

ASSIGN M:CI,(FILE,EDIT,:DUOC1)

METASYM CI,CN,L0

•SS R0,R1,R2,R3,R4,R5,R6,R7,R8,R9,R10,R11,R12,R13,R14,R15

•SS SR1,SR2,SR3,SR4,D1,D2,D3,D4,*

•END

* CONCORDANCE EXTENDED MEMORY MODE.

* REFERENCE COUNT = 6375. DISK OVERFLOW = 3315.

ABN	5492/M:SETDCB 6112/M:WRITE	5496/M:READ	5520/M:PRECOR	6073/M:SETDCB	6078/M:WRITE	6106/M:SETDCB	
ADDCDTPARAM	908/BAL	925/BAL	929/BAL	957/BAL	967/BAL	978/BAL	1070/BAL
	1086/BAL	1105/BAL	1125/BAL	1142/BAL	1156/BAL	1166/BAL	1180/BAL
	1198/BAL	1213/BAL	1218/BAL	1226/BAL	1241/BAL	1262/BAL	1305/BAL
	1313/BAL	1387/BAL	1373/BAL	1380/BAL	1410/BAL	1418/BAL	1429/BAL
	1436/BAL	1483/BAL	1498/BAL	1519/BAL	1531/BAL	1571/BAL	1577/BAL
	1642/BAL	3767*EQU					
ADJINT	1100/BAL	1118/BAL	1210/BAL	1238/BAL	1300/BAL	1346/BAL	1401/BAL
	1474/BAL	1490/BAL	1514/BAL	1527/BAL	1563/BAL	1637/BAL	1652-LW
ADJUSTALLFLAG	3506/BAL	3558/BAL	3593/BAL	3614/BAL	3668/BAL	4305*EQU	
AF	69/SET	124/CBM	124/CBM	124/CBM	124/CBM	131/D8	132/D8
	133/PSW	135/LCI	135/LCI	136/PSM	146/D8	147/SET	148/D8
	149/PLW	151/LCI	151/LCI	152/PLM	162/D8	163/D8	166/SET
	166/SET	197/SET	199/GEN	201/ERRBR	203/GEN	203/GEN	205/GEN
	205/GEN						
AFA	199/GEN	202/D8					
ALLFLAG	304=DATA 4308/STW	649/STW 4389/LW	699/STW 4444/STW	1764/STW 4458/STW	1872/MTW	3488/MTW	4306/MTW
ALLOK	305=DATA	1830/STW	3516/STW	3568/STW	3624/STW	4422/MTW	
ALPH	184=EGU 1158/NXTPRM	742/NXTPRM 1165/LI	910/NXTPRM 1202/NXTPRM	917/NXTPRM 1216/NXTPRM	931/NXTPRM 1217/LI	1068/NXTPRM 4095/LI	1069/LI
ANLZRIGHT	4323=EGU	4675/BAL	4752/BAL				

AR10	4326/BL	4335=LI					
AR10A	4342=CB	4347/BLE					
AR12	4337/BG	4351=LI					
AR15	4343/BE	4357=AI	4360/BF				
AR18	4353/B	4364=PULL	4375/B				
AR20	4339/BNEZ	4370=LW					
BA	1687/LI	1711/LI	1722/LI	2859/LI	3088/LI	3095/LI	3117/LI
	3441/LI	4019/EQU	4019/EQU	4046/EQU	4046/EQU	4256/EQU	4256/EQU
	5019/LI	5033/LI	5133/LI	5527/LI	6027/LI	6043/LI	
BADI6	5084=RES	5476/BNE	5514/BNE	6088/BNE	6122/BNE		
BADI61	5077/BNE	5086=RES	5240/BNE	5279/BNE	5387/BNE		
BDISPTBL	545=EQU	1702/CW					
BD10	4907=LI	4912/BDR					
BD20	4921/BNE	4925=EQU					
BD30	4919=EQU	4924/BDR					
BEGINEDITOR	74/DEF	250/DATA	636=EQU	661/LI	6145/END		
BGD10	638/B	653=LI	678=EQU				
BINTODEC	2860/BAL	3097/BAL	3442/BAL	4902=EQU	5020/BAL	5034/BAL	5134/BAL
	6028/BAL						
BL	229=EQU	1724/GEN4	2071/GEN4	2668/GEN4	2935/GEN4	3119/GEN4	5882/GEN4

BLANK						
BLANKBUF	190-EQU					
BLANKCNT	4934-PUSH	5457/BAL	5490/BAL			
	306-RES	4744/STW	4754/AWM	4759/CW	4770/SW	4775/STW
	4795/AW	4815/LW	4828/XW	4842/AW	4851/AWM	4870/AWM
BLD08	2046/BEZ	2052-MTW	2065-EQU			
BLD10	2054-CI	2066-EQU	2102/BLE			
BLD12	2053/BEZ	2055/BEZ	2070-BAL			
BLD25	2092/BLE	2098-BAL				
BLD30	2077/BE	2106/BEZ	2111-EQU	2123/B		
BLD40	2030/BCR	2120-BAL				
BLD5	2037/BEZ	2044-LI				
BPFLAG	307-DATA	1815/STW	3470/STW	3728/LW	3730/STW	4338/MTW
BPV0FF	2001/CW	2015-TEXTC	2361/CW	2378/CW		
BPV0N	1993/CW	2014-TEXTC	2354/CW	2372/CW		
BPV10	2002/BNE	2009-BAL	2363/B	2380/B		
BPV5	1994/BNE	2001-CW				
BR\$FPT	515-GEN	1750/CAL1				
BRK\$KEY	682/M:INT	1674-PUSH				
BRK30	1683/BEZ	1697-LI				

BRK40	1703/BE	1707=CI					
BRK50	1708/BLE	1720=LW					
BRK53	1732=LB	1736/BDR					
BRK55	1737=STB	1746/B					
BRK60	1722/BGEZ	1742=LW					
BRK80	1681/BNEZ 1740/B	1686/BLZ 1748=BAL	1694/B	1705/B	1710/BLZ	1718/B	1721/BLZ
BRK90	1758=LI	1789/LI					
BRK91	1766/BEZ	1775=EGU					
BRK99	1679/BLZ	1699/BEZ	1792=EGU				
BUF	607/M:DCB	621/M:DCB					
BUILDFLAG	450/MTW	500=DATA	2052/MTW	2113/MTW	5586/STW	5590/LW	
CARDIMG	308=RES 2088/STB 2944/LB 3363/STB 4525/CB 4706/STB 5617/STB 5728/STB 5804/LB	519/DATA 2678/LB 2952/STB 3364/STB 4557/STB 4801/LB 5621/STW 5739/LB 5805/STB	565/DATA 2680/STB 3330/LB 3376/CB 4640/CW 4802/STB 5626/DATA 5740/STB 5818/STB	581/DATA 2687/LB 3331/STB 3531/STB 4646/CB 4817/STB 5662/CB 5741/STB 5860/LB	607/M:DCB 2692/STB 3344/STB 4342/CB 4650/CB 4938/STW 5665/CB 5782/CB	621/M:DCB 2706/STB 3360/CB 4359/CB 4697/LB 5408/CB 5670/STB 5790/CB	2080/LB 2722/STB 3362/LB 4503/CB 4698/STB 5416/STB 5698/CB 5799/STB
CBRCHTBL	757/EXU	814=EGU					
CDT	309=RES	705/STW	711/STW	712/LI	738/MTW	763/AWM	1812/LI

CDTADR	3798/LW	4281/BR	5907/MTW				
	310-RES	713/STW	734/LB*	735/AWM	948/STB*	993/LB*	997/LB*
	1622/LW*	1624/STW*	1698/LB*	1813/STW	1832/LB*	1877/LB*	1878/AWM
	1885/LB*	1898/LW	1991/LB*	1992/LW*	2025/LB*	2026/AW	2036/LB*
	2039/LB*	2040/LW*	2045/LB*	2048/LB*	2049/LW*	2135/LB*	2136/AW
	2138/LB*	2139/AW	2159/LB*	2160/LW*	2167/LB*	2168/AW	2175/LB*
	2176/AW	2187/LB*	2208/LB*	2209/LW*	2212/LB*	2215/LB*	2216/LW*
	2251/LB*	2252/AW	2256/LB*	2257/AW	2269/LB*	2270/AW	2282/LB*
	2283/AW	2296/LB*	2297/AW	2300/LB*	2301/AW	2335/LB*	2336/AW
	2352/LB*	2353/LW*	2370/LB*	2371/LW*	2390/LB*	2391/AW	2420/LB*
	2421/AW	2472/LB*	2473/AW	2485/LB*	2490/LB*	2491/AW	2509/LB*
	2510/AW	2517/LB*	2518/AW	2533/LB*	2536/LB*	2537/AW	2580/LB*
	2581/AW	2619/LB*	2620/AW	2648/LB*	2649/LW*	2651/LB*	2652/LW*
	2767/LB*	2768/LW*	2770/LW*	2795/LB*	2796/LW*	2800/LW*	2803/LB*
	2804/AW	2911/LB*	2912/LW*	2917/LB*	2920/LB*	2921/LW*	3003/LB*
	3004/LW*	3006/LW*	3008/LB*	3009/LW*	3011/LW*	3014/LB*	3017/LB*
	3018/LW*	3143/LB*	3144/LW*	3146/LB*	3147/LW*	3183/LB*	3184/LW*
	3260/LB*	3261/LW*	3263/LW*	3350/LB*	3391/LB*	3392/LW*	3394/LW*
	3398/LW	3399/LB*	3467/STW	3520/LB*	3521/AW	3551/LB*	3552/AW
	3572/LB*	3573/LW*	3588/LB*	3589/AW	3606/LB*	3607/AW	3628/LB*
	3629/LW*	3643/LB*	3644/AW	3683/LB*	3684/LW*	3770/STB*	3772/LB*
	3773/STB*	3776/STB*	3778/AW	3785/LW*	4284/STW*	4288/STB*	4291/STW*
	4294/STW*	4393/LB*	4398/LB*	4402/LB*	4403/LW*	4418/LB*	4419/LW*
	4432/LB*	4433/AW	4592/LB*	4595/LB*	4596/LW*	4603/LB*	4606/LB*
	4607/LW*	5899/LB*	5958/LB*				
CFLAG							
	501-DATA	516/DATA	1753/LB	5571/MTW	5573/MTW		
CHARPSN							
	311-RES	706/STW	1621/MTW	3821/LW	3828/STW	3852/LW	3868/LW
	3884/STW	3916/LW	4002/STW	4060/LW			
CHECK1CDTENTRY							
	1066/BAL	1083/BAL	1153/BAL	1195/BAL	1259/BAL	1277/BAL	1295/BAL
	1354/BAL	1396/BAL	1456/BAL	1469/BAL	1509/BAL	1569/BAL	1639/BAL
	3797-EQU						
CHG:STG:CNT							
	353-DATA	708/STW	3435/LW	3447/STW	4436/MTW	4440/MTW	4534/MTW

CLOSE	1770/BAL 2321/BAL 2589/BAL	1785/BAL 2330/BAL 2594/BAL	2056/BAL 2415/BAL 2598/BAL	2107/BAL 2443/BAL 2600/BAL	2122/BAL 2456/BAL 2605/BAL	2236/BAL 2506/BAL 4949*EQU	2242/BAL 2530/BAL 5834/BAL
CLOSE2	1780/BAL 4957*EQU	2243/BAL	2313/BAL	2320/BAL	2333/BAL	2590/BAL	2595/BAL
CLOSE3	1773/BAL	2179/BAL	2237/BAL	2318/BAL	2339/BAL	4962*M;CLOSE	
CM	230*EQU	4016/DATA	4254/DATA				
CMNDTBL	1871/EXU	1935*EQU					
CMT10	2667*BAL	2734/B					
CMT15	2705-LB	2712/BL					
CMT20	2710/B	2720*CI	2724/B				
CMT30	2721/BGE	2728*BAL	2758/B				
CMT40	2654/BLZ	2656/BGE	2738*BAL				
CMT50	2658/BCS	2744*BAL					
CMT60	2732/BE	2750*BAL					
CMT70	2714/BNE	2756*BAL					
CM10A	2670/BEZ	2695*EQU					
CM10B	2687-LB	2690/BDR					
CNAMETBL	747/CW	771*EQU	810/EQU	1700/LW			
CNMRTBL	756/LB	856*EQU					

C0M	185-EQU 1229/NXTPRM 1431/NXTPRM 3856/NXTNAM	1088/NXTPRM 1307/NXTPRM 1485/NXTPRM 3871/NXTNAM	1097/NXTPRM 1367/NXTPRM 1521/NXTPRM	1111/NXTPRM 1375/NXTPRM 3826/NXTNAM	1182/NXTPRM 1412/NXTPRM 3839/NXTNAM	1201/NXTPRM 1423/NXTPRM 3854/NXTNAM	
COPYFL	312-DATA	1765/MTW	1875/STW	2133/STW	2234/MTW	2268/MTW	2331/MTW
CPY1	2145-AI	2162/B					
CPY1A	2149/BEZ	2156-CB					
CPY1B	2151/BNE	2158-LI					
CPY10	2188/BNEZ	2207-LI					
CPY15	2213/BEZ	2220-BAL	2231/BLE				
CPY2	2166-LI						
CPY20	2194/BE	2222/BE	2235/BEZ	2242-BAL			
CPY3	2178/BCS	2184-BAL	2261/BCS	2263/B	2305/BCS	2307/B	
CPY30	2162/BNE	2250-LI					
CPY32	2157/BE	2267-LI					
CPY35	2254/BCR	2311-BAL					
CPY36	2260/BCS	2304/BCS	2318-BAL				
CPY37	2320-BAL						
CPY40	2171/BCS	2319/B	2322-BAL	2476/BCS			
CPY5	2192-BAL	2203/B					

CPY5A							
	2197=LW						
CPY50	2196/BNE	2224/BCS	2328=BAL				
CPY56	2332/BNEZ	2339=BAL					
CPY58	2338/B	2340=B					
CPY60	2280/BNEZ	2293/BNEZ	2341=BAL				
CR	231=EGU	417/GEN4	4015/DATA	4250/DATA	5567/CI	5933/LI	5986/LI
CRFLAG	313=DATA	2358/STW	5403/LW				
CR3	2358=STW	2362/BE					
CR5	2355/BNE	2361=CW					
CT\$FLAG	355=DATA	709/STW	1290/MTW	1876/STW	2660/MTW	2669/MTW	
CTBLSZ	744/LI	810=EGU					
DELETE	2523/BAL	2773/BAL	3042/BAL	4975=EGU			
DELETEFILE	2337/BAL	2392/BAL	5056=EGU				
DELETERECORD	2824/BAL	3062/BAL	3082/BAL	3153/BAL	5012/BAL	5116=EGU	
DELNXT	347=DATA	4994/STW	4995/LW	4999/STB	5000/M:DELREC	5003/LW	
DF\$ABN	5058/LI	5074=RES					
DFLTINCR	314=DATA	2539/STW	2913/LW	2922/STW	3012/LW	3019/STW	
DFLTSEQ	215=EGU	2033/LI					
DIGITS							

	293-DATA	3932/CLM	4069/CLM	4161/CLM	4180/CLM	4213/CLM	
DLT10	2393/BCS	2403-BAL					
DLT5	2397-BAL						
DL10	4990-CW	5007/B					
DL15	4993/BLE	5011-BL					
DL17	5018/BE	5023-EQU					
DL20	5011/BL	5030-LW	5046/B				
DL25	5032/BLE	5037-EQU					
DL30	4991/BE	5044-BAL					
DMY\$TPM	458-BAL	5939/STW	5940/BAL				
DMY\$TYPECERR	440-EQU	3960/STW	3961/B				
DMY\$TYPEPEERR	446-EQU	3955/STW	3956/B				
DMYSTKDW	275-DATA	702/LD					
DO	116-EQU	2674/LW	2693/STW	2705/LB	2706/STB	3023/LW	3025/CLM
	3027/CLM	3029/LW	3031/CLM	3033/CLM	3340/STW	3367/CW	3374/CW
	3781/LW	3782/STW	3798/LW	3799/CI	4907/LI	4908/DW	4909/AI
	4910/STB	5138/LI	5139/CB	5148/CB	5155/LB	5156/STB	5161/LI
	5162/STB	5169/LB	5170/STB	5179/LB	5181/STB	5899/LB	5913/LW
	5914/AI	5915/SLS	5925/BR	5926/STB	5958/LB	5959/AI	5961/STB
	6029/LW	6031/SLD	6035/STW	6041/LI	6044/CB	6048/CB	
EDIT\$DCBT	475/DATA	487-EQU					
EDIT\$TCB	248/DATA	470-EQU					

EDIT\$TSTK							
EDITBASE	471/DATA	481-EQU					
EDT10	62-CSECT	73/DEF	247/ORG	658/LI			
EDT15	2423/BCS	2435-BAL					
EDT20	2437-LI	2446/B					
EDT5	2424/BCS	2443-BAL					
END	2414/BLZ	2419-LI					
	178-EQU	743/NXTPRM	970/NXTPRM	981/NXTPRM	1011/NXTPRM	1073/NXTPRM	
	1090/NXTPRM	1113/NXTPRM	1128/NXTPRM	1184/NXTPRM	1230/NXTPRM	1265/NXTPRM	
	1280/NXTPRM	1309/NXTPRM	1316/NXTPRM	1362/NXTPRM	1369/NXTPRM	1377/NXTPRM	
	1383/NXTPRM	1414/NXTPRM	1425/NXTPRM	1433/NXTPRM	1439/NXTPRM	1523/NXTPRM	
	1534/NXTPRM	1580/NXTPRM	1594/NXTPRM	1620/NXTPRM	1634/NXTPRM	3827/NXTNAM	
	3857/NXTNAM	3872/NXTNAM					
ENEDITOR							
E8DCLMN	669/LI	6144-EQU					
E8F	315-RES	4336/CW	4346/CW	4370/LW	4374/SW	4658/STW	5660/STW
	5672/MTW	5851/LW					
E8M	232-EQU	2193/CW	2221/CW	2481/LW	2501/CW	2551/CW	2583/LW
	2731/CW	2811/CW	3038/CW	3053/CW	3189/CW	3215/CW	3284/CW
	3459/CW	4990/CW	5515/LW				
ERR	233-EQU	417/GEN4	417/GEN4	2071/GEN4	2668/GEN4	2935/GEN4	3232/GEN4
	5529/GEN4	5533/CI	5882/GEN4	5982/CI	6000/LI		
ERRC1	5460/M:SETDCB	5462/M:READ	5521/M:READ				
ERRC10	365-TEXTC	4579/DATA	4826/DATA	4878/DATA			
	374-TEXTC	4475/DATA					

ERRC11							
	375-TEXTC	4623/DATA					
ERRC2							
	366-TEXTC	4687/DATA					
ERRC3							
	367-TEXTC	3699/DATA					
ERRC4							
	368-TEXTC	765/DATA	1909/DATA	3693/DATA	3719/DATA	3802/DATA	
ERRC5							
	369-TEXTC						
ERRC6							
	370-TEXTC	4469/DATA					
ERRC7							
	371-TEXTC	4425/DATA					
ERRC8							
	372-TEXTC	751/DATA	941/DATA				
ERRC9							
	373-TEXTC	739/NXTPRM	909/NXTPRM	916/NXTPRM	930/NXTPRM	1157/NXTPRM	1174/DATA
	1215/NXTPRM	1228/NXTPRM	1278/NXTPRM	1592/NXTPRM	1615/NXTPRM	1632/NXTPRM	
ERRM1							
	378-TEXTC	2751/DATA	2876/DATA	3111/DATA	3238/DATA	3315/DATA	3476/DATA
	5045/DATA	5527/LI	5532/LB	5539/STB			
ERRM12							
	384-TEXTC	2445/DATA					
ERRM13							
	385-TEXTC	1915/DATA					
ERRM14							
	386-TEXTC	2404/DATA	2436/DATA				
ERRM15							
	387-TEXTC	2121/DATA					
ERRM16							
	389-TEXTC	3133/DATA					
ERRM17							
	390-TEXTC	2602/DATA					
ERRM18							
	391-TEXTC	2607/DATA					
ERRM19							

	392-TEXTC	2342/DATA					
ERRM20	393-TEXTC	2104/DATA	2233/DATA	2570/DATA	2970/DATA		
ERRM21	394-TEXTC	3496/DATA					
ERRM3	379-TEXTC	2094/DATA	2757/DATA	2958/DATA			
ERRM4	380-TEXTC	3105/DATA					
ERRM5	381-TEXTC	2010/DATA					
ERRM6	382-TEXTC	2870/DATA	3321/DATA	3409/DATA			
ERRM8	383-TEXTC	1849/DATA					
ERRORCNT	316-RES	715/STW	1843/CW	1845/STW	3465/STW	4426/MTW	5894/MTW
ERRP1	396-TEXTC	2745/DATA	3159/DATA	3953/CI			
ERRP10	405-TEXTC	1134/DATA	4188/DATA				
ERRP11	406-TEXTC	4232/DATA					
ERRP12	407-TEXTC	2323/DATA					
ERRP13	408-TEXTC	2312/DATA					
ERRP14	409-TEXTC	410/EGU	2739/DATA				
ERRP14A	410-EQU						
ERRP15	411-TEXTC	4149/DATA					
ERRP16	412-TEXTC	2329/DATA					
ERRP17	413-TEXTC	1246/DATA	1321/DATA	1444/DATA	1539/DATA		

ERRP18	414=TEXTC	4143/DATA				
ERRP2	397=TEXTC	3165/DATA				
ERRP3	398=TEXTC 3869/NXTNAM	3813/NXTNAM	3818/LI	3837/NXTNAM	3853/NXTNAM	3861/NXTNAM
ERRP4	399=TEXTC 1087/NXTPRM* 1306/NXTPRM* 1411/NXTPRM* 1532/NXTPRM*	968/NXTPRM* 1110/NXTPRM* 1314/NXTPRM* 1422/NXTPRM* 1578/NXTPRM*	979/NXTPRM* 1126/NXTPRM* 1360/NXTPRM* 1430/NXTPRM* 3822/NXTNAM	1009/NXTPRM* 1181/NXTPRM* 1366/NXTPRM* 1437/NXTPRM*	1067/NXTPRM 1200/NXTPRM* 1374/NXTPRM* 1484/NXTPRM*	1071/NXTPRM* 1263/NXTPRM* 1381/NXTPRM* 1520/NXTPRM*
ERRP5	400=TEXTC 1397/NXTPRM 1606/NXTPRM	1095/NXTPRM 1470/NXTPRM	1205/NXTPRM 1486/NXTPRM	1233/NXTPRM 1510/NXTPRM	1296/NXTPRM 1524/NXTPRM	1339/NXTPRM 1559/NXTPRM
ERRP6	401=TEXTC	1115/NXTPRM				
ERRP7	402=TEXTC	1310/NXTPRM	1370/NXTPRM	1378/NXTPRM	1426/NXTPRM	1434/NXTPRM
ERRP8	403=TEXTC	964/NXTPRM	1002/DATA	1415/NXTPRM		
ERRP9	404=TEXTC	975/NXTPRM				
EXC10	1856/B	1871=EXU	1886/B	1893/BNEZ	1895/BLZ	1904/B
EXC15	1873/BGEZ	1884=LI				
EXC20	1847/BNEZ	1892=MTW				
EXC30	1855/BNEZ	1861/BNEZ	1908=BAI			
EXC40	1839/BLZ	1914=BAI				
EXC5	1837/BL	1841/BLE	1860=MTW			

EXC50	1833/BEZ	1920=MTW					
EXC55	1923/BEZ	1930=EGU					
EXC6	1835/BL	1854=MTW					
F:BLANK&PRESERV	1936/BAL	1989=EGU					
F:BUILD	1939/BAL	2022=EGU					
F:COPY	1940/BAL	2130=EGU					
F:CR	1938/BAL	2350=EGU					
F:DELETE	1941/BAL	2387=EGU					
F:EDIT	1942/BAL	2412=EGU					
F:EI	85/DEF 1783/CW 5461/M:READ 5521/M:READ	490/DATA 4950/M:CLOSE 5492/M:SETDCB 5654/LW*	562/GEN 5000/M:DELREC 5493/M:READ 5656/LW	600=CSECT 5117/M:DELREC 5498/LW 6106/M:SETDCB	601=M:DCB 5220/LW 5501/LW* 6107/M:WRITE	609=EGU 5460/M:SETDCB 5520/M:PRECARD 6137/M:WRITE	1768/CW
F:END	452/B	1943/BAL	2114/BEZ	2453=EGU			
F:EB	86/DEF 1778/CW	492/DATA 4958/M:CLOSE	579/GEN 4962/M:CLOSE	615=CSECT 5067/M:CLOSE	616=M:DCB 6073/M:SETDCB	624=EGU 6074/M:WRITE	1771/CW
F:LNK	109=EGU 1942/BAL 2011/B* 2343/B* 2603/B*	1936/BAL 1943/BAL 2116/B* 2359/B* 2608/B*	1937/BAL 1944/BAL 2238/B* 2376/B* 2634/B*	1938/BAL 1945/BAL 2246/B* 2399/B*	1939/BAL 1949/BAL 2314/B* 2405/B*	1940/BAL 1997/B* 2324/B* 2431/B*	1941/BAL 2005/B* 2340/B* 2439/B*
F:MERGE	1944/BAL	2468=BAL					
F:RP							

	1945/BAL	2368=EGU					
FITA	1937/BAL	2616=EGU					
FC10	4395/BG	4417=LI					
FC15	4390/BGEZ	4421/BG	4423/BFZ	4431=LI			
FC15A	4401/BE	4432=LB					
FC20	4413/B	4451=PULL					
FC30	4435/BCS	4457=LI					
FC35	4462=PULL	4470/B	4476/B				
FC40	4408/BGE	4468=BAL					
FC45	4406/BL	4474=BAL					
FF	234=EGU	4253/DATA					
FID1ADR	317=DATA	2474/STW	2541/LW				
FID2ADR	318=DATA	2511/STW	2543/LW				
FIELD CNT	319=RES	4743/STW	4758/MTW	4776/MTW	4806/MTW	4809/MTW	4852/MTW
FILE	601/MIDCB	616/MIDCB					
FILETYPE	320=DATA	640/MTW	1781/MTW	1838/MTW	2061/STW	2413/MTW	2426/STW
	2438/STW	2454/MTW	5687/MTW	5760/MTW	5831/MTW	5838/STW	
FINDCBLUMN	3499/BAL	3517/BAL	3546/BAL	3569/BAL	3585/BAL	3603/BAL	3625/BAL
	3640/BAL	4386=EGU					
FINDMATCH	2817/BAL	4434/BAL	4487=EGU				

FINISH\$STEP\$LOOP						
3192/B	3222=EGU	3688/B				
FIRST\$F:CMND						
220=EGU	1834/CI					
FIRST\$I:CMND						
221=EGU	1358/CI	1567/CI	1840/CI			
FIRST\$R:CMND						
222=EGU	1836/CI					
FIRST\$FROM						
322=DATA	2480/STW	2493/STW	2499/LW	2548/LW		
FIRST\$SET						
323=RES	1901/STW	2797/STW	2798/STW	2815/STW	2829/LW	2836/LW
2842/LW	3191/STW	3207/LW	3217/STW	3218/LW	3280/STW	3404/STW
3424/LW	3463/STW	3680/LW	3687/STW	3700/LW	3740/LW	3754/LW
FM10						
4495/BLE	4501=LI					
FM10A						
4503=CB	4507/BLE					
FM10B						
4505=AI	4526/BNE					
FM15						
4497/B	4512=LCI					
FM20						
4504/BE	4517=LI					
FM20A						
4522=AI	4527/BDR					
FM30						
4521/BEZ	4531=LW					
FNDTBL1						
2820/EXU	2880=EGU					
FNDTBL2						
2852/EXU	2886=EGU					
FNDTYP						
2817=BAI	4532/CI					
FND20						
2811=CW	2848/B					
FND30						

FND32	2824-BAL	2881/B					
FND35	2829-LW	2882/B					
FND40	2836-LW	2883/B					
FND50	2818/BCS	2825/B	2832/B	2842-LW			
FND60	2814/BG	2846/BE	2852-EXU	2877/B			
FND65	2856-LW	2887/B					
FND65A	2867-CI	2888/B	2889/B				
FND70	2857/BEZ	2869-BAL					
FRSTCLMN	2812/BE	2875-BAL					
GEN4	321-RES	2816/LW	3268/LW	3327/LW	4391/LW	4405/CW	4611/STW
GET\$CBL\$PAIR	124-CBM						
GET\$INCREMENT	1421-EQU	1572/B	1644/B				
GET\$SEQ\$INCR	1109-EQU	1243/B	1499/B				
GETFILEID	1094-EQU	1182/NXTPRM	1457/B				
GETNEXT\$ERROR	1084/BAL	1154/BAL	1179/BAL	1196/BAL	1224/BAL	1260/BAL	3810-EQU
GETNEXT\$FINISH	3819/B	3947-EQU	3996/B	4081/BBL			
GETNEXTNAME	3988-EQU	4009/B	4102/B	4138/B	4230/BGE	4244/B	
GETNEXTPARAM	193/CNAME	3914-EQU					
	194/CNAME	4058-EQU					

GF#PUSH#SUBR					
GF10	3820/BAL	3850/BAL	3866/BAL	3893=EQU	
GF15	3824/NXTNAM	3837=NXTNAM			
GF18	3838/NXTNAM	3847=LB			
GF20	3843/B	3854/NXTNAM	3861=NXTNAM		
GF30	3855/NXTNAM	3856/NXTNAM	3857/NXTNAM	3876=LI	
GF5	3833/B	3870/NXTNAM	3871/NXTNAM	3872/NXTNAM	3882=STW
GNTBL1	3818=LI	3849/BG	3865/BG		
GNTBL1SZ	2083/CB	2947/CB	3922/CB	4013=EQU	4019/EQU
GNTBL2	2081/LI	2945/LI	3921/LI	4019=EQU	
GNTBL2SZ	3939/CB	4034=EQU	4046/EQU		
GNTYTBL1	3938/LI	4046=EQU			
GN10	4008/LB	4024=EQU			
GN25	3930=CLM	3970/BNE			
GN30	3954/BL	3960=STW			
GN35	3931/BIL	3933/BIL	3936/BIL	3940/BE	3965=STB
GN45	3943/BG	3974=LI			
GN50	3993/BE	4000=SLS			
GPTBL	3923/BE	4008=LB			

GP1BLSZ	4066/CB	4248-EQU	4256/EQU	
GPTYTBL	4065/LI	4256-EQU		
GP10	4100/LB	4261-EQU		
GP20	4078/BIL	4085-STB	4091/BIL	4094/BIL
GP30	4067/BE	4100-LB		
GP30A	4075/BE	4106-LB	4114/B	
GP35	4112-STB	4121/BE		
GP40	4111/BE	4118-LB		
GP43	4096/B	4127-LI		
GP45	4133/BEZ	4142-BAL		
GP50	4109/BG	4148-BAL		
GP52	4070/BIL	4072/BE	4154-LI	
GP52A	4161-CLM	4174/B		
GP53	4162/BIL	4167-MI	4214/BIL	
GP53A	4166/B	4178-LB	4186/BGEZ	4218/B
GP55	4171/BGE	4187-BAL	4216/BNE	
GP60	4181/BBL	4193-CI		
GP63	4164/BNE	4194/BE	4200-CI	
	4201/BE	4222-CI		

GP66							
HEXCHAR	4203/BNE	4237=LI					
	277=TEXT	5088/LB	5091/LB	5097/LB	5101/LB		
I:DELETE	1969/BAL	3484=EGU					
I:DELETE01	3489/BLZ	3493/BNE	3498=EGU				
I:DELETE02	3491=LB	3494/BDR					
I:FOLLOW\$BY	1971/BAL	3545=EGU					
I:JUMP	1977/BAL	3677=EGU					
I:LNK	111=EGU	1968/BAL	1969/BAL	1970/BAL	1971/BAL	1972/BAL	1973/BAL
	1974/BAL	1975/BAL	1976/BAL	1977/BAL	1978/BAL	1979/BAL	1980/BAL
	1981/BAL	1982/BAL	3417/B*	3500/BCS*	3508/B*	3516/STW	3518/BCS*
	3538/B*	3547/BCS*	3560/B*	3568/STW	3570/BCS*	3574/BEZ*	3577/B*
	3586/BCS*	3595/B*	3604/BCS*	3616/B*	3624/STW	3626/BCS*	3630/BEZ*
	3633/B*	3641/BCS*	3670/B*	3694/B*	3702/B*	3714/B*	3720/B*
	3731/B*	3742/B*	3749/B*	3753/BEZ*	3758/B*		
I:NO\$CHANGE	1978/BAL	3709=EGU					
I:OVERWR\$EXTEND	1970/BAL	3515=EGU					
I:OVERWRITE	1973/BAL	3584=EGU					
I:PRECEDE\$BY	1974/BAL	3602=EGU					
I:REVERSE\$BPFLAG	1979/BAL	3727=EGU					
I:SET	1968/BAL	3387=EGU					
I:SHIFT\$LEFT	1972/BAL	3567=EGU					
I:SHIFT\$RIGHT							

1975/BAL	3623-EQU				
I:SUBSTITUTE					
1976/BAL	3639-EQU				
I:TS\$CMND\$NMR					
223-EQU	1625/LI				
I:TY\$CMND\$NMR					
224-EQU	1628/LI				
I:TYPE					
1981/BAL	3738-EQU				
I:TYPE\$SUP\$SEQ					
1980/BAL	3745-EQU				
I:TYPEX					
1982/BAL	3751-EQU				
ICBRCHTBL					
949/EXU	1042-EQU				
ICNAMETBL					
937/CW	1015-EQU	1024/EQU			
ICNMRTBL					
947/LB	1028-EQU				
ICS10					
910/NXTPRM	935-LI	959/B			
ICS20					
938/BE	946-LI				
ICS50					
917/NXTPRM	953-BAL				
ICS90					
995/BE	999/BE	1007-B			
ICTBLSZ					
935/LI	1024-EQU				
ILGL\$SEMICBLON					
761-EQU	1072/NXTPRM	1089/NXTPRM	1112/NXTPRM	1127/NXTPRM	1183/NXTPRM
1203/NXTPRM	1231/NXTPRM	1264/NXTPRM	1279/NXTPRM	1315/NXTPRM	1361/NXTPRM
1424/NXTPRM	1432/NXTPRM	1438/NXTPRM	1533/NXTPRM	1579/NXTPRM	3825/NXTNAM
ILGL\$SEQ2					
1099/NXTPRM	1124/BEZ	1132-EQU	1299/NXTPRM	1513/NXTPRM	1562/NXTPRM
INOUT					
603/M:DCB					

INSMSG							
INS10	2972/DATA	2982=TEXTC					
INS20	2918/BEZ	2926=BAL					
INS35	2933=B	2966/BL					
INS38	2956/BLE	2962=BAL					
INS40	2968/BLE	2971=BAL					
INS50	2941/BE	2973=EQU					
INTFLAG1	2937/EXU	2976=LI					
	502=DATA	697/STW	1685/LW	1709/LW	1720/LW	2200/STW	2226/STW
	2561/STW	2844/STW	3066/STW	3429/STW	3457/STW	5005/STW	5014/STW
INTFLAG2							
	503=DATA	698/STW	1727/LW	2201/STW	2227/STW	2562/STW	3067/STW
INTG							
	182=EQU	740/NXTPRM	924/LI	956/LI	976/NXTPRM	977/LI	
	1096/NXTPRM	1116/NXTPRM	1206/NXTPRM	1234/NXTPRM	1297/NXTPRM	1311/NXTPRM	1312/LI
	1340/NXTPRM	1371/NXTPRM	1372/LI	1379/NXTPRM	1398/NXTPRM	1427/NXTPRM	1428/LI
	1435/NXTPRM	1471/NXTPRM	1487/NXTPRM	1511/NXTPRM	1525/NXTPRM	1560/NXTPRM	
	1607/NXTPRM	1616/NXTPRM	4240/LI				
I0ERRC0D							
	428/DATA	430=TEXT	5089/STB	5093/STB	5099/STB	5102/STB	
I0ERRMSG							
	428=DATA	428/DATA	5104/DATA				
J:CCBUF							
	79/REF	5576/LB	5580/LB	5591/LB			
JB:CCARS							
	83/REF	5575/LB	5582/LB				
JMP10							
	3679/BEZ	3692=BAL					
JMP15							
	3686/BCS	3698=BAL					

KBUF	324-RES 6139/M:WRITE	5465/M:READ	5645/STW	5647/STB	6076/M:WRITE	6109/M:WRITE	
KEY	5000/M:DELREC	5117/M:DELREC	5465/M:READ	6076/M:WRITE	6109/M:WRITE	6139/M:WRITE	
KEYED	602/M:DCB	617/M:DCB					
KEYM	606/M:DCB	620/M:DCB					
KPE	263-DATA	6034/BR					
K1	261-DATA	3729/EOR					
K10	262-DATA	4908/DW					
L	1767/LW 2583/LW 3284/CW 5515/LW	1777/LW 2731/CW 3459/CW	2193/CW 2811/CW 3818/LI	2221/CW 3038/CW 4170/CW	2481/LW 3053/CW 4990/CW	2501/CW 3189/CW 5305/LW	2551/CW 3215/CW 5307/LW
LASTCLMN	325-RES 4612/STW	3270/LW	3334/CW	3339/LW	3529/CW	4407/CW	4490/LW
LASTFROM	326-RES 3060/STW	2482/STW 3069/XW	2496/STW 3071/XW	2503/CW 3123/LW	2553/CW	2556/SW	2593/STW
LASTKEY	327-DATA 5655/STW	644/STW	1759/STW	5117/M:DELREC	5435/LW	5519/STW	5525/LW
LASTSET	328-RES 3425/CW	2801/STW 3461/CW	2813/CW	2845/CW	3264/STW	3395/STW	3406/CW
LCLETTERS	296-DATA	3935/CLM	4077/CLM	4090/CLM			
LETTERS	294-DATA	3930/CLM	4080/CLM	4093/CLM			
LF							

	130-EQU 4018/DATA	145-EQU 4251/DATA	169-LI 5935/LI	198-BAL 5988/LI	235-EQU	417/GEN4	24 417/GEN4
LINK1	488/DATA	493-DATA					
LNK							
	103-EQU	198/BAL	441/BAL	447/BAL	458/BAL	524/PZE*	642/BAL
	655/BAL	661/LI	662/AI	663/SLS	664/STW	667/STB	672/SW
	724/BAL	727/BAL	750/BAL	764/BAL	905/BAL	908/BAL	919/BAL
	925/BAL	929/BAL	940/BAL	953/BAL	957/BAL	967/BAL	978/BAL
	1001/BAL	1064/BAL	1066/BAL	1070/BAL	1081/BAL	1083/BAL	1084/BAL
	1086/BAL	1100/BAL	1105/BAL	1118/BAL	1125/BAL	1133/BAL	1142/BAL
	1151/BAL	1153/BAL	1154/BAL	1156/BAL	1166/BAL	1173/BAL	1179/BAL
	1180/BAL	1193/BAL	1195/BAL	1196/BAL	1198/BAL	1210/BAL	1211/BAL
	1213/BAL	1218/BAL	1224/BAL	1226/BAL	1238/BAL	1239/BAL	1241/BAL
	1245/BAL	1257/BAL	1259/BAL	1260/BAL	1262/BAL	1275/BAL	1277/BAL
	1293/BAL	1295/BAL	1300/BAL	1305/BAL	1313/BAL	1320/BAL	1331/BAL
	1337/BAL	1346/BAL	1350/BAL	1354/BAL	1357/BAL	1373/BAL	1380/BAL
	1394/BAL	1396/BAL	1401/BAL	1405/BAL	1410/BAL	1418/BAL	1429/BAL
	1436/BAL	1443/BAL	1454/BAL	1456/BAL	1467/BAL	1469/BAL	1474/BAL
	1478/BAL	1483/BAL	1490/BAL	1494/BAL	1498/BAL	1507/BAL	1509/BAL
	1514/BAL	1519/BAL	1527/BAL	1531/BAL	1538/BAL	1551/BAL	1557/BAL
	1563/BAL	1569/BAL	1571/BAL	1577/BAL	1590/BAL	1604/BAL	1613/BAL
	1630/BAL	1637/BAL	1638/BAL	1639/BAL	1642/BAL	1655/B*	1662/B*
	1676/BAL	1688/BAL	1692/BAL	1712/BAL	1716/BAL	1723/BAL	1738/BAL
	1743/BAL	1748/BAL	1751/BAL	1770/BAL	1773/BAL	1780/BAL	1785/BAL
	1848/BAL	1902/BAL	1903/BAL	1908/BAL	1914/BAL	2009/BAL	2023/BAL
	2029/BAL	2056/BAL	2058/BAL	2070/BAL	2075/BAL	2093/BAL	2098/BAL
	2099/BAL	2103/BAL	2107/BAL	2120/BAL	2122/BAL	2131/BAL	2170/BAL
	2177/BAL	2179/BAL	2180/BAL	2184/BAL	2192/BAL	2198/BAL	2220/BAL
	2223/BAL	2232/BAL	2236/BAL	2237/BAL	2242/BAL	2243/BAL	2244/BAL
	2253/BAL	2259/BAL	2298/BAL	2303/BAL	2311/BAL	2313/BAL	2318/BAL
	2320/BAL	2321/BAL	2322/BAL	2328/BAL	2330/BAL	2333/BAL	2337/BAL
	2339/BAL	2341/BAL	2388/BAL	2392/BAL	2397/BAL	2403/BAL	2415/BAL
	2422/BAL	2435/BAL	2443/BAL	2444/BAL	2456/BAL	2468/BAL	2475/BAL
	2500/BAL	2506/BAL	2512/BAL	2523/BAL	2525/BAL	2530/BAL	2542/BAL
	2544/BAL	2546/BAL	2549/BAL	2558/BAL	2569/BAL	2574/BAL	2589/BAL
	2590/BAL	2594/BAL	2595/BAL	2598/BAL	2600/BAL	2601/BAL	2605/BAL

