	002	16CA	6642	
ISLCSA	UNDEFINED	SYMBOL		9174
ISNUHC	UNDEFINED	SYMBOL		6516*
ITPHTC	001	1CFA	9476	
RKM120	003	1C6C	9172	
RXR	UNDEFINED	SYMBOL		2911
STRAD2	001	0DF5	4179	4099
STRAOP	002	0DDF	4154	4040* 4049 4130* 4137
STRBOP	002	0DF0	4172	4140*
STRCA2	002	0DF6	4180	4097*
STRCOP	002	0DE2	4157	4049*
STRCWR	001	0DE5	4162	4065
STRFN2	001	0DE8	4165	4068
STRFOP	002	0DF3	4175	4137*
STRPBA	002	0DF9	4182	4083 4097
STRSB1	001	0DEE	4171	4141
STRSC1	001	0DEB	4168	4145
STRSTA	001	0DDD	4153	4047 4131
STRSTC	001	0DE0	4156	4050
STRSTF	001	0DF1	4174	4138 4143
STRSTX	001	0DE3	4159	4057
STRUSF	001	0DF4	4177	4135
STRWOP	002	0DE7	4163	4066*
STRXOP	001	0DE4	4160	
STR1OP	002	0DED	4169	4129* 4140
TRMAOP	002	0EDF	4306	4211*
TRMBIC	001	0ED8	4300	4195
TRMBNI	UNDEFINED	SYMBOL		4202
TRMBN1	002	0EDC	4303	4271
TRMBOP	002	0EE4	4312	
TRMBRC	001	0EE2	4311	4265
TRMFN1	001	0EE5	4314	4251
TRMSTA	001	0EDD	4305	4210
TRMSTX	001	0EE0	4308	4246
TRMUSC	001	0EE8	4317	4253
TRNBOP	UNDEFINED	SYMBOL		4264*
TWOAD2	001	108D	4704	4661
TWOCA2	002	108E	4705	
V\$APWR	001	0800	2185	2330
V\$BFR1	001	5400	2248	2438
V\$BFR2	001	5500	2249	2439
V\$CBNZ	001	0CB2	2257	2337
V\$CCON	001	5120	2264	2435 4166
V\$CDCV	001	3100	2261	2390
V\$CDSY	001	2E00	2260	2387
V\$CFPZ	001	0C70	2255	2336
V\$CNXZ	001	0470	2258	2325
V\$CSSR	001	5100	2263	2434 4315 4595

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 223

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$CZFP	001	04AD	2256	2326
V\$DTLN	001	4600	2270	2422
V\$DTVR	001	4700	2271	2423
V\$FABS	001	1761	2156	2354
V\$FACS	001	1400	2172	2346
V\$FASN	001	1413	2171	2347
V\$FATN	001	1100	2170	2343
V\$FCOS	001	0A00	2167	2332
V\$FCOT	001	0D00	2165	2338
V\$FCSC	001	1725	2169	2353
V\$FDEG	001	17DA	2176	2358
V\$FDET	001	4540	2179	2421
V\$FEXP	001	0500	2163	2327
V\$FHCS	001	1500	2175	2348
V\$FHSN	001	1557	2174	2349
V\$FHTN	001	1593	2173	2350
V\$FINT	001	176C	2157	2355
V\$FLGT	001	0200	2161	2320
V\$FLOG	001	0219	2160	2322
V\$FLTW	001	020B	2162	2321
V\$FRAD	001	17CB	2177	2357
V\$FRND	001	1800	2178	2359
V\$FSEC	001	1700	2168	2352
V\$FSGN	001	17A7	2158	2356
V\$FSIN	001	0A1A	2166	2333
V\$FSQR	001	0900	2159	2331
V\$FTAN	001	0D28	2164	2339
V\$IFCI	001	1B00	2148	2363
V\$IFIO	001	1A00	2150	2362
V\$ISDN	001	1900	2149	2360
V\$KBTL	001	1EAC	2292	
V\$KBTS	001	0DAC	2291	
V\$LPRB	001	4F00	2246	2432
V\$LPRT	001	4D00	2244	2430
V\$LPR2	001	4E00	2245	2431
V\$MADD	001	4007	2193	2410 3568
V\$MASN	001	43A0	2191	2417 3554
V\$MCON	001	4324	2198	2415 3592
V\$MIDN	001	4300	2199	2414 3596
V\$MINV	001	4500	2203	2420 3580
V\$MMPY	001	4100	2195	2411 3576
V\$MSMY	001	4264	2196	2413 3551
V\$MSUB	001	4000	2194	2409 3572
V\$MTRN	001	4400	2202	2419 3584
V\$MZER	001	432B	2200	2416 3588
V\$PCH1	001	5200	2284	2436
V\$PCH2	001	5300	2285	2437
V\$SCDI	001	2A00	2241	2381
V\$SCDO	001	2A96	2242	2382
V\$SFA2	001	5000	2226	2433
V\$SFD1	001	0000	2236	2318
V\$SFD2	001	0100	2237	2319
V\$SKEY	001	2500	2240	2376
V\$SPRT	001	2800	2239	2379
V\$VMPL	001	4C06	2278	2429
V\$VMPS	001	4C00	2277	2428