2606/BAL	2628/BAL	2657/BAL	2667/BAL	2696/BAL	2728/BAL	2729/BAL
2730/BAL	2738/BAL	2744/BAL	2750/BAL	2756/BAL	2773/BAL	2806/BAL
2807/BAL	2817/BAL	2824/BAL	2830/BAL	2831/BAL	2837/BAL	2847/BAL
2860/BAL	2861/BAL	2869/BAL	2875/BAL	2926/BAL	2928/BAL	2934/BAL
2939/BAL	2957/BAL	2962/BAL	2963/BAL	2969/BAL	2971/BAL	3036/BAL
3042/BAL	3044/BAL	3047/BAL	3062/BAL	3064/BAL	3070/BAL	3072/BAL
3082/BAL	3084/BAL	3089/BAL	3093/BAL	3097/BAL	3098/BAL	3104/BAL
3110/BAL	3118/BAL	3124/BAL	3128/BAL	3132/BAL	3148/BAL	3151/BAL
3153/BAL	3158/BAL	3164/BAL	3186/BAL	3187/BAL	3208/BAL	3214/BAL
3223/BAL	3226/BAL	3231/BAL	3237/BAL	3266/BAL	3278/BAL	3291/BAL
3292/BAL	3293/BAL	3295/BAL	3304/BAL	3305/BAL	3306/BAL	3314/BAL
3320/BAL	3354/B	3369/B	3375/BGE	3397/BAL	3402/BAL	3408/BAL
3416/BAL	3427/BAL	3442/BAL	3443/BAL	3455/BAL	3458/BAL	3468/BAL
3475/BAL	3495/BAL	3499/BAL	3504/BAL	3506/BAL	3507/BAL	3517/BAL
3522/BAL	3537/BAL	3546/BAL	3554/BAL	3556/BAL	3558/BAL	3559/BAL
3569/BAL	3575/BAL	3576/BAL	3585/BAL	3590/BAL	3593/BAL	3594/BAL
3603/BAL	3609/BAL	3611/BAL	3614/BAL	3615/BAL	3625/BAL	3631/BAL
3632/BAL	3640/BAL	3651/BAL	3659/BAL	3665/BAL	3668/BAL	3669/BAL
3681/BAL	3685/BAL	3692/BAL	3698/BAL	3701/BAL	3718/BAL	3741/BAL
3748/BAL	3755/BAL	3790/B	3800/BE	3801/BAL	3818/LI	3820/BAL
3850/BAL	3866/BAL	3889/B	3903/B	3948/LW	3990/LB*	3992/CB*
4001/LW*	4142/BAL	4148/BAL	4187/BAL	4231/BAL	4283/BR	4285/LW
4297/B	4307/BLZ	4309/B	4331/B	4366/B	4387/PUSH	4424/BAL
4434/BAL	4451/PULL	4453/B	4462/PULL	4464/B	4468/BAL	4474/BAL
4513/B	4532/CI	4537/B	4547/PUSH	4565/PULL	4566/B	4578/BAL
4620/B	4622/BAL	4624/LI	4625/STW	4626/STW	4636/BEZ	4662/B
4675/BAL	4686/BAL	4710/B	4740/BGE	4752/BAL	4821/B	4825/BAL
4877/BAL	4927/B	4934/PUSH	4935/LI	4938/STW	4939/BDR	4941/PULL
4942/B	4951/B	4959/B	4963/B	4986/BAL	4997/BAL	4998/LI
4999/STB	5012/BAL	5020/BAL	5021/BAL	5026/B	5034/BAL	5035/BAL
5040/B	5044/BAL	5065/BAL	5070/B	5080/B	5103/BAL	5118/B
5130/PUSH	5131/LW	5134/BAL	5186/PULL	5188/B	5218/BAL	5227/B
5233/B	5243/B	5268/BAL	5272/B	5285/B	5358/B	5376/BAL
5380/B	5393/B	5400/PUSH	5402/BAL	5403/LW	5405/LW	5406/AI
5408/CB	5411/AI	5412/CI	5414/LI	5415/STW	5416/STB	5419/PULL
5420/B	5432/PUSH	5433/BAL	5437/PULL	5439/B	5443/BAL	5444/PULL
5446/B	5456/PUSH	5457/BAL	5458/BAL	5466/BAL	5467/PULL	5469/B

5477/PULL	5479/B	5489/PUSH	5490/BAL	5497/BAL	5506/PULL	5507/B
5522/BAL	5528/BAL	5541/PULL	5542/B	5586/STW	5587/BAL	5590/LW
5613/B	5637/B	5648/B	5653/PUSH	5656/LW	5657/SLS	5658/STW
5659/AI	5660/STW	5662/CB	5665/CB	5667/BAL	5670/STB	5671/STW
5674/BAL	5675/PULL	5676/B	5688/BLZ	5690/BNEZ	5691/PUSH	5704/PULL
5705/B	5715/BAL	5761/BLZ	5763/BEZ	5769/B	5832/BLZ	5833/PUSH
5834/BAL	5835/BAL	5839/PULL	5840/B	5849/PUSH	5867/BAL	5875/BAL
5877/PULL	5878/B	5881/BAL	5895/BLZ	5896/PUSH	5900/LW	5905/PUSH
5906/LW	5942/PULL	5943/B	5975/PUSH	5977/LW	5991/LW	5991/LW
5993/LB*	5996/CB*	6001/CB*	6011/BAL	6014/PULL	6015/B	6025/PUSH
6026/LW	6028/BAL	6056/BAL	6058/PULL	6059/B	6069/PUSH	6070/BAL
6071/BAL	6080/PULL	6082/B	6090/PULL	6092/B	6102/PUSH	6103/BAL
6104/BAL	6113/PULL	6115/B	6123/PULL	6125/B	6134/PUSH	6135/BAL
6136/BAL	6142/PULL	6143/B				
LP						
237=EGU	1724/GEN4	3119/GEN4	4017/DATA			
LPAR						
187=EGU	3824/NXTNAM					
M:EI						
81/REF	609/EGU					
M:EB						
82/REF	624/EGU					
M:TRTN						
1757=M:TRTN	1791/B	1797/B				
M:UC						
80/REF	515/GEN	517/GEN	522/GEN	527/GEN	2635/M:DEVICE	
2636/M:DEVICE	2637/M:DEVICE	2638/M:DEVICE	5609/LW	5707/LB	5766/LB	5770/LB
5777/LB	5787/LB					
MASTEREXECUTIVE						
743/NXTPRM	970/NXTPRM	981/NXTPRM	1011/NXTPRM	1073/NXTPRM	1090/NXTPRM	
1113/NXTPRM	1128/NXTPRM	1184/NXTPRM	1265/NXTPRM	1280/NXTPRM	1316/NXTPRM	
1362/NXTPRM	1369/NXTPRM	1377/NXTPRM	1383/NXTPRM	1425/NXTPRM	1433/NXTPRM	
1439/NXTPRM	1534/NXTPRM	1580/NXTPRM	1594/NXTPRM	1634/NXTPRM	1811=EGU	
MASTERPARSER						
443/B	451/BNEZ	454/B	693=EGU	752/B	766/B	942/B
1003/B	1135/B	1175/B	1247/B	1322/B	1445/B	1540/B
1774/B	1782/BGZ	1786/B	1795/LI	1850/B	1910/B	1916/B

	1926/B	1931/B	1937/BAL	2630/B	2740/B	2746/B	2752/B
	3200/B	3227/B	3233/B	3412/B	3423/BGZ	3451/B	3497/B
	3803/B	4144/B	4150/B	4189/B	4233/B	4627/B	
MAXCLMN							
	216=EGU	308/RES	331/RES	344/RES	530/DATA	2091/CI	2655/CI
	2711/CI	2720/CI	2955/CI	3271/CI	3341/CI	3346/CI	4325/CI
	4351/LI	4373/LI	4550/CI	4559/CI	4591/LI	4617/CI	4638/LI
	4724/CI	4739/CI	4746/CI	4882/LI	4884/LI	4892/LI	4935/LI
	5412/CI	5414/LI	5464/M:READ	5495/M:READ	5521/M:READ	5560/LI	
MAXSEQ							
	329=DATA	2101/CW	2230/CW	2567/CW	2967/CW		
MODE							
	60=EGU	69/SET	77/D0	246/D0	282/D0	295/D0	449/D0
	468/D0	588/D0	599/D0	614/D0	639/D0	694/D0	718/D0
	773/D01	816/D01	859/D01	1673/D0	1776/D01	1816/D0	1864/D0
	2027/D01	2031/D01	2051/D0	2072/D0	2112/D0	2199/D0	2225/D0
	2427/D0	2457/D0	2560/D0	2617/D0	2659/D0	2681/D0	2771/D01
	2843/D01	2898/D0	2907/D01	2914/D01	2936/D0	2975/D0	3065/D0
	3267/D0	3428/D01	3456/D01	3851/D01	3867/D01	3883/D01	3934/D0
	4044/D01	4076/D0	4089/D0	4252/D01	4265/D01	5002/D0	5013/D01
	5105/D0	5401/D01	5459/D01	5491/D01	5563/D0	5624/D0	5673/D01
	5685/D0	5850/D0	5857/D0	5883/D01	5927/D0	5976/D0	6072/D01
	6105/D01						
MOVESEQ							
	1688/BAL	1712/BAL	1723/BAL	1743/BAL	3089/BAL	3118/BAL	3124/BAL
	5129=EGU	5528/BAL					
MOVESTRING							
	3522/BAL	3556/BAL	3590/BAL	3611/BAL	3665/BAL	4546=EGU	
MQ10							
	5140/BNE	5146=LI					
MQ20							
	5149/BNEZ	5155=LB	5160/BLE				
MQ25							
	5167=CW	5173/B					
MQ30							
	5168/BL	5177=LI					
MQ30A							

MRG10	5179-LB	5184/BDR					
MRG13	2488/BNE	2499-LW					
MRG14	2530-BAL						
MRG15	2531-LI	2584/B					
MRG17	2534/BEZ	2541-LW					
MRG20	2542-LW						
MRG25	2551-CW	2575/B					
MRG30	2562/BLE	2572-CW					
MRG35	2513/BCS	2578-EQU					
MRG55	2583-LW						
MRG56	2552/BGE	2554/BG	2571/B	2587-SW			
MRG65	2565/BEZ	2589-BAL					
MRG70	2573/BGE	2593-STW					
MRG80	2502/BGE	2504/BG	2598-BAL				
MRG82	2477/BCS	2600-BAL					
MSG0	2514/BCS	2605-BAL					
MSG1	417-GEN4	656/DATA	1677/DATA	1752/DATA	5868/DATA	5876/DATA	6012/DATA
MSG2	418-TEXTC	2185/DATA					
	419-TEXTC	2245/DATA					

MSG3							
	420=TEXTC	2398/DATA					
MSG4							
	421=TEXTC	5836/DATA					
MSG5							
	422=TEXTC	2547/DATA					
MSG6							
	423=TEXTC	2859/LI	2862/DATA	5019/LI	5022/DATA	5033/LI	5036/DATA
MSG7							
	424=TEXTC	3095/LI	3099/DATA				
MSG8							
	425=TEXTC	3441/LI	3444/DATA				
MS10							
	4565=PULL	4571/BEZ	4577/B	4580/B			
MS20							
	4560/BGE	4570=AI					
MS20A							
	4551/BGE	4573=AI	4576/BDR				
MS20B							
	4575/BNE	4578=BAI					
MS5							
	4555=AI	4561/BDR					
MVD:REC:CNT							
	352=DATA	707/STW	2470/STW	2559/MTW	3001/STW	3063/MTW	3083/MTW
	3096/LW						
MVEMSG1							
	433=TEXTC	3088/LI	3092/STB	3094/DATA			
MVEMSG2							
	435=TEXTC	3117/LI	3127/STB	3129/DATA			
MVE10							
	3002=LI						
MVE20							
	3015/BEZ	3023=LW					
MVE30							
	3053=CW	3073/B					
MVE35							
	3056/BLE	3077=BL					

MVE40	2591/B	3077/B	3088-LI	3112/B			
MVE50	3026/BIL	3028/BIL	3032/BIL	3034/BIL	3104-BAL		
MVE53	3054/BE	3110-BAL					
MVE56	2596/B	3059/BGE	3080/BGE	3116-SW			
MVE58	2599/B	3039/BE	3041/BG	3132-BAL			
NAME	179-EQU	198/BAL	1085/LI	1155/LI	1197/LI	1225/LI	1261/LI
	3814/NXTNAM	3823/NXTNAM	3838/NXTNAM	3855/NXTNAM	3862/NXTNAM	3870/NXTNAM	3983/LI
NCG10	3711/BEZ	3712-BAL					
NEWCDENTRY	905/BAL	919/BAL	953/BAL	1064/BAL	1081/BAL	1151/BAL	1193/BAL
	1257/BAL	1275/BAL	1293/BAL	1331/BAL	1337/BAL	1394/BAL	1454/BAL
	1467/BAL	1507/BAL	1551/BAL	1557/BAL	1590/BAL	1604/BAL	1613/BAL
	1630/BAL	4278-EQU					
NEWKEY	6077/MIWRITE	6110/MIWRITE					
N6CHGFLG	330-DATA	645/STW	1760/STW	3205/MTW	3713/STW		
N6PR0MPT\$FPT	513-EQU	2032/CAL1	2662/CAL1	2908/CAL1			
NUM	131/D8	132/D8	146/D8	147/SET	148/D8	162/D8	163/D8
	197/SET	201/ERR0R					
NXTNAM	193-CNAME						
NXTPRM	194-CNAME						
0\$ABN	564/DATA	5213/LI	5237-RES				
0\$ACCT	573-RES	5216/LI	5374/LI				

0\$FPT	562-GEN 5389/STW	5212/STW 5390/CAL1	5214/STW 5636/CAL1	5219/CAL1	5370/STW	5372/STW	5377/CAL1
0\$NAME							
0\$PASS	571-RES	5215/LI	5373/LI				
0EX10	575-RES	5217/LI	5375/LI				
0EX20	3529-CW	3533/B					
0N\$ABN	3530/BGE	3537-BAL					
0NEWKEY	5369/LI	5384-RES					
0PEN	6140/M:WRITE						
0PENINIT	2058/BAL	2422/BAL	2512/BAL	5200-EQU			
0PENNEW	5065/BAL	5218/BAL	5268/BAL	5297-EQU	5376/BAL		
0PEN1	2029/BAL	5367-EQU					
0PEN2	2170/BAL	2259/BAL	2303/BAL	2475/BAL	2542/BAL	5206-EQU	
0PEN3	2177/BAL	2253/BAL	2544/BAL	5258-EQU			
0UT	2180/BAL	2292/BAL	5253-PUSH				
02\$ABN	618/M:DCB						
02\$ACCT	581/DATA	5262/LI	5276-RES				
02\$FPT	585-RES	5063/LI	5266/LI				
02\$NAME	579-GEN 5281/STW	5059/STW 5282/CAL1	5061/STW	5066/CAL1	5261/STW	5263/STW	5269/CAL1

583-RES	5062/LI	5265/LI				
625PASS						
587-RES	5064/LI	5267/LI				
PARAMBUF						
331-RES	745/LW	915/LW	921/XW	926/STW	936/LW	955/XW
958/STW	1123/MTW	1140/STW	1159/LW	1163/STW	1167/LW	1220/LW
1652/LW	1654/STW	1659/LW	1660/STW	3781/LW	3815/LB	3847/LB
3863/LB	3886/STW	3894/LB	3899/LW	3965/STB	3976/STB	3980/STB
4085/STB	4112/STB	4129/STB	4134/STB	4207/STW	4225/STW	4229/CW
4241/STW						
PARAMPSN						
332-RES	737/STW	1000/MTW	3769/LW	3774/MTW	3951/MTW	5913/LW
PARSE:BP						
815/B	1061-EQU					
PARSE:BUILD						
819/B	846/B	1080-EQU				
PARSE:CM						
840/B	1292-EQU					
PARSE:COPY						
820/B	847/B	1150-EQU				
PARSE:CR						
818/B	1062-EQU					
PARSE:CT						
841/B	1289-EQU					
PARSE:DE						
826/B	1330-EQU					
PARSE:DELETE						
821/B	848/B	1255-EQU				
PARSE:EDIT						
822/B	849/B	1256-EQU				
PARSE:END						
823/B	850/B	1273-EQU				
PARSE:FD						
827/B	1391-EQU					
PARSE:FS						
839/B	1392-EQU					
PARSE:FT						

828/B	1393-EQU	
PARSE: I: CMND\$INTG		
740/NXTPRM	913-EQU	
PARSE: I: CMND\$STRG		
741/NXTPRM	903-EQU	
PARSE: IN		
829/B	1452-EQU	
PARSE: IS		
830/B	1453-EQU	
PARSE: JU		
843/B	1553/B	1556-EQU
PARSE: MD		
831/B	1465-EQU	
PARSE: MERGE		
824/B	851/B	1192-EQU
PARSE: MK		
832/B	1466-EQU	
PARSE: NO		
844/B	1274-EQU	
PARSE: RF		
845/B	1589-EQU	
PARSE: RN		
833/B	1506-EQU	
PARSE: RP		
825/B	1060-EQU	
PARSE: SE		
842/B	1333/B	1336-EQU
PARSE: SS		
834/B	1549-EQU	
PARSE: ST		
835/B	1550-EQU	
PARSE: TA		
817/B	1063-EQU	
PARSE: TC		
838/B	1603-EQU	
PARSE: TS		
836/B	1611-EQU	

PARSE:TX				
	852/B	1588-EQU		
PARSE:TY				
	837/B	1612-EQU		
PASS				
	604/M:DCB	622/M:DCB		
PATCH				
	78/DEF	553-RES		
PBU05				
	1097/NXTPRM	1139-LI		
PBU10				
	1098/NXTPRM	1104-LI		
PBU20				
	1117/NXTPRM	1122-LI		
PBU30				
	1114-EQU	1143/B		
PCM10				
	1298/NXTPRM	1304-LI		
PCM20				
	1308/NXTPRM	1309/NXTPRM	1320-BAL	
PC910				
	1169/BE	1171/BE	1179-BAL	
PC93				
	1161/BNE	1164-EQU		
PC95				
	1172-EQU	1222/BNE		
PDE10				
	1341/NXTPRM	1350-BAL		
PDE15				
	1342/NXTPRM	1354-BAL		
PDE20				
	1359/BGE	1366-NXTPRM		
PDE5				
	1340/NXTPRM	1346-BAL		
PERIOD				
	189-EQU	3824/NXTNAM	3839/NXTNAM	3854/NXTNAM
PFD10				

PFD15	1399/NXTPRM	1405=BAL					
PFD20	1400/NXTPRM	1409=LI					
PMD10	1413/NXTPRM	1414/NXTPRM	1443=BAL				
PMD15	1472/NXTPRM	1478=BAL					
PMD20	1473/NXTPRM	1482=LI					
PMD25	1488/NXTPRM	1494=BAL					
PME15	1489/NXTPRM	1498=BAL					
PME20	1208/NXTPRM	1212=LI					
PME30	1202/NXTPRM	1217=LI					
PME35	1235/NXTPRM	1239=BAL					
PME40	1236/NXTPRM	1240=LI					
PME5	1230/NXTPRM	1245=BAL					
PP10	1207/NXTPRM	1211=BAL					
PP20	4593/BEZ	4602=AI					
PP25	4604/BEZ	4611=STW					
PR	4614/BGE	4616/BL	4618/BG	4622=BAL			
PRMBUFSZ	236=EGU	4017/DATA					
PRN10	333=RES	923/XW	927/STW	1661/MTW	3775/AW	3777/SW	3779/LW
	3882/STW	3989/STW					

	1512/NXTPRM	1518-LI					
PRN20	1526/NXTPRM	1531-BAL					
PRN30	1522/NXTPRM	1523/NXTPRM	1538-BAL				
PROCESSCOL#PAIR	2806/BAL	3186/BAL	3266/BAL	3397/BAL	4588-EQU		
PROMPT\$FPT	509-EQU	695/CAL1					
PROMPT2\$FPT	511-EQU	2899/CAL1					
PRS10	748/BE	756-LB					
PSS10	1561/NXTPRM	1567-CI					
PSS20	1568/BGE	1576-LI					
PTY10	1608/NXTPRM	1617/NXTPRM	1638-BAL				
PTY15	1609/NXTPRM	1618/NXTPRM	1639-BAL				
PTY5	1607/NXTPRM	1616/NXTPRM	1637-BAL				
PULL	142-CNAME						
PURGE	158-CNAME						
PUSH	127-CNAME						
PUTCR	5400-PUSH	6071/BAL	6104/BAL	6136/BAL			
PUTCR2	5404/BNEZ	5409/BE	5419-PULL				
P1	101-EQU	756/LB	904/LI	907/LI	914/LI	924/LI	928/LI
	947/LB	948/STB	956/LI	966/LI	977/LI	1069/LI	1085/LI
	1104/LI	1122/LI	1139/LI	1140/STW	1141/LI	1155/LI	1165/LI

1197/LI	1212/LI	1217/LI	1225/LI	1240/LI	1261/LI	1304/LI
1312/LI	1355/LW	1356/LI	1372/LI	1409/LI	1417/LI	1428/LI
1482/LI	1518/LI	1567/CI	1570/LI	1576/LI	1626/CI	1629/LW
1641/LI	1685/LW	1709/LW	1720/LW	1727/LW	1732/LB	1733/STB
1900/LW	1901/STW	2025/LB	2026/AW	2028/LW	2033/LI	2040/LW
2057/XW	2059/LW	2100/AW	2101/CW	2135/LB	2136/AW	2140/SLS
2145/AI	2148/LB	2167/LB	2168/AW	2175/LB	2176/AW	2197/LW
2209/LW	2227/STW	2229/AW	2230/CW	2251/LB	2252/AW	2256/LB
2257/AW	2269/LB	2270/AW	2271/LB*	2274/AW	2275/LB*	2278/AW
2279/LW*	2282/LB	2283/AW	2284/LB*	2287/AW	2288/LB*	2291/AW
2292/LW*	2296/LB	2297/AW	2300/LB	2301/AW	2335/LB	2336/AW
2390/LB	2391/AW	2420/LB	2421/AW	2472/LB	2473/AW	2474/STW
2479/LI	2480/STW	2481/LW	2482/STW	2485/LB	2487/CI	2490/LB
2491/AW	2492/LW*	2494/AI	2495/LW*	2499/LW	2509/LB	2510/AW
2511/STW	2519/LW	2527/STW	2533/LB	2541/LW	2543/LW	2548/LW
2550/LW	2562/STW	2566/AW	2567/CW	2572/CW	2587/SW	2619/LB
2620/AW	2622/LW	2622/LW*	2624/CW	2649/LW	2733/LW	2768/LW
2796/LW	2797/STW	2816/LW	2829/LW	2836/LW	2858/LW	2912/LW
2964/AW	2965/CW	2967/CW	3009/LW	3025/CLM	3029/LW	3035/XW
3037/XW	3046/XW	3048/LW	3049/SW	3057/AW	3058/CW	3067/STW
3069/XW	3071/XW	3078/AW	3079/CW	3096/LW	3116/SW	3123/LW
3144/LW	3150/LW	3184/LW	3188/STW	3207/LW	3218/LW	3261/LW
3288/LW	3290/LI	3301/LW	3303/LI	3392/LW	3405/STW	3424/LW
3425/CW	3429/STW	3435/LW	3436/CI	3446/LI	3447/STW	3448/STW
3449/STW	3457/STW	3501/AW	3505/SW	3524/AW	3529/CW	3531/STB
3532/AI	3548/AW	3557/AW	3592/AW	3612/AW	3613/AW	3646/LW
3647/AW	3663/LW	3667/AW	3680/LW	3684/LW	3687/STW	3700/LW
3740/LW	3747/LI	3754/LW	3756/LI	3757/STW	3768/PUSH	3770/STB
3772/LB	3773/STB	3775/AW	3776/STB	3777/SW	3778/AW	3782/STW*
3788/STW*	3789/PULL	3894/LB	3895/AI	3896/SLS	3897/AW	3902/PDR
3917/LB	3919/CI	3922/CB	3930/CLM	3932/CLM	3935/CLM	3939/CB
3948/LW	3949/CW	3952/AND	3953/CI	3955/STW	3960/STW	3965/STB
3967/LB	3969/CI	3975/LI	3976/STB	3983/LI	3992/CB	4008/LB
4061/LB	4063/CI	4066/CB	4069/CLM	4071/CI	4074/CI	4077/CLM
4080/CLM	4085/STB	4087/LB	4090/CLM	4093/CLM	4095/LI	4100/LB
4106/LB	4110/CI	4112/STB	4118/LB	4120/CI	4122/LI	4161/CLM
4163/CI	4168/AI	4169/AW	4172/LB	4178/LB	4180/CLM	4183/AI

4184/AW	4202/CI	4211/LB	4213/CLM	4215/CI	4226/LI	4237/LI
4240/LI	4279/PUSH	4280/SLS	4281/BR	4282/SLS	4283/BR	4284/STW
4289/AND	4290/AI	4291/STW	4292/LI	4294/STW	4296/PULL	4308/STW
4324/PUSH	4325/CI	4329/PURGE	4336/CW	4342/CB	4345/AI	4346/CW
4352/SW	4358/AI	4359/CB	4364/PULL	4371/SW	4389/LW	4391/LW
4403/LW	4404/AI	4405/CW	4407/CW	4437/LW	4438/AI	4444/STW
4445/AI	4494/CW	4503/CB	4505/AI	4506/CW	4518/LW	4531/LW
4550/CI	4557/STB	4558/AI	4559/CI	4590/LI	4596/LW	4597/AI
4611/STW	4613/CW	4615/CI	4673/PUSH	4674/AW	4677/SW	4683/LW
4688/SW	4697/LB	4699/AI	4709/PULL	4723/SW	4724/CI	4726/LW
4739/CI	4745/AW	4746/CI	4756/AW	4762/LW	4764/AW	4765/AI
4871/LW	4903/PUSH	4905/LW	4906/LI	4912/BDR	4918/LI	4920/CB
4926/PULL	4976/PUSH	4995/LW	4996/AI	5003/LW	5004/AND	5005/STW
5016/LW	5017/CI	5024/PULL	5030/LW	5031/CI	5038/PULL	5146/LI
5148/CB	5150/AI	5167/CW	5309/LB*	5311/LB*	5317/AW	5318/LB*
5320/LB*	5325/LB*	5328/AW	5333/AI	5339/LB*	5344/LB*	5517/PUSH
5518/LI	5519/STW	5525/LW	5526/AND	5530/LI	5532/LB	5535/AI
5538/AI	5539/STB	5540/PULL	5645/STW	5697/LI	5698/CB	5727/LI
5728/STB	5740/STB	5781/LI	5782/CB	5790/CB	5818/STB	5854/CI
5858/LI	5865/BDR	5869/LI	5884/LI	5896/PUSH	5900/LW	5905/PUSH
5906/LW	5919/SLS	5920/LW	5928/LB	5929/AI	5930/LB	5938/SLS
5939/STW	5942/PULL	5949/LB*	5953/LB*	5962/LI	6032/LB	6033/SLS
6034/BR	6036/STH	6038/LW	6039/SLS	6040/STW	6054/LI	6055/STB
102-EQU	1687/LI	1711/LI	1722/LI	1725/AW	1733/STB	1734/AI
2138/LB	2139/AW	2141/SLS	2147/AI	2150/CB	2156/CB	2492/LW
2493/STW	2495/LW	2496/STW	2521/LW	2528/STW	2770/LW	2803/LB
2804/AW	2859/LI	3011/LW	3027/CLM	3030/LW	3088/LI	3095/LI
3117/LI	3120/AW	3263/LW	3264/STW	3286/CW	3394/LW	3395/STW
3441/LI	3491/LB	3492/CI	3501/AW	3502/LW	3503/LI	3520/LB
3521/AW	3523/LB*	3548/AW	3549/LI	3555/LW	3588/LB	3589/AW
3591/LB*	3610/XW	3647/AW	3648/SW	3650/LI	3658/LI	3664/LW
3666/LB*	3829/LI	3830/PUSH	3831/PUSH	3840/LI	3841/PUSH	3876/LI
3877/PUSH	3885/PULL	3886/STW	3899/LW	3900/PUSH	3915/PUSH	3916/LW
3917/LB	3918/AI	3967/LB	3968/AI	3984/AI	4002/STW	4003/PULL
4059/PUSH	4060/LW	4061/LB	4062/AI	4087/LB	4088/AI	4106/LB
4107/AI	4108/CW	4118/LB	4119/AI	4137/AI	4172/LB	4173/AI

4178/LB	4179/AI	4211/LB	4212/AI	4228/AI	4243/AI	4279/PUSH
4285/LW	4286/AI	4287/SLS	4288/STB	4291/STW	4294/STW	4295/BDR
4296/PULL	4324/PUSH	4329/PURGE	4341/LI	4342/CB	4359/CB	4364/PULL
4409/LI	4432/LB	4433/AW	4446/LB	4446/LB*	4489/STW	4491/LB
4492/SW	4519/LB	4520/AI	4527/BDR	4548/SLS	4549/LB	4555/AI
4556/LB	4573/AI	4574/CB	4589/PUSH	4591/LI	4607/LW	4612/STW
4613/CW	4617/CI	4619/PULL	4674/AW	4677/SW	4682/AW	4683/LW
4684/SW	4688/SW	4689/AW	4691/LI	4698/STB	4700/AI	4706/STB
4707/AI	4715/AW	4716/SW	4718/LI	4723/SW	4726/LW	4727/SW
4745/AW	4784/AH	4785/AI	4786/STH	4861/AW	4865/STH	4870/AWM
4879/SW	4885/LW	4890/AW	4891/LW	4903/PUSH	4904/AI	4910/STB
4911/AI	4914/STB	4915/AI	4920/CB	4922/STB	4923/AI	4926/PULL
4992/CW	5019/LI	5033/LI	5058/LI	5059/STW	5060/LI	5061/STW
5132/STW	5133/LI	5135/LW	5156/STB	5157/AI	5162/STB	5163/AI
5170/STB	5171/AI	5181/STB	5182/AI	5185/LW	5187/SW	5202/LI
5208/LI	5212/STW	5213/LI	5214/STW	5254/LI	5260/LI	5261/STW
5262/LI	5263/STW	5369/LI	5370/STW	5371/LI	5372/STW	5527/LI
5532/LB	5533/CI	5739/LB	5741/STB	5764/PUSH	5768/PULL	5798/LI
5799/STB	5804/LB	5805/STB	6027/LI			

P3

106-EQU	2531/LI	2536/LB	2537/AW	2538/LW	2538/LW*	2539/STW
2566/AW	2587/SW	2793/LI	2819/AI	2856/LW	2858/LW	2867/CI
3012/LW	3018/LW	3019/STW	3049/SW	3057/AW	3078/AW	3116/SW
3502/LW	3505/SW	3553/LB	3557/AW	3573/LW	3608/LB	3613/AW
3629/LW	3645/LB	3648/SW	3657/LCW	3657/LCW	4684/SW	4708/BDR
4715/AW	4717/AW	4727/SW	4759/CW	4770/SW	4771/AW	4829/SW
4879/SW	4881/STW	4883/SW	4890/AW	5057/PUSH	5064/LI	5068/PULL
5075/LB	5078/PULL	5094/LB	5201/PUSH	5207/PUSH	5217/LI	5225/PULL
5231/PULL	5238/LB	5241/PULL	5253/PUSH	5259/PUSH	5267/LI	5270/PULL
5277/LB	5283/PULL	5300/STW*	5303/STW*	5308/STW*	5346/STB*	5354/STB*
5368/PUSH	5375/LI	5378/PULL	5385/LB	5391/PULL	5456/PUSH	5467/PULL
5474/LB	5477/PULL	5489/PUSH	5506/PULL	5512/LB	5541/PULL	6069/PUSH
6080/PULL	6086/LB	6090/PULL	6102/PUSH	6113/PULL	6120/LB	6123/PULL

R: COMMENTARY

1963/BAL

2646-EQU

R: DELETE

1949/BAL

2765-EQU

R: FIND\$DELETE						
1950/BAL	2786=EQU					
R: FIND\$SEQUENCE						
1962/BAL	2781=EQU					
R: FIND\$TYPE						
1951/BAL	2783/B	2788/B	2791=EQU			
R: INSERT						
1952/BAL	2900/B	2902/B	2906=EQU			
R: INSERT\$SUP\$SEQ						
1953/BAL	2896=EQU					
R: LNK						
110=EQU	1950/BAL	1951/BAL	1952/BAL	1953/BAL	1954/BAL	1955/BAL
1956/BAL	1957/BAL	1958/BAL	1959/BAL	1960/BAL	1961/BAL	1962/BAL
1963/BAL	2692/BEZ*	2774/B*	2863/B*	2868/BNE*	2871/B*	2974/B*
3100/B*	3106/B*	3130/B*	3134/B*	3154/B*	3160/B*	3166/B*
3380/B*						
R: MOVE\$DELETE						
1954/BAL	2989=EQU					
R: MOVE\$KEEP						
1955/BAL	2991/B	2994=EQU				
R: RENUMBER						
1956/BAL	3141=EQU					
R: SET\$STEP						
1957/BAL	3173=EQU					
R: SET\$STEP\$TYPE						
1958/BAL	3175/B	3178=EQU				
R: TS\$CMND\$NMR						
225=EQU	1626/CI					
R: TY\$CMND\$NMR						
226=EQU	3351/CI					
R: TYPE						
1960/BAL	3247=EQU					
R: TYPE\$COMPRESSED						
1961/BAL	3245=EQU					
R: TYPE\$SUP\$SEQ						
1959/BAL	3249/B	3252=EQU				
READNXTRANDOM						

2500/BAL	2549/BAL	2807/BAL	2926/BAL	3036/BAL	3047/BAL	3187 ⁴¹ /BAL
3278/BAL	3402/BAL	4986/BAL	4997/BAL	5431-EQU		
READRANDOM						
1902/BAL	2657/BAL	3070/BAL	3148/BAL	3685/BAL	3701/BAL	5433/BAL
5455-EQU						
READSEQUEN						
2192/BAL	2220/BAL	2525/BAL	2574/BAL	2730/BAL	2847/BAL	2928/BAL
3044/BAL	3072/BAL	3214/BAL	3295/BAL	3458/BAL	5443/BAL	5488-EQU
READTELETYPE						
2075/BAL	2939/BAL	5556-EQU				
READTELETYPE2						
727/BAL	2696/BAL	5550-EQU				
RECSIZE						
334-DATA	520/PZE*	2090/STW	2674/LW	2675/STW	2676/MTW	2693/STW
2954/STW	4660/STW	5405/LW	5415/STW	5417/MTW	5658/STW	5671/STW
5692/LW	5701/CW	5731/LW	5737/STW	5738/MTW	5778/CW	5808/CW
5811/STW	5817/STW	6079/M:WRITE	6111/M:WRITE	6141/M:WRITE		
REL						
4962/M:CLOSE	5067/M:CLOSE					
RELATIVE						
589-DATA	681/CAL1					
REOPEN						
642/BAL	5635-EQU					
REPSEQ						
1211/BAL	1239/BAL	1350/BAL	1405/BAL	1478/BAL	1494/BAL	1638/BAL
1659-LW						
RESTART\$EXECUTIVE						
1820/BNEZ	1828-EQU	1879/B	3471/B			
RESUME\$PARSING						
733-EQU	969/NXTPRM	980/NXTPRM	1010/NXTPRM	1368/NXTPRM	1376/NXTPRM	
1382/NXTPRM	1593/NXTPRM	1633/NXTPRM				
REV						
5520/M:PRECARD						
RNM10						
3149/BCS	3158-BAL					
RNM13						
3152/BCS	3164-BAL					

RP	238-EQU	1744/GEN4	3125/GEN4	4018/DATA		
RP\$FLAG	354-DATA	2375/STW	4635/MTW	5814/MTW		
RPAR	188-EQU 3872/NXTNAM	3855/NXTNAM	3856/NXTNAM	3857/NXTNAM	3870/NXTNAM	3871/NXTNAM
RP3	2375-STW	2379/BE				
RP5	2373/BNE	2378-CW				
RR\$ERR	5460/MISSETDCB	5462/MIREAD	5473-RES			
RS\$ABN	5492/MISSETDCB	5496/MIREAD	5511-RES			
RS\$ABNABN	5520/MIPRECORD	5521/MIREAD	5523-EQU			
RS\$ABNEOM	5531-EQU	5536/B				
RS\$ABNOUT	5534/BE	5537-EQU				
RT\$FPT	527-GEN	5607/CAL1				
RTADDTBL	5606/LW	5625-EQU				
RTSTBTBL	5565/EXU	5577/EXU	5581/EXU	5616-EQU		
RTSTWTBL	5561/EXU	5620-EQU				
RT10	5572/BNEZ	5606-LW				
RT15	5609-LW					
RT17	5584/BE	5604/B	5612-PULL			
RT5	5553/B	5559-LW	5593/BNE	5598/BGE		

RT8	5597-CW	5602/B					
RT9	5600/BNE	5603-AI					
S	66-FNAME 635/CSECT 3856/NXTNAM 4028/DATA	231/EQU 1937/BAL 3857/NXTNAM 4029/DATA	235/EQU 3824/NXTNAM 3870/NXTNAM	417/GEN4 3839/NXTNAM 3871/NXTNAM	417/GEN4 3854/NXTNAM 3872/NXTNAM	417/GEN4 3855/NXTNAM 4017/DATA	4018/DATA
SAVE	605/M;DCB	619/M;DCB	4950/M;CLOSE	4958/M;CLOSE			
SBS10	3649/BLEZ	3656-BE					
SBS15	3652/B	3656/BE	3663-LW				
SC	239-EQU	4255/DATA					
SC0L	186-EQU 1112/NXTPRM 1279/NXTPRM 1382/NXTPRM 1533/NXTPRM	969/NXTPRM 1127/NXTPRM 1308/NXTPRM 1413/NXTPRM 1579/NXTPRM	980/NXTPRM 1183/NXTPRM 1315/NXTPRM 1424/NXTPRM 1593/NXTPRM	1010/NXTPRM 1203/NXTPRM 1361/NXTPRM 1432/NXTPRM 1619/NXTPRM	1072/NXTPRM 1231/NXTPRM 1368/NXTPRM 1438/NXTPRM 1633/NXTPRM	1089/NXTPRM 1264/NXTPRM 1376/NXTPRM 1522/NXTPRM 3825/NXTNAM	
SECT1	75/DEF	260-EQU					
SECT5	76/DEF	637-EQU					
SEQ	180-EQU 1207/NXTPRM 1472/NXTPRM 1576/LI	1098/NXTPRM 1235/NXTPRM 1488/NXTPRM 1608/NXTPRM	1104/LI 1298/NXTPRM 1512/NXTPRM 1617/NXTPRM	1117/NXTPRM 1304/LI 1518/LI 4237/LI	1122/LI 1341/NXTPRM 1526/NXTPRM	1141/LI 1399/NXTPRM 1561/NXTPRM	1570/LI
SEQLIM	217-EQU	329/DATA					
SEQ2	181-EQU 1299/NXTPRM	1099/NXTPRM 1342/NXTPRM	1208/NXTPRM 1356/LI	1212/LI 1400/NXTPRM	1236/NXTPRM 1409/LI	1240/LI 1473/NXTPRM	1482/LI

	1489/NXTPRM	1513/NXTPRM	1562/NXTPRM	1609/NXTPRM	1618/NXTPRM	1641/LI	2487/CI ⁴⁴
SET\$LOOP	4226/LI						
SETADR	1925/BNEZ	3421=EGU					
SETEBD	335=RES	1899/STW	3401/STW	3466/LW			
	1903/BAL	2098/BAL	2728/BAL	2830/BAL	2962/BAL	3223/BAL	3292/BAL
	3305/BAL	3416/BAL	3468/BAL	3507/BAL	3537/BAL	3559/BAL	3576/BAL
	3594/BAL	3615/BAL	3632/BAL	3669/BAL	4634=EGU	5667/BAL	
SETFLAG							
	336=DATA	646/STW	1682/MTW	1761/STW	1819/MTW	1846/MTW	1863/STW
	1894/MTW	1897/STW	1924/MTW	3181/STW	3198/STW	3255/STW	3323/STW
	3389/STW	3411/STW	3422/MTW	3450/STW	4625/STW		
SETKEY							
	5458/BAL	5644=EGU	6070/BAL	6103/BAL	6135/BAL		
SETK2							
	5663/BE	5666/BE	5669=LI				
SETK6							
	5668/B	5673=D81					
SETLASTKEY							
	5466/BAL	5497/BAL	5522/BAL	5652=EGU			
SET10							
	3407/BLE	3416=BAL					
SHIFLEFT							
	3504/BAL	3575/BAL	3659/BAL	4672=EGU			
SHIFTRIGHT							
	3554/BAL	3609/BAL	3631/BAL	3651/BAL	4738=EGU		
SIZE							
	5464/M:READ	5495/M:READ	5521/M:READ	6079/M:WRITE	6111/M:WRITE	6141/M:WRITE	
SL10							
	4696/BE	4705=LI	4719/B	4728/B			
SL20							
	4690/BLEZ	4715=AW					
SL3							
	4682=AW	4725/BL					
SL30							

	4676/BCS	4723=SW		
SL5	4685/BGEZ	4695=CI		
SL5A	4697-LB	4701/BDR		
SPL10	3206/BEZ	3214=BAL		
SPL15	3225/BGZ	3231=BAL		
SPL20	3190/BE	3216/BE	3237=BAL	
SRS10	4641/BNE	4654=SLS		
SRS15	4647/BNE	4651/BNE	4653/B	4658=STW
SRS5	4646-CB	4656/B		
SR10	4778/BEZ	4784=AH	4811/B	
SR12	4777/BGZ	4790=LH	4810/BGZ	
SR12A	4780/B	4792=AND		
SR15	4801-LB	4805/BDR	4886/B	
SR15A	4797/BF	4806=MTW		
SR20	4807/BLZ	4815=LW	4873/B	
SR20A	4816-LI	4893/B		
SR5	4752=BAL	4766/B		
SR50	4753/BCS	4825=BAL		
SR52	4835-AH	4857/B		

SR52A	4837=SH	4866/B					
SR55	4836/BLEZ	4848=LH					
SR58	4853/BEZ	4861=AW					
SR60	4862/BLEZ	4870=AWM					
SR70	4747/BE	4877=BAL					
SR72	4880/BLEZ	4890=AW					
SR8	4760/BLE	4770=SW					
SRRA	4772=LW	4844/B					
STACK	275/DATA	337=RES	350/DATA				
STACKDW	133/PSW	136/PSM	149/PLW	152/PLM	170/MSP	350=DATA	703/STD
STACKSZ	218=EGU	276/DATA	337/RES	351/DATA			
STEP\$LOOP	1921/BNEZ	3204=EGU					
STEPFLAG	338=DATA	647/STW	716/MTW	1680/MTW	1762/STW	1854/MTW	1860/MTW
	1892/MTW	1920/MTW	3180/STW	3199/STW	3224/MTW	3678/MTW	3710/MTW
	4626/STW						
STL10	3426/BNE	3455=BAL					
STL20	3460/BE	3475=BAL					
STL30	3437/BE	3440/BNEZ	3445=EGU				
STL5	3434=LI	3462/BG	3477/B				
STOPCLMN							

STOPLASTCMD	339-RES	4493/STW	4494/CW	4506/CW		
STP10	1755/BE	1787-PULL				
STRG	3197-LI	3210/BEZ	3239/B			
SVBPFLAG	183-EGU 998/CI	741/NXTPRM 1416/NXTPRM	907/LI 1417/LI	918/NXTPRM 4122/LI	928/LI 4400/CI	965/NXTPRM 966/LI
SV1STSET	341-DATA	1814/LW	1996/STW	2004/STW	3469/LW	
TAB	340-RES	1900/LW	3279/STW	3403/STW		
TABCFLAG	2635/M:DEVICE	2636/M:DEVICE	2637/M:DEVICE	2638/M:DEVICE		
TABC10	505-DATA	1822/STW	1865/STW	5696/STW	5729/MTW	5762/MTW
TABC13	5402/BAL	5759-EGU				
TABC15	5770-LB	5774/BL				
TABC17	5767/BNEZ	5772-AI				
TABC18	5771/BEZ	5775-AI	5779/BG	5783/BNE	5802/BE	5812/B
TABC20	5786/BLZ	5789-LW				
TABC25	5790-CB	5794/BG				
TABC30	5791/BNE	5796-AI				
TABC35	5804-LB	5809/BL				
TABC5	5776/BLZ	5814-MTW				
	5818-STB	5821/BL				

TABERRFLAG	5768-PULL	5815/BNEZ	5822/B				
TABEXPAND	504-DATA	2063/STW	2429/STW	5712/MTW	5714/MTW		
TABSET	5674/BAL	5686-EGU					
TABXFLAG	2632/EXU	2635-MIDVICE					
TABX10	506-DATA	1823/STW	1866/STW	2772/MTW	2915/MTW	3275/MTW	5689/MTW
TABX12	5699/BE	5707-LB	5724/BL				
TABX15	5712-MTW						
TABX17	5708/BNEZ	5719-AI					
TABX19	5721/BG	5727-LI					
TABX4	5739-LB	5745/BGE					
TABX5	5697-LI	5735/BEZ	5749/B				
TABX7	5698-CB	5702/BL					
TA5	5704-PULL	5711/BNEZ	5713/BNEZ	5717/B	5725/B		
TC10	2625/BE	2632-EXU					
TC15	5860-LB	5865/BDR	5870/B				
TC25	5864/BLEZ	5875-BAL					
TC5	5855/BGE	5881-BAL					
TEMPBLCK	5859-LI	5873-CAL1	5885/B				
	342-RES	5132/STW	5133/LI	5135/LW	5139/CB	5148/CB	5155/LB

	5169/LB	5948/STW	5952/STB	5954/STB	5961/STB	5962/LI	6027/LI ⁴⁹
	6029/LW	6032/LB	6035/STW	6036/STW	6037/STW	6040/STW	6043/LI
	6055/STB	6057/DATA					
TESTED	ACTIVE						
TEXT	2023/BAL	2131/BAL	2388/BAL	2468/BAL	5830-EQU		
	343-RES	3490/LB*	3491/LB*	4489/STW	4491/LB*	4502/LB*	4519/LB*
	4524/LB*						
TM4							
	6002/BNE	6007-CAL1					
TM5							
	5983/BE	6009/BLZ	6014-PULL				
TPC#FPT							
	517-GEN	2682/CAL1	5873/CAL1				
TPMSG							
	5947/LW	5966-TEXTC					
TP10							
	5901/B	5919-SLS					
TP20							
	5908/BGZ	5947-LW					
TP5							
	5912-LI	5963/B					
TRECSIZE							
	357-DATA	5693/STW	5816/LW				
TSADDR							
	507-DATA	679/STW					
TS10							
	6045/BNE	6049/BNE	6054-LI				
TTYIMG							
	344-RES	2705/LB	2713/CB	3917/LB	3967/LB	4061/LB	4087/LB
	4106/LB	4118/LB	4172/LB	4178/LB	4211/LB	5599/CB	5618/STB
	5622/STW	5627/DATA					
TTYIMGSZ							
	345-RES	729/STW	1922/MTW	3209/MTW	4108/CW		
TXFLAG							
	356-DATA	3449/STW	3752/MTW	3757/STW	4412/MTW	4441/MTW	
TYPE#ALPHA							

963=EGU	1008/B	1044/B	1045/B	1047/B	1048/B	
TYPE\$BETA						
974=EGU	1046/B	1049/B				
TYPE\$I:CMND\$D						
985=EGU	1043/B					
TYPE\$I:CMND\$S						
987/B	990=EGU	1050/B				
TYPECARD						
2831/BAL	3226/BAL	3293/BAL	3306/BAL	3741/BAL	3748/BAL	3755/BAL
5848=EGU						
TYPECERR						
441/BAL	750/BAL	764/BAL	940/BAL	1173/BAL	1908/BAL	3692/BAL
3698/BAL	3718/BAL	3801/BAL	4424/BAL	4468/BAL	4474/BAL	4578/BAL
4686/BAL	4825/BAL	4877/BAL	5893=EGU			
TYPEMSG						
458/BAL	655/BAL	724/BAL	1676/BAL	1692/BAL	1716/BAL	1738/BAL
1748/BAL	1751/BAL	1848/BAL	1914/BAL	2009/BAL	2093/BAL	2103/BAL
2120/BAL	2184/BAL	2232/BAL	2244/BAL	2311/BAL	2322/BAL	2328/BAL
2341/BAL	2397/BAL	2403/BAL	2435/BAL	2444/BAL	2546/BAL	2569/BAL
2601/BAL	2606/BAL	2628/BAL	2738/BAL	2744/BAL	2750/BAL	2756/BAL
2861/BAL	2869/BAL	2875/BAL	2957/BAL	2969/BAL	2971/BAL	3093/BAL
3098/BAL	3104/BAL	3110/BAL	3128/BAL	3132/BAL	3158/BAL	3164/BAL
3237/BAL	3314/BAL	3320/BAL	3408/BAL	3443/BAL	3475/BAL	3495/BAL
4622/BAL	5021/BAL	5035/BAL	5044/BAL	5103/BAL	5587/BAL	5715/BAL
5835/BAL	5867/BAL	5875/BAL	5974=EGU	6011/BAL	6056/BAL	
TYPEPERR						
447/BAL	1001/BAL	1133/BAL	1245/BAL	1320/BAL	1443/BAL	1538/BAL
4142/BAL	4148/BAL	4187/BAL	4231/BAL	5904=EGU		
TYPESEQ						
2070/BAL	2667/BAL	2837/BAL	2934/BAL	3231/BAL	5881/BAL	6024=EGU
TYPM\$FPT						
522=GEN	6007/CAL1					
TYP10						
3284=CW	3296/B					
TYP15						
3287/BLE	3300=BL					
TYP17						

TYP20	3300/BL	3308=AI					
TYP25	3285/BE	3314=BAL					
TYP25A	3309/BLEZ	3320=BAL					
TYP40	3317/BLEZ	3322=LI					
TYP42	3291/BAL	3304/BAL	3327=LW				
TYP45	3330=LB	3336/B					
TYP5	3335/BGE	3337=STW					
TYP50	3269/BNEZ	3272/BL	3274/BE	3276=RES			
TYP55	3328/BEZ	3339=LW					
TYP60	3344=STB	3347/BLE					
TYP65	3342/BGE	3349=LI					
TYP70	3354=B						
TYP72	3352/BG	3357=LI					
TYP75	3360=CB	3368/BL					
TYP80	3362=LB	3378/B					
TYP82	3361/BE	3372=AI					
TYP90	3373=AI	3377/BE					
T1	3310/B	3318/B	3324/B	3379=EQU			
	104=EQU	643/LI	644/STW	645/STW	646/STW	647/STW	648/LI

649/STW	658/LI	659/SLS	666/SW	669/LI	670/AI	671/SLS
672/SW	675/STB	696/LI	697/STW	698/STW	699/STW	700/STW
702/LD	703/STD	704/LI	705/STW	706/STW	707/STW	708/STW
709/STW	710/LI	711/STW	712/LI	713/STW	714/LI	715/STW
734/LB	735/AWM	736/LI	737/STW	745/LW	746/BR	747/CW
762/LI	763/AWM	915/LW	921/XW	926/STW	936/LW	937/CW
955/XW	958/STW	993/LB	994/CI	997/LB	998/CI	1159/LW
1160/CW	1162/LW	1163/STW	1167/LW	1168/CW	1170/CW	1220/LW
1221/CW	1355/LW	1358/CI	1622/LW	1623/AND	1624/STW	1625/LI
1628/LI	1629/LW	1659/LW	1660/STW	1742/LW	1745/AW	1758/LI
1759/STW	1760/STW	1761/STW	1762/STW	1763/LI	1764/STW	1789/LI
1790/STS	1795/LI	1796/STS	1812/LI	1813/STW	1814/LW	1815/STW
1817/LI	1818/STW	1821/LI	1822/STW	1823/STW	1842/LI	1843/CW
1862/LI	1863/STW	1865/STW	1866/STW	1874/LI	1875/STW	1876/STW
1877/LB	1878/AWM	1896/LI	1897/STW	1898/LW	1899/STW	1992/LW
1993/CW	1995/LI	1996/STW	2001/CW	2003/LI	2004/STW	2034/LI
2049/LW	2100/AW	2160/LW	2161/CW	2210/LI	2216/LW	2229/AW
2353/LW	2354/CW	2357/LI	2358/STW	2361/CW	2371/LW	2372/CW
2374/LI	2375/STW	2378/CW	2425/LI	2426/STW	2428/LI	2429/STW
2437/LI	2438/STW	2517/LB	2518/AW	2519/LW*	2520/AI	2521/LW*
2527/STW	2550/LW	2580/LB	2581/AW	2582/LW	2582/LW*	2652/LW
2653/AI	2655/CI	2675/STW	2677/LW	2685/LW	2700/LW	2798/STW
2800/LW	2801/STW	2913/LW	2921/LW	2922/STW	2964/AW	3004/LW
3023/LW	3031/CLM	3035/XW	3037/XW	3046/XW	3048/LW	3122/LW
3126/AW	3127/STB	3147/LW	3150/LW	3197/LI	3198/STW	3199/STW
3322/LI	3323/STW	3388/LI	3389/STW	3410/LI	3411/STW	3434/LI
3450/STW	3464/LI	3465/STW	3466/LW	3467/STW	3469/LW	3470/STW
3523/LB	3524/AW	3525/LI	3531/STB	3646/LW	3663/LW	3666/LB
3667/AW	3712/LI	3713/STW	3728/LW	3729/BR	3730/STW	3811/PUSH
3815/LB	3816/CI	3821/LW	3828/STW	3847/LB	3848/CI	3852/LW
3863/LB	3864/CI	3868/LW	3884/STW	3888/PULL	4387/PUSH	4388/LI
4393/LB	4394/CI	4397/LI	4419/LW	4420/CI	4427/LI	4439/BR
4442/CI	4451/PULL	4457/LI	4458/STW	4462/PULL	4502/LB	4503/CB
4697/LB	4698/STB	4705/LI	4706/STB	4742/LI	4743/STW	4744/STW
4774/LI	4775/STW	4779/LH	4790/LH	4791/AI	4796/CI	4805/BR
4815/LW	4819/BR	4827/LI	4828/XW	4829/SW	4835/AH	4838/AW
4841/SW	4843/STH	4861/AW	4885/LW	4891/LW	4976/PUSH	4977/LI

	5001/AI 5215/LI	5015/AI 5265/LI	5016/LW 5310/STB*	5024/PULL 5312/STB*	5030/LW 5314/LB*	5038/PULL 5373/LI	5062/LI 53
T2	105-EQU 2555/STW 2713/CB 3040/CW 4535/PULL 4850/AND 5306/STW*	922/LI 2556/SW 2722/STB 3055/CW 4801/LB 4851/AWM 5322/STB*	923/XW 2564/AI 2929/LW 4488/PUSH 4802/STB 5063/LI 5335/STB*	927/STW 2685/LW 2965/CW 4496/PURGE 4816/LI 5216/LI 5374/LI	1788/LI 2686/AI 3006/LW 4508/PULL 4817/STB 5266/LI	1794/LI 2690/BDR 3024/LW 4524/LB 4848/LH 5299/STW*	2528/STW 2701/LI 3033/CLM 4525/CB 4849/AW 5302/STW*
UTSM1	534-TEXTC	5588/DATA					
UTSM2	535-TEXTC	725/DATA					
UTSM3	536-TEXTC	2629/DATA					
UTSM4	537-TEXTC	1687/LI	1691/STB	1693/DATA			
UTSM5	539-TEXTC	1711/LI	1715/STB	1717/DATA	1722/LI	1737/STB	1739/DATA
UTSM6	541-TEXTC	1749/DATA					
UTSM7	542-TEXTC	1729/LB	1732/LB				
UTSM8	543-TEXTC	5716/DATA					
WAIT	5463/M:READ	5494/M:READ	6075/M:WRITE	6108/M:WRITE	6138/M:WRITE		
WNR#ABN	6106/M:SETDCB	6112/M:WRITE	6119-RES				
WRITENEWRANDOM	3151/BAL	6101-EQU					
WRITERANDOM	2099/BAL	2729/BAL	2963/BAL	3064/BAL	3084/BAL	3208/BAL	3427/BAL
	3455/BAL	3681/BAL	6133-EQU				
WRITE2	2198/BAL	2223/BAL	2558/BAL	6068-EQU			

W2#ABN	6073/M;SETDCB	6078/M;WRITE	6085-RES				
X: C	286-TEXTC						
X: F	283-TEXTC	2624/CW					
X: INT0	280-GEN4	1221/CW					
X: M	284-TEXTC						
X: 0N	278-TEXTC	1162/LW	1168/CW				
X: 0VER	279-GEN4	1170/CW	2161/CW				
X: S	285-TEXTC						
X: T0	281-TEXTC	1160/CW					
XEQFLAG	508-DATA	700/STW	1678/MTW	1818/STW			
XF	265-DATA	5222/AND	5500/AND				
XFFFF	268-DATA	4763/AND	4792/AND	4840/AND	4850/AND		
XFFFFFF	270-DATA	5436/AND	5502/AND	5526/AND			
XFF00	267-DATA	1623/AND	3786/AND	4289/AND			
XFO	266-DATA	5925/0R					
X1	99-EQU	460/B	529/PZE*	744/LI	747/CW	749/BDR	756/LB
	757/EXU	935/LI	937/CW	939/BDR	947/LB	949/EXU	992/LI
	993/LB	996/LI	997/LB	1697/LI	1698/LB	1700/LW	1702/CW
	1729/LB	1730/AW	1736/BDR	1753/LB	1754/CI	1829/LI	1830/STW
	1831/LI	1832/LB	1845/STW	1884/LI	1885/LB	1990/LI	1991/LB
	2024/LI	2025/LB	2035/LI	2036/LB	2038/AI	2039/LB	2044/LI

2045/LB
2135/LB
2159/LB
2207/LI
2251/LB
2273/AI
2281/LI
2289/SLS
2300/LB
2389/LI
2472/LB
2509/LB
2579/AI
2632/EXU
2680/STB
2766/LI
2910/LI
2947/CB
3008/LB
3145/LI
3183/LB
3268/LW
3337/STW
3349/LI
3391/LB
3494/BDR
3587/AI
3628/LB
3770/STB
3787/AI
3878/AI
3921/LI
3978/BDR
4066/CB
4208/LI
4410/LI
4496/PURGE

2047/AI
2137/LI
2166/LI
2208/LB
2255/LI
2274/AW
2282/LB
2290/AI
2334/LI
2390/LB
2484/AI
2516/AI
2580/LB
2647/LI
2692/STB
2767/LB
2911/LB
2949/BDR
3013/LI
3146/LB
3185/LI
3270/LW
3339/LW
3350/LB
3396/LI
3519/AI
3588/LB
3642/AI
3771/AI
3788/STW
3882/STW
3922/CB
3990/LB
4068/BDR
4392/LI
4417/LI
4501/LI

2048/LB
2138/LB
2167/LB
2211/LI
2256/LB
2275/LB
2284/LB
2291/AW
2335/LB
2419/LI
2485/LB
2517/LB
2618/LI
2648/LB
2699/LI
2794/LI
2916/LI
3000/LI
3014/LB
3174/LI
3254/LI
3271/CI
3340/STW
3357/LI
3398/LW
3520/LB
3591/LB
3643/LB
3773/STB
3789/PULL
3886/STW
3924/BDR
3995/BDR
4100/LB
4393/LB
4418/LB
4502/LB

2081/LI
2148/LB
2174/LI
2212/LB
2267/LI
2276/SLS
2285/SLS
2292/LW
2351/LI
2420/LB
2486/AI
2532/AI
2619/LB
2650/LI
2705/LB
2795/LB
2917/LB
3001/STW
3016/AI
3179/LI
3255/STW
3327/LW
3341/CI
3364/STB
3400/AW
3550/AI
3592/AW
3682/LI
3779/LW
3811/PUSH
3887/BDR
3938/LI
4003/PULL
4127/LI
4396/LI
4431/LI
4508/PULL

2083/CB
2150/CB
2175/LB
2214/AI
2269/LB
2277/AI
2286/AI
2295/LI
2352/LB
2469/LI
2490/LB
2533/LB
2623/LI
2651/LB
2707/AI
2802/LI
2919/AI
3002/LI
3017/LB
3180/STW
3259/LI
3330/LB
3344/STB
3365/AI
3401/STW
3551/LB
3605/AI
3683/LB
3784/BDR
3812/LI
3888/PULL
3939/CB
4008/LB
4131/BDR
4398/LB
4432/LB
4517/LI

2085/BDR
2156/CB
2186/LI
2215/LB
2271/LB
2278/AW
2287/AW
2296/LB
2369/LI
2470/STW
2497/AI
2535/AI
2624/CW
2677/LW
2713/CB
2803/LB
2920/LB
3003/LB
3142/LI
3181/STW
3260/LB
3333/AI
3345/AI
3372/AI
3490/LB
3571/AI
3606/LB
3768/PUSH
3785/LW
3832/AI
3897/AW
3941/BDR
4059/PUSH
4154/LI
4399/LI
4447/AI
4522/AI

2134/LI
2158/LI
2187/LB
2250/LI
2272/SLS
2279/LW
2288/LB
2299/LI
2370/LB
2471/LI
2508/AI
2536/LB
2626/BDR
2678/LB
2715/AI
2805/LI
2945/LI
3007/LI
3143/LB
3182/LI
3265/AI
3334/CW
3346/CI
3390/LI
3491/LB
3572/LB
3627/AI
3769/LW
3786/AND
3842/AI
3915/PUSH
3974/LI
4065/LI
4200/CI
4402/LB
4488/PUSH
4524/LB

4535/PULL
4589/PUSH
4605/AI
4645/LI
4659/AI
4803/AI
4924/BDR
5085/LW
5101/LB
5159/CI
5201/PUSH
5231/PULL
5277/LB
5300/STW
5311/LB
5346/STB
5388/LI
5566/AI
5599/CB
5622/STW
5741/STB
5780/AI
5808/CW
5862/AI
5921/AI
5950/LW
5975/PUSH
5996/CB

4547/PUSH
4592/LB
4606/LB
4646/CB
4660/STW
4808/PULL
4926/PULL
5087/SCS
5102/STB
5167/CW
5207/PUSH
5238/LB
5278/CI
5302/STW
5312/STB
5353/LI
5389/STW
5569/LW
5601/AI
5707/LB
5742/AI
5782/CB
5818/STB
5877/PULL
5922/CB
5953/LB
5977/LW
6005/LI

4549/LB
4594/AI
4619/PULL
4648/BDR
4661/PULL
4820/PULL
5057/PUSH
5088/LB
5136/LI
5169/LB
5220/LW
5239/CI
5280/LI
5303/STW
5320/LB
5354/STB
5391/PULL
5575/LB
5606/LW
5719/AI
5747/LW
5789/LW
5819/AI
5896/PUSH
5924/AI
5956/BDR
5978/SLS
6008/AI

4561/BDR
4595/LB
4637/PUSH
4652/LI
4741/PUSH
4882/LI
5068/PULL
5090/SLS
5139/CB
5172/AI
5221/SLS
5241/PULL
5281/STW
5305/LW
5322/STB
5368/PUSH
5551/PUSH
5576/LB
5612/PULL
5720/CW
5766/LB
5797/AI
5820/CW
5897/LI
5926/STB
5957/LI
5979/LB
6014/PULL

4565/PULL
4598/AI
4638/LI
4654/SLS
4772/LW
4883/SW
5075/LB
5091/LB
5141/AI
5177/LI
5222/AND
5253/PUSH
5283/PULL
5306/STW
5334/LI
5378/PULL
5557/PUSH
5578/BDR
5617/STB
5734/SW
5770/LB
5801/CW
5849/PUSH
5899/LB
5940/BAL
5958/LB
5980/AI

4570/AI
4602/AI
4640/CW
4655/AI
4794/LW
4903/PUSH
5076/CI
5097/LB
5155/LB
5179/LB
5223/CI
5259/PUSH
5298/LW
5307/LW
5335/STB
5385/LB
5560/LI
5595/LI
5618/STB
5736/AW
5777/LB
5804/LB
5859/LI
5905/PUSH
5942/PULL
5960/LI
5981/LB

56
4576/BDR
4603/LB
4642/BDR
4658/STW
4801/LB
4916/LI
5078/PULL
5099/STB
5158/AI
5183/AI
5225/PULL
5270/PULL
5299/STW
5308/STW
5344/LB
5386/CI
5562/BDR
5597/CW
5621/STW
5737/STW
5778/CW
5807/AI
5860/LB
5920/LW
5949/LB
5961/STB
5992/LI

X1FFFF

269=DATA

X2

100=EGU
1702/CW
1834/CI
2039/LB
2088/STB
2352/LB

525/PZE*
1704/BDR
1836/CI
2040/LW
2159/LB
2353/LW

946/LI
1707/CI
1840/CI
2048/LB
2160/LW
2370/LB

948/STB
1731/LI
1871/EXU
2049/LW
2208/LB
2371/LW

1698/LB
1732/LB
1885/LB
2078/LW
2209/LW
2648/LB

1700/LW
1735/AI
1991/LB
2079/AI
2215/LB
2649/LW

1701/LI
1832/LB
1992/LW
2080/LB
2216/LW
2651/LB

2652/LW
2706/STB
2768/LW
2911/LB
2952/STB
3010/AI
3147/LW
3329/LI
3351/CI
3373/AI
3551/LB
3607/AW
3644/AW
3782/STW
3942/CI
3981/AI
4001/LW
4129/STB
4165/LI
4222/CI
4402/LB
4523/AI
4595/LB
4640/CW
4786/STH
4804/AI
4892/LI
5096/SCS
5151/BDR
5306/STW
5314/LB
5322/STB
5339/LB
5552/LI
5606/LW
5743/AI
5799/STB

2679/LI
2708/AI
2769/AI
2912/LW
3003/LB
3011/LW
3183/LB
3331/STB
3359/LI
3374/CW
3552/AW
3608/LB*
3645/LB*
3783/AI
3965/STB
3982/SLS
4073/LI
4130/AI
4185/AI
4227/LI
4403/LW
4525/CB
4596/LW
4646/CB
4790/LH
4808/PULL
5088/LB
5097/LB
5178/LI
5308/STW
5315/AI
5323/AI
5344/LB
5557/PUSH
5612/PULL
5744/CW
5800/AI

2680/STB
2711/CI
2770/LW
2920/LB
3004/LW
3017/LB
3184/LW
3332/AI
3360/CB
3376/CB
3553/LB*
3610/XW
3664/LW
3788/STW
3966/AI
3989/STW
4085/STB
4132/AI
4193/CI
4238/CI
4418/LB
4556/LB
4603/LB
4650/CB
4792/AND
4817/STB
5089/STB
5100/SLS
5184/BDR
5309/LB
5316/SLS
5325/LB
5345/AI
5558/LI
5731/LW
5789/LW
5801/CW

2684/LI
2720/CI
2795/LB
2921/LW
3005/AI
3018/LW
3260/LB
3337/STW
3362/LB
3391/LB
3555/LW
3612/AW
3683/LB
3898/LI
3976/STB
3991/LI
4086/AI
4134/STB
4196/BDR
4242/LI
4419/LW
4557/STB
4606/LB
4661/PULL
4793/AWM
4818/AI
5091/LB
5101/LB
5301/LI
5310/STB
5317/AW
5326/AI
5346/STB
5561/EXU
5732/AI
5790/CB
5805/STB

2687/LB
2722/STB
2796/LW
2942/LW
3006/LW
3143/LB
3261/LW
3343/LI
3363/STB
3392/LW
3572/LB
3628/LB
3684/LW
3899/LW
3977/AI
3992/CB
4101/LI
4135/AI
4204/CI
4387/PUSH
4451/PULL
4572/LI
4607/LW
4773/LW
4794/LW
4871/LW
5093/STB
5137/LI
5302/STW
5311/LB
5318/LB
5327/SLS
5347/AI
5565/EXU
5736/AW
5792/AI
5806/AI

2689/AI
2723/AI
2799/AI
2943/AI
3008/LB
3144/LW
3262/AI
3344/STB
3366/AI
3393/AI
3573/LW
3629/LW
3780/LI
3901/AI
3979/AI
3994/AI
4112/STB
4136/SLS
4209/LI
4398/LB
4462/PULL
4574/CB
4637/PUSH
4779/LH
4795/AW
4872/AI
5094/LB
5142/BDR
5303/STW
5312/STB
5320/LB
5328/AW
5354/STB
5577/EXU
5739/LB
5793/CW
5811/STW

2700/LW
2767/LB
2800/LW
2944/LB
3009/LW
3146/LB
3263/LW
3350/LB
3367/CW
3394/LW
3606/LB
3643/LB
3781/LW
3925/LI
3980/STB
4000/SLS
4113/AI
4155/LI
4217/LI
4400/CI
4518/LW
4592/LB
4639/LW
4784/AH
4802/STB
4884/LI
5095/SLS
5147/LI
5304/LI
5313/BDR
5321/AI
5335/STB
5551/PUSH
5581/EXU
5740/STB
5796/AI
5816/LW

	5817/STW 5922/CB 5955/AI 6001/CB 6043/LI	5820/CW 5928/LB 5975/PUSH 6004/AI 6044/CB	5851/LW 5932/BDR 5979/LB 6014/PULL 6046/STB	5852/AI 5950/LW 5985/BDR 6025/PUSH 6047/AI	5863/AI 5951/AI 5993/LB 6026/LW 6048/CB	5898/LI 5952/STB 5996/CB 6030/LW 6050/STB	58 5912/LI 5954/STB 5998/BDR 6038/LW 6058/PULL
X3	97-EQU 1793/PULL 2678/LB 5094/LB 5704/PULL 5765/LI 5777/LB	674/LI 1796/STS 2692/STB 5098/AI 5707/LB 5766/LB 5784/LW	675/STB 2132/LI 3045/LW 5099/STB 5710/AI 5768/PULL	1674/PUSH 2133/STW 3058/CW 5691/PUSH 5722/AI 5770/LB	1756/PULL 2526/LW 3079/CW 5692/LW 5723/CI 5772/AI	1787/PULL 2572/CW 5092/LI 5693/STW 5748/AI 5773/CI	1790/STS 2583/LW 5093/STB 5694/LI 5764/PUSH 5775/AI
X4	98-EQU 2258/LI 2820/EXU 2995/LI 3302/B 5698/CB 5744/CW 5793/CW	986/LI 2262/LI 2852/EXU 3061/B 5130/PUSH 5700/AI 5747/LW	991/LI 2302/LI 2897/LI 3081/B 5131/LW 5701/CW 5784/LW	1007/B 2306/LI 2909/LI 3248/LI 5179/LB* 5720/CW 5785/AI	2169/LI 2782/LI 2933/B 3253/LI 5186/PULL 5728/STB 5787/LB	2173/LI 2787/LI 2937/FXU 3273/CI 5695/LI 5730/AI 5787/LB	2195/CI 2792/LI 2990/LI 3289/B 5696/STW 5734/SW 5788/AI
X800000	271-DATA	3949/CW					
ZEROISTGIFLG	358-DATA	3439/MTW	3448/STW	3739/MTW	3746/MTW	4411/MTW	
4BLNKS	273-DATA	4639/LW	4936/LW	5298/LW	5559/LW		

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37

```

*****
*M* EDIT IS A FILE BUILDING/MAINTENANCE UTILITY FOR ON-LINE CP-V USERS
*****
*P*      NAME      EDIT
*P*
*P*      PURPOSE:  THE EDIT PROCESSOR IS A FILE MANIPULATION
*P*                  UTILITY AVAILABLE TO ON-LINE CP-V USERS.
*P*                  ITS CAPABILITIES INCLUDE THE BUILDING, DELETING
*P*                  COPYING AND MERGING OF ENTIRE FILES AND EDITING
*P*                  RECORDS WITHIN FILES AS WELL AS EDITING OF DATA
*P*                  WITHIN RECORDS.
*P*
*P*      REFERENCE:  EDIT SUBSYSTEM TECHNICAL MANUAL.  THIS 105
*P*                  PAGE DOCUMENT CONTAINS DETAILED FLOWCHARTS, TABLES
*P*                  AND SUBROUTINE DESCRIPTIONS, AND JUST ABOUT
*P*                  EVERYTHING A SYSTEMS PROGRAMMER NEEDS
*P*                  TO GET INTO EDIT FOR MODIFICATIONS OR DEBUGGING.
*P*
*P*      REFERENCE:  CP-V EDIT REFERENCE CARD (COMMAND STRUCTURE)
*P*                  THIS CARD CONTAINS THE COMMAND STRUCTURES FOR
*P*                  ALL THE EDIT FUNCTIONS AND IS VERY
*P*                  HANDY TO KEEP BY THE USERS TERMINAL.
*P*
*P*      REFERENCE:  CP-V TIME SHARING REFERENCE MANUAL.
*P*                  THIS MANUAL HAS AN EDIT SECTION WHICH SHOWS
*P*                  IN GREAT DETAIL ALL THE EDIT COMMANDS ALONG
*P*                  WITH EXAMPLES OF USAGE.  IT WOULD BE THE USERS
*P*                  BEST INITIAL INTRODUCTION TO EDIT.
*P*
*P*      REFERENCE:  CP-V TIME-SHARING USER'S GUIDE.
*P*                  THIS MANUAL CONTAINS A SECTION ON EDIT
*P*                  SIMILAR TO THE TIME-SHARING REFERENCE MANUAL.
*P*
*P*      DESCRIPTION:  EDIT IS ORGANIZED IN A HIGHLY MODULAR FASHON.
*P*                  UPON ENTRY, 'BEGINEDITOR' PERFORMS INITIALIZATION
*P*                  AFTER WHICH 'MASTERPARSER' CONTROLS INPUT COMMAND

```

38 *P*
 39 *P*
 40 *P*
 41 *P*
 42 *P*
 43 *P*
 44 *P*
 45 *P*
 46 *P*
 47 *P*
 48 *P*
 49 *P*
 50 *P*
 51 *P*
 52 *P*
 53 *P*
 54 *P*
 55 *P*
 56 *P*
 57 *
 58 *
 59 *
 60 00000002 *
 61 *
 62 02 00000 *S*
 63 *S*
 64 00000001 *
 65 *
 66 00000000 S
 67 *
 68 *
 69 *
 70 *
 71 *
 72 *
 73 *
 74 *
 S

SCAN OF A LINE OF USER COMMANDS. FROM A LINE OF INPUT COMMAND(S) THE COMMAND DESCRIPTION TABLE (CDT) IS BUILT. ERROR CHECKS ARE MADE AND WARNINGS GIVEN TO THE USER IF NECESSARY. 'MASTERPARSER' USES A NUMBER OF SUBROUTINES TO BUILD THE CDT: 'GETNAME' AND 'GETNEXTPARAM' TO BREAK DOWN TEXT STRINGS; 'PARSE;I;CMND\$INTG;' TO PROCESS INTEGER STRINGS; 'PARSE;I;CMND\$STRG;' TO PROCESS ALPHABETIC STRINGS IN SLASHES; AND ROUTINES OF THE FORM 'PARSE;CMND;' FOR COMMAND PROCESSING.

ON ENCOUNTERING A CARRIAGE RETURN CHARACTER, CONTROL IS PASSED TO THE 'MASTEREXECUTIVE' ROUTINE TO PERFORM THE COMMANDS WHICH THEN RESIDE IN THE CDT. 'MASTEREXECUTIVE' SERVES AS A DRIVER FOR COMMAND PROCESSING USING 'F;' ROUTINES FOR FILE COMMANDS, 'R;' ROUTINES FOR RECORD COMMANDS AND 'I;' ROUTINES FOR INTRA-RECORD COMMAND PROCESSING.