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 224

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$XKAF	001	1C00	2225	2364
V\$XKCA	001	2400	2229	2372
V\$XKCL	001	240A	2228	2373
V\$XKIN	001	2B00	2224	2383
V\$XKLP	001	24AD	2230	
V\$XKRS	001	240D	2227	2374
V\$XMGT	001	3E06	2218	2404 8034
V\$XMIN	001	3D00	2217	2402 6780
V\$XMPL	001	3F06	2221	2407 8591
V\$XMPS	001	3F00	2220	2406 8588
V\$XMPT	001	3E0C	2219	2405 8991
V\$XMPU	001	3F13	2222	2408 9666
V\$XMRD	001	3E00	2216	2403 7162
V\$XSGT	001	2100	2211	2369
V\$XSGY	UNDEFINED	SYMBOL		7664
V\$XSIN	001	2B6E	2210	2384 2837
V\$XSPR	001	3400	2213	2393
V\$XSPT	001	1D00	2212	2365
V\$XSPU	001	3800	2214	2397
V\$XSRD	001	3300	2209	2392 3732
V\$00E1	001	0000	2318	
V\$01E1	001	0100	2319	
V\$02E1	001	0200	2320	
V\$02E2	001	020B	2321	
V\$02F3	001	0219	2322	
V\$03CC	001	0300	2323	
V\$04CC	001	0400	2324	
V\$04E1	001	0470	2325	
V\$04E2	001	04AD	2326	
V\$05E1	001	0500	2327	
V\$06CC	001	0600	2328	
V\$07CC	001	0700	2329	
V\$08E1	001	0800	2330	
V\$09E1	001	0900	2331	
V\$10E1	001	0A00	2332	
V\$10E2	001	0A1A	2333	
V\$11CC	001	0B00	2334	
V\$12CC	001	0C00	2335	
V\$12E1	001	0C70	2336	
V\$12E2	001	0CB2	2337	
V\$13E1	001	0D00	2338	
V\$13E2	001	0D28	2339	
V\$14CC	001	0E00	2340	
V\$15CC	001	0F00	2341	
V\$16CC	001	1000	2342	
V\$17E1	001	1100	2343	
V\$18CC	001	1200	2344	
V\$19CC	001	1300	2345	
V\$20E1	001	1400	2346	
V\$20E2	001	1413	2347	
V\$21E1	001	1500	2348	
V\$21E2	001	1557	2349	
V\$21E3	001	1593	2350	
V\$22CC	001	1600	2351	
V\$23E1	001	1700	2352	
V\$23E2	001	1725	2353	