MODE = 1 FOR BTM VERSION
 = 2 FOR UTS VERSION

MODE	EGU	2
EDITBASE	CSECT	0
	SYSTEM	SIG7FD
	SYSTEM	BPM
S	FNAME	
	PROC	
	LOCAL	A
A	SET	AF(MODE)
	PEND	A
DEF	EDITBASE	DATA AREA FOR EDIT
DEF	BEGINEDITOR	EDITOR START ADDRESS

H01 20:44 SEP 08, 1975

61

75
76
77 00000001
78
79
80
81
82
83
84
85 *S*
86 *S*
87
88 *

DEF SECT1
DEF SECT5
DB MODE#2
DEF PATCH
REF J:CCBUF
REF M:UC
REF M:EI
REF M:EO
REF JB:CCARS
ELSE
DEF F:EI
DEF F:EO
FIN

DATA AREA ADDRESS (FOR GENMDS)
PURE PROCEDURE ADDR (FOR GENMDS)

PATCH AREA (FOR GENMDS)
TTY BUFFER PASSED FROM TEL
DCB FOR USER TERMINAL
EDIT INPUT DCB
EDIT OUTPUT DCB
BYTE COUNT OF STRING IN J:CCBUF

BTM DCB (INPUT)
BTM DCB (OUTPUT)

PAGE

* REGISTER ALLOCATION *

*
*
* REGISTERS 1-13 MUST BE PRESERVED BY ANY SUBR WHICH USES THEM
*

97 00000001
98 00000002
99 00000003
100 00000004
101 00000005
102 00000006
103 00000007
104 00000008
105 00000009
106 0000000A
107 0000000B
108 0000000C
109 0000000D
110 0000000E
111 0000000F

X3 EQU 1
X4 EQU 2
X1 EQU 3
X2 EQU 4
P1 EQU 5
P2 EQU 6
LNK EQU 7
T1 EQU 8
T2 EQU 9
P3 EQU 10
R1 EQU 11
R2 EQU 12
F:LNK EQU 13
R:LNK EQU 13
I:LNK EQU 13

*
* REGISTERS 0,14-15 ARE NEVER SAVED BY SUBRS
*

112
113
114
115 00000000
116 0000000E
117 0000000F

R0 EQU 0
D0 EQU 14
D1 EQU 15

118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154

00000000

00000000

```

PAGE
*****
* SYSTEM PROCEDURES *
*****
*
*
GEN4      COM,8,8,8,8      AF(1),AF(2),AF(3),AF(4)
*
*
PUSH      CNAME
          PROC
          LOCAL      I
LF        EQU        $
I         DB         NUM(AF)
          DB         NUM(AF(I))+1
          PSW,AF(I) STACKDW
          ELSE
          LCI        (AF(I,2)=AF(I,1)+1)&X'F'
          PSM,AF(I,1) STACKDW
          FIN
          FIN
          PEND
*
*
PULL      CNAME
          PROC
          LOCAL      I,K
LF        EQU        $
I         DB         NUM(AF)
K         SET        NUM(AF)=I+1
          DB         NUM(AF(K))+1
          PLW,AF(K) STACKDW
          ELSE
          LCI        (AF(K,2)=AF(K,1)+1)&X'F'
          PLM,AF(K,1) STACKDW
          FIN
          FIN
    
```

H01 20:44 SEP 08, '75

64

```
155                                PEND
156                                *
157                                *
158                                PURGE      CNAME
159                                PROC
160                                LOCAL      I,N
161                                N          SET      0
162                                I          D0      NUM(AF)
163                                D0      NUM(AF(I))=1
164                                N          SET      N+1
165                                ELSE
166                                N          SET      N+((AF(I,2)=AF(I,1))&X'F')+1
167                                FIN
168                                FIN
169                                LF          LI,0      =N
170                                MSP,0      STACKDW
171                                PEND
```

172
 173
 174
 175
 176
 177
 178 00000000
 179 00000001
 180 00000002
 181 00000003
 182 00000004
 183 00000005
 184 00000006
 185 00000007
 186 00000008
 187 00000009
 188 0000000A
 189 0000000B
 190 0000000C
 191
 192
 193 FR
 194 FR
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208

PAGE

 * PARSE PROCEDURES *

 *
 *
 END EQU 0
 NAME EQU 1
 SEQ EQU 2
 SEQ2 EQU 3
 INTG EQU 4
 STRG EQU 5
 ALPH EQU 6
 COM EQU 7
 SCOL EQU 8
 LPAR EQU 9
 RPAR EQU 10
 PERIOD EQU 11
 BLANK EQU 12

UTS FILE SEPARATOR

*
 *
 NXTNAM CNAME GETNEXTNAME
 NXTPRM CNAME GETNEXTPARAM
 PROC
 LOCAL I,N
 N SET NUM(AF)=1
 LF BAL, LNK NAME(1)
 GEN,8,1,23 N, AFA(1), AF(1)
 I DB N
 ERROR,1, NUM(AF(I+1)) =2 'ILGL SYNTAX'
 DB AFA(I+1,2)=1
 GEN,8,24 AF(I+1,1), *N=I+AF(I+1,2)+1
 FLSE
 GEN,8,24 AF(I+1,1), AF(I+1,2)
 FIN
 FIN
 PEND

209			PAGE		
210			*****		
211			* ADJUSTABLE PARAMETERS *		
212			*****		
213			*		
214			*		
215	000003E8	DFLTSEQ EQU	1000	DEFAULT STARTING SEQ. #	
216	0000008C	MAXCLMN EQU	140		
217	0098967F	SEQLIM EQU	9999999	FOR MAX. SEQ. NO.	
218	00000070	STACKSZ EQU	125	SIZE OF TEMP STACK	
219		*			
220	00000004	FIRST\$F:CMND EQU	4		
221	0000001E	FIRST\$I:CMND EQU	30		
222	0000000B	FIRST\$R:CMND EQU	11		
223	0000002A	I:TS\$CMND\$NMR EQU	42		
224	0000002B	I:TY\$CMND\$NMR EQU	43		
225	00000015	R:TS\$CMND\$NMR EQU	21		
226	00000016	R:TY\$CMND\$NMR EQU	22		
227		*			
228		*			
229	TEXT	BL EQU	' '		
230	TEXT	CM EQU	'.'		
231	00000000	CR EQU	S(X'15',X'0D')		
232	00989680	EOF EQU	10000000		
233	00000008	EOM EQU	X'08'		
234	0000000C	FF EQU	X'0C'		
235	00000015	LF EQU	S(X'25',X'15')		
236	TEXT	PR EQU	'.'		
237	TEXT	LP EQU	'('		
238	TEXT	RP EQU	')'		
239	TEXT	SC EQU	' '		

240
241
242
243
244
245
246
247
248
249
250
251
252
253

00000000

PAGE

* EDIT/BTM INTERFACE CONTROL BLOCK *

*
*

S DB MODE=1
S BRG EDITBASE
S DATA EDIT\$TCB
S DATA,1A 0,0
S DATA BEGINEDITBR
S DB1 X'40'-10
S DATA 0
FIN

			PAGE		
254					
255			*****		
256			* CONSTANT DATA *		
257			*****		
258			*		
259			*		
260		02 0000U	SECT1	EQU	\$
261	02	00000	K1	DATA	1
262	02	00001	K10	DATA	10
263	02	00002	KPE	DATA	1.1
264			*		
265	02	00003	XF	DATA	X'F'
266	02	00004	XFO	DATA	X'FO'
267	02	00005	XFF0C	DATA	X'FF00'
268	02	00006	XFFFF	DATA	X'FFFF'
269	02	00007	X1FFFF	DATA	X'1FFFF'
270	02	00008	XFFFFFFF	DATA	X'FFFFFF'
271	02	00009	X800000	DATA	X'800000'
272			*		
273	02	0000A	4BLNKS	DATA	1 1
274				BOUND	8
275	02	0000C	DMYSTKDw	DATA	STACK
276	02	0000D		DATA,2	STACKSZ,0
	02	0000D 2			
277	02	0000E	HEXCHAR	TEXT	'0123456789ABCDEF'
	02	0000F			
	02	00010			
	02	00011			
278	02	00012	X:BN	TEXTC	'BN'
279	02	00013	X:OVER	GEN4	4,'B','V','E'
280	02	00014	X:INTB	GEN4	4,'I','N','T'
281	02	00015	X:TB	TEXTC	'TB'
282		00000001		D0	MODE=2
283	02	00016	X:F	TEXTC	'F'
284	02	00017	X:M	TEXTC	'M'
285	02	00018	X:S	TEXTC	'S'
286	02	00019	X:C	TEXTC	'C'

H01 20:44 SEP 08, '75

287
 288
 289
 290
 291
 292
 293 02 0001A 000000F0 A
 02 0001B 000000F9 A
 294 02 0001C 000000C1 A
 02 0001D 000000E9 A
 295 00000001
 296 02 0001E 00000081 A
 02 0001F 000000A9 A
 297

*
 * FIN
 *
 * SPECIAL LIMITS
 *
 BOUND 8
 DIGITS DATA '0','9'
 LETTERS DATA 'A','Z'
 DB MODE=2.
 LCLETTERS DATA 'X','9'
 FIN

				PAGE		
298						
299					*****	
300					* VARIABLE DATA *	
301					*****	
302					*	
303					*	
304	02 00020	FFFFFFFF	A	ALLFLAG	DATA	=1 GLOBAL: >=0 IF ALL USED ON I:CMND
305	02 00021	00000000	A	ALLOK	DATA	0 GLOBAL: =0 IF 'ALL' IS OK.
306	02 00022			BLANKCNT	RES	1 SHFTRGHT: # OF BLANKS TO COMPRESS
307	02 00023	00000000	A	BPFLAG	DATA	0 GLOBAL: BLANK PRESERVATION FLAG, ON=1
308	02 00024			CARDIMG	RES	MAXCLMN/4+1 GLOBAL: HOLDS ACTIVE CARD IMAGE.
309	02 00048			CDT	RES	100 GLOBAL: COMMAND DESCRIPTION TABLE
310	02 000AC			CDTADR	RES	1 GLOBAL: ADR OF CURRENT CMND IN CDT
311	02 000AD			CHARPSN	RES	1 PARSER: PSN OF NEXT CHAR TO SCAN
312	02 000AE	00000000	A	COPYFL	DATA	0 F: COPY = FID1=FID2 IF 1
313	02 000AF	00000001	A	CRFLAG	DATA	1 GLOBAL: 0= INCLUDE TERM. IN OUTPUT
314	02 000B0	000003E8	A	DFLTINCR	DATA	1000 GLOBAL: DEFAULT VALUE FOR INCREMENT
315	02 000B1			ENDCLMN	RES	1 GLOBAL: COL. # OF LAST NON-BLANK
316	02 000B2			ERRRCNT	RES	1 GLOBAL: # OF ERROR MSGS TO PRINT
317	02 000B3	00000000	A	FID1ADR	DATA	
318	02 000B4	00000000	A	FID2ADR	DATA	
319	02 000B5			FIELD CNT	RES	1 SHFTRGHT: # OF FIELDS TO COMPRESS
320	02 000B6	FFFFFFFF	A	FILETYPE	DATA	=1 GLOBAL: SPECIFIES TYPE OF INP FILE
321	02 000B7			FRSTCLMN	RES	1 FINDMATCH: FIRST COL. TO START AT
322	02 000B8	00000000	A	FIRSTFROM	DATA	
323	02 000B9			FIRSTSET	RES	1 GLOBAL: FIRST SEQ. # FOR SET CMND
324	02 000BA			KBUF	RES	1 I/O: HOLDS KEY FOR CURRENT I/O
325	02 000BB			LASTCLMN	RES	1 FINDMATCH: LAST COL. TO STOP IN
326	02 000BC			LASTFROM	RES	1 F: MOVE: LAST IFROM: SEQ # READ
327	02 000BD	00000000	A	LASTKEY	DATA	0 I/O: HOLDS LAST READ KEY
328	02 000BE			LASTSET	RES	1 GLOBAL: LAST SEQ. # FOR SET CMND
329	02 000BF	0098967F	A	MAXSEQ	DATA	SEQLIM GLOBAL: MAX. SEQ. NO. ALLOWED
330	02 000C0	00000000	A	NBCHGFLG	DATA	0 GLOBAL: ON(1) IF NO CHANGE CMND READ
331	02 000C1			PARAMBUF	RES	MAXCLMN/4+1
332	02 000E5			PARAMPSN	RES	1 ADD&NEWCDT=: PSN OF NXT PARAM IN CDT
333	02 000E6			PRMBUFSZ	RES	1 PARSER: # OF WORDS IN PARAMBUF
334	02 000E7	0000008C	A	RECSIZE	DATA	140 GLOBAL: OUTPUT RECORD SIZE.

H01 20:44 SEP 08, 1975

335	02	000E8		SETADR	RES	1	GLOBAL; ADR OF LAST SET CMND IN CDT
336	02	000E9	00000000	SETFLAG	DATA	0	GLOBAL; 0N(1) IF SET CMND ACTIVE
337	02	000EA		STACK	RES	STACKSZ	GLOBAL; STACK USED FOR PUSH/PULL
338	02	00167	00000000	STEPFLAG	DATA	0	GLOBAL; 0N(1) IF STEP CMND ACTIVE
339	02	00168		STOPCLMN	RES	1	FINDMTC; COL. # TO STOP MATCHING AT
340	02	00169		SV1STSET	RES	1	GLOBAL; INITIAL 1ST SEQ # FOR SET
341	02	0016A	00000000	SVBPFLAG	DATA	0	GLOBAL; HOLDS DFLT VALUE OF BPFLAG
342	02	0016B		TEMPBLCK	RES	10	GLOBAL; HOLDS EBCDIC TEXT FOR TYPMSG
343	02	00175		TEXTCADR	RES	1	FINDMTC; ADR OF TEXTC_STRG TO MATCH
344	02	00176		TTYIMG	RES	MAXCLMN/4+1	GLOBAL; HOLDS TELETYPE INPUT IMAGE.
345	02	0019A		TTYIMGSZ	RES	1	GLOBAL; HOLDS SIZE OF TELETYPE IMAGE
346				*			
347	02	0019B	00000000	DELNXT	DATA	0	DELETE TEMP
348				*			
349					BOUND	8	
350	02	0019C	000000EA	STACKDW	DATA	STACK	GLOBAL; DW FOR HARDWARE PSW/PLW
351	02	0019D	007D		DATA,2	STACKSZ,0	
	02	0019D	2 0000				
352	02	0019E	00000000	MVD;REC;CNT	DATA	0	COUNT OF RECORDS MOVED
353	02	0019F	00000000	CHG;STG;CNT	DATA	0	COUNT OF STRINGS CHANGED
354	02	001A0	00000001	RP#FLAG	DATA	1	RECORD SIZE PRESERVATION (OFF)
355	02	001A1	00000000	CT#FLAG	DATA	0	TYPE BEFORE ICM; INPUT
356	02	001A2	00000000	TXFLAG	DATA	0	TX RECORD CHANGED FLAG
357	02	001A3	00000000	TRECSIZE	DATA	0	TAB RECORD SIZE (FOR RP=0N)
358	02	001A4	00000000	ZER0;STG;FLG	DATA	0	FLAG TO INDICATE ZER0 STRINGS

71

359
 360
 361
 362
 363
 364
 365 02 001A5 0D6060C3 A
 02 001A6 F17AD6E5 A
 02 001A7 C5D9C6D3 A
 02 001A8 D6E64040 A
 366 02 001A9 0E6060C3 A
 02 001AA F17AF4D5 A
 02 001AB C4C5D9C6 A
 02 001AC D3D6F640 A
 367 02 001AD 0F60C3F1 A
 02 001AE 7AD5D640 A
 02 001AF E2E4C3C8 A
 02 001B0 4UD9C5C3 A
 368 02 001B1 1260C3F1 A
 02 001B2 7AC3D4D5 A
 02 001B3 C440C9D3 A
 02 001B4 C7D340C8 A
 02 001B5 C5D9C540 A
 369 02 001B6 116060C3 A
 02 001B7 F17AD5D6 A
 02 001B8 40E2E4C3 A
 02 001B9 C840E2E3 A
 02 001BA D9C74040 A
 370 02 001BB 0E6060C3 A
 02 001BC F17AC3D6 A
 02 001BD D36ED3C9 A
 02 001BE D4C9E340 A
 371 02 001BF 126060C3 A
 02 001C0 F17A7DC1 A
 02 001C1 D3D37D40 A
 02 001C2 C9C7D5D6 A
 02 001C3 D9C5C440 A

PAGE

 * ERROR MESSAGES *

*

*

ERRC1 TEXTC I=C1:OVERFLOW!

ERRC2 TEXTC I=C1:UNDERFLOW!

ERRC3 TEXTC I=C1:NO SUCH REC!

ERRC4 TEXTC I=C1:CMND ILGL HERE!

ERRC5 TEXTC I=C1:NO SUCH STRG!

ERRC6 TEXTC I=C1:COL>LIMIT!

ERRC7 TEXTC I=C1:!!ALL!! IGNORED!

20144 SEP 08, 1975

372	02 001C4	0V60C3F1 A	ERRC8	TEXTC	!-C1;UNKN CMND!
	02 001C5	7AE4D5D2 A			
	02 001C6	D540C3D4 A			
	02 001C7	D5C44040 A			
373	02 001C8	0V60C3F1 A	ERRC9	TEXTC	!-C1;ILGL SYNTAX!
	02 001C9	7AC9D3C7 A			
	02 001CA	D340E2E8 A			
	02 001CB	D5E3C1E7 A			
374	02 001CC	0E6060C3 A	ERRC10	TEXTC	!-C1;CBL<LIMIT!
	02 001CD	F17AC3D6 A			
	02 001CE	D34CD3C9 A			
	02 001CF	D4C9F340 A			
375	02 001D0	1260C2C1 A	ERRC11	TEXTC	!-BAD CBL. NO. PAIR!
	02 001D1	C440C3D6 A			
	02 001D2	D34B40D5 A			
	02 001D3	D64B40D7 A			
	02 001D4	C1C9D940 A			
376			*		
377			*		
378	02 001D5	186060C5 A	ERRM1	TEXTC	!-EOF HIT AFTER YYYY.YYY!
	02 001D6	D6C640C8 A			
	02 001D7	C9E340C1 A			
	02 001D8	C6E3C5D9 A			
	02 001D9	4UE8F8E8 A			
	02 001DA	E84BE8E8 A			
	02 001DB	E8404040 A			
379	02 001DC	0A6060D6 A	ERRM3	TEXTC	!-OVERFLOW!
	02 001DD	E5C5D9C6 A			
	02 001DE	D3D6E640 A			
380	02 001DF	0E60D9D5 A	ERRM4	TEXTC	!-RNG OVERLAP!
	02 001E0	C740D6E5 A			
	02 001E1	C5D9D3C1 A			
	02 001E2	D7404040 A			
381	02 001E3	0E60D5D6 A	ERRM5	TEXTC	!-NOT ON/OFF!
	02 001E4	E340D6D5 A			
	02 001E5	61D6C6C6 A			
382	02 001E6	066060D5 A	ERRM6	TEXTC	!-NONE!

MO1 20144 SEP 08, 1975

74

	02	001E7	D6D5C540	A			
383	02	001E8	0B60D4C9	A	ERRM8	TEXTC	!-MISSING SE!
	02	001E9	E2E2C9D5	A			
	02	001EA	C740F2C5	A			
384	02	001EB	1A60C6C9	A	ERRM12	TEXTC	!-FILE NOT KEYED; MUST COPY!
	02	001EC	D3C540D5	A			
	02	001ED	D6E340D2	A			
	02	001EE	C5E8C5C4	A			
	02	001EF	5E40D4E4	A			
	02	001F0	E2E340C3	A			
	02	001F1	D6D7E840	A			
385	02	001F2	0E60D5D6	A	ERRM13	TEXTC	!-NO FILE NAMED!
	02	001F3	40C6C9D3	A			
	02	001F4	C540D5C1	A			
	02	001F5	D4C5C440	A			
386	02	001F6	0D60D5D6	A	ERRM14	TEXTC	!-NO SUCH FILE!
	02	001F7	40E2E4C3	A			
	02	001F8	C840C6C9	A			
	02	001F9	D3C54040	A			
387	02	001FA	1960C6C9	A	ERRM15	TEXTC	!-FILE EXISTS; CAN'T BUILD!
	02	001FB	D3C540C5	A			
	02	001FC	E7C9E2E3	A			
	02	001FD	E25E40C3	A			
	02	001FE	C1D57DE3	A			
	02	001FF	40C2E4C9	A			
	02	00200	D3C44040	A			
388					*		
389	02	00201	1060D5D6	A	ERRM16	TEXTC	!-NOTHING TO MOVE!
	02	00202	E3C8C9D5	A			
	02	00203	C740E3D6	A			
	02	00204	40D4D6E5	A			
	02	00205	C5404040	A			
390	02	00206	1760D4C5	A	ERRM17	TEXTC	!-MERGE SOURCE NOT KEYED!
	02	00207	D9C7C540	A			
	02	00208	E2D6E4D9	A			
	02	00209	C3C540D5	A			
	02	0020A	D6E340D2	A			

H01 20144 SEP 08, 1975

75

391	02 0020B	C5E8C5C4 A			
	02 0020C	1C60D4C5 A	ERRM18	TEXTC	!-MERGE DESTINATION NOT KEYED!
	02 0020D	D9C7C540 A			
	02 0020E	C4C5F2E3 A			
	02 0020F	C9D5C1E3 A			
	02 00210	C9D6D540 A			
	02 00211	D5D6E340 A			
	02 00212	D2C5E8C5 A			
	02 00213	C4404040 A			
392	02 00214	2360F2D6 A	ERRM19	TEXTC	!-SORRY... NO PASSWORD ALLOWED HERE.!
	02 00215	D9D9F84B A			
	02 00216	4B4B40D5 A			
	02 00217	D640D7C1 A			
	02 00218	E2E2E6D6 A			
	02 00219	D9C440C1 A			
	02 0021A	D3D3D6E6 A			
	02 0021B	C5C440C8 A			
	02 0021C	C5D9C54B A			
393	02 0021D	1760D4C1 A	ERRM20	TEXTC	!-MAX. SEQ. NO. EXCEEDED!
	02 0021E	E74B40E2 A			
	02 0021F	C5D84B40 A			
	02 00220	D5D64B40 A			
	02 00221	C5E7C3C5 A			
	02 00222	C5C4C5C4 A			
394	02 00223	2960C9C1 A	ERRM21	TEXTC	!-CAN NOT DELETE ALL OCCURRENCES OF BLANKS.!
	02 00224	D540D5D6 A			
	02 00225	E340C4C5 A			
	02 00226	D3C5F3C5 A			
	02 00227	4UC1D3D3 A			
	02 00228	4UD6C3C3 A			
	02 00229	D9D9C5D5 A			
	02 0022A	C3C5E240 A			
	02 0022B	D6C640C2 A			
	02 0022C	D3C1D5D2 A			
	02 0022D	E24B4040 A			
395			*		
396	02 0022E	0F60D7F1 A	ERRP1	TEXTC	!-P1!NO SUCH REC!

	02	0022F	7AD5D640	A			
	02	00230	E2E4C3C8	A			
	02	00231	40D9C5C3	A			
397	02	00232	0E60D7F2	A	ERRP2	TEXTC	I-P2;REC EXISTS!
	02	00233	7AD9C5C3	A			
	02	00234	40C5F7C9	A			
	02	00235	E2E3E240	A			
398	02	00236	0E60D7F1	A	ERRP3	TEXTC	I-P1;BAD FID!
	02	00237	7AC2C1C4	A			
	02	00238	40C6C9C4	A			
399	02	00239	0E60D7F1	A	ERRP4	TEXTC	I-P1;ILGL SYNTAX!
	02	0023A	7AC9D3C7	A			
	02	0023B	D340E2E8	A			
	02	0023C	D5E3C1E7	A			
400	02	0023D	0E60D7F1	A	ERRP5	TEXTC	I-P1;NOT SEQ #!
	02	0023E	7AD5D6E3	A			
	02	0023F	40E2C5D8	A			
	02	00240	4U7B4040	A			
401	02	00241	0E60D7F1	A	ERRP6	TEXTC	I-P1;NOT INCR!
	02	00242	7AD5D6E3	A			
	02	00243	40C9D5C3	A			
	02	00244	D9404040	A			
402	02	00245	0E60D7F1	A	ERRP7	TEXTC	I-P1;NOT COL #!
	02	00246	7AD5D6E3	A			
	02	00247	40C3D6D3	A			
	02	00248	4U7B4040	A			
403	02	00249	0E60D7F1	A	ERRP8	TEXTC	I-P1;NOT STRG!
	02	0024A	7AD5D6E3	A			
	02	0024B	40E2F3D9	A			
	02	0024C	C7404040	A			
404	02	0024D	0E60D7F1	A	ERRP9	TEXTC	I-P1;NOT CNT!
	02	0024E	7AD5D6E3	A			
	02	0024F	40C3D5E3	A			
405	02	00250	0E60D7F1	A	ERRP10	TEXTC	I-P1;ILGL SEQ #!
	02	00251	7AC9D3C7	A			
	02	00252	D340E2C5	A			
	02	00253	D8407B40	A			

```

HO1  20:44  SEP 08, 1975
406   02 00254  0U60D7F1 A  ERRP11  TEXTC  1-P1;SEQ2<SEQ1
      02 00255  7AE2C5D8 A
      02 00256  F24CF2C5 A
      02 00257  D8F14040 A
407   02 00258  1U60D7F1 A  ERRP12  TEXTC  1-P1;NO SUCH FILE
      02 00259  7AD5D640 A
      02 0025A  E2E4C3C8 A
      02 0025B  4UC6C9D3 A
      02 0025C  C5404040 A
408   02 0025D  0P60D7F2 A  ERRP13  TEXTC  1-P2;FILE EXISTS
      02 0025E  7AC6C9D3 A
      02 0025F  C540C5E7 A
      02 00260  C9E2F3E2 A
409   02 00261  0U60D7F2 A  ERRP14  TEXTC  1-P2;COL ERROR
      02 00262  7AC3D6D3 A
      02 00263  4UC5D9D9 A
      02 00264  D6D94040 A
410   02 00261  ERRP14A  FGU     ERRP14
411   02 00265  0U60D7F1 A  ERRP15  TEXTC  1-P1;ILGL STRG
      02 00266  7AC9D3C7 A
      02 00267  D340E2E3 A
      02 00268  D9C74040 A
412   02 00269  1U60D7F1 A  ERRP16  TEXTC  1-P1;FILE NOT KEYED & P3 NULL
      02 0026A  7AC6C9D3 A
      02 0026B  C540D5D6 A
      02 0026C  E340D2C5 A
      02 0026D  E8C5C440 A
      02 0026E  5U40D7F3 A
      02 0026F  4UD5F4D3 A
      02 00270  D3404040 A
413   02 00271  1160D7F1 A  ERRP17  TEXTC  1-P1;PARAM MISSING
      02 00272  7AD7C1D9 A
      02 00273  C1D440D4 A
      02 00274  C9E2F2C9 A
      02 00275  D5C74040 A
414   02 00276  0U60D7F1 A  ERRP18  TEXTC  1-P1;NULL STRG
      02 00277  7AD5F4D3 A

```

H01 20:44 SEP 08, 1975

78

	02	00278	D340E2E3	A			
	02	00279	D9C74040	A			
415					*		
416					*		
417	02	0027A	03150800	A	MSG0	GEN4	3,S(CR,LF),S(LF,EOM),S(EOM,0)
418	02	0027B	094B4BC3	A	MSG1	TEXTC	!..COPYING!
	02	0027C	D6D7E8C9	A			
	02	0027D	D5C74040	A			
419	02	0027E	0B4B4BC3	A	MSG2	TEXTC	!..COPY DONE!
	02	0027F	D6D7E840	A			
	02	00280	C4D6D5C5	A			
420	02	00281	094B4BC4	A	MSG3	TEXTC	!..DELETED!
	02	00282	C5D3C5E3	A			
	02	00283	C5C44040	A			
421	02	00284	0E4B4BC5	A	MSG4	TEXTC	!..EDIT STOPPED!
	02	00285	C4C9E340	A			
	02	00286	E2E3D6D7	A			
	02	00287	D7C5C440	A			
422	02	00288	0F4B4BD4	A	MSG5	TEXTC	!..MERGE STARTED!
	02	00289	C5D9C7C5	A			
	02	0028A	40E2E3C1	A			
	02	0028B	D9E3C5C4	A			
423	02	0028C	18404040	A	MSG6	TEXTC	! RECORDS DELETED!
	02	0028D	40404040	A			
	02	0028E	4040D9C5	A			
	02	0028F	C3D6D9C4	A			
	02	00290	E240C4C5	A			
	02	00291	D3C5E3C5	A			
	02	00292	C4404040	A			
424	02	00293	1640F0F0	A	MSG7	TEXTC	! 0000000 RECORDS MOVED!
	02	00294	F0F0F0F0	A			
	02	00295	F040D9C5	A			
	02	00296	C3D6D9C4	A			
	02	00297	E240D4D6	A			
	02	00298	E5C5C440	A			
425	02	00299	18404040	A	MSG8	TEXTC	! STRINGS CHANGED!
	02	0029A	40404040	A			

H01 20:44 SEP 08, 1975

02 0029B 4040F2E3 A
 02 0029C D9C9D5C7 A
 02 0029D E240C3C8 A
 02 0029E C1D5C7C5 A
 02 0029F C4404040 A

426

427

428 02 002A0 1060C2C1 A

429 02 002A1 C440C961 A

02 002A2 D65E40C1 A

02 002A3 C2D540C3 A

02 002A4 D6C4C540 A

430 02 002A5 40404040 A

02 002A6 40404040 A

431

432

433 02 002A7 0A6060C4 A

02 002A8 D6D5C540 A

02 002A9 C1E34040 A

434 02 002AA

435 02 002AD 046060C3 A

02 002AE E4E3D6C6 A

02 002AF C640C1E3 A

02 002B0 40404040 A

436 02 002B1

437

438

439

440 02 002B6

441 02 002B6 6A700D8F 05

442 02 002B7 00000000 A

443 02 002B8 68000004 05

444

445

446 02 002B9

447 02 002B9 6A700D99 05

448 02 002BA 00000000 A

```

*
*
IBERRMSG DATA      X'0060C2C1'+(((IBERRC0D+2*IBERRMSG)*4=1)**24
TEXT                'D I/8; ABN C0DE'

IBERRC0D TEXT      1          1          ABN C0DE=SUBC0DE PUT HERE

*
*
MVEMSG1 TEXTC      1=-DONE AT 1

RES                3
MVEMSG2 TEXTC      1=-CUTOFF AT 1

RES                5

*
* DUMMY CALLS FOR TYPECERR AND TYPEPERR
*
DMY$TYPECERR      EQU $
                   BAL,LNK TYPECERR      TYPE ERRCN
                   DATA      0
                   B          MASTERPARSER GO TO PARSE

*
*
DMY$TYPEPERR      EQU $
                   BAL,LNK TYPEPERR      TYPE ERRPN
                   DATA      0

```

HC1

20144 SEP 08, 175

449		00000001		D0	MODE=2	
450	02 0028B	330002C1		MTW,0	BUILDFLAG	WAS ENTRY FROM TEL (B)
451	02 0028C	69300004	05	BNEZ	MASTERPARSER	NO
452	02 0028D	680004BA	05	B	FIN	YES, EXIT
453				ELSE		
454			*S*	B	MASTERPARSER	CONTINUE
455				FIN		
456			*			
457			*			
458	02 0028E	6A700DC4	05	DMY\$TPM	BAL, LNK	TYPE MSG
459	02 0028F	00000000	A		DATA	0
460	02 002C0	68060000	A		B	0, X1
461			*			RETURN

462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498

00000000

```

PAGE
*****
* EDIT TASK CONTROL BLOCK, ETC. *
*****
*
*
*          D0          MODE=1
*S*          BBUND          8
*S* EDIT$TCB EQU          $
*S*          DATA          EDIT$TSTK
*S*          DATA,2          20,0
*S*          DATA,16         0
*S*          DATA,16         0
*S*          DATA          EDIT$DCBT
*S*          DATA,16         0
*S*          DATA          0
*S* *
*S* * USER'S TEMPORARY STACK
*S* *
*S* EDIT$TSTK          EQU $
*S*          D01          5
*S*          DATA,16         0
*S* *
*S* * DCB NAME TABLE
*S* *
*S* EDIT$DCBT          EQU $
*S*          DATA          LINK1
*S*          TEXTC          'F:EI'
*S*          DATA          F:EI
*S*          TEXTC          'F:EO'
*S*          DATA          F:EO
*S* LINK1          DATA          0
*S*          ELSE
*
*****
*          UTS INTERFACE PARAMETERS AND MESSAGES. *
*****

```

```

499
500 02 002C1 00000000 A BUILDFLAG DATA 0 IF ZERO, ENTERED BY BUILD
501 02 002C2 00000000 A CFLAG DATA
502 02 002C3 FFFFFFFF A INTFLAG1 DATA =1 /INTERRUPT SEQ INDICATORS FOR
503 02 002C4 FFFFFFFF A INTFLAG2 DATA =1 /THOSE COMMANDS WHICH DISPLAY.
504 02 002C5 00000000 A TABERRFLAG DATA
505 02 002C6 00000001 A TABCFLAG DATA 1
506 02 002C7 00000001 A TABXFLAG DATA 1
507 02 002C8 00000000 A TSADDR DATA 0 TEMP STACK ADDRESS
508 02 002C9 FFFFFFFF A XEQFLAG DATA =1 MINUS ONE IF NOT IN EXECUTION.
509 02 002CA PROMPT$FPT EQU $
510 02 002CA 2000005C A GEN,8,24 X'2C',1*1
511 02 002CB PROMPT2$FPT EQU $
512 02 002CB 2000004B A GEN,8,24 X'2C',1.1
513 02 002CC N$PROMPT$FPT EQU $
514 02 002CC 20000000 A GEN,8,24 X'2C',0
515 02 002CD 10000000 N BR$FPT GEN,8,24 X'10',M:UC
516 02 002CE 34000010 A DATA X'34000010',CFLAG,1,0
02 002CF 000002C2
02 002D0 00000001 A
02 002D1 00000000 A
517 02 002D2 11000000 N TPC$FPT GEN,8,24 X'11',M:UC
518 02 002D3 34000010 A DATA X'34000010'
519 02 002D4 00000024 DATA CARDIMG
520 02 002D5 800000E7 PZE *RECSIZE
521 02 002D6 00000000 A DATA 0
522 02 002D7 11000000 N TYPM$FPT GEN,8,24 X'11',M:UC
523 02 002D8 34000010 A DATA X'34000010'
524 02 002D9 80000007 A PZE *LNK
525 02 002DA 80000004 A PZE *X2
526 02 002DB 00000001 A DATA 1
527 02 002DC 10000000 N RT$FPT GEN,8,24 X'10',M:UC
528 02 002DD 34000010 A DATA X'34000010'
529 02 002DE 80000003 A PZE *X1
530 02 002DF 0000008C A DATA MAXCLMN
531 02 002E0 00000000 A DATA 0
532

```

```

533
534 02 002E1 09C5C4C9 A *
      02 002E2 E34CC8C5 A UTSM1 TEXTC 'EDIT HERE'
      02 002E3 D9C54040 A
535 02 002E4 025C0840 A UTSM2 TEXTC 'H' * + EBM
536 02 002E5 0C60D5D6 A UTSM3 TEXTC 'NOT F/M/S/C'
      02 002E6 E34CC661 A
      02 002E7 D461F261 A
      02 002E8 C3404040 A
537 02 002E9 246060C9 A UTSM4 TEXTC '==INTRA-RECORD COMMAND INTERRUPT AT '
      02 002EA D5E3D9C1 A
      02 002EB 60D9C5C3 A
      02 002EC D6D9C440 A
      02 002ED C3D6D4D4 A
      02 002EE C1D5C440 A
      02 002EF C9D5F3C5 A
      02 002F0 D9D9F4D7 A
      02 002F1 E340C1E3 A
      02 002F2 40404040 A
538 02 002F3 RES 3
539 02 002F6 176C60C3 A UTSM5 TEXTC '==COMMAND INTERRUPT AT '
      02 002F7 D6D4D4C1 A
      02 002F8 D5C440C9 A
      02 002F9 D5E3C5D9 A
      02 002FA D9E4D7E3 A
      02 002FB 4UC1E340 A
540 02 002FC RES 7
541 02 00303 0E606040 A UTSM6 TEXTC '== X TO ABORT.'
      02 00304 E740F3D6 A
      02 00305 4UC1C2D6 A
      02 00306 D9E34B40 A
542 02 00307 0FE6C8C9 A UTSM7 TEXTC '(WHILE DELETING)'
      02 00308 D3C540C4 A
      02 00309 C5D3C5E3 A
      02 0030A C9D5C75D A
543 02 0030B 326060E3 A UTSM8 TEXTC '==TAB CHAR. FOUND; ITA; NEEDED FOR COL. SIMULATION'
      02 0030C C1C240C3 A

```

H01 20:44 SEP 08, '75

84

02 0030D C8C1D94B A
 02 0030E 40C6D6E4 A
 02 0030F D5C45E40 A
 02 00310 70E3C17D A
 02 00311 40D5C5C5 A
 02 00312 C4C5C440 A
 02 00313 C6D6D940 A
 02 00314 C3D6D34B A
 02 00315 40E2C9D4 A
 02 00316 E4D3C1E3 A
 02 00317 C9D6D540 A

544

545 02 00317
 546 02 00318 04C3D6D7 A
 547 02 00319 05D4C5D9 A
 548 02 0031A 02D4D240 A
 549 02 0031B 02D4C440 A
 550 02 0031C 02C4C540 A
 551 02 0031D 02C6C440 A
 552 02 0031E 02C6F340 A
 553 02 0031F
 554
 555

*
 BDISPTBL EGU \$=1
 GEN,8,24 4,ICOP!
 GEN,8,24 5,IMER!
 TEXTC IMK!
 TEXTC IMD!
 TEXTC IDE!
 TEXTC IFD!
 TEXTC IFT!
 PATCH RES 50
 *
 FIN

 THESE COMMANDS REQUIRE DISPLAY
 OF SEQ. NUMBERS SET UP IN
 INTFLAG1 AND INTFLAG2, WHEN
 INTERRUPTED BY THE BREAK KEY

PAGE

 * OPEN FPTS (HAND-CODED TO AVOID PROBLEMS) *

556							
557							
558							
559							
560							
561							
562	02	00351	14000000	N	0\$FPT	GEN,8,24	X'14',FIEI
563	02	00352	65480001	A		GEN,32	X'65480001'
564	02	00353	00000BE4	05		DATA	0\$ABN ABN
565	02	00354	00000024			DATA	CARDIMG BUF
566	02	00355	00000002	A		DATA	2 KEYED
567	02	00356	00000004	A		DATA	4 INOUT
568	02	00357	00000002	A		DATA	2 SAVE
569	02	00358	00000003	A		DATA	3 MAX KEY LENGTH
570	02	00359	01000808	A		DATA	X'01000808'
571	02	0035A			0\$NAME	RES	8
572	02	00362	02000202	A		DATA	X'02000202'
573	02	00363			0\$ACCT	RES	2 ACCOUNT
574	02	00365	03010202	A		DATA	X'03010202'
575	02	00366			0\$PASS	RES	2
576							
577							
578							
579	02	00368	14000000	N	02\$FPT	GEN,8,24	X'14',FIE0
580	02	00369	65480001	A		GEN,32	X'65480001'
581	02	0036A	00000BFE	05		DATA	02\$ABN,CARDIMG,2,4,2,3
	02	0036B	00000024				
	02	0036C	00000002	A			
	02	0036D	00000004	A			
	02	0036E	00000002	A			
	02	0036F	00000003	A			
582	02	00370	01000808	A		DATA	X'01000808'
583	02	00371			02\$NAME	RES	8
584	02	00379	02000202	A		DATA	X'02000202'
585	02	0037A			02\$ACCT	RES	2
586	02	0037C	03010202	A		DATA	X'03010202'
587	02	0037D			02\$PASS	RES	2

SAME PARAMETERS AS ABOVE

H01

20:44 SEP 08, '75

588 00000001

D0

MODE=2

589 02 0037F 06 A

RELATIVE DATA,1

6,32,0,0

THIS IS AN FPT FOR

02 0037F 1 20 A

02 0037F 2 00 A

02 0037F 3 00 A

590 02 00380 20000000 A

DATA

1**29

SETTING RELATIVE TABS.

591 02 00381 0080 A

DATA,2

X'80',X'80'

02 00381 2 0080 A

592

FIN

593
 594
 595
 596
 597
 598
 599 00000000
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609 EXT
 610
 611
 612
 613
 614 00000000
 615
 616
 617
 618
 619
 620
 621
 622
 623
 624 EXT
 625

PAGE

 * DCB'S FOR UPDATE AND COPY FILES *

 *
 *
 DB MODE=1
 S F:EI CSECT 0
 S F:EI M:DCB (FILE),,
 S (KEYED),,
 S (INOUT),,
 S (PASS,'SECRET'),,
 S (SAVE),,
 S (KEYM,3),,
 S (BUF,CARDIMG)
 FLSE
 F:EI EQU M:EI
 FIN
 *
 * COPY FILE DCB
 *
 DB MODE=1
 S F:EO CSECT 0
 S F:EO M:DCB (FILE),,
 S (KEYED),,
 S (OUT),,
 S (SAVE),,
 S (KEYM,3),,
 S (BUF,CARDIMG),,
 S (PASS,'SECRET')
 FLSE
 F:EO EQU M:EO
 FIN

626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662

05 00000
05 00000
05 00000
05 00000 68000001
00000000

PAGE

*
* BEGIN EDITOR *
*

*
*
*

CSECT	S(0,1)	
BEGINEDITOR	EQU *	
SECTS	EQU	*
B	BGD10	:::ENTER HERE AT NORMAL START
DB	MODE=1	
MTW,0	FILETYPE	:::ENTER HERE AT BREAK
BLZ	\$+2	RE-OPEN FILE IF ONE WAS OPEN
BAL,LNK	REOPEN	
LI,T1	0	RESET ASSORTED FLAGS, ETC.
STW,T1	LASTKEY	
STW,T1	NOCHGFLG	
STW,T1	SETFLAG	
STW,T1	STEPFLAG	
LI,T1	=1	
STW,T1	ALLFLAG	
S	*	
S	* FINISH INITIALIZATION	
S	*	
BGD10	LI,R0 4	SET ACTIVATION TYPE = 4
CAL3,2	0	
BAL,LNK	TYPMSG	TYPE: L/F + C/R
DATA	MSG0	
S	*	
LI,T1	EDITBASE	CONVERT BASE TO
SLS,T1	=9	PAGES.
S	*	
LI,LNK	BEGINEDITOR	CONVERT PROGRAM TO NEXT
AI,LNK	X'1FF'	HIGHER PAGE, PUT

663		*S*	SLS, LNK	=9	
664		*S*	STW, LNK	RO	IN RO.
665		*S* *			
666		*S*	SW, RO	T1	CONVERT RO TO PROGRAM DATA
667		*S*	STB, LNK	RO	PAGE COUNT.
668		*S* *			
669		*S*	LI, T1	ENEDITOR	COMPUTE PURE PROCEDURE PAGE COUNT
670		*S*	AI, T1	X11FF1	
671		*S*	SLS, T1	=9	
672		*S*	SW, T1	LNK	END-BEGIN
673		*S* *			
674		*S*	LI, X3	0	SET UP REGISTER 1,
675		*S*	STB, T1	X3	
676		*S*	CAL3, 11	0	SET SWAP SIZE
677			ELSE		
678			EGU	*	
679	05 00001	05 00001	350002C8	02	BGD10
680					*
681	05 00002	0480037F	02		
682	05 00003	04800000	03	CAL1, 8	RELATIVE
				MIINT	BRK*KEY
683	03 00000	0E00028F	05		TABING
				FIN	

```

684
685
686
687
688
689
690
691
692
693      05 00004
694      00000001
695      05 00004 041002CA 02
696      05 00005 228FFFFFF A
697      05 00006 358002C3 02
698      05 00007 358002C4 02
699      05 00008 35800020 02
700      05 00009 358002C9 02
701
702      05 0000A 1280000C 02
703      05 0000B 1580019C 02
704      05 0000C 22800000 A
705      05 0000D 35800048 02
706      05 0000E 358000AD 02
707      05 0000F 3580019E 02
708      05 00010 3580019F 02
709      05 00011 358001A1 02
710      05 00012 22800100 A
711      05 00013 35800049 02
712      05 00014 22800049 02
713      05 00015 358000AC 02
714      05 00016 2287A120 A
715      05 00017 358000B2 02
716      05 00018 33000167 02
717      05 00019 6830001C
718      00000000
719
720

```

```

PAGE
*****
*
*   M A S T E R   P A R S E R   *
*
*****
*
*
*
MASTERPARSER      EQU $
                   DB      MODE=2
                   CAL1,1  PROMPT$FPT
                   LI,T1   =1
                   STW,T1  INTFLAG1
                   STW,T1  INTFLAG2
                   STW,T1  ALLFLAG
                   STW,T1  XEQFLAG
                   FIN
                   LD,T1   DMYSTKDW
                   STD,T1  STACKDW
                   LI,T1   0
                   STW,T1  CDT
                   STW,T1  CHARPSN
                   STW,T1  MVD:REC:CNT
                   STW,T1  CHG:STG:CNT
                   STW,T1  CT$FLAG
                   LI,T1   X'0100'
                   STW,T1  CDT+1
                   LI,T1   CDT+1
                   STW,T1  CDTADR
                   LI,T1   50000
                   STW,T1  ERRBRCNT
                   MTW,0   STEPFLAG
                   BEZ     $+3
                   DB      MODE=1
                   LI,R0   1*1
                   CAL3,1  0

```

```

RESET ALL FLAG
PURGE STACK
SET # OF CMNDS = 0
SET NEXT CHAR TO SCAN = 0
SET MVD:REC:CNT = 0
SET CHG:STG:CNT = 0
SET TYPE 'CM' FLAG OFF
PUT 'END OF CDT' MARKER IN CDT
INIT CDTADR=1ST CMND ADDR
SET TO PRINT ALL ERROR MSGS
IS SYSTEM IN STEP MODE
YES = TYPE; 1*1

```

```

*S*
*S*

```

721			*S*	LI,R0	1*1	TYPE PR8MPT: 1*1
722			*S*	CAL3,1	0	
723				ELSE		
724	05	0001A	6A700DC4	BAL,LNK	TYPMSG	YES = TYPE 1*1
725	05	0001B	000002E4 02	DATA	UTSM2	
726				FIN		
727	05	0001C	6A700CAE	BAL,LNK	READTELETYPE2	READ IN COMMANDS
728	05	0001D	20BFFFFF A	AI,R1	=1	SAVE CNT OF # OF CHARS INPUT,
729	05	0001E	35B0019A 02	STW,R1	TTYIMGSZ	LESS C/R
730						
731						
732						
733		05	0001F	RESUME*PARSING	EQU *	(ENTER HERE AFTER SEMI-COLON FOUND)
734	05	0001F	F28000AC 02	LB,T1	*CDTADR	INCR CDTADR TO NEXT ENTRY
735	05	00020	668000AC 02	AWM,T1	CDTADR	
736	05	00021	22800004 A	LI,T1	4	SET PSN OF NEXT PARAM = 1
737	05	00022	358000E5 02	STW,T1	PARAMPSN	
738	05	00023	33100048 02	MTW,1	CDT	INCR 8CUNT OF # OF ENTRIES
739				NXTPRM	ERRC9,,	
740					(INTG,PARSE:1:CMND*INTG),,	
741					(STRG,PARSE:1:CMND*STRG),,	
742					(ALPH,*),,	
743	05	00024	6A7008D4		(END,MASTEREXECUTIVE)	
	05	00025	040001C8 02			
	05	00026	04000096			
	05	00027	0500008E			
	05	00028	0600002A			
	05	00029	00000301			
744	05	0002A	22300025 A	LI,X1	CTBLSZ	
745	05	0002B	328000C1 02	LW,T1	PARAMBUF	SEARCH FOR COMMAND NAME IN TABLE
746	05	0002C	49800E25	8R,T1	=X'00404040'	CONVERT CMND TO UPPER CASE
747	05	0002D	31860039	CW,T1	CNAMETBL,X1	
748	05	0002E	68300033	BE	PRS10	FOUND = GO PROCESS
749	05	0002F	6430002D	BDR,X1	=2	LOOP
750	05	00030	6A700D8F	BAL,LNK	TYPPECRR	NOT IN TBL = TYPE: 1*CN:UNKN CMND'
751	05	00031	000001C4 02	DATA	ERRC8	
752	05	00032	68000004	B	MASTERPARSER	GO TO PARSER

753				*			
754				*	COMMAND FOUND; GO PROCESS ITS PARAMETERS		
755				*			
756	05	00033	72560084	PRS10	LB,P1	CNMRTBL,X1	SET P1=CMND NUMBER
757	05	00034	6706005E		EXU	CBRCHTBL,X1	GO PROCESS CMND PARAMS
758				*			
759				*			
760				*			
761		05	00035	ILGL\$SEMICOLON		EQU *	(ENTER HERE IF ; AFTER F; OR R:CMND)
762	05	00035	22800100 A		LI,T1	X101001	INCR TO TYPE # OF NEXT CMND
763	05	00036	66800048 02		AWM,T1	CDT	
764	05	00037	6A700D8F		BAL,LNK	TYPECERR	TYPE; !=CN;CMND ILGL HERE!
765	05	00038	000001B1 02		DATA	ERRC4	
766	05	00039	68000004		B	MASTERPARSER	

PAGE

*
* COMMAND NAME TABLE
*

767
768
769
770
771 05 00039
772 05 0003A 02C2D740 A
773 05 0003B
774 05 0003B 02E3C140 A
775 05 0003C 02C3D940 A
776 05 0003D 05C2E4C9 A
777 05 0003E 04C3D6D7 A
778 05 0003F 06C4C5D3 A
779 05 00040 04C5C4C9 A
780 05 00041 03C5D5C4 A
781 05 00042 05D4C5D9 A
782 05 00043 02D9D740 A
783 05 00044 02C4C540 A
784 05 00045 02C6C440 A
785 05 00046 02C6F340 A
786 05 00047 02C9D540 A
787 05 00048 02C9E240 A
788 05 00049 02D4C440 A
789 05 0004A 02D4D240 A
790 05 0004B 02D9D540 A
791 05 0004C 02E2E240 A
792 05 0004D 02E2F340 A
793 05 0004E 02E3F240 A
794 05 0004F 02E3F840 A
795 05 00050 02E3C340 A
796 05 00051 02C6F240 A
797 05 00052 02C3D440 A
798 05 00053 02C3F340 A
799 05 00054 02E2C540 A
800 05 00055 02D1F440 A
801 05 00056 02D5D640 A
802 05 00057 02D9C640 A
803 05 00058 01C24040 A

CNAMETBL	EGU	\$=1	
	TEXTC	'BP'	1: BP
	D01	MODE=2	
	TEXTC	'TA'	2: TAB
	TEXTC	'CR'	3: CR
	GEN,8,24	5, 'BUI'	4: BUILD
	GEN,8,24	4, 'COP'	5: COPY
	GEN,8,24	6, 'DEL'	6: DELETE
	GEN,8,24	4, 'EDI'	7: EDIT
	TEXTC	'END'	8: END
	GEN,8,24	5, 'MER'	9: MERGE
	TEXTC	'RP'	10: RP
	TEXTC	'DE'	11: DE
	TEXTC	'FD'	12: FD
	TEXTC	'FT'	13: FT
	TEXTC	'IN'	14: IN
	TEXTC	'IS'	15: IS
	TEXTC	'MD'	16: MD
	TEXTC	'MK'	17: MK
	TEXTC	'RN'	18: RN
	TEXTC	'SS'	19: SS
	TEXTC	'ST'	20: ST
	TEXTC	'TS'	21: TS
	TEXTC	'TY'	22: TY
	TEXTC	'TC'	23: TC
	TEXTC	'FS'	24: FS
	TEXTC	'CM'	25: CM
	TEXTC	'CT'	25: CT
	TEXTC	'SE'	30: SE
	TEXTC	'JU'	39: JU
	TEXTC	'NB'	40: NB
	TEXTC	'RF'	41: RF
	TEXTC	'B'	4: BUILD (SHORT FORM)

H01 20144 SEP 08, '75

804	05	00059	01C34040	A	TEXTC	IC'	5: COPY (SHORT FORM)
805	05	0005A	01C44040	A	TEXTC	ID'	6: DELETE (SHORT FORM)
806	05	0005B	01C54040	A	TEXTC	IE'	7: EDIT (SHORT FORM)
807	05	0005C	01E74040	A	TEXTC	IX'	8: END (SHORT FORM)
808	05	0005D	01D44040	A	TEXTC	IM'	9: MERGE (SHORT FORM)
809	05	0005E	02E3F740	A	TEXTC	ITX'	44: TX

810 00000025

CTBLSZ EQU *-CNAMETBL=1

*
* COMMAND BRANCH TABLE

814 05 0005E

CBRCHTBL EQU *-1

815	05	0005F	680000F7	B	PARSE:BP	1: BP
816	05	00060		DB1	MODE=2	
817	05	00060	680000F7	B	PARSE:TA	2: TAB
818	05	00061	680000F7	B	PARSE:CR	3: CR
819	05	00062	68000103	B	PARSE:BUILD	4: BUILD
820	05	00063	68000131	B	PARSE:COPY	5: COPY
821	05	00064	68000181	B	PARSE:DELETE	6: DELETE
822	05	00065	68000181	B	PARSE:EDIT	7: EDIT
823	05	00066	6800018B	B	PARSE:END	8: END
824	05	00067	68000150	B	PARSE:MERGE	9: MERGE
825	05	00068	680000F7	B	PARSE:RP	10: RP
826	05	00069	680001AF	B	PARSE:DE	11: DE
827	05	0006A	680001DC	B	PARSE:FD	12: FD
828	05	0006B	680001DC	B	PARSE:FT	13: FT
829	05	0006C	6800020C	B	PARSE:IN	14: IN
830	05	0006D	6800020C	B	PARSE:IS	15: IS
831	05	0006E	68000210	B	PARSE:MD	16: MD
832	05	0006F	68000210	B	PARSE:MK	17: MK
833	05	00070	68000228	B	PARSE:RN	18: RN
834	05	00071	68000245	B	PARSE:SS	19: SS
835	05	00072	68000245	B	PARSE:ST	20: ST
836	05	00073	68000269	B	PARSE:TS	21: TS
837	05	00074	68000269	B	PARSE:TY	22: TY
838	05	00075	68000262	B	PARSE:TC	23: TC
839	05	00076	680001DC	B	PARSE:FS	24: FS
840	05	00077	68000193	B	PARSE:CM	25: CM

841	05	00078	68000192	B	PARSE:CT	25: CT
842	05	00079	680001B2	B	PARSE:SE	30: SE
843	05	0007A	68000248	B	PARSE:JU	39: JU
844	05	0007B	6800018B	B	PARSE:NO	40: NO
845	05	0007C	6800025C	B	PARSE:RF	41: RF
846	05	0007D	68000103	B	PARSE:BUILD	4: BUILD
847	05	0007E	68000131	B	PARSE:COPY	5: COPY
848	05	0007F	68000181	B	PARSE:DELETE	6: DELETE
849	05	00080	68000181	B	PARSE:EDIT	7: EDIT
850	05	00081	6800018B	B	PARSE:END	8: END
851	05	00082	68000150	B	PARSE:MERGE	9: MERGE
852	05	00083	6800025C	B	PARSE:TX	44: TX

*
* COMMAND NUMBER TABLE
*

					CNMRTBL	EQU	\$	
853								
854								
855								
856		05	00084					
857	05	00084	00	A		DATA,1	0	(FILLER)
858	05	00084	1 01	A		DATA,1	1	1: BP
859	05	00084	2			DB1	MODE=2	
860	05	00084	2 02	A		DATA,1	2	2: TAB
861	05	00084	3 03	A		DATA,1	3	3: CR
862	05	00085	04	A		DATA,1	4	4: BUILD
863	05	00085	1 05	A		DATA,1	5	5: COPY
864	05	00085	2 06	A		DATA,1	6	6: DELETE
865	05	00085	3 07	A		DATA,1	7	7: EDIT
866	05	00086	08	A		DATA,1	8	8: END
867	05	00086	1 09	A		DATA,1	9	9: MERGE
868	05	00086	2 0A	A		DATA,1	10	10: RP
869	05	00086	3 0B	A		DATA,1	11	11: DE
870	05	00087	0C	A		DATA,1	12	12: FD
871	05	00087	1 0D	A		DATA,1	13	13: FT
872	05	00087	2 0E	A		DATA,1	14	14: IN
873	05	00087	3 0F	A		DATA,1	15	15: IS
874	05	00088	10	A		DATA,1	16	16: MD
875	05	00088	1 11	A		DATA,1	17	17: MK
876	05	00088	2 12	A		DATA,1	18	18: RN
877	05	00088	3 13	A		DATA,1	19	19: SS

H01 20144 SEP 08, '75

878	05	00089	14	A	
879	05	00089	1	15	A
880	05	00089	2	16	A
881	05	00089	3	17	A
882	05	0008A	18	A	
883	05	0008A	1	19	A
884	05	0008A	2	19	A
885	05	0008A	3	19	A
886	05	0008B	27	A	
887	05	0008B	1	28	A
888	05	0008B	2	29	A
889	05	0008B	3	04	A
890	05	0008C	05	A	
891	05	0008C	1	06	A
892	05	0008C	2	07	A
893	05	0008C	3	08	A
894	05	0008D	09	A	
895	05	0008D	1	20	A
896					

DATA,1	20
DATA,1	21
DATA,1	22
DATA,1	23
DATA,1	24
DATA,1	25
DATA,1	25
DATA,1	30
DATA,1	39
DATA,1	40
DATA,1	41
DATA,1	4
DATA,1	5
DATA,1	6
DATA,1	7
DATA,1	8
DATA,1	9
DATA,1	44
BBUND	4

201	ST
21:	TS
22:	TY
23:	TC
24:	FS
25:	CM
25:	CT
30:	SE
39:	JU
40:	NO
41:	RF
4:	BUILD (SHORT FORM)
5:	COPY (SHORT FORM)
6:	DELETE (SHORT FORM)
7:	EDIT (SHORT FORM)
8:	END (SHORT FORM)
9:	MERGE (SHORT FORM)
44:	TX

MO1 20144 SEP 08, 1975
 929 05 000A6 6A70081B
 930
 931 05 000A7 6A7008D4
 05 000A8 010001C8 02
 05 000A9 060000AA
 932
 933
 934
 935 05 000AA 22300008 A
 936 05 000AB 328000C1 02
 937 05 000AC 318600E3
 938 05 000AD 683000B2
 939 05 000AE 643000AC
 940 05 000AF 6A700D8F
 941 05 000B0 000001C4 02
 942 05 000B1 68000004
 943
 944
 945
 946 05 000B2 22400001 A
 947 05 000B3 725600EC
 948 05 000B4 F55800AC 02
 949 05 000B5 670600EE
 950
 951
 952
 953 05 000B6 6A700963
 954 05 000B7 00000002 A
 955 05 000B8 468000C1 02
 956 05 000B9 22500004 A
 957 05 000BA 6A70081B
 958 05 000BB 358000C1 02
 959 05 000BC 680000AA
 960
 961
 962
 963 05 000BU

BAL, LNK ADDCDTPARAM
 NXTPRM ERRC9,
 (ALPH,*)
 *
 * COMMAND NAME FOUND: IDENTIFY IT
 *
 ICS10 LI, X1 ICTBLSZ
 LW, T1 PARAMBUF SEARCH TABLE FOR CMND NAME
 CW, T1 ICNAMETBL, X1
 BE ICS20 FOUND - GO PROCESS
 BDR, X1 #=2 LOOP
 BAL, LNK TYPECERR TYPE: !-CN:UNKN CMND!
 DATA ERRC8
 B MASTERPARSER GO TO PARSER
 *
 * COMMAND IDENTIFIED: GO PROCESS LAST PARAMETER
 *
 ICS20 LI, X2 1 PUT CMND NUMBER IN CDT
 LB, P1 ICNMRTBL, X1
 STB, P1 *CDTADR, X2
 FXU ICBRCHTBL, X1 GO PROCESS LAST PARAM
 *
 * FORM FOUND IS: C X . , PROCESS THIS
 *
 ICS50 BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY WITH CMND=0
 DATA 2
 XW, T1 PARAMBUF PUT INTG IN PARAMBUF AND SAVE NAME
 LI, P1 INTG PUT INTG IN CDT
 BAL, LNK ADDCDTPARAM
 STW, T1 PARAMBUF RESTORE CMND NAME
 B ICS10 GO IDENTIFY CMND
 *
 * FINISH TYPE ALPHA: - X /STR2/
 *
 TYPE\$ALPHA EQU \$

```

964
965 05 000BD 6A7008D4
      05 000BE 01000249 02
      05 000BF 090000C0
966 05 000C0 22500005 A
967 05 000C1 6A70081B
968
969
970 05 000C2 6A7008D4
      05 000C3 02800239 02
      05 000C4 0800001F
      05 000C5 00000301

```

```

NXTPRM  ERRP8,,
          (STRG,*)

LI,P1    STRG          PUT STRING IN CDT
BAL,LNK  ADDCDTPARAM
NXTPRM   *ERRP4,,
          (SCBL,RESUME*PARSING),,
          (END,MASTEREXECUTIVE)

```

```

971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990

```

```

*
* FINISH TYPE BETA: = X N
*
TYPE$BETA
NXTPRM  EQU $
          ERRP9,,
          (INTG,*)

LI,P1    INTG          PUT COUNT IN CDT
BAL,LNK  ADDCDTPARAM
NXTPRM   *ERRP4,,
          (SCBL,RESUME*PARSING),,
          (END,MASTEREXECUTIVE)

```

```

982
983
984
985
986
987
988
989
990

```

```

*
* INTRALINE COMMANDS 'D' OR 'S' FOUND: CHECK THAT FORM IS: /STR1/ D(S)
*
TYPE$I:CMND$D  EQU $
LI,X4         1
B             TYPE$I:CMND$S+1
          USE X4=1 FOR 'D'

*
*
TYPE$I:CMND$S  EQU $

```

05 000D1

991	05	000D1	22200000	A	LI,X4	0	USE X4=0 FOR 'S'
992	05	000D2	22300003	A	LI,X1	3	
993	05	000D3	F28600AC	02	LB,T1	*CDTADR,X1	GET # OF PARAMS IN CDT
994	05	000D4	21800003	A	CI,T1	3	IS # OF PARAMS = 3
995	05	000D5	683000DE		BE	ICS90	YES = FORM MUST BE: N /ST1/ D(S) =
996	05	000D6	22300004	A	LI,X1	4	
997	05	000D7	F28600AC	02	LB,T1	*CDTADR,X1	NO = GET TYPE OF PARAM1
998	05	000D8	21800005	A	CI,T1	STRG	IS TYPE='STRING'
999	05	000D9	683000DE		BE	ICS90	YES = FORM MUST BE: /ST1/ D(S) =
1000	05	000DA	33E000E5	02	MTW,-2	PARAMPSN	NO = ADJ PARAM PSN FOR ERROR MSG
1001	05	000DB	6A7000D9		BAL,LNK	TYPEPERR	TYPE: !=P1;NOT STRNG;
1002	05	000DC	00000249	02	DATA	ERRP8	
1003	05	000DD	68000004		B	MASTERPARSER	GO TO PARSER

1004
1005 * FORM OF 'D' OR 'S' IS OK: GO PARSE FURTHER

1006							
1007	05	000DE	680400DF		ICS90	B	*+1,X4
1008	05	000DF	680000BD			B	TYPE\$ALPHA
1009						NXTPRM	*ERRP4,1
1010							(SCBL,RESUME\$PARSING),1
1011	05	000E0	6A7000D4				(END,MASTEREXECUTIVE)
	05	000E1	02800239	02			
	05	000E2	0800001F				
	05	000E3	00000301				

1012
1013 * INTRALINE COMMAND NAME TABLE

1014							
1015		05	000E3		ICNAMETBL	EQU	\$=1
1016	05	000E4	01C44040	A	TEXTC	ID'	31: D
1017	05	000E5	01C54040	A	TEXTC	IE'	32: E
1018	05	000E6	01C64040	A	TEXTC	IF'	33: F
1019	05	000E7	01D34040	A	TEXTC	IL'	34: L
1020	05	000E8	01D64040	A	TEXTC	IO'	35: O
1021	05	000E9	01D74040	A	TEXTC	IP'	36: P
1022	05	000EA	01D94040	A	TEXTC	IR'	37: R
1023	05	000EB	01E24040	A	TEXTC	IS'	38: S
1024			00000008		ICTBLSZ	EQU	\$=ICNAMETBL-1

1025
 1026
 1027
 1028 05 000EC
 1029 05 000EC 00 A
 1030 05 000EC 1 17 A
 1031 05 000EC 2 20 A
 1032 05 000EC 3 21 A
 1033 05 000ED 22 A
 1034 05 000ED 1 23 A
 1035 05 000ED 2 24 A
 1036 05 000ED 3 25 A
 1037 05 000EE 26 A

*
 * INTRALINE COMMAND NUMBER TABLE
 *

ICNMRTBL EQU \$
 DATA,1 0 (FILLER)
 DATA,1 31 31: D
 DATA,1 32 32: E
 DATA,1 33 33: F
 DATA,1 34 34: L
 DATA,1 35 35: 0
 DATA,1 36 36: P
 DATA,1 37 37: R
 DATA,1 38 38: S
 BOUND 4

1038
 1039
 1040
 1041
 1042 05 000EE
 1043 05 000EF 680000CF
 1044 05 000F0 680000BD
 1045 05 000F1 680000BD
 1046 05 000F2 680000C6
 1047 05 000F3 680000BD
 1048 05 000F4 680000BD
 1049 05 000F5 680000C6
 1050 05 000F6 680000D1

*
 * INTRALINE COMMAND BRANCH TABLE
 *

ICBRCHTBL EQU \$-1
 B TYPE#I;CMND#D 31: D
 B TYPE\$ALPHA 32: E
 B TYPE\$ALPHA 33: F
 B TYPE\$BETA 34: L
 B TYPE\$ALPHA 35: 0
 B TYPE\$ALPHA 36: P
 B TYPE\$BETA 37: R
 B TYPE#I;CMND#S 38: S

PAGE

1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073

05 000F7
05 000F7
05 000F7
05 000F7
05 000F7 6A700963
05 000F8 00000001 A
05 000F9 6A700834
05 000FA 6A7008D4
05 000FB 01000239 02
05 000FC 060000FD
05 000FD 22500006 A
05 000FE 6A70081B
05 000FF 6A7008D4
05 00100 02800239 02
05 00101 08000035
05 00102 00000301

```
*****  
* PARSE FORM: BP BN(0FF) *  
* PARSE FORM: TA F(M,S) *  
* PARSE FORM: RP BN(0FF) *  
* PARSE FORM: CR BN(0FF) *  
*****  
*  
*  
PARSE:RP EQU $  
PARSE:BP EQU $  
PARSE:CR EQU $  
PARSE:TA EQU $  
BAL,LNK NEWCDTENTRY BUILD NEW CDT ENTRY  
DATA 1  
BAL,LNK CHECK1CDTENTRY MAKE SURE !BP! IS FIRST CMND  
NXTPRM ERRP4,,  
(ALPH,*)  
  
LI,P1 ALPH PUT ALPHA TEXT IN CDT  
BAL,LNK ADDCDTPARAM  
NXTPRM *ERRP4,,  
(SCOL,ILGL$SEMICOLON),,  
(END,MASTEREXECUTIVE)
```

PAGE

* PARSE FORM: BUILD FID(,N(,I)) *

1074
1075
1076
1077
1078
1079
1080 05 00103
1081 05 00103 6A700963
1082 05 00104 00000003 A
1083 05 00105 6A700834
1084 05 00106 6A70083A
1085 05 00107 22500001 A
1086 05 00108 6A70081B
1087
1088
1089
1090 05 00109 6A7008D4
05 0010A 03800239 02
05 0010B 0700010E
05 0010C 08000035
05 0010D 00000301

1091
1092
1093
1094 05 0010E
1095
1096
1097
1098
1099 05 0010E 6A7008D4
05 0010F 0400023D 02
05 00110 04000114
05 00111 0700012C
05 00112 02000115
05 00113 03000129
1100 05 00114 6A700287
1101

PARSE:BUILD EQU \$
BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
DATA 3
BAL, LNK CHECK1CDTENTRY MAKE SURE 'BUILD' IS FIRST CMND
BAL, LNK GETFILEID GET FILE ID
LI, P1 NAME PUT IT IN CDT
BAL, LNK ADDCDTPARAM
NXTPRM *ERRP4,,
(COM,*),,
(SCBL, ILGL\$SEMICOLON),,
(END, MASTEREXECUTIVE)

*
*
*
GET\$SEQ\$INCR EQU \$ (ENTER HERE FOR FORM: N(,I))
NXTPRM ERRP5,,
(INTG,*),,
(COM, PBU05),,
(SEQ, PBU10),,
(SEQ2, ILGL\$SEQ2)

BAL, LNK ADJINT
*

```

1102      * PUT SEQ # IN CDT
1103      *
1104      05 00115      22500002 A  PBU10      LI,P1      SEQ          PUT SEQ # IN CDT
1105      05 00116      6A70081B      BAL,LNK      ADDCDTPARAM
1106      *
1107      *
1108      *
1109      05 00117      GET$INCREMENT      EGU $          (ENTER HERE FOR FORM: (I) )
1110      NXTPRM      *ERRP4,,
1111      (COM,*),,
1112      (SCOL,ILGL$SEMICOLON),,
1113      05 00117      6A7008D4      (END,MASTEREXECUTIVE)
1113      05 00118      03800239 02
1113      05 00119      0700011C
1113      05 0011A      08000035
1113      05 0011B      00000301
1114      05 0011C      PBU30      FGU          $
1115      05 0011D      NXTPRM      ERRP6,,
1116      05 0011E      (INTG,*),,
1117      05 0011C      6A7008D4      (SEQ,PBU20)
1117      05 0011D      02000241 02
1117      05 0011E      04000120
1117      05 0011F      02000121
1118      05 00120      6A700287      BAL,LNK      ADJINT
1119      *
1120      * PUT INCREMENT IN CDT
1121      *
1122      05 00121      22500002 A  PBU20      LI,P1      SEQ          PUT INCR IN CDT
1123      05 00122      330000C1 02      MTW,0      PARAMBUF      MAY NOT BE ZERO.
1124      05 00123      68300129      BEZ          ILGL$SEQ2
1125      05 00124      6A70081B      BAL,LNK      ADDCDTPARAM
1126      05 00125      6A7008D4      NXTPRM      *ERRP4,,
1127      05 00126      02800239 02      (SCOL,ILGL$SEMICOLON),,
1127      05 00127      08000035      (END,MASTEREXECUTIVE)
1128      05 00128      00000301

```

1129				*			
1130				*			
1131				*			
1132		05 00129		ILGL SEQ2	EGU *		
1133	05	00129	6A700D99	BAL, LNK	TYPEPERR	TYPE: IPN: ILGL SEQ #	
1134	05	0012A	0U000250 02	DATA	ERRP10		
1135	05	0012B	68000004	B	MASTERPARSER	GO TO PARSER	
1136				*			
1137				*	ENTER HERE FOR FORM: (, , I)		
1138				*			
1139	05	0012C	225003E8 A	PBU05	LI, P1	1000	ONE IS DEFAULT SEQ #
1140	05	0012D	355000C1 02		STW, P1	PARAMBUF	
1141	05	0012E	22500002 A		LI, P1	SEQ	PUT SEQ # IN CDT
1142	05	0012F	6A70081B		BAL, LNK	ADDCDTPARAM	
1143	05	00130	6800011C		B	PBU30	GET INCREMENT

PAGE

 * PARSE FORM: COPY FID1 TO FID2(,N(,I)) *

*
 *

1144
 1145
 1146
 1147
 1148
 1149
 1150 05 00131
 1151 05 00131 6A700963
 1152 05 00132 00000005 A
 1153 05 00133 6A700834
 1154 05 00134 6A70083A
 1155 05 00135 22500001 A
 1156 05 00136 6A70081B
 1157
 1158 05 00137 6A7008D4
 05 00138 010001C8 02
 05 00139 0600013A
 1159 05 0013A 328000C1 02
 1160 05 0013B 31800015 02
 1161 05 0013C 6930013F
 1162 05 0013D 32800012 02
 1163 05 0013E 358000C1 02
 1164 05 0013F 0013F
 1165 05 0013F 22500006 A
 1166 05 00140 6A70081B
 1167 05 00141 328000C1 02
 1168 05 00142 31800012 02
 1169 05 00143 68300149
 1170 05 00144 31800013 02
 1171 05 00145 68300149
 1172 05 00146 00146
 1173 05 00146 6A700D8F
 1174 05 00147 000001C8 02
 1175 05 00148 68000004
 1176
 1177
 1178

PARSE: COPY EQU \$
 BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
 DATA 5
 BAL, LNK CHECK1CDTENTRY MAKE SURE 'COPY' IS FIRST CMND
 BAL, LNK GETFILEID GET FILE ID 1
 LI, P1 NAME PUT IT IN CDT
 BAL, LNK ADDCDTPARAM
 NXTPRM ERRC9, ,
 (ALPH, *)
 LW, T1 PARAMBUF
 CW, T1 X: T8
 BNE PC03
 LW, T1 X: 0N
 STW, T1 PARAMBUF
 PC03 EQU \$
 LI, P1 ALPH PUT '0N(0VER)' IN CDT
 BAL, LNK ADDCDTPARAM
 LW, T1 PARAMBUF
 CW, T1 X: 0N DOES PARAM2= '0N' OR '0VER'
 BE PC010
 CW, T1 X: 0VER
 BE PC010
 PC05 EQU \$
 BAL, LNK TYPECERR NO = TYPE! 'CN: ILGL SYNTAX!
 DATA ERRC9
 B MASTERPARSER EXIT TO PARSER
 *
 * GET 2ND FID AND THEN GO PROCESS FORM: (,N(,I))
 *

MO1 20144 SEP 08, 175

1179 05 00149 6A70083A
 1180 05 0014A 6A70081B
 1181
 1182
 1183
 1184 05 0014B 6A7008D4
 05 0014C 03800239 02
 05 0014D 0700010E
 05 0014E 08000035
 05 0014F 00000301

PC010

BAL, LNK GETFILEID GET FILE ID 2
 BAL, LNK ADDCDTPARAM PUT IT IN CDT
 NXTPRM *ERRP4,;
 (COM, GET*SEQ*INCR),;
 (SC0L, ILGL*SEMIC0L0N),;
 (END, MASTEREXECUTIVE)

1185

*

PAGE

1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203

1204
1205
1206
1207
1208

1209
1210
1211
1212
1213
1214

05 00150 6A700963
05 00151 00000006 A
05 00152 6A700834
05 00153 6A70083A
05 00154 22500001 A
05 00155 6A70081B

05 00156 6A7008D4
05 00157 03800239 02
05 00158 0700015B
05 00159 06000167
05 0015A 08000035

05 0015B 6A7008D4
05 0015C 0300023D 02
05 0015D 04000160
05 0015E 02000161
05 0015F 03000162

05 00160 6A700287
05 00161 6A70028B
05 00162 22500003 A
05 00163 6A70081B

```
*****
* PARSE FORM: MERGE FID1(,N1(=N2)) INTO FID2,N3(=N4)(,I) *
*****
*
*
* PARSE:MERGE EQU $
BAL,LNK NEWCDTENTRY SET UP NEW ENTRY.
DATA 6
BAL,LNK CHECK1CDTENTRY MUST BE FIRST.
BAL,LNK GETFILEID GET THE FID,
LI,P1 NAME AND ADD IT TO THE CDT.
BAL,LNK ADDCDTPARAM
*
NXTPRM *ERRP4,, CHECK FOR SPECIFIC RECORD RANGE.
(COM,*),,
(ALPH,PME20),,
(SCOL,ILGL,SEMICOLON)
*
NXTPRM ERRP5,, CONVERT SEQUENCE SPECIFICATION.
(INTG,*),,
(SEQ,PME5),,
(SEQ2,PME15)
*
BAL,LNK ADJINT ADJUST INTEGER,
PME5 BAL,LNK REPSEQ DUPLICATE SINGLE VALUE.
PME15 LI,P1 SEQ2
BAL,LNK ADDCDTPARAM PUT SEQ # PAIR IN CDT.
*
```

1215					NXTPRM	ERRC9,, (ALPH,*)	VERIFY 'INT0' NEXT.
1216	05 00164	6A7008D4					
	05 00165	010001C8	02				
	05 00166	06000167					
1217	05 00167	22500006	A	PME20	LI,P1	ALPH	ADD STRING TO CDT.
1218	05 00168	6A70081B			BAL,LNK	ADDCDTPARAM	
1219				*			
1220	05 00169	328000C1	02		LW,T1	PARAMBUF	MAKE SURE OF PARAMETER.
1221	05 0016A	31800014	02		CW,T1	X:INT0	
1222	05 0016B	69300146			BNE	PC05	BRANCH ON ERROR.
1223				*			
1224	05 0016C	6A70083A			BAL,LNK	GETFILEID	COLLECT FID2
1225	05 0016D	22500001	A		LI,P1	NAME	AND ADD TO CDT.
1226	05 0016E	6A70081B			BAL,LNK	ADDCDTPARAM	
1227				*			
1228					NXTPRM	ERRC9,, (COM,*),, (END,PME40),, (SC0L,ILGL&SEMIC0L0N)	VERIFY PRESENCE OF DESTINATION SEQ #.
1229							
1230							
1231	05 0016F	6A7008D4					
	05 00170	030001C8	02				
	05 00171	07000174					
	05 00172	0000017E					
	05 00173	08000035					
1232				*			
1233					NXTPRM	ERRP5,, (INTG,*),, (SEQ,PME30),, (SEQ2,PME35)	CONVERT SPECIFICATION.
1234							
1235							
1236	05 00174	6A7008D4					
	05 00175	0300023D	02				
	05 00176	04000179					
	05 00177	0200017A					
	05 00178	0300017B					
1237				*			
1238	05 00179	6A700287			BAL,LNK	ADJINT	
1239	05 0017A	6A70028B		PME30	BAL,LNK	REPSEQ	
1240	05 0017B	22500003	A	PME35	LI,P1	SEQ2	ADD TO CDT.
1241	05 0017C	6A70081B			BAL,LNK	ADDCDTPARAM	

H01 20144 SEP 08, '75

110

1242			*			
1243	05 0017D	68000117		B	GET*INCREMENT	GO PROCESS POSSIBLE INCREMENT.
1244			*			
1245	05 0017E	6A700D99	PME40	BAL, LNK	TYPEPERR	
1246	05 0017F	00000271 02		DATA	FRRP17	
1247	05 00180	68000004		B	MASTERPARSER	

PAGE

1266
 1267
 1268
 1269
 1270
 1271
 1272
 1273 05 0018B
 1274 05 0018B
 1275 05 0018B 6A700963
 1276 05 0018C 00000000 A
 1277 05 0018D 6A700834
 1278
 1279
 1280 05 0018E 6A7008D4
 05 0018F 020001C8 02
 05 00190 08000035
 05 00191 00000301
 1281

```
*****
*  PARSE FORMS:  END  *
*                   NO  *
*****
*
*
PARSE:END           EQU $
PARSE:NO            EQU $
          BAL, LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY
          DATA       0
          BAL, LNK   CHECK1CDTENTRY     MAKE SURE (END(NO),) IS FIRST CMND
          NXTPRM     ERRC9,,
                     (SCOL, ILGL$SEMICOLON),,
                     (END, MASTEREXECUTIVE)
```

*

PAGE

```

1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310

```

```

*****
* PARSE FORM: CM N,C *
* PARSE FORM: CT N,C *
*****
*
*
PARSE:CT FGU          $
05 00192 331001A1 02  MTW,1    CT$FLAG          SET TYPE FLAG ON
AND TURN CMND INTO ICM:
*
PARSE:CM FGU          $
05 00193 6A700963     BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
05 00194 00000002 A   DATA      2
05 00195 6A700834     BAL,LNK  CHECK1CDTENTRY     MAKE SURE ICM: IS FIRST CMND
NXTPRM          ERRP5,,
                    (INTG,*),,
                    (SEQ,PCM10),,
                    (SEQ2,ILGL$SEQ2)
05 00196 6A7008D4
05 00197 0300023D 02
05 00198 0400019B
05 00199 0200019C
05 0019A 03000129
05 0019B 6A700287     BAL,LNK  ADJINT
*
* SEQ # GIVEN: PUT IT IN CDT AND PROCESS COLUMN NUMBER
*
PCM10  LI,P1    SEQ          PUT SEQ # IN CDT
05 0019C 22500002 A   BAL,LNK  ADDCDTPARAM
05 0019D 6A70081B     NXTPRM   *ERRP4,,
                    (COM,*),,
                    (SCBL,PCM20),,
                    (END,PCM20)
05 0019E 6A7008D4
05 0019F 03800239 02
05 001A0 070001A3
05 001A1 080001AC
05 001A2 000001AC
NXTPRM          ERRP7,,

```

HC1 20144 SEP 08, 175

1311 05 001A3 6A7008D4
 05 001A4 01000245 02
 05 001A5 040001A6
 1312 05 001A6 22500004 A
 1313 05 001A7 6A70081B
 1314
 1315
 1316 05 001A8 6A7008D4
 05 001A9 02800239 02
 05 001AA 08000035
 05 001AB 00000301

(INTG,*)

LI,P1 INTG PUT COL, # IN CDT
 BAL,LNK ADDCDTPARAM
 NXTPRM *ERRP4,)
 (SCOL,ILGL*SEMICOLON),;
 (END,MASTEREXECUTIVE)