## CROSS REFERENCE

SYMBOL   LEN   VALUE   DEFN   REFERENCES   VER 15, MOD 00   20/07/20   PAGE 225

V\$23E3	001	1761	2354	
V\$23E4	001	176C	2355	
V\$23E5	001	17A7	2356	
V\$23E6	001	17CB	2357	
V\$23E7	001	17DA	2358	
V\$24E1	001	1800	2359	
V\$25E1	001	1900	2360	
V\$26E1	001	1A00	2362	
V\$27E1	001	1B00	2363	
V\$28E1	001	1C00	2364	
V\$29E1	001	1D00	2365	
V\$30CC	001	1E00	2366	
V\$31CC	001	1F00	2367	
V\$32CC	001	2000	2368	
V\$33E1	001	2100	2369	
V\$34CC	001	2200	2370	
V\$35CC	001	2300	2371	
V\$36CC	001	2400	2375	
V\$36E1	001	2400	2372	
V\$36E2	001	240A	2373	
V\$36E3	001	240D	2374	
V\$37E1	001	2500	2376	
V\$38CC	001	2600	2377	
V\$39CC	001	2700	2378	
V\$40E1	001	2800	2379	
V\$41CC	001	2900	2380	
V\$42E1	001	2A00	2381	
V\$42E2	001	2A96	2382	
V\$43E1	001	2B00	2383	
V\$43E2	001	2B6E	2384	
V\$44CC	001	2C00	2385	
V\$45CC	001	2D00	2386	
V\$46E1	001	2E00	2387	
V\$47CC	001	2F00	2388	
V\$48CC	001	3000	2389	
V\$49E1	001	3100	2390	
V\$50CC	001	3200	2391	
V\$51E1	001	3300	2392	
V\$52E1	001	3400	2393	
V\$53CC	001	3500	2394	
V\$54CC	001	3600	2395	
V\$55CC	001	3700	2396	
V\$56E1	001	3800	2397	
V\$57CC	001	3900	2398	
V\$58CC	001	3A00	2399	
V\$59CC	001	3B00	2400	
V\$60CC	001	3C00	2401	
V\$61E1	001	3D00	2402	
V\$62E1	001	3E00	2403	
V\$62E2	001	3E06	2404	
V\$62E3	001	3E0C	2405	
V\$63E1	001	3F00	2406	
V\$63E2	001	3F06	2407	
V\$63E3	001	3F13	2408	
V\$64E1	001	4000	2409	
V\$64E2	001	4007	2410	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 226

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V\$65E1	001	4100	2411	
V\$66CC	001	4200	2412	
V\$66E1	001	4264	2413	
V\$67E1	001	4300	2414	
V\$67E2	001	4324	2415	
V\$67E3	001	432B	2416	
V\$67E4	001	43A0	2417	
V\$68E1	001	4400	2419	
V\$69E1	001	4500	2420	
V\$69E2	001	4540	2421	
V\$70E1	001	4600	2422	
V\$71E1	001	4700	2423	
V\$72CC	001	4800	2424	
V\$73CC	001	4900	2425	
V\$74CC	001	4A00	2426	
V\$75CC	001	4B00	2427	
V\$76E1	001	4C00	2428	
V\$76E2	001	4C06	2429	
V\$77CC	001	4D00	2430	
V\$78CC	001	4E00	2431	
V\$79CC	001	4F00	2432	
V\$80E1	001	5000	2433	
V\$81E2	001	5100	2434	
V\$81E3	001	5120	2435	
V\$82E1	001	5200	2436	
V\$83E2	001	5300	2437	
V\$84E1	001	5400	2438	
V\$85E2	001	5500	2439	
V@CDPT	001	0007	2450	
V@CHGH	001	0008	2555	
V@CMIC	001	0002	2451	
V@CMNI	001	00FF	2448	
V@CMUL	001	0007	2556	
V@CNIX	001	0080	2449	
V@COEX	001	001E	2446	
V@CPLS	001	00F0	2453	
V@CPRC	001	000A	2455	
V@CSQR	001	0003	2553	
V@CSTR	001	0002	2554	
V@CTTA	001	0027	2456	
V@DCAD	001	0002	2476	2477
V@DEXP	001	0000	2481	
V@DMAN	001	000D	2483	2484
V@DMN1	001	0001	2482	
V@DPDF	001	0002	2471	
V@DSAD	001	0001	2472	
V@DSGN	001	000D	2484	
V@DVAD	001	0004	2477	
V@EART	001	0001	2454	
V@ECRT	001	0038	2527	
V@EFUL	001	00F4	2526	
V@EINV	001	00F7	2522	
V@EIPR	001	00F1	2523	
V@ENSV	001	00F3	2524	
V@ENUL	001	0000	2521	
V@ERPC	001	0020	2452	