1317
 1318
 1319
 1320 05 001AC 6A700D99
 1321 05 001AD 00000271 02
 1322 05 001AE 68000004

*
 * ERROR: SECOND PARAMETER MISSING
 *

PCM20 BAL,LNK TYPEPERR TYPE: !.PN:PARAM MISSING!
 DATA ERRP17
 B MASTERPARSER GO TO PARSER

H01	20144	SEP 08,	175			
1356	05	001BD	22500003	A	LI,P1	SEQ2 PUT SEQ # PAIR IN CDT
1357	05	001BE	6A70081B		BAL, LNK	ADDCDTPARAM
1358	05	001BF	2180001E	A	CI, T1	FIRST, IICMND IS CMND='DE'
1359	05	001C0	681001C5		BGE	PDE20 NB = CMND='SE'
1360					NXTPRM	*ERRP4,,
1361						(SC0L, ILGL\$SEMIC0L0N),,
1362	05	001C1	6A7008D4			(END, MASTEREXECUTIVE)
	05	001C2	02800239	02		
	05	001C3	08000035			
	05	001C4	00000301			
1363					*	
1364					* FINISH UP 'SF'	
1365					*	
1366					PDE20	NXTPRM *ERRP4,,
1367						(C0M,*),,
1368						(SC0L, RESUME\$PARSING),,
1369	05	001C5	6A7008D4			(END, MASTEREXECUTIVE)
	05	001C6	03800239	02		
	05	001C7	070001CA			
	05	001C8	0800001F			
	05	001C9	00000301			
1370					NXTPRM	ERRP7,,
1371	05	001CA	6A7008D4			(INTG,*)
	05	001CB	01000245	02		
	05	001CC	040001CD			
1372	05	001CD	22500004	A	LI,P1	INTG
1373	05	001CE	6A70081B		BAL, LNK	ADDCDTPARAM PUT 1ST C0L # IN CDT
1374					NXTPRM	*ERRP4,,
1375						(C0M,*),,
1376						(SC0L, RESUME\$PARSING),,
1377	05	001CF	6A7008D4			(END, MASTEREXECUTIVE)
	05	001D0	03800239	02		
	05	001D1	070001D4			
	05	001D2	0800001F			
	05	001D3	00000301			
1378					NXTPRM	ERRP7,,
1379	05	001D4	6A7008D4			(INTG,*)

HO1 20:44 SEP 08, 175

117

	05	001D5	01000245	02
	05	001D6	0*0001D7	
1380	05	001D7	6A70081B	
1381				
1382				
1383	05	001D8	6A7008D4	
	05	001D9	02800239	02
	05	001DA	0800001F	
	05	001DB	0U000301	

```
BAL, LNK ADDCDTPARAM PUT 2ND COL # IN CDT
NXTPRM *ERRP4, )
(SCOL, RESUME*PARSING), )
(END, MASTEREXECUTIVE)
```


HC1 20144 SEP 08, 175

119

	05 001EB	08000209			
	05 001EC	00000209			
1415				NXTPRM	ERRP8,,
1416	05 001ED	6A7008D4			(STRG,*)
	05 001EE	01000249	02		
	05 001EF	050001F0			
1417	05 001F0	22500005	A	LI,P1	STRG
1418	05 001F1	6A70081B		BAL,LNK	ADDCDTPARAM
1419					
1420					
1421	05 001F2			GET\$CBL#\$PAIR	EQU *
1422				NXTPRM	*ERRP4,,
1423					(COM,*),,
1424					(SCBL,ILGL\$SEMICOLON),,
1425	05 001F2	6A7008D4			(END,MASTEREXECUTIVE)
	05 001F3	03800239	02		
	05 001F4	070001F7			
	05 001F5	08000035			
	05 001F6	00000301			
1426				NXTPRM	ERRP7,,
1427	05 001F7	6A7008D4			(INTG,*)
	05 001F8	01000245	02		
	05 001F9	040001FA			
1428	05 001FA	22500004	A	LI,P1	INTG
1429	05 001FB	6A70081B		BAL,LNK	ADDCDTPARAM
1430				NXTPRM	*ERRP4,,
1431					(COM,*),,
1432					(SCBL,ILGL\$SEMICOLON),,
1433	05 001FC	6A7008D4			(END,MASTEREXECUTIVE)
	05 001FD	03800239	02		
	05 001FE	07000201			
	05 001FF	08000035			
	05 00200	00000301			
1434				NXTPRM	ERRP7,,
1435	05 00201	6A7008D4			(INTG,*)
	05 00202	01000245	02		
	05 00203	04000204			

HC1 20:44 SEP 08, '75

1436 05 00204 6A70081B
 1437
 1438
 1439 05 00205 6A7008D4
 05 00206 02800239 02
 05 00207 08000035
 05 00208 0U000301

BAL, LNK ADDCDTPARAM PUT 2ND 'COL #' IN CDT
 NXTPRM *ERRP4,,
 (SCOL, ILGL\$SEMICOLON),,
 (END, MASTEREXECUTIVE)

1440
 1441
 1442
 1443 05 00209 6A700D99
 1444 05 0020A 0U000271 02
 1445 05 0020B 68000004

*
 * ERROR: SECOND PARAMETER MISSING
 *
 PFD20 BAL, LNK TYPEPERR TYPE: '=PN;PARAM MISSING'
 DATA ERRP17
 B MASTERPARSER GO TO PARSE

PAGE

* PARSE FORM: IN N(I) *

*
*

1446
1447
1448
1449
1450
1451
1452 05 0020L
1453 05 0020L
1454 05 0020C 6A700963
1455 05 0020D 0000002 A
1456 05 0020E 6A700834
1457 05 0020F 6800010E

PARSE:IN EGU \$
PARSE:IS EGU \$
BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
DATA 2
BAL, LNK CHECK1CDTENTRY MAKE SURE 'IN' IS FIRST CMND
B GET*SEQ*INCR GO PROCESS FORM: N(I)

PAGE

1458
 1459
 1460
 1461
 1462
 1463
 1464
 1465 05 0021U
 1466 05 0021U
 1467 05 00210 6A700963
 1468 05 00211 0000003 A
 1469 05 00212 6A700834
 1470
 1471
 1472
 1473 05 00213 6A7008D4
 05 00214 0300023D 02
 05 00215 04000218
 05 00216 02000219
 05 00217 0300021A
 1474 05 00218 6A700287
 1475
 1476
 1477
 1478 05 00219 6A70028B
 1479
 1480
 1481
 1482 05 0021A 22500003 A
 1483 05 0021B 6A70081B
 1484
 1485 05 0021C 6A7008D4
 05 0021D 01800239 02
 05 0021E 0700021F
 1486
 1487
 1488

```

*****
* PARSE FORMS: MD N(=M),K(=L)(,I) *
* MK N(=M),K(=L)(,I) *
*****
*
*
* PARSE:MD EQU $
* PARSE:MK EQU $
* BAL,LNK NEWCDTENTRY BUILD NEW CDT ENTRY
* DATA 3
* BAL,LNK CHECK1CDTENTRY MAKE SURE 'MD(MK)' IS FIRST CMND
* NXTPRM ERRP5,,
* (INTG,*),,
* (SEQ,PMD10),,
* (SEQ2,PMD15)
*
* BAL,LNK ADJINT
*
* ONLY ONE SEQ. # GIVEN: DUPLICATE IT
*
* PMD10 BAL,LNK REPSEQ
*
* PUT FIRST SEQ # PAIR IN CDT AND GET 2ND PARAMETER
*
* PMD15 LI,P1 SEQ2 PUT 'SEQ # PAIR' PARAM IN CDT
* BAL,LNK ADDCDTPARAM
* NXTPRM *ERRP4,,
* (COM,*)
*
* NXTPRM ERRP5,,
* (INTG,*),,
* (SEQ,PMD20),,

```

MO1 20144 SEP 08, 175

1489 05 0021F 6A7008D4
 05 00220 0300023D 02
 05 00221 04000224
 05 00222 02000225
 05 00223 03000226
 1490 05 00224 6A700287

(SEQ2,PMD25)

BAL, LNK ADJINT

*
 * ONLY ONE SEQ. # GIVEN; DUPLICATE IT
 *

1491
 1492
 1493
 1494 05 00225 6A70028B

PMD20 BAL, LNK REPSEQ

*
 * PUT 2ND SEQ # PAIR IN CDT AND GO PROCESS INCREMENT
 *

1495
 1496
 1497
 1498 05 00226 6A70081B
 1499 05 00227 68000117

PMD25 BAL, LNK ADDCDTPARAM PUT 1SEQ # PAIR IN CDT
 B GET\$INCREMENT GO PROCESS INCR

PAGE

* PARSE FORM: RN N,K *

*
*

1500
1501
1502
1503
1504
1505
1506 05 0022B
1507 05 0022B 6A700963
1508 05 00229 00000002 A
1509 05 0022A 6A700834
1510
1511
1512
1513 05 0022B 6A7008D4
05 0022C 0300023D 02
05 0022D 04000230
05 0022E 02000231
05 0022F 03000129
1514 05 00230 6A700287
1515
1516
1517
1518 05 00231 22500002 A
1519 05 00232 6A70081B
1520
1521
1522
1523 05 00233 6A7008D4
05 00234 03800239 02
05 00235 07000238
05 00236 08000242
05 00237 00000242
1524
1525
1526 05 00238 6A7008D4
05 00239 0200023D 02
05 0023A 0400023C

PARSE:RN EGU *
BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
DATA 2
BAL, LNK CHECK1CDTENTRY MAKE SURE 'RN' IS FIRST CMND
NXTPRM ERRP5,,
(INTG,*),,
(SEQ,PRN10),,
(SEQ2,ILGL#SEQ2)

BAL, LNK ADJINT

*
* PUT SEQ # IN CDT AND GET 2ND SEQ #

*
PRN10 LI,P1 SEQ PUT SEQ # IN CDT
BAL, LNK ADDCDTPARAM
NXTPRM *ERRP4,,
(COM,*),,
(SCBL,PRN30),,
(END,PRN30)

NXTPRM ERRP5,,
(INTG,*),,
(SEQ,PRN20)

1527 05 0023B 0200023D
 1528 05 0023C 6A700287
 1529
 1530
 1531 05 0023D 6A70081B
 1532
 1533
 1534 05 0023E 6A7008D4
 05 0023F 02800239 02
 05 00240 08000035
 05 00241 0U000301
 1535
 1536
 1537
 1538 05 00242 6A700D99
 1539 05 00243 0U000271 02
 1540 05 00244 68000004

BAL, LNK ADJINT
 *
 * PUT 2ND SEQ # IN CDT AND FINISH UP
 *
 PRN20 BAL, LNK ADDCDTPARAM PUT 2ND SEQ # IN CDT
 NXTPRM *ERRP4,,
 (SCOL, ILGL#SEMICOLON),,
 (END, MASTEREXECUTIVE)
 *
 * ERROR: SECOND PARAMETER MISSING
 *
 PRN30 BAL, LNK TYPEPERR TYPE: !-PN;PARAM MISSING!
 DATA ERRP17
 B MASTERPARSER GO TO PARSER

PAGE

```

1541
1542
1543
1544
1545
1546
1547
1548
1549      05 00245
1550      05 00245
1551      05 00245      6A700963
1552      05 00246      0U000003 A
1553      05 00247      6800024A
1554
1555
1556      05 00248
1557      05 00248      6A700963
1558      05 00249      0U000001 A
1559
1560
1561
1562      05 0024A      6A7008D4
1563      05 0024B      0300023D 02
1564      05 0024C      0400024F
1565      05 0024D      02000250
1566      05 0024E      03000129
1567      05 0024F      6A700287
1568
1569
1570
1571
1572      05 00255      680001F2
1573

```

```

*****
* PARSE FORMS:  SS N(,C(,D)) *
*                ST N(,C(,D)) *
*                JU N          *
*****
*
* PARSE:ISS EGU          $
* PARSE:ST EGU          $
*                BAL,LNK NEWCDTENTRY      BUILD NEW CDT ENTRY
*                DATA    3
*                B        PARSE:JU+2
*
*
* PARSE:JU EGU          $
*                BAL,LNK NEWCDTENTRY      BUILD NEW CDT ENTRY
*                DATA    1
*                NXTPRM  ERRP5,,
*                        (INTG,*,),
*                        (SEQ,PSS10),,
*                        (SEQ2,ILGL$SEQ2)
*
*                BAL,LNK  ADJINT
*
* PUT SEQ # IN CDT AND MAKE SURE CMND IS FIRST FOR 'SS' AND 'ST'
*
PSS10  CI,P1  FIRST$I:CMND  IS CMND='JU'
      BGE    PSS20
      BAL,LNK CHECK1CDTENTRY  NO - MAKE SURE ,SS(ST), IS 1ST CMND
      LI,P1  SEQ          PUT SEQ # IN CDT
      BAL,LNK ADDCDTPARAM
      B      GET$COL#$PAIR
*

```

MO1 20:44 SEP 08, '75

127

1574
 1575
 1576 05 00256 22500002 A
 1577 05 00257 6A70081B
 1578
 1579
 1580 05 00258 6A7008D4
 05 00259 02800239 02
 05 0025A 08000035
 05 0025B 0U000301

* PUT SEQ # FOR 'JU' IN CDT, BUT IT NEED NOT BE FIRST
 *
 PSS20 LI,P1 SEQ PUT SEQ # IN CDT
 BAL, LNK ADDCDTPARAM
 NTPRM *ERRP4,,
 (SC0L, ILGL*SEMICOLON),,
 (END, MASTEREXECUTIVE)

PAGE

1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588 05 0025C
 1589 05 0025C
 1590 05 0025C 6A700963
 1591 05 0025D 00000000 A
 1592
 1593
 1594 05 0025E 6A7008D4
 05 0025F 020001C8 02
 05 00260 0800001F
 05 00261 00000301

```

*****
* PARSE FORM: RF *
* PARSE FORM: TX *
*****
*
*
PARSE:TX EGU      $
PARSE:RF EGU      $
           BAL, LNK NEWCDENTRY
           DATA   0
           NXTPRM  ERRCS,,
                (SCBL,RESUME$PARSING),,
                (END,MASTEREXECUTIVE)

```

PAGE

```
*****
* PARSE FORMS: TS N(=M) & TS *
* TY N(=M) & TN *
* TC N(=M) *
*****
```

```
1595
1596
1597
1598
1599
1600
1601
1602
1603      05 00262
1604 05 00262 6A700963
1605 05 00263 00000003 A
1606
1607
1608
1609 05 00264 6A7008D4
      05 00265 0300023D 02
      05 00266 04000281
      05 00267 02000282
      05 00268 03000283
```

```
PARSE:TC EQU $
BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
DATA 3
NXTPRM ERRP5,, 'TC' MUST SPECIFY RECORD.
(INTG,PTY5),,
(SEQ,PTY10),,
(SEQ2,PTY15)
```

```
1610
1611      05 00269
1612      05 00269
1613 05 00269 6A700963
1614 05 0026A 00000003 A
1615
1616
1617
1618
1619
1620 05 0026B 6A7008D4
      05 0026C 050001C8 02
      05 0026D 04000281
      05 0026E 02000282
      05 0026F 03000283
      05 00270 08000272
      05 00271 00000272
1621 05 00272 33F000AD 02
```

```
*
PARSE:TS EQU $
PARSE:TY EQU $
BAL, LNK NEWCDTENTRY BUILD NEW CDT ENTRY
DATA 3
NXTPRM ERRC9,,
(INTG,PTY5),,
(SEQ,PTY10),,
(SEQ2,PTY15),,
(SCBL,*),,
(END,*)
MTW=-1 CHARPSN SET TO RESCAN LAST CHAR
```

HC1 20:44 SEP 08, '75

190

1622 05 00273 B28000AC 02
 1623 05 00274 40800005 02
 1624 05 00275 B08000AC 02
 1625 05 00276 2280002A A
 1626 05 00277 21500015 A
 1627 05 00278 6030027A
 1628 05 00279 2280002B A
 1629 05 0027A 32500008 A
 1630 05 0027B 6A700963
 1631 05 0027C 00000000 A
 1632
 1633
 1634 05 0027D 6A7008D4
 05 0027E 020001C8 02
 05 0027F 0800001F
 05 00280 00000301
 1635
 1636
 1637 05 00281 6A700287
 1638 05 00282 6A70028B
 1639 05 00283 6A700834
 1640
 1641 05 00284 22500003 A
 1642 05 00285 6A70081B
 1643
 1644 05 00286 600001F2

*
 *
 *
 *
 *
 *
 *

LW,T1 *CDTADR
 AND,T1 XFF00
 STW,T1 *CDTADR
 LI,T1 I;TS*CMND*NMR
 CI,P1 R;TS*CMND*NMR
 BE **2
 LI,T1 I;TY*CMND*NMR
 LW,P1 T1
 BAL,LNK NEWCDTENTRY
 DATA 0
 NXTPRM ERRC9,,
 (SCBL,RESUME*PARSING),,
 (END,MASTEREXECUTIVE)
 BAL,LNK ADJINT
 BAL,LNK REPSEQ
 BAL,LNK CHECK1CDTENTRY
 LI,P1 SEQ2
 BAL,LNK ADDCDTPARAM
 B GET*COL*PAIR

MUST BE INTRALINE ITS! OR ITY! S0
 WIPE OUT CDT ENTRY JUST BUILT
 IS CMND ITS!
 NO - MUST BE ITY!
 BUILD ENTRY IN CDT FOR THIS CMND
 SCALE INTEGER TO SEQ #.
 REPLICATE SINGLE SEQ #.
 INSURE TY(TS) IS FIRST COMMAND.
 ADD SEQ # PAIR TO COMMAND TABLE.
 NOW GET OPTIONAL COLUMN NUMBERS.

PAGE

* PARSE UTILITY ROUTINES *

*
* FORM SEQUENCE NUMBER AS INTEGER*1000.

1645						
1646						
1647						
1648						
1649						
1650						
1651						
1652	05	00287	32F000C1 02	ADJINT	LW,D1	PARAMBUF
1653	05	00288	24F003E8 A		MI,D1	1000
1654	05	00289	35F000C1 02		STW,D1	PARAMBUF
1655	05	0028A	E8000007 A		B	*LNK
1656						
1657						
1658						
1659	05	0028B	328000C1 02	REPSEQ	LW,T1	PARAMBUF
1660	05	0028C	358000C2 02		STW,T1	PARAMBUF+1
1661	05	0028D	331000E6 02		MTW,1	PRMBUFSZ
1662	05	0028E	E8000007 A		B	*LNK
1663						
1664						

*
* REPLICATE SINGLE SEQUENCE NUMBER IN PARAMBUF+1.
*

PAGE

```

1665
1666
1667
1668
1669
1670
1671
1672
1673          00000001
1674 05 0028F 0910019C 02 BRK$KEY DB      MODE=2
1675 05 00290 04800E26          PUSH      X3          SAVE POINTER OF PSD IN STACK
1676 05 00291 6A700DC4          CAL1,8    #'X'06700002'  PURGE TERMINAL OUTPUT BUFFERS
1677 05 00292 0000027A 02     BAL,LNK   TYPMSG      MOVE TO A CLEAN LINE ON USER
1678 05 00293 390002C9 02     DATA    MSGO        TERMINAL.
1679 05 00294 691002FC          MTW,0    XEQFLAG     IF NOT EXECUTING, GET NEXT COMMAND.
1680 05 00295 39000167 02     BLZ      BRK99
1681 05 00296 693002D2          MTW,0    STEPFLAG    IF STEPPING, SKIP DISPLAY CHECK.
1682 05 00297 390000E9 02     BNEZ     BRK80
1683 05 00298 683002A3          MTW,0    SETFLAG    IS SYSTEM IN SET MODE=
1684          *          BEZ      BRK30    ZERO SAYS NO.
1685 05 00299 325002C3 02     *          LW,P1     INTFLAG1   IF DISPLAY FLAG SET,
1686 05 0029A 691002D2          BLZ      BRK80
1687 05 0029B 22600BC9 02     LI,P2    BA(UTSM4)+37  BUILD SEQ # INTO MESSAGE AND
1688 05 0029C 6A700B97          BAL,LNK  MOVESEG     SEND IT OUT.
1689 05 0029D 00000000 A      GEN4     0,0,0,0
1690 05 0029E 20800024 A      AI,R1    36          ADJUST COUNT OF FULL STRING
1691 05 0029F 758002E9 02     STB,R1   UTSM4
1692 05 002A0 6A700DC4          BAL,LNK  TYPMSG
1693 05 002A1 000002E9 02     DATA    UTSM4
1694 05 002A2 680002D2          B        BRK80    NOW ASK ABOUT CONTINUE.
1695
1696          *
1697 05 002A3 22300001 A      BRK30    LI,X1     1          EXECUTING FILE OR EDIT COMMAND.
1698 05 002A4 F24600AC 02     LB,X2    *CDTADR,X1  GET COMMAND NUMBER AND RETRIEVE
1699 05 002A5 683002FC          BEZ      BRK99    ORIGINAL EBCDIC.
1700 05 002A6 32380039          LW,X1    CNAMETBL,X2  NOW CHECK DISPLAY TABLE FOR
1701 05 002A7 22400007 A      LI,X2    7          PRESENCE OF THIS COMMAND.

```

HC1 20144 SEP 08, 175

1702	05	002A8	31380317	02		CW,X1	BDISPTBL,X2
1703	05	002A9	683002AC			BE	BRK40
1704	05	002AA	644002A8			BDR,X2	*-2
1705	05	002AB	680002D2			B	BRK80
1706					*		
1707	05	002AC	21400004	A	BRK40	CI,X2	4
1708	05	002AD	682002B8			BLE	BRK50
1709	05	002AE	325002C3	02		LW,P1	INTFLAG1
1710	05	002AF	691002D2			BLZ	BRK80
1711	05	002B0	22600BF0	02		LI,P2	BA(UTSM5)+24
1712	05	002B1	6A700B97			BAL,LNK	MOVESEQ
1713	05	002B2	00000000	A		GEN4	0,0,0,0
1714	05	002B3	20B00017	A		AI,R1	23
1715	05	002B4	75B002F6	02		STB,R1	UTSM5
1716	05	002B5	6A700DC4			BAL,LNK	TYPEMSG
1717	05	002B6	000002F6	02		DATA	UTSM5
1718	05	002B7	680002D2			B	BRK80
1719					*		
1720	05	002B8	325002C3	02	BRK50	LW,P1	INTFLAG1
1721	05	002B9	691002D2			BLZ	BRK80
1722	05	002BA	22600BF0	02		LI,P2	BA(UTSM5)+24
1723	05	002BB	6A700B97			BAL,LNK	MOVESEQ
1724	05	002BC	40400000	A		GEN4	BL,LP,0,0
1725	05	002BD	3060000B	A		AW,P2	R1
1726	05	002BE	20B00017	A		AI,R1	23
1727	05	002BF	325002C4	02		LW,P1	INTFLAG2
1728	05	002C0	681002CD			BGEZ	BRK60
1729	05	002C1	72300307	02		LB,X1	UTSM7
1730	05	002C2	30B00003	A		AW,R1	X1
1731	05	002C3	22400001	A		LI,X2	1
1732	05	002C4	72580307	02	BRK53	LB,P1	UTSM7,X2
1733	05	002C5	755C0000	A		STB,P1	0,P2
1734	05	002C6	20600001	A		AI,P2	1
1735	05	002C7	20400001	A		AI,X2	1
1736	05	002C8	643002C4			BDR,X1	BRK53
1737	05	002C9	75B002F6	02	BRK55	STB,R1	UTSM5
1738	05	002CA	6A700DC4			BAL,LNK	TYPEMSG

IF NOT FOUND,
ASK TO CONTINUE.

THESE COMMANDS TAKE SINGLE SEQUENCE
NUMBER = DE,FD,FT

THESE COMMANDS TAKE A DOUBLE SEQ. #
DISPLAY

SET UP DDD.DD ()
INCR MSG BYTE ADDR
AND MSG LENGTH
IF SECOND SEQ. # NOT SET UP,
WE MUST BE DELETING.

THEREFORE, INSERT DELETING
MESSAGE.

ADJUST BYTE COUNT OF TOTAL
MESSAGE.

1739	05	002CB	000002F6	02		DATA	UTSM5	
1740	05	002CC	680002D2			B	BRK80	THEN ASK ABOUT CONTINUE.
1741								
1742	05	002CD	3280000B	A	* BRK60	LW,T1	R1	SAVE MSG LENGTH
1743	05	002CE	6A700B97			BAL, LNK	MOVESEQ	MOVE SECOND SEQ # NUMBER INTO
1744	05	002CF	50000000	A		GEN4	RP,0,0,0	MESSAGE.
1745	05	002D0	30B00008	A		AW,R1	T1	INCREMENT MSG LENGTH
1746	05	002D1	680002C9			B	BRK55	
1747								
1748	05	002D2	6A700DC4		* BRK80	BAL, LNK	TYPMSG	ASK FOR A CHARACTER.
1749	05	002D3	00000303	02		DATA	UTSM6	
1750	05	002D4	041002CD	02		CAL1,1	BR#FPT	READ IT.
1751	05	002D5	6A700DC4			BAL, LNK	TYPMSG	RETURN CARRIAGE.
1752	05	002D6	0000027A	02		DATA	MSG0	
1753	05	002D7	723002C2	02		LB,X1	CFLAG	IF CHARACTER IS NOT X
1754	05	002D8	213000E7	A		CI,X1	'X'	CONTINUE COMMAND.
1755	05	002D9	683002F7			BE	STOPLASTCMD	
1756	05	002DA	0810019C	02		PULL	X3	STRAIGHTEN OUT STACK
1757	05	002DB	04900005	A	M:TRTN	M:TRTN		CONTINUE
1758	05	002DC	22800000	A	BRK90	LI,T1	0	START CLEAN UP
1759	05	002DD	358000BD	02		STW,T1	LASTKEY	
1760	05	002DE	358000C0	02		STW,T1	NOCHGFLG	
1761	05	002DF	358000E9	02		STW,T1	SETFLAG	
1762	05	002E0	35800167	02		STW,T1	STEPFLAG	
1763	05	002E1	228FFFFFF	A		LI,T1	=1	
1764	05	002E2	35800020	02		STW,T1	ALLFLAG	
1765	05	002E3	330000AE	02		MTW,0	COPYFL	DOES FID1=FID2
1766	05	002E4	683002ED			BEZ	BRK91	NO
1767	05	002E5	32B00E27			LW,R1	L('X'00200000')	CHECK IF FILES GOT OPEN
1768	05	002E6	31B00000	X		CW,R1	F:EI	(INPUT)
1769	05	002E7	684002E9			BAZ	*+2	
1770	05	002E8	6A700B33			BAL, LNK	CLOSE	CLOSE,SAVE OLD FILE
1771	05	002E9	31B00000	X		CW,R1	F:EO	(OUTPUT)
1772	05	002EA	684002EC			BAZ	*+2	
1773	05	002EB	6A700B37			BAL, LNK	CLOSE3	CLOSE,RLS NEW FILE
1774	05	002EC	68000004			B	MASTERPARSER	
1775		05 002ED			BRK91	EGU	*	

1776	05	002ED		D81	MODE=2	
1777	05	002ED	32800E27	LW,R1	L(X'00200000')	IF OPEN FOR OUTPUT,
1778	05	002EE	31800000 X	CW,R1	F:EO	
1779	05	002EF	684002F1	BAZ	*+2	
1780	05	002FO	6A700B35	BAL,LNK	CLOSE2	CLOSE ANY COPY OR MERGE FILE.
1781	05	002F1	330000B6 02	MTW,0	FILETYPE	CLOSE INPUT FILE, UNLESS OPEN
1782	05	002F2	69200004	BGZ	MASTERPARSER	FOR EDIT.
1783	05	002F3	31800000 X	CW,R1	F:EI	
1784	05	002F4	684002F6	BAZ	*+2	
1785	05	002F5	6A700B33	BAL,LNK	CLOSE	
1786	05	002F6	68000004	B	MASTERPARSER	
1787	05	002F7	0810019C 02	STOPLASTCMD PULL	X3	POINTER OF PSD IN STACK.
1788	05	002F8	2291FFFF A	LI,T2	X'1FFFF'	SET A MASK
1789	05	002F9	228002DC	LI,T1	BRK90	RETURN ADR. WANTED.
1790	05	002FA	47820000 A	STS,T1	0,X3	
1791	05	002FB	680002DB	B	M:TRTN	
1792	05	002FC		BRK99 EQU	*	PREPARE A CLEAN EXIT.
1793	05	002FC	0810019C 02	PULL	X3	GET THE STACK POINTER.
1794	05	002FD	2291FFFF A	LI,T2	X'1FFFF'	MASK
1795	05	002FE	22800004	LI,T1	MASTERPARSER	ADR. OF RETURN.
1796	05	002FF	47820000 A	STS,T1	0,X3	
1797	05	00300	680002DB	B	M:TRTN	
1798				FIN		

PAGE

```
*****
*
*   M A S T E R   P R O G R A M
*
*   T O   E X E C U T E
*
*   E D I T   C O M M A N D S
*
*****
```

```
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811      05 00301
1812 05 00301 22800049 02
1813 05 00302 358000AC 02
1814 05 00303 3280016A 02
1815 05 00304 35800023 02
1816      00000001
1817 05 00305 22800001 A
1818 05 00306 358002C9 02
1819 05 00307 330000E9 02
1820 05 00308 6930030C
1821 05 00309 22800000 A
1822 05 0030A 358002C6 02
1823 05 0030B 358002C7 02
1824
1825
1826
1827
1828      05 0030C
1829 05 0030C 22300000 A
1830 05 0030D 35300021 02
1831 05 0030E 22300001 A
1832 05 0030F F24600AC 02
1833 05 00310 6830034A
1834 05 00311 21400004 A
1835 05 00312 69100322
```

```
MASTEREXECUTIVE EQU *
LI,T1 CDT+1 SET CDTADR,FIRST COMMAND IN CDT
STW,T1 CDTADR
LW,T1 SVBPFLAG RESTORE LAST DFLT VALUE OF BPFLAG
STW,T1 BPFLAG
DB MODE=2
LI,T1 1
STW,T1 XEGFLAG
MTW,0 SETFLAG ARE WE IN SET MODE
BNEZ RESTART$EXECUTIVE YES, DON'T RESET TAB FLAGS
LI,T1 0
STW,T1 TABCFLAG
STW,T1 TABXFLAG
FIN

*
*
*
RESTART$EXECUTIVE EQU *
LI,X1 0 (INTRALINE CMND LOOP ENTERS HERE)
STW,X1 ALL0K INDICATE 'ALL' MODE IS
LI,X1 1 POTENTIALLY LEGAL
LB,X2 *CDTADR,X1 GET NUMBER OF COMMAND
BEZ EXC50 IS CMND=0 (END OF CDT)
CI,X2 FIRST$FICMND IS IT A FILE COMMAND
BL EXC6 NO - ITS A CONTRL CMND
```

1836	05	00313	2140000B A	CI,X2	FIRST,R:CMND	NO - IS IT A FILE COMMAND	
1837	05	00314	69100325	BL	EXC5	YES - SKIP TEST	
1838	05	00315	390000B6 02	MTW,0	FILETYPE	NO - IS INP FILE PRESENT AND KEYED	
1839	05	00316	69100347	BLZ	EXC40	NO - ERROR	
1840	05	00317	2140001E A	CI,X2	FIRST,I:CMND	IS IT AN I:CMND (EXCEPT 'SE')	
1841	05	00318	68200325	BLE	EXC5		
1842	05	00319	228FFFFE A	LI,T1	=2	IF ERRORCNT =2 OR LESS	
1843	05	0031A	318000B2 02	CW,T1	ERRORCNT	SET LOOP EXECUTED ONCE	
1844	05	0031B	6810031D	BGE	*+2	DONT TYPE ANY MORE CERRS	
1845	05	0031C	353000B2 02	STW,X1	ERRORCNT	ONE CERR PER I:COMMAND	
1846	05	0031D	390000E9 02	MTW,0	SETFLAG	YES - IS SYSTEM IN SET MODE	
1847	05	0031E	69300337	BNEZ	EXC20	YES - GO CHECK ON CMND	
1848	05	0031F	6A700DC4	BAL,LNK	TYPEMSG	NO - TYPE: 'MISSING SET'	
1849	05	00320	000001E8 02	DATA	ERRM8		
1850	05	00321	68000004	B	MASTERPARSER	EXIT TO PARSER	
1851				*			
1852				*	CONTROL COMMAND = 'BP', 'TA', 'CR'		
1853				*			
1854	05	00322	39000167 02	EXC6	MTW,0	STEPFLAG	IS SYSTEM IN STEP MODE
1855	05	00323	69300344	BNEZ	EXC30	YES - ERROR	
1856	05	00324	6800032B	B	EXC10	CONTINUE	
1857				*			
1858				*	F:CMND, R:CMND, OR 'SE': CHECK TO SEE THAT SYSTEM IS NOT IN STEP MODE		
1859				*			
1860	05	00325	39000167 02	EXC5	MTW,0	STEPFLAG	IS SYSTEM IN STEP MODE
1861	05	00326	69300344	BNEZ	EXC30	YES - ERROR	
1862	05	00327	22800000 A	LI,T1	0	TURN OFF 'SET MODE' FLAG	
1863	05	00328	358000E9 02	STW,T1	SETFLAG		
1864		00000001		D0	MODE=2		
1865	05	00329	358002C6 02	STW,T1	TABCFLAG	AND RESET TAB FLAGS	
1866	05	0032A	358002C7 02	STW,T1	TABXFLAG		
1867				FIN			
1868				*			
1869				*	EXECUTE CURRENT COMMAND IN CDT		
1870				*			
1871	05	0032B	67080351	EXC10	EXU	CMNDTBL,X2	EXECUTE COMMAND
1872	05	0032C	39000020 02	MTW,0	ALLFLAG	WAS CMND AN I:CMND WITH PARAM1=ALL	

1873	05	0032D	68100334	BGEZ	EXC15	YES - EXECUTE IT UNTIL FLAG GOES OFF
1874	05	0032E	22800000 A	LI,T1	0	TURN OFF COPY FLAG
1875	05	0032F	358000AE 02	STW,T1	COPYFL	
1876	05	00330	358001A1 02	STW,T1	CT\$FLAG	TURN OFF 'CT' FLAG
1877	05	00331	F28000AC 02	LB,T1	*CDTADR	INCR CDTADR TO NEXT COMMAND
1878	05	00332	668000AC 02	AWM,T1	CDTADR	
1879	05	00333	6800030C	B	RESTART\$EXECUTIVE GO PROCESS NEW COMMAND	
1880				*		
1881				*	ALLFLAG SET: RE-EXECUTE INTRALINE COMMAND UNTIL ALL OCCURRENCES ARE	
1882				*	PROCESSED	
1883				*		
1884	05	00334	22300001 A	EXC15	LI,X1 1	GET NUMBER OF COMMAND
1885	05	00335	F24600AC 02		LB,X2 *CDTADR,X1	
1886	05	00336	6800032B		B EXC10	GO EXECUTE COMMAND
1887				*		
1888				*	COMMAND IS INTRALINE (EXCEPT 'SE'): TURN ON 'I:CMND EXECUTED' FLAG;	
1889				*	IF COMMAND IS FIRST IN CDT DO A DUMMY I:SET USING PARAMETERS FROM	
1890				*	LAST ACTUAL I:SET	
1891				*		
1892	05	00337	39000167 02	EXC20	MTW,0 STEPFLAG	IS SYSTEM IN STEP MODE
1893	05	00338	6930032B		BNEZ EXC10	
1894	05	00339	390000E9 02		MTW,0 SETFLAG	NO - MUST SET LOOP BE INITIALIZED
1895	05	0033A	6910032B		BLZ EXC10	NO - GO EXECUTE I:CMND
1896	05	0033B	228FFFFFF A		LI,T1 *1	SET SETFLAG=*1 TO INDICATE THAT SET
1897	05	0033C	358000E9 02		STW,T1 SETFLAG	LOOP HAS BEEN INITIALIZED
1898	05	0033D	328000AC 02		LW,T1 CDTADR	SAVE ADDR OF CMND IN CDT (IN
1899	05	0033E	358000E8 02		STW,T1 SETADR	SETADR) FOR LATER I:CMND LOOP
1900	05	0033F	32500169 02		LW,P1 SV1STSET	
1901	05	00340	35500089 02		STW,P1 FIRSTSET	INITIALIZE LOOP PER LAST I:SET
1902	05	00341	6A700C6D		BAL,LNK READRANDOM	READ FIRST RECORD TO ALTER
1903	05	00342	6A700A56		BAL,LNK SETEBD	SET EBD MARKER
1904	05	00343	6800032B		B EXC10	GO EXECUTE COMMAND
1905				*		
1906				*	ERROR: GIVEN COMMAND IS ILLEGAL WHEN SYSTEM IS IN STEP MODE	
1907				*		
1908	05	00344	6A700D8F	EXC30	BAL,LNK TYPECERR	TYPE: 'CN:CMND ILGL HERE'
1909	05	00345	000001B1 02		DATA ERR4	

```

1910 05 00346 68000004 B MASTERPARSER EXIT TO PARSER
1911 *
1912 * ERROR: NO SOURCE FILE NAMED
1913 *
1914 05 00347 6A700DC4 EXC40 BAL,LNK TYPEMSG TYPE: I-NO FILE NAMED!
1915 05 00348 000001F2 02 DATA ERRM13
1916 05 00349 68000004 B MASTERPARSER EXIT TO PARSER
1917 *
1918 * END OF CDT: IF IN SET OR STEP MODES, GO TO APPROPRIATE LOOP
1919 *
1920 05 0034A 33000167 02 EXC50 MTW,0 STEPFLAG IS SYSTEM IN STEP MODE
1921 05 0034B 693006AD BNEZ STEP,LOOP YES = GO TO STEP LOOP
1922 05 0034C 3300019A 02 MTW,0 TTYIMGSZ WAS INPUT LINE NULL
1923 05 0034D 68300351 BEZ EXC55 YES = ERROR
1924 05 0034E 330000E9 02 MTW,0 SETFLAG IS SYSTEM IN SET MODE
1925 05 0034F 69300748 BNEZ SET,LOOP YES = GO TO SET LOOP
1926 05 00350 68000004 B MASTERPARSER EXIT TO PARSER
1927 *
1928 * ERROR: NULL COMMAND
1929 *
1930 05 00351 EXC55 EQU $
1931 05 00351 68000004 B MASTERPARSER EXIT TO PARSER
1932 *
1933 * FILE COMMANDS CAN ONLY APPEAR ONE PER LINE
1934 *
1935 05 00351 CMNDTBL EQU $=1
1936 05 00352 6AD0037E BAL,F;LNK F;BLANK,PRESERV 1: BP
1937 05 00353 6AD00534 BAL,F;LNK S(MASTERPARSER,F:TA) TA
1938 05 00354 6AD00481 BAL,F;LNK F;CR 3: CR
1939 05 00355 6AD00390 BAL,F;LNK F;BUILD 4: BUILD
1940 05 00356 6AD003D9 BAL,F;LNK F;COPY 5: COPY
1941 05 00357 6AD00497 BAL,F;LNK F;DELETE 6: DELETE
1942 05 00358 6AD004A3 BAL,F;LNK F;EDIT 7: EDIT
1943 05 00359 6AD004BA BAL,F;LNK F;END 8: END
1944 05 0035A 6AD004BE BAL,F;LNK F;MERGE 9: MERGE
1945 05 0035B 6AD0048C BAL,F;LNK F;RP 10: RP
1946 *

```

1947
 1948
 1949 05 0035C 6AD0058C
 1950 05 0035D 6AD00596
 1951 05 0035E 6AD00598
 1952 05 0035F 6AD005DB
 1953 05 00360 6AD005D8
 1954 05 00361 6AD00614
 1955 05 00362 6AD00616
 1956 05 00363 6AD00686
 1957 05 00364 6AD00699
 1958 05 00365 6AD0069B
 1959 05 00366 6AD006C5
 1960 05 00367 6AD006C3
 1961 05 00368 6AD006C3
 1962 05 00369 6AD00594
 1963 05 0036A 6AD00545
 1964 05 0036B
 1965
 1966
 1967
 1968 05 0036F 6AD0072D
 1969 05 00370 6AD00773
 1970 05 00371 6AD00787
 1971 05 00372 6AD00798
 1972 05 00373 6AD007A7
 1973 05 00374 6AD007B1
 1974 05 00375 6AD007BC
 1975 05 00376 6AD007CA
 1976 05 00377 6AD007D4
 1977 05 00378 6AD007ED
 1978 05 00379 6AD00800
 1979 05 0037A 6AD00808
 1980 05 0037B 6AD00810
 1981 05 0037C 6AD0080C
 1982 05 0037D 6AD00814

* RECORD COMMANDS CAN ONLY APPEAR ONE PER LINE

*
 BAL,R;LNK R:DELETE 11: DE
 BAL,R;LNK R:FIND\$DELETE 12: FD
 BAL,R;LNK R:FIND\$TYPE 13: FT
 BAL,R;LNK R:INSERT 14: IN
 BAL,R;LNK R:INSERT\$SUP\$SEQ 15: IS
 BAL,R;LNK R:MOVE\$DELETE 16: MD
 BAL,R;LNK R:MOVE\$KEEP 17: MK
 BAL,R;LNK R:RENUMBER 18: RN
 BAL,R;LNK R:SET\$STEP 19: SS
 BAL,R;LNK R:SET\$STEP\$TYPE 20: ST
 BAL,R;LNK R:TYPE\$SUP\$SEQ 21: TS
 BAL,R;LNK R:TYPE 22: TY
 BAL,R;LNK R:TYPE\$COMPRESSED 23: TC
 BAL,R;LNK R:FIND\$SEQUENCE 24: FS
 BAL,R;LNK R:COMMENTARY 25: CM\$CT
 RES 4

*
 * INTRALINE COMMANDS MAY BE COMPOUNDED ON ONE LINE
 *

BAL,I;LNK I:SET 30: SE (MUST BE FIRST I:CMND)
 BAL,I;LNK I:DELETE 31: D
 BAL,I;LNK I:OVERWR\$EXTEND 32: E
 BAL,I;LNK I:FOLLOW\$BY 33: F
 BAL,I;LNK I:SHIFT\$LEFT 34: L
 BAL,I;LNK I:OVERWRITE 35: O
 BAL,I;LNK I:PRECEDE\$BY 36: P
 BAL,I;LNK I:SHIFT\$RIGHT 37: R
 BAL,I;LNK I:SUBSTITUTE 38: S
 BAL,I;LNK I:JUMP 39: JU
 BAL,I;LNK I:NO\$CHANGE 40: NO
 BAL,I;LNK I:REVERSE\$BPFLAG 41: RF
 BAL,I;LNK I:TYPE\$SUP\$SEQ 42: TS
 BAL,I;LNK I:TYPE 43: TY
 BAL,I;LNK I:TYPEX 44: TX

PAGE

1983
 1984
 1985
 1986
 1987
 1988
 1989 05 0037E
 1990 05 0037E 22300005 A
 1991 05 0037F F24600AC 02
 1992 05 00380 B28800AC 02
 1993 05 00381 3180038E
 1994 05 00382 69300386
 1995 05 00383 22800001 A
 1996 05 00384 3580016A 02
 1997 05 00385 E800000D A
 1998
 1999
 2000
 2001 05 00386 3180038F
 2002 05 00387 6930038B
 2003 05 00388 22800000 A
 2004 05 00389 3580016A 02
 2005 05 0038A E800000D A
 2006
 2007
 2008
 2009 05 0038B 6A700DC4
 2010 05 0038C 000001E3 02
 2011 05 0038D E800000D A
 2012
 2013
 2014 05 0038E 02D6D540 A
 2015 05 0038F 03D6C6C6 A

```

*****
* FILE COMMAND: SET BLANK PRESERVATION *
*****
*
*
* F:BLANKSPRESERV EGU *
LI,X1 5 SET TO GET PARAMETER FROM CDT
LB,X2 *CDTADR,X1
LW,T1 *CDTADR,X2 GET 'ON' OR 'OFF' AS A TEXTC=STRING
CW,T1 BPV0N
BNE BPV5 IS STRING='ON'
LI,T1 1 YES = SET BPFLAG=1
STW,T1 SVBPFLAG
B *FILNK EXIT
*
* TEST FOR 'OFF'
*
BPV5 CW,T1 BPV0FF
BNE BPV10 IS STRING='OFF'
LI,T1 0 YES = SET BPFLAG=0
STW,T1 SVBPFLAG
B *FILNK EXIT
*
* ERROR: NOT ON OR OFF
*
BPV10 BAL,LNK TYPEMSG TYPE: !=NOT ON/OFF!
DATA ERRMS
B *FILNK EXIT
*
*
BPV0N TEXTC 'ON'
BPV0FF TEXTC 'OFF'
```

PAGE

 * FILE COMMAND: BUILD *

2016						
2017						
2018						
2019						
2020						
2021						
2022		05 00390		F:BUILD	EGU	*
2023	05 00390	6A700D78		BAL, LNK	TESTEDITACTIVE	CHECK IF EDIT FILE ACTIVE
2024	05 00391	22300005	A	LI, X1	5	
2025	05 00392	F25600AC	02	LB, P1	*CDTADR, X1	SET P1=ADR OF FID IN CDT
2026	05 00393	305000AC	02	AW, P1	CDTADR	
2027	05 00394			D01	MODE=2	
2028	05 00394	32C00005	A	LW, R2	P1	SAVE FID ADDRESS
2029	05 00395	6A700C36		BAL, LNK	OPENNEW	OPEN OUTPUT ONLY FILE
2030	05 00396	688003D5		BCR, 8	BLD40	DOES FILE ALREADY EXIST
2031	05 00397			D01	MODE=2	
2032	05 00397	041002CC	02	CAL1, 1	N0PR0MPT, FPT	
2033	05 00398	225003E8	A	LI, P1	DFLTSEG	N0 = SET P1=DEFAULT SEQ #
2034	05 00399	228003E8	A	LI, T1	1000	T1=1 (DEFAULT INCR)
2035	05 0039A	22300006	A	LI, X1	6	
2036	05 0039B	F28600AC	02	LB, R1	*CDTADR, X1	GET PARAM2 TYPE
2037	05 0039C	683003A0		BEZ	BLD5	TEST IF PARAM2 PRESENT
2038	05 0039D	20300001	A	AI, X1	1	
2039	05 0039E	F24600AC	02	LB, X2	*CDTADR, X1	YES = SET P1=SEQ # FROM CDT
2040	05 0039F	B25800AC	02	LW, P1	*CDTADR, X2	
2041						
2042						
2043						
2044	05 003A0	22300008	A	BLD5	LI, X1	8
2045	05 003A1	F28600AC	02	LB, R1	*CDTADR, X1	GET PARAM3 TYPE
2046	05 003A2	683003A6		BEZ	BLD08	TEST IF PARAM3 PRESENT
2047	05 003A3	20300001	A	AI, X1	1	
2048	05 003A4	F24600AC	02	LB, X2	*CDTADR, X1	YES = SET T1=INCR FROM CDT
2049	05 003A5	B28800AC	02	LW, T1	*CDTADR, X2	
2050						
2051		00000C01			D0	MODE=2
2052	05 003A6	330002C1	02	BLD08	MTW, 0	BUILDFLAG

2053	05	003A7	683003B2		BEZ	BLD12	
2054	05	003A8	21C00000	A	BLD10	CI,R2	0
2055	05	003A9	683003B2		BEZ	BLD12	HAVE WE BUILT FIRST LINE
2056	05	003AA	6A700B33		BAL, LNK	CLOSE	YES==BYPASS RE=OPEN
2057	05	003AB	4650000C	A	XW,P1	R2	CLOSE AND SAVE BUILD FILE
2058	05	003AC	6A700BC8		BAL, LNK	OPEN	POSITION FID FOR OPENING
2059	05	003AD	3250000C	A	LW,P1	R2	REOPEN IN INPUT SO ESC LEAVES INTACT
2060	05	003AE	22C00001	A	LI,R2	1	RESET P1 TO SEQ. #
2061	05	003AF	35C000B6	02	STW,R2	FILETYPE	NOW MARK FILE AS IF WE ARE IN
2062	05	003B0	22C00000	A	LI,R2	0	EDIT MODE = RECORD COMM O.K.
2063	05	003B1	35C002C5	02	STW,R2	TABERRFLAG	AND R2 SO WILL NOT RE=OPEN EB
2064					ELSE		EDIT WOULD DO THIS, SO SHALL BUILD
2065					*S* BLD08	EGU	\$
2066					*S* BLD10	EGU	\$
2067					FIN		
2068					* TYPE NEXT SEQ # AND READ INPUT LINE		
2069					*		
2070	05	003B2	6A700DD8		BLD12	BAL, LNK	TYPESEG TYPE 'DDDD,DDD'
2071	05	003B3	40080000	A	GEN4	BL,E0M,0,0	
2072			00000001		D0	MODE=2	
2073	05	003B4	22F00009	A	LI,D1	9	
2074					FIN		
2075	05	003B5	6A700CB2		BAL, LNK	READTELETYPE	READ INPUT LINE
2076	05	003B6	21B00001	A	CI,R1	1	
2077	05	003B7	683003D2		BE	BLD30	
2078	05	003B8	3240000B	A	LW,X2	R1	GET BYTE CNT., INTO INDEX REG.
2079	05	003B9	204FFFFFF	A	AI,X2	=1	MAKE X2 A BINARY COUNT
2080	05	003BA	72F80024	02	LB,D1	CARDIMG,X2	GET LAST BYTE INPUT
2081	05	003BB	22300004	A	LI,X1	GNTBL1SZ	GET THE COUNT OF DIFFERENT TYPE
2082					*		OF LINE TERMINATORS.
2083	05	003BC	71F608CD		CB,D1	GNTBL1,X1	FIND A CARRAGE RETURN
2084	05	003BD	683003C0		BE	\$*3	
2085	05	003BE	643003BC		BDR,X1	\$*2	
2086	05	003BF	680003C3		B	\$*4	
2087	05	003C0	22F00040	A	LI,D1	' '	BLANK OUT C/R
2088	05	003C1	75F80024	02	STB,D1	CARDIMG,X2	
2089	05	003C2	33F0000B	A	MTW,-1	R1	IF CR DECREMENT CHAR. COUNT.

HC1 20144 SEP 08, 175

144

2090 05 003C3 35B000E7 02
 2091 05 003C4 21B0008C A
 2092 05 003C5 682003C8
 2093 05 003C6 6A700DC4
 2094 05 003C7 000001DC 02
 2095
 2096
 2097
 2098 05 003C8 6A700A56
 2099 05 003C9 6A700E1F
 2100 05 003CA 30500008 A
 2101 05 003CB 315000BF 02
 2102 05 003CC 682003A8
 2103 05 003CD 6A700DC4
 2104 05 003CE 0000021D 02
 2105 05 003CF 21C00000 A
 2106 05 003D0 683003D2
 2107 05 003D1 6A700B33
 2108
 2109
 2110
 2111 05 003D2
 2112 00000001
 2113 05 003D2 330002C1 02
 2114 05 003D3 683004BA
 2115
 2116 05 003D4 E800000D A
 2117
 2118
 2119
 2120 05 003D5 6A700DC4
 2121 05 003D6 000001FA 02
 2122 05 003D7 6A700B33
 2123 05 003D8 680003D2

STW,R1 RECSIZE
 CI,R1 MAXCLMN
 BLE BLD25
 BAL,LNK TYPMSG NO - TYPE; 1=-OVERFLOW;
 DATA ERRM3
 *
 * WRITE INPUT LINE AND INCREMENT SEQ. #
 *
 BLD25 BAL,LNK SETEBD FINDS COL. OF LAST NON-BLANK
 BAL,LNK WRITERANDBM WRITE CARD IMAGE; P1 CONTAINS SEQ. #
 AW,P1 T1 INCREMENT SEQ. #
 CW,P1 MAXSEQ IS SEQ. NO. TOO BIG
 BLE BLD10 NO. GO READ MORE INPUT
 BAL,LNK TYPMSG YES.
 DATA ERRM20
 CI,R2 0 IF THIS IS BANG BUILD,
 BEZ BLD30 R2 IS NON-ZERO
 BAL,LNK CLOSE
 *
 * NULL INPUT LINE: EXIT
 *
 BLD30 FGU \$
 D0 MODE=2
 MTW,0 BUILDFLAG IF ENTERED BY BUILD COMMAND, EXIT
 BEZ FIEND TO TEL.
 FIN
 B *F;LNK EXIT
 *
 * ERROR: NAMED FILE ALREADY EXISTS
 *
 BLD40 BAL,LNK TYPMSG TYPE: 1=FILE EXISTS; CAN'T BUILD!
 DATA ERRM15
 BAL,LNK CLOSE CLOSE F;EI
 B BLD30

PAGE

 * FILE COMMAND: COPY *

2124
 2125
 2126
 2127
 2128
 2129
 2130 05 003D9
 2131 05 003D9 6A700D78
 2132 05 003DA 22100000 A
 2133 05 003DB 301000AE 02
 2134 05 003DC 22300005 A
 2135 05 003DD F25600AC 02
 2136 05 003DE 305000AC 02
 2137 05 003DF 22300009 A
 2138 05 003E0 F26600AC 02
 2139 05 003E1 306000AC 02
 2140 05 003E2 20500002 A
 2141 05 003E3 20600002 A
 2142
 2143
 2144
 2145 05 003E4 20500001 A
 2146
 2147 05 003E5 20600001 A
 2148 05 003E6 723A0000 A
 2149 05 003E7 683003EB
 2150 05 003E8 713C0000 A
 2151 05 003E9 693003ED
 2152 05 003EA 680003E4
 2153
 2154
 2155
 2156 05 003EB 713C0000 A
 2157 05 003EC 6830043E
 2158 05 003ED 22300007 A
 2159 05 003EE F24600AC 02
 2160 05 003EF B28800AC 02

*
 * F: COPY EQU *
 BAL, LNK TESTEDITACTIVE CHECK IF EDIT FILE ACTIVE
 LI, X3 0 INITIALIZE FLAG FOR
 STW, X3 COPYFL FID1=FID2
 LI, X1 5 OBTAIN FID 1 AND FID 2
 LB, P1 *CDTADR, X1 AS
 AW, P1 CDTADR BYTE
 LI, X1 9 ADDRESSES
 LB, P2 *CDTADR, X1 IN
 AW, P2 CDTADR REGISTERS
 SLS, P1 2 P1 AND
 SLS, P2 2 P2
 *
 * SEARCH LOOP TO DETERMINE IF FID1 = FID2
 *
 CPY1 AI, P1 1
 * (OK TO BYPASS TEXTC BYTE IN COMPR)
 AI, P2 1
 LB, X1 0, P1 GET FID 1 BYTE
 BEZ CPY1A QUIT WHEN END OF FID
 CB, X1 0, P2
 BNE CPY1B OR WHEN NOT EQUAL
 B CPY1 LOOP
 *
 * FINISH FID COMPARISON - FID STRING HAS ENDED
 *
 CPY1A CB, X1 0, P2 CHECK LAST BYTE
 BE CPY32
 CPY1B LI, X1 7 FIND OUT WHETHER BN
 LB, X2 *CDTADR, X1 OR OVER SPECIFIED
 LW, T1 *CDTADR, X2 T1='BN' OR 'OVER'

2198	05 0040C	6A700DFD	BAL, LNK	WRITE2	WRITE RECORD IN COPY FILE
2199		00000001	D8	MODE=2	
2200	05 0040D	35B002C3 02	STW, R1	INTFLAG1	
2201	05 0040E	35B002C4 02	STW, R1	INTFLAG2	
2202			FIN		
2203	05 0040F	68000406	B	CPY5	NO = LOOP
2204					
2205					
2206					
2207	05 00410	22300008 A	CPY10	LI, X1	11
2208	05 00411	F24600AC 02		LB, X2	*CDTADR, X1
2209	05 00412	B25800AC 02		LW, P1	*CDTADR, X2
2210	05 00413	228003E8 A		LI, T1	1000
2211	05 00414	2230000C A		LI, X1	12
2212	05 00415	F2C600AC 02		LB, R2	*CDTADR, X1
2213	05 00416	6830041A		BEZ	CPY15
2214	05 00417	20300001 A		AI, X1	1
2215	05 00418	F24600AC 02		LB, X2	*CDTADR, X1
2216	05 00419	B28800AC 02		LW, T1	*CDTADR, X2
2217					
2218					
2219					
2220	05 0041A	6A700C7F	CPY15	BAL, LNK	READSEQUEN
2221	05 0041B	31B00E28		CW, R1	L(EOF)
2222	05 0041C	6830042B		BE	CPY20
2223	05 0041D	6A700DFD		BAL, LNK	WRITE2
2224	05 0041E	69800471		BGS, 8	CPY50
2225		00000001		D8	MODE=2
2226	05 0041F	35B002C3 02		STW, R1	INTFLAG1
2227	05 00420	355002C4 02		STW, P1	INTFLAG2
2228				FIN	
2229	05 00421	30500008 A		AW, P1	T1
2230	05 00422	315000BF 02		CW, P1	MAXSEQ
2231	05 00423	6820041A		BLE	CPY15
2232	05 00424	6A700DC4		BAL, LNK	TYPMSG
2233	05 00425	0000021D 02		DATA	ERRM20
2234	05 00426	330000AE 02		MTW, 0	COPYFL

*
* PROCESS STARTING SEQ. # AND INCREMENT PARAMETERS
*

SET P1=STARTING SEQ #
T1=1 (DEFAULT INCR)

GET PARAM4 TYPE
TEST IF PARAM4 PRESENT

YES = SET T1=INCR FROM CDT

*
* COPY AND RESSEQUENCE SOURCE FILE THROUGH EOF
*

READ SOURCE RECORD
IS IT AN EOF
YES = GO FINISH UP
WRITE RECORD IN COPY FILE
DOES RECORD ALREADY EXIST

NO = INCR SEQ #
IS SEQ. NO. TOO BIG
NO.
YES.
DOES FID1=FID2

2235	05	00427	6830042B	BEZ	CPY20	NO, SAVE BOTH	
2236	05	00428	6A700B33	BAL, LNK	CLOSE	OTHERWISE, SAVE ORIGINAL	
2237	05	00429	6A700B37	BAL, LNK	CLOSE3	AND DELETE COPY	
2238	05	0042A	E800000D A	B	*FILNK	EXIT	
2239				*			
2240				*	EOF FOUND; CLOSE COPY FILE AND EXIT		
2241				*			
2242	05	0042B	6A700B33	CPY20	BAL, LNK	CLOSE	CLOSE INPUT FILE
2243	05	0042C	6A700B35		BAL, LNK	CLOSE2	CLOSE COPY FILE
2244	05	0042D	6A700DC4		BAL, LNK	TYPEMSG	TYPE: ..., COPY DONE
2245	05	0042E	0000027E 02		DATA	MSG2	
2246	05	0042F	E800000D A		B	*FILNK	EXIT
2247				*			
2248				*	OPEN FOR COPY A ON B		
2249				*			
2250	05	00430	22300009 A	CPY30	LI, X1	9	
2251	05	00431	F25600AC 02		LB, P1	*CDTADR, X1	P1=ADR OF FID2 IN CDT
2252	05	00432	305000AC 02		AW, P1	CDTADR	
2253	05	00433	6A700BEF		BAL, LNK	OPEN2	OPEN INPUT=CHNGD TO OUT
2254	05	00434	68800466		BCR, 8	CPY35	ERROR IF FILE 2 EXISTS
2255	05	00435	22300005 A		LI, X1	5	OBTAIN FID 1 AND FID 2
2256	05	00436	F25600AC 02		LB, P1	*CDTADR, X1	AS
2257	05	00437	305000AC 02		AW, P1	CDTADR	BYTE
2258	05	00438	22200000 A		LI, X4	0	X4=4 MEANS NOT KEYED
2259	05	00439	6A700BCC		BAL, LNK	OPEN1	
2260	05	0043A	6980046A		BCS, 8	CPY36	IF FILE DOES NOT EXIST
2261	05	0043B	69400401		BCS, 4	CPY3	IS FILE KEYED
2262	05	0043C	22200001 A		LI, X4	1	MARK AS KEYED
2263	05	0043D	68000401		B	CPY3	GO TO BODY OF COPY
2264				*			
2265				*	OPEN FOR COPY A OVER A OR A ON A		
2266				*			
2267	05	0043E	22300005 A	CPY32	LI, X1	5	CHECK FID1 FOR PASSWORD
2268	05	0043F	331000AE 02		MTW, 1	COPYFL	SET TO SHOW FID1=FID2
2269	05	00440	F25600AC 02		LB, P1	*CDTADR, X1	
2270	05	00441	305000AC 02		AW, P1	CDTADR	P1 = FILE NAME
2271	05	00442	F2300005 A		LB, X1	*P1	BYTE CNT OF FILE NAME

2272	05	00443	2530007E A	SLS,X1	=2	BYTE TO WORD COUNT
2273	05	00444	20300001 A	AI,X1	1	GET NEXT WORD
2274	05	00445	30500003 A	AW,P1	X1	
2275	05	00446	F2300005 A	LB,X1	*P1	BYTE CNT OF ACCOUNT
2276	05	00447	2530007E A	SLS,X1	=2	
2277	05	00448	20300001 A	AI,X1	1	POINT TO PASSWORD
2278	05	00449	30500003 A	AW,P1	X1	
2279	05	0044A	B2300005 A	LW,X1	*P1	FETCH PASSWORD
2280	05	0044B	6930047E	BNEZ	CPY60	PASSWORD GIVEN = ERROR
2281	05	0044C	22300009 A	LI,X1	9	NOW GO TO SAME FOR FID2
2282	05	0044D	F25600AC 02	LB,P1	*CDTADR,X1	
2283	05	0044E	305000AC 02	AW,P1	CDTADR	P1 = FILE NAME
2284	05	0044F	F2300005 A	LB,X1	*P1	BYTE CNT OF FILE NAME
2285	05	00450	2530007E A	SLS,X1	=2	BYTE TO WORD COUNT
2286	05	00451	20300001 A	AI,X1	1	GET NEXT WORD
2287	05	00452	30500003 A	AW,P1	X1	
2288	05	00453	F2300005 A	LB,X1	*P1	BYTE CNT OF ACCOUNT
2289	05	00454	2530007E A	SLS,X1	=2	
2290	05	00455	20300001 A	AI,X1	1	POINT TO PASSWORD
2291	05	00456	30500003 A	AW,P1	X1	
2292	05	00457	B2300005 A	LW,X1	*P1	FETCH PASSWORD
2293	05	00458	6930047E	BNEZ	CPY60	PASSWORD GIVEN = ERROR
2294						
2295	05	00459	22300005 A	LI,X1	5	OBTAIN FID 1 AND FID 2
2296	05	0045A	F25600AC 02	LB,P1	*CDTADR,X1	AS
2297	05	0045B	305000AC 02	AW,P1	CDTADR	BYTE
2298	05	0045C	6A700BEB	BAL,LNK	OPEN3	OPEN FOR OUTPUT
2299	05	0045D	22300005 A	LI,X1	5	OBTAIN FID1 AND FID2
2300	05	0045E	F25600AC 02	LB,P1	*CDTADR,X1	AS
2301	05	0045F	305000AC 02	AW,P1	CDTADR	BYTE
2302	05	00460	22200000 A	LI,X4	0	X4=4 MEANS NOT KEYED
2303	05	00461	6A700BCC	BAL,LNK	OPEN1	OPEN1 OPEN1 IN. CONTINUE
2304	05	00462	6980046A	BCS,8	CPY36	IF FILE DOES NOT EXIST
2305	05	00463	69400401	BCS,4	CPY3	IS FILE KEYED
2306	05	00464	22200001 A	LI,X4	1	MARK AS KEYED
2307	05	00465	68000401	B	CPY3	GO TO BODY OF COPY
2308						

2309				* ERROR: COPY FILE EXISTS AND PARAMETER 2 IS 'BN'	
2310				*	
2311	05	00466	6A700DC4	CPY35	BAL, LNK TYPEMSG TYPE: 'P2:FILE EXISTS'
2312	05	00467	0000025D 02		DATA ERRP13
2313	05	00468	6A700B35		BAL, LNK CLOSE2
2314	05	00469	E800000D A		B *FILNK EXIT
2315				*	
2316				* ERROR: SOURCE FILE NAMED DOESN'T EXIST	
2317				*	
2318	05	0046A	6A700B37	CPY36	BAL, LNK CLOSE3
2319	05	0046B	6800046E		B CPY40
2320	05	0046C	6A700B35	CPY37	BAL, LNK CLOSE2 CLOSE E8 WITH SAVE
2321	05	0046D	6A700B33		BAL, LNK CLOSE
2322	05	0046E	6A700DC4	CPY40	BAL, LNK TYPEMSG TYPE: 'P1:NO SUCH FILE'
2323	05	0046F	00000258 02		DATA ERRP12
2324	05	00470	E800000D A		B *FILNK EXIT
2325				*	
2326				* ERROR: DUPLICATE RECORD COPIED	
2327				*	
2328	05	00471	6A700DC4	CPY50	BAL, LNK TYPEMSG TYPE: 'P1:FILE NOT SEQD & P3 NULL'
2329	05	00472	00000269 02		DATA ERRP16
2330	05	00473	6A700B33		BAL, LNK CLOSE CLOSE INPUT FILE
2331	05	00474	330000AE 02		MTW, 0 COPYFL DON'T DELETE INPUT FILE IF
2332	05	00475	6930047C		BNEZ CPY56 FID1=FID2
2333	05	00476	6A700B35		BAL, LNK CLOSE2 CLOSE COPY FILE
2334	05	00477	22300009 A		LI, X1 9
2335	05	00478	F25600AC 02		LB, P1 *CDTADR, X1 SET P1=ADR OF FID2 IN CDT
2336	05	00479	305000AC 02		AW, P1 CDTADR
2337	05	0047A	6A700B6A		BAL, LNK DELETEDFILE DELETE COPY FILE
2338	05	0047B	6800047D		B CPY58 EXIT IF FID1 NOT= FID2
2339	05	0047C	6A700B37	CPY56	BAL, LNK CLOSE3 DELETE COPY FILE, FID1=FID2
2340	05	0047D	E800000D A	CPY58	B *FILNK EXIT
2341	05	0047E	6A700DC4	CPY60	BAL, LNK TYPEMSG TYPE: 'PASSWORD ERROR'
2342	05	0047F	00000214 02		DATA ERRM19
2343	05	00480	E800000D A		B *FILNK EXIT

PAGE

* FILE COMMAND: SET TERMINATOR (X'15') MODE *

2344
2345
2346
2347
2348
2349
2350 05 00481
2351 05 00481 22300005 A
2352 05 00482 F24600AC 02
2353 05 00483 B28800AC 02
2354 05 00484 3180038E
2355 05 00485 69300489
2356
2357 05 00486 22800000 A
2358 05 00487 358000AF 02
2359 05 00488 E800000D A
2360
2361 05 00489 3180038F
2362 05 0048A 68300487
2363 05 0048B 6800038B
2364
2365
2366
2367
2368 05 0048C
2369 05 0048C 22300005 A
2370 05 0048D F24600AC 02
2371 05 0048E B28800AC 02
2372 05 0048F 3180038E
2373 05 00490 69300494
2374 05 00491 22800000 A
2375 05 00492 358001A0 02
2376 05 00493 E800000D A
2377
2378 05 00494 3180038F
2379 05 00495 68300492
2380 05 00496 6800038B

*
*
F:CR EQU \$
LI,X1 5
LB,X2 *CDTADR,X1
LW,T1 *CDTADR,X2 GET PARAMETER AS A TEXTC STRING.
CW,T1 BPV0N CHECK FOR '0N'
BNE CR5
*
LI,T1 0 TURN '0N'
CR3 STW,T1 CRFLAG SET FLAG TO INCLUDE TERMINATOR
B *F:LNK IN OUTPUT RECORDS.
*
CR5 CW,T1 BPV0FF CHECK FOR '0FF'
BE CR3 TURN '0FF'
B BPV10 ERROR: =NOT 0N/0FF

* FILE COMMAND: RECORD SIZE PRESERVATION *

*
*
F:RP EQU \$
LI,X1 5
LB,X2 *CDTADR,X1
LW,T1 *CDTADR,X2 GET PARAMETER AS A TEXTC STRING
CW,T1 BPV0N CHECK FOR '0N'
BNE RP5
LI,T1 0 TURN '0N'
RP3 STW,T1 RPFLAG FLAG TO RETAIN RECORD SIZE
B *F:LNK EVEN WITH TRAILING BLANKS
*
RP5 CW,T1 BPV0FF CHECK FOR '0FF'
BE RP3 TURN '0FF'
B BPV10 ERROR: =NOT 0N/0FF

PAGE

2381
 2382
 2383
 2384
 2385
 2386
 2387 05 00497
 2388 05 00497 6A700D78
 2389 05 00498 22300005 A
 2390 05 00499 F25600AC 02
 2391 05 0049A 305000AC 02
 2392 05 0049B 6A700B6A
 2393 05 0049C 698004A0
 2394
 2395
 2396
 2397 05 0049D 6A700DC4
 2398 05 0049E 00000281 02
 2399 05 0049F E800000D A
 2400
 2401
 2402
 2403 05 004A0 6A700DC4
 2404 05 004A1 000001F6 02
 2405 05 004A2 E800000D A

 * FILE COMMAND: DELETE *

 *
 *
 F:DELETE EGU \$
 BAL, LNK TESTEDITACTIVE
 LI, X1 S
 LB, P1 *CDTADR, X1 SET P1,ADR OF FID IN CDT
 AW, P1 CDTADR
 BAL, LNK DELETEFILE DELETE FILE
 BCS, B DLT10 DID FILE EXIST
 *
 * TYPE MESSAGE AND EXIT
 *
 DLT5 BAL, LNK TYPEMSG TYPE: '...DELETED'
 DATA MSG3
 B *FILNK YES = EXIT
 *
 * ERROR: FILE TO DELETE DOESN'T EXIST
 *
 DLT10 BAL, LNK TYPEMSG TYPE: 'NO SUCH FILE'
 DATA ERRM14
 B *FILNK

PAGE

* FILE COMMAND: EDIT *

2406
2407
2408
2409
2410
2411
2412 05 004A3
2413 05 004A3 330000B6 02
2414 05 004A4 691004A6
2415 05 004A5 6A700B33
2416
2417
2418
2419 05 004A6 22300005 A
2420 05 004A7 F25600AC 02
2421 05 004A8 305000AC 02
2422 05 004A9 6A700BC8
2423 05 004AA 698004B1
2424 05 004AB 694004B6
2425 05 004AC 22800001 A
2426 05 004AD 358000B6 02
2427 00000001
2428 05 004AE 22800000 A
2429 05 004AF 358002C5 02
2430
2431 05 004B0 E800000D A
2432
2433
2434
2435 05 004B1 6A700DC4
2436 05 004B2 000001F6 02
2437 05 004B3 228FFFFFF A
2438 05 004B4 358000B6 02
2439 05 004B5 E800000D A
2440
2441
2442

*
*
F:EDIT FGU \$
MTW,0 FILETYPE FILETYPE=1 NEVER OPENED
BLZ EDT5 *1 OPENED AS INPUT, KEYED
BAL,LNK CLOSE CLOSE FILE IF EVER OPENED
*
* OPEN FILE AND SET FILE TYPE
*
EDT5 LI,X1 5
LB,P1 *CDTADR,X1 SET P1=ADR OF FID IN CDT
AW,P1 CDTADR
BAL,LNK OPEN OPEN FILE
BCS,8 EDT10 DOES FILE EXIST
BCS,4 EDT20 YES = IS IT KEYED
LI,T1 1 YES = SET FILETYPE=1
STW,T1 FILETYPE
DB MODE=2
LI,T1 0
STW,T1 TABERRFLAG
FIN
B *F:LNK EXIT
*
* ERROR: SOURCE FILE DOESN'T EXIST
*
EDT10 BAL,LNK TYREMSG TYPE: !-NO SUCH FILE!
DATA ERRM14
EDT15 LI,T1 =1 SHOW UNSUCCESSFUL OPEN
STW,T1 FILETYPE
B *F:LNK
*
* FILE EXISTS BUT IS NOT KEYED
*

W01 20:44 SEP 08, '75

2443 05 004B6 6A700B33
2444 05 004B7 6A700DC4
2445 05 004B8 0U0001EB 02
2446 05 004B9 680004B3

EDT20

BAL, LNK CLOSE
BAL, LNK TYPMSG
DATA ERRM12
B EDT15

154
CLOSE FILE
TYPE: 1-FILE NOT KEYED; MUST COPY;
EXIT.

PAGE

* FILE COMMAND: END *

2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461

05 004BA 390000B6 02
05 004BB 691004BD
05 004BC 6A700B33
00000000
S
05 004BD 04900001 A

FI:END FGU \$
MTW,0 FILETYPE
BLZ \$+2
BAL, LNK CLOSE
DB MODE=1
CAL3,6 0
ELSE
MEXIT
FIN

WAS INPUT FILE EVER NAMED
NO - SKIP CLOSE
CLOSE INPUT FILE
EXIT TO BTM
EXIT TO UTS.

PAGE

 * FILE COMMAND: MERGE *

2462						
2463						
2464						
2465						
2466						
2467						
2468	05	004BE	6A700D78	F:MERGE	BAL, LNK	TESTEDITACTIVE
2469	05	004BF	22300000 A		LI, X1	0
2470	05	004C0	3530019E 02		STW, X1	MVD:REC:CNT
2471	05	004C1	22300005 A		LI, X1	5
2472	05	004C2	F25600AC 02		LB, P1	*CDTADR, X1
2473	05	004C3	305000AC 02		AW, P1	CDTADR
2474	05	004C4	355000B3 02		STW, P1	FID1ADR
2475	05	004C5	6A700BCC		BAL, LNK	OPEN1
2476	05	004C6	6980046E		BCS, 8	CPY40
2477	05	004C7	6940052C		BCS, 4	MRG80
2478				*		
2479	05	004C8	22500000 A		LI, P1	0
2480	05	004C9	355000B8 02		STW, P1	FIRSTFROM
2481	05	004CA	32500E28		LW, P1	L(EOF)
2482	05	004CB	355000BC 02		STW, P1	LASTFROM
2483				*		
2484	05	004CC	20300001 A		AI, X1	1
2485	05	004CD	F25600AC 02		LB, P1	*CDTADR, X1
2486	05	004CE	20300001 A		AI, X1	1
2487	05	004CF	21500003 A		CI, P1	SEQ2
2488	05	004D0	693004D9		BNE	MRG10
2489				*		
2490	05	004D1	F25600AC 02		LB, P1	*CDTADR, X1
2491	05	004D2	305000AC 02		AW, P1	CDTADR
2492	05	004D3	B2600005 A		LW, P2	*P1
2493	05	004D4	356000B8 02		STW, P2	FIRSTFROM
2494	05	004D5	20500001 A		AI, P1	1
2495	05	004D6	B2600005 A		LW, P2	*P1
2496	05	004D7	356000BC 02		STW, P2	LASTFROM
2497	05	004D8	20300002 A		AI, X1	2
2498				*		

RESET THE RECORD CNT.
 SET P1 TO ADDRESS OF FID1 IN CDT.
 OPEN MERGE SOURCE IN INPUT MODE.
 ERROR IF NON-EXISTENT
 OR NOT HEYED.
 SET UP INPUT RANGE AS DEFAULT
 ENTIRE FILE.
 BUT READJUST IF SPECIFIL RANGE
 GIVEN
 COMPUTE ADDRESS OF SEQUENCE PAIR
 AND STORE THEM AWAY.
 STEP AROUND !INT0!

H01 20:44 SEP 08, '75

2499	05	004D9	325000B8	02	MRG10	LW,P1	FIRSTFROM	VERIFY EXISTENCE OF RECORDS TO
2500	05	004DA	6A700C61			BAL,LNK	READNXRANDOM	MOVE.
2501	05	004DB	31B00E28			CW,R1	L(EOF)	IF RECORD READ WAS 'EOF',
2502	05	004DC	6810052A			BGE	MRG70	OR GREATER THAN LAST FROM, THEN
2503	05	004DD	31B000BC	02		CW,R1	LASTFROM	
2504	05	004DE	6920052A			BG	MRG70	NOTHING TO MOVE!
2505					*			
2506	05	004DF	6A700B33			BAL,LNK	CLOSE	YES, CLOSE FILE SO WE CAN
2507					*			USE F:EI ROUTINES TO DELETE
2508	05	004E0	2U300002	A		AI,X1	2	IT0! RANGE.
2509	05	004E1	F25600AC	02		LB,P1	*CDTADR,X1	STEP TO FID2 AND OPEN
2510	05	004E2	3U5000AC	02		AW,P1	CDTADR	
2511	05	004E3	355000B4	02		STW,P1	FID2ADR	
2512	05	004E4	6A700BC8			BAL,LNK	OPEN	
2513	05	004E5	6980051C			BCS,8	MRG30	IF NON-EXISTENT,CREATE NEW FILE.
2514	05	004E6	69400530			BCS,4	MRG82	ERROR IF NOT KEYED
2515					*			
2516	05	004E7	2U300002	A		AI,X1	2	NOW GET SEQUENCE NUMERS OF IT0!
2517	05	004E8	F28600AC	02		LB,T1	*CDTADR,X1	RANGE
2518	05	004E9	3U8000AC	02		AW,T1	CDTADR	
2519	05	004EA	B2500008	A		LW,P1	*T1	IN P1,PL = TEMPORARILY.
2520	05	004EB	2U800001	A		AI,T1	1	
2521	05	004EC	B2600008	A		LW,P2	*T1	
2522					*			
2523	05	004ED	6A700B39			BAL,LNK	DELETE	DELETE IT0! RANGE
2524	05	004EE	698004F0			BCS,8	*+2	GET IT0P! SEQ # IF LAST IT0!
2525	05	004EF	6A700C7F			BAL,LNK	READSEQUEN	NOT HIT EXACTLY.
2526	05	004F0	3210000B	A		LW,X3	R1	IT0P! SEQ # TO X3
2527	05	004F1	35500008	A		STW,P1	T1	
2528	05	004F2	35600009	A		STW,P2	T2	
2529					*			
2530	05	004F3	6A700B33		MRG13	BAL,LNK	CLOSE	CLOSE FID2 AS F:EI
2531	05	004F4	22A003E8	A	MRG14	LI,P3	1000	DEFAULT INCREMENT
2532	05	004F5	2U300001	A		AI,X1	1	
2533	05	004F6	F25600AC	02		LB,P1	*CDTADR,X1	
2534	05	004F7	683004FD			BEZ	MRG15	
2535	05	004F8	2U300001	A		AI,X1	1	INCREMENT GIVEN.