## CROSS REFERENCE

VER 15, MOD 00 20/07/20 PAGE 227

SYMBOL	LEN	VALUE	DEFN	REFERENCES
V@ESAV	001	00F2	2525	
V@FEHN	001	0002	2551	
V@FEPL	001	0091	2547	
V@FERS	001	0003	2550	
V@FPGS	001	0081	2546	
V@FRET	001	0015	2549	
V@FSPC	001	0040	2548	
V@FTAB	001	0000	2552	
V@KADD	001	004E	2537	
V@KCLE	001	006E	2534	
V@KDIV	001	0061	2540	
V@KEMN	001	006C	2532	
V@KEPL	001	006B	2531	
V@KMUL	001	005C	2539	
V@KPER	001	004B	2542	
V@KPST	001	007B	2536	
V@KPWR	001	005A	2541	
V@KSQR	001	006F	2533	
V@KSTO	001	006D	2535	
V@KSUB	001	0060	2538	
V@LAIP	001	0003	2502	2503
V@LDEX	001	0002	2505	
V@LETE	001	0003	2509	
V@LEXP	001	0001	2499	2501
V@LFKO	001	0006	2504	
V@LINI	001	0200	2508	
V@LLKS	001	0010	2501	
V@LMAN	001	000F	2500	2501
V@LNOP	001	0015	2506	
V@LTBE	001	0007	2503	
V@LVPG	001	0100	2507	2508
V@MCHS	001	00C0	2488	
V@MCRD	001	0010	2464	
V@MDEF	001	0008	2465	
V@MEXC	001	0080	2462	
V@MEXT	001	0004	2491	
V@MICC	001	0010	2447	
V@MIPC	001	0080	2489	
V@MIPL	001	0020	2495	
V@MLST	001	0040	2463	
V@MPND	001	0000	2494	
V@MPOF	001	0080	2492	
V@MPRC	001	0020	2461	
V@MSFU	001	0002	2466	
V@MSTN	001	0004	2460	
V@OALL	001	00F4	2517	
V@ONUL	001	00F0	2513	2514
V@OPM1	001	00F2	2515	2516
V@ORTN	001	00F1	2514	2515
V@OSTK	001	00F3	2516	2517
V@PEOF	001	0002	2490	
V@PSQ2	001	0014	2493	

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 114

OL105 I THE CODE LENGTH OF #BOVLY IS 8183 DECIMAL.  
THE CODE LENGTH OF #BOVLY IS 8183 DECIMAL.

0020 2495  
V@MLST 001 0040 2463  
V@MPND 001 0000 2494  
V@MPOF 001 0080 2492  
V@MPRC 001 0020 2461  
V@MSFU 001 0002 2466  
V@MSTN 001 0004 2460  
V@OALL 001 00F4 2517  
V@ONUL 001 00F0 2513 2514  
V@OPM1 001 00F2 2515 2516  
V@ORTN 001 00F1 2514 2515  
V@OSTK 001 00F3 2516 2517  
V@PEOF 001 0002 2490  
V@PSQ2 001 0014 2493

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 126

OL105 I THE CODE LENGTH OF #BOVLY IS 7935 DECIMAL.  
GTH OF #BOVLY IS 7935 DECIMAL.  
020 2461

V@MSFU 001 0002 2466  
V@MSTN 001 0004 2460  
V@OALL 001 00F4 2517  
V@ONUL 001 00F0 2513 2514  
V@OPM1 001 00F2 2515 2516  
V@ORTN 001 00F1 2514 2515  
V@OSTK 001 00F3 2516 2517  
V@PEOF 001 0002 2490  
V@PSQ2 001 0014 2493

TOTAL STATEMENTS IN ERROR IN THIS ASSEMBLY = 141

OL105 I THE CODE LENGTH OF #BOVLY IS 7935 DECIMAL.