2536	05	004F9	F2A600AC	02		LB,P3	*CDTADR,X1	
2537	05	004FA	30A000AC	02		AW,P3	CDTADR	
2538	05	004FB	B2A0000A	A		LW,P3	*P3	
2539	05	004FC	35A000B0	02		STW,P3	DFLTINCR	
2540					*			
2541	05	004FD	325000B3	02	MRG15	LW,P1	FID1ADR	R-OPEN FILES IN PROPER MODE.
2542	05	004FE	6A700BCC			BAL,LNK	OPEN1	SOURCE IN INPUT.
2543	05	004FF	325000B4	02		LW,P1	FID2ADR	
2544	05	00500	6A700BEF			BAL,LNK	OPEN2	
2545					*			
2546	05	00501	6A700DC4			BAL,LNK	TYPEMSG	
2547	05	00502	00000288	02		DATA	MSG5	
2548	05	00503	325000B8	02	MRG17	LW,P1	FIRSTFROM	GET FIRST FROM RECORD IN FILE 1.
2549	05	00504	6A700C61			BAL,LNK	READNXTRANDM	
2550	05	00505	32500008	A		LW,P1	T1	FIRST FROM SEG # TO P1.
2551	05	00506	31B00E28		MRG20	CW,R1	L(EOF)	IF EOF READ,
2552	05	00507	68100522			BGE	MRG55	WE'RE DONE.
2553	05	00508	31B000BC	02		CW,R1	LASTFROM	IF SEQ # READ GREATER THAN LAST
2554	05	00509	69200522			BG	MRG55	FROM WE'RE DONE.
2555	05	0050A	35B00009	A		STW,R1	T2	
2556	05	0050B	389000BC	02		SW,T2	LASTFROM	
2557					*			
2558	05	0050C	6A700DFD			BAL,LNK	WRITE2	WRITE RECORD INTO FILE2.
2559	05	0050D	3310019E	02		MTW,1	MVD:REC:CNT	COUNT REC.S MOVED.
2560		00000001				D0	MODE=2	
2561	05	0050E	35B002C3	02		STW,R1	INTFLAG1	
2562	05	0050F	355002C4	02		STW,P1	INTFLAG2	
2563						FIN		
2564	05	00510	20900000	A		AI,T2	0	
2565	05	00511	68300523			BEZ	MRG56	
2566	05	00512	3050000A	A		AW,P1	P3	INCREMENT WRITE SEQ #.
2567	05	00513	315000BF	02		CW,P1	MAXSEQ	IS SEQ. NO. TOO BIG
2568	05	00514	68200518			BLE	MRG25	NO.
2569	05	00515	6A700DC4			BAL,LNK	TYPEMSG	YES.
2570	05	00516	0000021D	02		DATA	ERRM20	
2571	05	00517	68000522			B	MRG55	
2572	05	00518	31500001	A	MRG25	CW,P1	X3	IF CURRENT WRITE SE # MEETS

2573	05	00519	68100526		BGE	MRG65				
2574	05	0051A	6A700C7F		BAL, LNK	READSEQUEN				!STOP! SEQ # WE'RE CUT OFF.
2575	05	0051B	68000506		B	MRG20				GET NEXT FROM RECORD.
2576					*					
2577					*					
2578		05	0051C		MRG30	EQU	\$			OUTPUT FILE DOESN'T EXIST.
2579	05	0051C	20300002	A		AI, X1	2			GET STARTING OUTPUT SEQUENCE.
2580	05	0051D	F28600AC	02		LB, T1	*CDTADR, X1			
2581	05	0051E	308000AC	02		AW, T1	CDTADR			
2582	05	0051F	B2800008	A		LW, T1	*T1			
2583	05	00520	32100E28		MRG35	LW, X3	L(E0F)			SET !STOP! SEQUENCE TO EOF.
2584	05	00521	680004F4			B	MRG14			
2585					*					
2586					*					
2587	05	00522	3850000A	A	MRG55	SW, P1	P3			SUCCESSFUL MERGE, MOVE DEST SEQ #
2588					*					BACK TO LAST USED. THEN USE
2589	05	00523	6A700B33		MRG56	BAL, LNK	CLOSE			!MK! CODE AFTER CLOSING.
2590	05	00524	6A700B35			BAL, LNK	CLOSE2			
2591	05	00525	68000661			B	MVE40			
2592					*					
2593	05	00526	35B000BC	02	MRG65	STW, R1	LASTFROM			SET LAST SEQ # READ.
2594	05	00527	6A700B33			BAL, LNK	CLOSE			
2595	05	00528	6A700B35			BAL, LNK	CLOSE2			
2596	05	00529	68000674			B	MVE56			THEN USE !MK! CODE.
2597					*					
2598	05	0052A	6A700B33		MRG70	BAL, LNK	CLOSE			CLOSE INPUT FILE
2599	05	0052B	68000683			B	MVE58			THEN USE !MK! ROUTINE
2600	05	0052C	6A700B33		MRG80	BAL, LNK	CLOSE			
2601	05	0052D	6A700DC4			BAL, LNK	TYPMSG			
2602	05	0052E	00000206	02		DATA	ERRM17			!SOURCE NOT KEYED!
2603	05	0052F	E800000D	A		B	*F:LNK			
2604					*					
2605	05	00530	6A700B33		MRG82	BAL, LNK	CLOSE			
2606	05	00531	6A700DC4			BAL, LNK	TYPMSG			
2607	05	00532	0000020C	02		DATA	ERRM18			DEST. NOT KEYED
2608	05	00533	E800000D	A		B	*F:LNK			
2609					*					

PAGE

```

2610
2611
2612
2613
2614
2615
2616      05 00534
2617      00000001
2618      05 00534 22300005 A
2619      05 00535 F25600AC 02
2620      05 00536 305000AC 02
2621
2622      05 00537 B2500005 A
2623      05 00538 22300004 A
2624      05 00539 31560015 02
2625      05 0053A 6830053F
2626      05 00538 64300539
2627
2628      05 0053C 6A7000DC4
2629      05 0053D 000002E5 02
2630      05 0053E 68000004
2631
2632      05 0053F 67060540
2633
2634      05 00540 E800000D A
2635      05 00541 04100001 03
          03 00001 28000000 X
          03 00002 80000000 A
          03 00003 04 A
          03 00003 1 07 A
          03 00003 2 00 A
          03 00003 3 00 A
          03 00004 00 A
2636      05 00542 04100005 03
          03 00005 28000000 X
          03 00006 80000000 A
          03 00007 04 A

```

```

*****
* FILE COMMAND: TA *
*****
*
*
F:TA      FGU      $
          D8      MBDE=2
          LI,X1    5      COMPUTE ADDRESS OF TAB SPECIFIER
          LB,P1    *CDTADR,X1  IN CDT.
          AW,P1    CDTADR
*
          LW,P1    *P1      GET SPECIFIER
          LI,X1    4      AND CHECK VALIDITY
          CW,P1    X:F=1,X1
          BE      TA5
          BDR,X1  $=2
*
          BAL,LNK  TYPMSG   ERROR: NOT F,M,S.
          DATA   UTSM3
          B       MASTERPARSER
*
          TAS     EXU      TABSET=1,X1  CHANGE MUC TABS FOR F,M OR S
*
          B       *F:LNK   RETURN
          TABSET  M:DEVICE M:UC,(TAB,7,0,0,0) FTABS

```

```

          M:DEVICE M:UC,(TAB,10,19,37,0) MTABS

```

H01 20:44 SEP 08, '75

03	00007	1	0A	A
03	00007	2	13	A
03	00007	3	25	A
03	00008		00	A

2637 05 00543 0*100009 03

M:DEVICE M:UC,(TAB,8,16,30,0) STABS

03	00009		28000000	X
03	0000A		80000000	A
03	0000B		04	A
03	0000B	1	08	A
03	0000B	2	10	A
03	0000B	3	1E	A
03	0000C		00	A

2638 05 00544 0*10000D 03

M:DEVICE M:UC,(TAB,8,12,36,0) CTABS

03	0000D		28000000	X
03	0000E		80000000	A
03	0000F		04	A
03	0000F	1	08	A
03	0000F	2	0C	A
03	0000F	3	24	A
03	00010		00	A

2639

FIN

PAGE

2640
 2641
 2642
 2643
 2644
 2645
 2646 05 00545
 2647 05 00545 22300005 A
 2648 05 00546 F24600AC 02
 2649 05 00547 B25800AC 02
 2650 05 00548 22300007 A
 2651 05 00549 F24600AC 02
 2652 05 0054A B28800AC 02
 2653 05 0054B 208FFFFFF A
 2654 05 0054C 69100580
 2655 05 0054D 2180008C A
 2656 05 0054E 68100580
 2657 05 0054F 6A70006D
 2658 05 00550 69800583
 2659 00000001
 2660 05 00551 330001A1 02
 2661 05 00552 68300554
 2662 05 00553 041002CC 02
 2663
 2664
 2665
 2666
 2667 05 00554 6A7000DB
 2668 05 00555 4U080000 A
 2669 05 00556 330001A1 02
 2670 05 00557 68300562
 2671
 2672
 2673
 2674 05 00558 32E000E7 02
 2675 05 00559 358000E7 02
 2676 05 0055A 331000E7 02

```

*****
* RECORD COMMAND: ADD COMMENTARY *
*****
*
*
R:COMMENTARY      EQU *
                  LI,X1      5
                  LB,X2      *CDTADR,X1      SET P1=STARTING SEQ #
                  LW,P1      *CDTADR,X2
                  LI,X1      7
                  LB,X2      *CDTADR,X1      SET T1=STARTING COLUMN #
                  LW,T1      *CDTADR,X2
                  AI,T1      =1              ADJ TO INTERNAL COL. #
                  BLZ        CMT40
                  CI,T1      MAXCLMN        IS COL. # >= MAX COL. #
                  BGE        CMT40         YES = ERROR
                  BAL,LNK    READRANDOM     READ FIRST RECORD
                  BCS,8      CMT50         DOES IT EXIST (IF NO, ERROR)
                  DB        MODE=2
                  MTW,0      CT$FLAG       IS TYPE FLAG ON
                  BEZ        $+2           NO
                  CAL,1      N$PR$MPT$FPT YES, TURN OFF PROMPT
                  FIN
*
* TYPE SEQ. # AND READ IN COMMENTARY
*
CMT10  BAL,LNK  TYPESEQ      TYPE: !DDDD.DDD!
        GEN4    BL,E0M,C,0
        MTW,0   CT$FLAG      IS TYPE FLAG ON
        BEZ     CM10A        NO, CONTINUE
*
* CT SPECIFIED, TYPE CARD TO COLUMN REQUESTED
*
LW,D0    RECSIZE      SAVE RECSIZE TEMPORARILY
STW,T1   RECSIZE      SET NEW RECSIZE (COLUMN NO)
MTW,1    RECSIZE
    
```

H01 20:44 SEP 08, '75

2677	05	0055B	32300008	A	LW,X1	T1	COLUMN FOR PROMPT
2678	05	0055C	72160024	02	LB,X3	CARDIMG,X1	
2679	05	0055D	2240005C	A	LI,X2	'*'	
2680	05	0055E	75460024	02	STB,X2	CARDIMG,X1	ADD TO TYPE IMAGE
2681		00000001			D0	MODE=2	
2682	05	0055F	041002D2	02	CAL1,1	TPC#FPT	TYPE CARD TO COMMENTARY
2683					ELSE		
2684				*S*	LI,X2	0	
2685				*S*	LW,T2	T1	GET NO. CHARS TO TYPE
2686				*S*	AI,T2	1	
2687				*S* CM10B	LB,R0	CARDIMG,X2	GET NEXT CHARACTER
2688				*S*	CAL3,1	0	TYPE IT
2689				*S*	AI,X2	1	
2690				*S*	BDR,T2	CM10B	CONTINUE UNTIL ALL CHARS. TYPED
2691					FIN		
2692	05	00560	75160024	02	STB,X3	CARDIMG,X1	RESTORE CHARACTER TO CARD IMAGE
2693	05	00561	35E000E7	02	STW,D0	RECSIZE	
2694							
2695		05 00562			* CM10A	EGU	*
2696	05	00562	6A700CAE		BAL,LNK	READTELETYPE2	READ COMMENTARY
2697	05	00563	20BFFFFF	A	AI,R1	=1	SET R1=# OF CHARS READ, LESS C/R
2698	05	00564	E830000D	A	BEZ	*R;LNK	IF ONLY C/R READ = EXIT
2699	05	00565	22300000	A	LI,X1	0	
2700	05	00566	32400008	A	LW,X2	T1	
2701	05	00567	22900040	A	LI,T2	' '	
2702							
2703							
2704							
2705	05	00568	72E60176	02	* CMT15	LB,D0	TTYIMG,X1
2706	05	00569	75E80024	02	STB,D0	CARDIMG,X2	MOVE COMMENTARY INTO SPECIFIED COLUMN
2707	05	0056A	20300001	A	AI,X1	1	
2708	05	0056B	20400001	A	AI,X2	1	
2709	05	0056C	64B0056E		BDR,R1	*+2	TEST IF ANY MORE CHARS LEFT TO MOVE
2710	05	0056D	68000574		B	CMT20	NO = GO FINISH UP
2711	05	0056E	2140008C	A	CI,X2	MAXCLMN	YES = TEST IF ANY ROOM LEFT ON CARD
2712	05	0056F	69100568		BL	CMT15	YES = LOOP
2713	05	00570	71960176	02	CB,T2	TTYIMG,X1	NO = TEST IF REMAINING CHARS ARE ALL

2714	05	00571	69300589	BNE	CMT70	BLANKS (IF NOT, ERROR)	
2715	05	00572	20300001 A	AI,X1	1		
2716	05	00573	64800570	BDR,R1	*-3	LOOP	
2717				*			
2718				*	BLANK OUT REST OF CARD AFTER NEW COMMENTARY		
2719				*			
2720	05	00574	2140008C A	CMT20	CI,X2	MAXCLMN	BLANK OUT REST OF CARD
2721	05	00575	68100579	BGE	CMT30		
2722	05	00576	75980024 02	STB,T2	CARDIMG,X2		
2723	05	00577	20400001 A	AI,X2	1		
2724	05	00578	68000574	B	CMT20		
2725				*			
2726				*	WRITE NEW RECORD AND THEN GET NEXT RECORD TO PROCESS		
2727				*			
2728	05	00579	6A700A56	CMT30	BAL,LNK	SETEOF	
2729	05	0057A	6A700E1F	BAL,LNK	WRITERAND8M		
2730	05	0057B	6A700C7F	BAL,LNK	READSEQUEN	READ NEXT RECORD	
2731	05	0057C	31B00E28	CW,R1	L(EOF)	WAS IT AN EOF	
2732	05	0057D	68300586	BE	CMT60	YES = ERROR	
2733	05	0057E	3250000B A	LW,P1	R1	SET P1=SEQ # OF RECORD	
2734	05	0057F	68000554	B	CMT10	GO GET MORE COMMENTARY	
2735				*			
2736				*	ERROR: SPECIFIED COLUMN NUMBER > MAX COLUMN NUMBER		
2737				*			
2738	05	00580	6A700DC4	CMT40	BAL,LNK	TYPMSG	TYPE: 'P2:COL>72'
2739	05	00581	00000261 02	DATA	ERRP14		
2740	05	00582	68000004	B	MASTERPARSER	EXIT TO PARSER	
2741				*			
2742				*	ERROR: INITIAL SEQ. # DOESN'T EXIST		
2743				*			
2744	05	00583	6A700DC4	CMT50	BAL,LNK	TYPMSG	TYPE: 'P1:NO SUCH REC'
2745	05	00584	0000022E 02	DATA	ERRP1		
2746	05	00585	68000004	B	MASTERPARSER	EXIT TO PARSER	
2747				*			
2748				*	ERROR: EOF HIT		
2749				*			
2750	05	00586	6A700DC4	CMT60	BAL,LNK	TYPMSG	TYPE: '=-EOF HIT'

H01 20:44 SEP 08, '75

2751 05 00587 000001D5 02
 2752 05 00588 68000004
 2753
 2754
 2755
 2756 05 00589 6A700DC4
 2757 05 0058A 000001DC 02
 2758 05 0058B 68000579

DATA ERRM1
 B MASTERPARSER EXIT TO PARSER
 *
 * ERROR: COMMENTARY OVERFLOWS CARD
 *
 CMT70 BAL, LNK TYPEMSG TYPE: ==OVERFLOW!
 DATA ERRM3
 B CMT30 GO CONTINUE WITH NEXT RECORD

PAGE

2759
 2760
 2761
 2762
 2763
 2764
 2765 05 0058C
 2766 05 0058C 22300005 A
 2767 05 0058D F24600AC 02
 2768 05 0058E B25800AC 02
 2769 05 0058F 20400001 A
 2770 05 00590 B26800AC 02
 2771 05 00591
 2772 05 00591 331002C7 02
 2773 05 00592 6A700B39
 2774 05 00593 E800000D A

 * RECORD COMMAND: DELETE *

 *
 *
 R:DELETE EGU *
 LI,X1 5
 LB,X2 *CDTADR,X1
 LW,P1 *CDTADR,X2
 AI,X2 1
 LW,P2 *CDTADR,X2
 DB1 MODE=2
 MTW,1 TABXFLAG
 BAL,LNK DELETE
 B *R;LNK

GET ADDR OF FIRST SEQ # IN CDT
 SET P1=FIRST SEQ #
 P2=LAST SEQ #

DELETE ALL BETWEEN THESE SEQ #'S
 EXIT

PAGE

 * RECORD COMMANDS: FIND AND DELETE(TYPE) *

2775							
2776							
2777							
2778							
2779							
2780							
2781		05 00594		R:FIND\$SEQUENCE	EQU #		
2782	05 00594	22200002	A	LI,X4	2		USE X4=2 FOR 'FS'.
2783	05 00595	68000599		B	R:FIND\$TYPE+1		
2784							
2785							
2786		05 00596		R:FIND\$DELETE	EQU #		
2787	05 00596	22200000	A	LI,X4	0		USE X4=0 FOR 'FD'
2788	05 00597	68000599		B	R:FIND\$TYPE+1		
2789							
2790							
2791		05 00598		R:FIND\$TYPE	EQU #		
2792	05 00598	22200001	A	LI,X4	1		USE X4=1 FOR 'FT'
2793	05 00599	22A00000	A	LI,P3	0		USE P3 TO COUNT # OF MATCHES FOUND
2794	05 0059A	22300005	A	LI,X1	5		
2795	05 0059B	F24600AC	02	LB,X2	*CDTADR,X1		
2796	05 0059C	B25800AC	02	LW,P1	*CDTADR,X2		SET P1=FIRST SEQ # IN CDT
2797	05 0059D	355000B9	02	STW,P1	FIRSTSET		FIRSTSET=1ST SEQ # IN CDT
2798	05 0059E	358000B9	02	STW,T1	FIRSTSET		SET FIRSTSET=1ST SEQ # IN CDT
2799	05 0059F	20400001	A	AI,X2	1		LASTSET=2ND SEQ # IN CDT
2800	05 005A0	B28800AC	02	LW,T1	*CDTADR,X2		
2801	05 005A1	358000BE	02	STW,T1	LASTSET		
2802	05 005A2	22300007	A	LI,X1	7		
2803	05 005A3	F26600AC	02	LB,P2	*CDTADR,X1		SET P2=ABSOLUTE ADDR OF STRING TO
2804	05 005A4	306000AC	02	AW,P2	CDTADR		MATCH
2805	05 005A5	22300008	A	LI,X1	8		
2806	05 005A6	6A700A34		BAL,LNK	PROCESSCOL#PAIR		PROCESS COL # PARAMS
2807	05 005A7	6A700C61		BAL,LNK	READNXRANDOM		READ FIRST SEQ # OR NEXT HIGHEST
2808							
2809							
2810							
2811	05 005A8	31B00E28		FND20	CW,R1	L(EOF)	WAS IT AN EOF

2812	05	005A9	683005CF		BE	FND70	YES - ERROR
2813	05	005AA	31B000BE	02	CW,R1	LASTSET	WAS INPUT SEQ # > LAST SEQ #
2814	05	005AB	692005C1		BG	FND50	YES - FINISH UP
2815	05	005AC	35B000B9	02	STW,R1	FIRSTSET	NO - SAVE NEW SEQ #
2816	05	005AD	325000B7	02	LW,P1	FRSTCLMN	CHECK IF REC CONTAINS STRING
2817	05	005AE	6A7009EF		FNDTYP	BAL,LNK	FINDMATCH
2818	05	005AF	698005BB		BCS,8	FND40	
2819	05	005B0	20A00001	A	AI,P3	1	YES - INCR MATCH COUNT
2820	05	005B1	670405D2		EXU	FNDTBL1,X4	GO PERFORM APPRB ACTION
2821					*		
2822					*	'FD' USED: DELETE RECORD	
2823					*		
2824	05	005B2	6A700B95		FND30	BAL,LNK	DELETERECORD
2825	05	005B3	680005BB		B	FND40	GO ON TO NEXT RECORD
2826					*		
2827					*	'FT' USED: TYPE SEQ #, AND RECORD	
2828					*		
2829	05	005B4	325000B9	02	FND32	LW,P1	FIRSTSET
2830	05	005B5	6A700A56		BAL,LNK	SETE0D	
2831	05	005B6	6A700D82		BAL,LNK	TYPECARD	
2832	05	005B7	680005BB		B	FND40	
2833					*		
2834					*	'FS' USED: TYPE SEQ #	
2835					*		
2836	05	005B8	325000B9	02	FND35	LW,P1	FIRSTSET
2837	05	005B9	6A700DDB		BAL,LNK	TYPESEQ	GET SEQ #
2838	05	005BA	00000000	A	GEN4	0,0,0,0	TYPE: 'DDDD.DDD'
2839					*		
2840					*	TEST IF LAST RECORD HIT: IF YES, GO FINISH UP	
2841					*		
2842	05	005BB	32B000B9	02	FND40	LW,R1	FIRSTSET
2843	05	005BC			D01	MODE=2	TEST IF LAST SEQ # = SEQ # TO STOP
2844	05	005BC	35B002C3	02	STW,R1	INTFLAG1	
2845	05	005BD	31B000BE	02	CW,R1	LASTSET	AT
2846	05	005BE	683005C1		BE	FND50	
2847	05	005BF	6A700C7F		BAL,LNK	READSEQUEN	NO - READ NEXT RECORD
2848	05	005C0	680005A8		B	FND20	LOOP

```

2849
2850
2851
2852 05 005C1 670405D5 FND50 EXU FNDTBL2,X4 GO FINISH UP
2853
2854
2855
2856 05 005C2 32F0000A A FND60 LW,D1 P3
2857 05 005C3 683005CC BEZ FND65A WERE ANY MATCHES FOUND
2858 05 005C4 3250000A A LW,P1 P3 GET RECORD COUNT IN P1
2859 05 005C5 22600A31 02 LI,P2 BA(MSG6)+1 GET BYTE ADR. OF PLACE TO PUT CNT.
2860 05 005C6 6A700B11 BAL,LNK BINTODEC GO PUT THE NUMBER THER
2861 05 005C7 6A700DC4 BAL,LNK TYPEMSG
2862 05 005C8 0000028C 02 DATA MSG6
2863 05 005C9 E800000D A B *R:LNK EXIT
2864
2865
2866
2867 05 005CA 21A00000 A FND65 CI,P3 0 WERE ANY MATCHES FOUND
2868 05 005CB E930000D A BNE *R:LNK YES - EXIT
2869 05 005CC 6A700DC4 FND65A BAL,LNK TYPEMSG NO - TYPE: !-NONE!
2870 05 005CD 000001E6 02 DATA ERRM6
2871 05 005CE E800000D A B *R:LNK EXIT
2872
2873
2874
2875 05 005CF 6A700DC4 FND70 BAL,LNK TYPEMSG TYPE: !-EOF HIT!
2876 05 005D0 00000105 02 DATA ERRM1
2877 05 005D1 680005C1 B FND50 GO FINISH UP
2878
2879
2880 05 005D2 FNDTBL1 EQU $
2881 05 005D2 680005B2 B FND30
2882 05 005D3 680005B4 B FND32
2883 05 005D4 680005B8 B FND35
2884
2885

```

H01 20:44 SEP 08, 175

2886 05 005D5

FNDTBL2 EQU

*

2887 05 005D5 680005C2

B

FND60

2888 05 005D6 680005CA

B

FND65

2889 05 005D7 680005CA

B

FND65

PAGE

 * RECORD COMMANDS: INSERT(SUPPRESSING SEQ. NUMBERS) *

2890						
2891						
2892						
2893						
2894						
2895						
2896		05 005D8		R:INSERT\$SUP\$SEQ	EQU \$	
2897	05	005D8	22200002 A	LI,X4	2	USE X4=2 FOR 'IS'
2898			00000001	D0	MODE=2	
2899	05	005D9	041002CB 02	CAL1,1	PROMPT2\$FPT	
2900	05	005DA	680005DD	B	R:INSERT+2	
2901				ELSE		
2902			*S*	B	R:INSERT+1	
2903				FIN		
2904				*		
2905				*		
2906		05 005DB		R:INSERT	EQU	\$
2907	05	005DB		D01	MODE=2	
2908	05	005DB	041002CC 02	CAL1,1	N0PROMPT\$FPT	
2909	05	005DC	22200000 A	LI,X4	0	USE X4=0 FOR 'IN'
2910	05	005DD	22300005 A	LI,X1	5	
2911	05	005DE	F24600AC 02	LB,X2	*CDTADR,X1	SET P1=STARTING SEQ #
2912	05	005DF	B25800AC 02	LW,P1	*CDTADR,X2	
2913	05	005E0	328000B0 02	LW,T1	DFLTINCR	SET T1=LAST INCR USED
2914	05	005E1		D01	MODE=2	
2915	05	005E1	331002C7 02	MTW,1	TABXFLAG	
2916	05	005E2	22300006 A	LI,X1	6	
2917	05	005E3	F2B600AC 02	LB,R1	*CDTADR,X1	GET PARAM2 TYPE
2918	05	005E4	683005E9	BEZ	INS10	TEST IF PARAM2 PRESENT
2919	05	005E5	24300001 A	AI,X1	1	
2920	05	005E6	F24600AC 02	LB,X2	*CDTADR,X1	YES = SET T1=INCR FROM CDT
2921	05	005E7	B28800AC 02	LW,T1	*CDTADR,X2	
2922	05	005E8	358000B0 02	STW,T1	DFLTINCR	SET NEW DEFAULT INCR
2923				*		
2924				*	GET SEQ. # AT WHICH TO STOP INSERTING	
2925				*		
2926	05	005E9	6A700C61	INS10	BAL,LNK	READNXTRAND0M
						READ 1ST SEQ # OR NEXT HIGHEST

H01 20144 SEP 08, '75

2964 05 00606 30500008 A
 2965 05 00607 31500009 A
 2966 05 00608 691005ED
 2967 05 00609 315000BF 02
 2968 05 0060A 6820060D
 2969 05 0060B 6A700DC4
 2970 05 0060C 0000021D 02
 2971 05 0060D 6A700DC4
 2972 05 0060E 00000613
 2973 05 0060F
 2974 05 0060F E800000D A
 2975 00000C01
 2976 05 00610 22F00009 A
 2977 05 00611 02000000 A
 2978 05 00612 22F00001 A
 2979
 2980
 2981
 2982 05 00613 03070708 A

INS38
 INS40
 INS50
 *
 *
 INSMMSG

AW,P1 T1
 CW,P1 T2
 BL INS20
 CW,P1 MAXSEQ
 BLE INS38
 BAL,LNK TYPEMSG
 DATA ERRM20
 BAL,LNK TYPEMSG
 DATA INSMMSG
 EGU *
 B *R:LNK
 DB MODE#2
 LI,D1 9
 NOP 0
 LI,D1 1
 FIN
 TEXTC 'GGH'

INCR SEQ #
 IS NEW SEQ # > SEQ # TO STOP AT
 IS SEQ. NO. TOO BIG
 NO.
 YES.
 RING BELL TWICE
 RETURN
 X4 IS NEVER ONE
 OFFSET FOR PROMPT ONLY
 X'07'+X'07'+E8M

PAGE

```

2983
2984 *****
2985 * RECORD COMMANDS: MOVE AND DELETE(KEEP) *
2986 *****
2987 *
2988 *
2989 05 00614 R:MOVE$DELETE EQU $
2990 05 00614 22200000 A LI,X4 0 USE X4=0 TO SIGNAL MD
2991 05 00615 68000617 B R:MOVE$KEEP+1
2992 *
2993 *
2994 05 00616 R:MOVE$KEEP EQU $
2995 05 00616 22200001 A LI,X4 1 USE X4=1 TO SIGNAL MK
2996 *
2997 * GET 'FROM' SEQ. # PAIR IN T1=2, 'TO' SEQ # PAIR IN P1=2, AND
2998 * INCREMENT IN P3
2999 *
3000 05 00617 22300000 A LI,X1 0
3001 05 00618 3530019E 02 STW,X1 MVD:REC:CNT ZERO BUT MOVED REC. COUNT.
3002 05 00619 22300005 A MVE10 LI,X1 5
3003 05 0061A F24600AC 02 LB,X2 *CDTADR,X1 GET ADDR OF 1ST 'FROM' SEQ # IN CDT
3004 05 0061B B28800AC 02 LW,T1 *CDTADR,X2 SET P1=FIRST 'FROM' SEQ #
3005 05 0061C 20400001 A AI,X2 1 P2=LAST 'FROM' SEQ #
3006 05 0061D B29800AC 02 LW,T2 *CDTADR,X2
3007 05 0061E 22300007 A LI,X1 7
3008 05 0061F F24600AC 02 LB,X2 *CDTADR,X1 GET ADDR OF 1ST 'TO' SEQ # IN CDT
3009 05 00620 B25800AC 02 LW,P1 *CDTADR,X2 SET P1=FIRST 'TO' SEQ #
3010 05 00621 20400001 A AI,X2 1 P2=LAST 'TO' SEQ #
3011 05 00622 B26800AC 02 LW,P2 *CDTADR,X2
3012 05 00623 32A000B0 02 LW,P3 DFLTINCR SET P3=LAST INCR USED
3013 05 00624 22300008 A LI,X1 8
3014 05 00625 F28600AC 02 LB,R1 *CDTADR,X1 GET PARAM3 TYPE
3015 05 00626 6830062B BEZ MVE20 TEST IF PARAM3 PRESENT
3016 05 00627 20300001 A AI,X1 1
3017 05 00628 F24600AC 02 LB,X2 *CDTADR,X1 YES - SET P3=INCR FROM CDT
3018 05 00629 B2A800AC 02 LW,P3 *CDTADR,X2
3019 05 0062A 35A000B0 02 STW,P3 DFLTINCR SET NEW DEFAULT INCR

```

```

3020
3021
3022
3023 05 0062B 32E00008 A
3024 05 0062C 32F00009 A
3025 05 0062D 1950000E A
3026 05 0062E 6890066E
3027 05 0062F 1960000E A
3028 05 00630 6890066E
3029 05 00631 32E00005 A
3030 05 00632 32F00006 A
3031 05 00633 1980000E A
3032 05 00634 6890066E
3033 05 00635 1990000E A
3034 05 00636 6890066E
3035 05 00637 46800005 A
3036 05 00638 6A700C61
3037 05 00639 46800005 A
3038 05 0063A 31B00E28
3039 05 0063B 68300683
3040 05 0063C 31B00009 A
3041 05 0063D 69200683
3042 05 0063E 6A700B39
3043 05 0063F 69800641
3044 05 00640 6A700C7F
3045 05 00641 3210000B A
3046 05 00642 46500008 A
3047 05 00643 6A700C61
3048 05 00644 32500008 A
3049 05 00645 3850000A A
3050
3051
3052
3053 05 00646 31B00E28
3054 05 00647 68300671
3055 05 00648 3190000B A
3056 05 00649 68200659
    
```

```

*
* CHECK FOR OVERLAPPING SEQ #'S AND SET UP MOVE
*
MVE20  LW,D0  T1  PUT 'FROM' SEQ #'S IN DW
        LW,D1  T2
        CLM,P1 D0  MAKE SURE 'TO' AND 'FROM' RANGES
        BIL    MVE50 ARE MUTUALLY EXCLUSIVE
        CLM,P2 D0
        BIL    MVE50
        LW,D0  P1
        LW,D1  P2
        CLM,T1 D0
        BIL    MVE50
        CLM,T2 D0
        BIL    MVE50
        XW,T1  P1  EXCHANGE FIRST 'FROM' AND 'TO'
        BAL,LNK READNXTRANDM CHECK 'FROM' RANGE
        XW,T1  P1  RESTORE
        CW,R1  L(EOF) M IF RECORD READ WAS AN EOF,
        BE    MVE58
        CW,R1  T2  OR SEQUENCE GREATER THAN SECOND
        BG    MVE58 'FROM', NOTHING TO MOVE
        BAL,LNK DELETE DELETE 'TO' RECORDS
        BCS,8  *+2 WAS LAST 'TO' SEQ # HIT BY DELETE
        BAL,LNK READSEQUEN YES = READ NEXT RECORD
        LW,X3  R1  SET X3=SEQ # AT WHICH TO STOP MOVE
        XW,P1  T1
        BAL,LNK READNXTRANDM READ 1ST 'FROM' REC OR NEXT HIGHEST
        LW,P1  T1  SET P1=NEW 'TO' SEQ # = INCR
        SW,P1  P3
*
* READ EACH 'FROM' RECORD AND WRITE UNDER 'TO' SEQ #
*
MVE30  CW,R1  L(EOF) WAS AN EOF READ
        BE    MVE53 YES = GO TYPE ERROR MESSAGE
        CW,T2  R1  WAS 'FROM' SEQ # >= LAST 'FROM' SEQ
        BLE   MVE35 YES = GO FINISH UP
    
```

3057	05	0064A	3050000A	A	AW,P1	P3	INCR ITB! SEQ #
3058	05	0064B	31500001	A	CW,P1	X3	IS NEW ITB! SEQ # > SEQ # TO STOP AT
3059	05	0064C	68100674		BGE	MVE56	YES = GO TYPE ERROR MESSAGE
3060	05	0064D	35B000BC	02	STW,R1	LASTFROM	
3061	05	0064E	6804064F		B	*+1,X4	DELETE (FROM) RECORD AS REQD
3062	05	0064F	6A700B95		BAL,LNK	DELETERECORD	
3063	05	00650	3310019E	02	MTW,P1	MVD:REC:CNT	INCRMENT REC. CNT
3064	05	00651	6A700E1F		BAL,LNK	WRITERANDOM	WRITE RECORD WITH NEW ITB! SEQ #
3065			00000001		DB	MODE=2	
3066	05	00652	35B002C3	02	STW,R1	INTFLAG1	
3067	05	00653	355002C4	02	STW,P1	INTFLAG2	
3068					FIN		
3069	05	00654	465000BC	02	XW,P1	LASTFROM	MUST REREAD LAST (FROM) RECORD TO
3070	05	00655	6A700C6D		BAL,LNK	READRANDOM	GET DCB BACK IN SEQ
3071	05	00656	465000BC	02	XW,P1	LASTFROM	RESTORE P1 AND LASTFROM
3072	05	00657	6A700C7F		BAL,LNK	READSEQUEN	READ NEXT (FROM) RECORD
3073	05	00658	68000646		B	MVE30	LOOP
3074							
3075							* LAST (FROM) SEQ # HIT OR PASSED: FINISH UP
3076							*
3077	05	00659	69100661		MVE35	BL	MVE40
3078	05	0065A	3050000A	A	AW,P1	P3	WAS LAST (FROM) SEQ # PASSED
3079	05	0065B	31500001	A	CW,P1	X3	NO, WAS HIT = INCR ITB! SEQ #
3080	05	0065C	68100674		BGE	MVE56	IS NEW ITB! SEQ # > SEQ # TO STOP AT
3081	05	0065D	6804065E		B	*+1,X4	YES = GO TYPE ERROR MESSAGE
3082	05	0065E	6A700B95		BAL,LNK	DELETERECORD	DELETE (FROM) REC AS REQD
3083	05	0065F	3310019E	02	MTW,P1	MVD:REC:CNT	INCRMENT RECORD COUNT.
3084	05	00660	6A700E1F		BAL,LNK	WRITERANDOM	WRITE REC WITH NEW ITB! SEQ #
3085							*
3086							* TYPE OUT LAST ITB! SEQ # AND EXIT
3087							*
3088	05	00661	22600AA7	02	MVE40	LI,P2	BA(MVEMSG1)+11
3089	05	00662	6A700B97		BAL,LNK	MVESEQ	BUILD MSG: I=-DONE AT DD.D! * NL
3090	05	00663	00000000	A	GEN4	0,0,0,0	FROM LAST ITB! SEQ #
3091	05	00664	20B0000A	A	AI,R1	10	ADJ CNT OF TEXTC=STRING
3092	05	00665	75B002A7	02	STB,R1	MVEMSG1	
3093	05	00666	6A700DC4		BAL,LNK	TYPMSG	TYPE MSG

HC1 20:44 SEP 08, '75

3094	05	00667	000002A7	02	DATA	MVEMSG1		
3095	05	00668	22600A4D	02	LI,P2	BA(MSG7)+1		
3096	05	00669	3250019E	02	LW,P1	MVD:REC:CNT	GET THE NUMBER OF REC.S MOVED	
3097	05	0066A	6A700B11		BAL,LNK	BINT0DEC	CONVERT IT, STUFF IT AWAY	
3098	05	0066B	6A700DC4		BAL,LNK	TYPMSG	AND PRINT IT OUT.	
3099	05	0066C	00000293	02	DATA	MSG7		
3100	05	0066D	E800000D	A	B	*R:LNK	EXIT	
3101								
3102								
3103								
3104	05	0066E	6A700DC4		MVE50	BAL,LNK	TYPMSG	TYPE; !=RNG OVERLAP!
3105	05	0066F	000001DF	02	DATA	ERRM4		
3106	05	00670	E800000D	A	B	*R:LNK	EXIT	
3107								
3108								
3109								
3110	05	00671	6A700DC4		MVE53	BAL,LNK	TYPMSG	TYPE; !=EOF HIT!
3111	05	00672	000001D5	02	DATA	ERRM1		
3112	05	00673	68000661		B	MVE40	GO EXIT	
3113								
3114								
3115								
3116	05	00674	3850000A	A	MVE56	SW,P1	P3	ADJ P1 TO LAST !T0! SEQ #
3117	05	00675	22600AC1	02	LI,P2	BA(MVEMSG2)+13	BUILD MSG; !=CUTOFF AT DDD.D (
3118	05	00676	6A700B97		BAL,LNK	MOVESEQ	WITH LAST !T0! SEQ #	
3119	05	00677	404D0000	A	GEN4	BL,LP,0,0		
3120	05	00678	3060000B	A	AW,P2	R1	INCR MSG BYTE ADDR	
3121	05	00679	20B0000C	A	AI,R1	12	CALC AND SAVE MSG LENGTH	
3122	05	0067A	3280000B	A	LW,T1	R1		
3123	05	0067B	325000BC	02	LW,P1	LASTFROM	BUILD: (DD.DD); + NL FROM LAST	
3124	05	0067C	6A700B97		BAL,LNK	MOVESEQ	'FROM' SEQ #	
3125	05	0067D	50000000	A	GEN4	RP,0,0,0		
3126	05	0067E	3080000B	A	AW,T1	R1	ADJ CNT OF TEXTC=STRING	
3127	05	0067F	758002AD	02	STB,T1	MVEMSG2		
3128	05	00680	6A700DC4		BAL,LNK	TYPMSG	TYPE; !=CUTOFF AT DDD.D (DD.DD); +	
3129	05	00681	000002AD	02	DATA	MVEMSG2	NL	
3130	05	00682	E800000D	A	B	*R:LNK	EXIT	

H01 20144 SEP 08, 175

178

3131				*		
3132	05	00683	6A700DC4	MVE58	BAL, LNK	TYPEMSG
3133	05	00684	00000201 02		DATA	ERRM16
3134	05	00685	E800000D A		B	*R; LNK

PAGE

 * RECORD COMMAND: RENUMBER *

3135
 3136
 3137
 3138
 3139
 3140
 3141 05 00686
 3142 05 00686 22300005 A
 3143 05 00687 F24600AC 02
 3144 05 00688 B25800AC 02
 3145 05 00689 22300007 A
 3146 05 0068A F24600AC 02
 3147 05 0068B B28800AC 02
 3148 05 0068C 6A700C6D
 3149 05 0068D 69800693
 3150 05 0068E 32500008 A
 3151 05 0068F 6A700E0E
 3152 05 00690 69800696
 3153 05 00691 6A700B95
 3154 05 00692 E800000D A
 3155
 3156
 3157
 3158 05 00693 6A700DC4
 3159 05 00694 0000022E 02
 3160 05 00695 E800000D A
 3161
 3162
 3163
 3164 05 00696 6A700DC4
 3165 05 00697 00000232 02
 3166 05 00698 E800000D A

R:RENUMBER EQU #
 LI,X1 5
 LB,X2 *CDTADR,X1 SET P1=OLD SEQ #
 LW,P1 *CDTADR,X2
 LI,X1 7
 LB,X2 *CDTADR,X1 SET T1=NEW SEQ #
 LW,T1 *CDTADR,X2
 BAL,LNK READRANDOM READ OLD RECORD
 BCS,8 RNM10 DID IT EXIST
 LW,P1 T1 YES = SET P1=NEW SEQ #
 BAL,LNK WRITENEWRANDOM WRITE RECORD UNDER NEW SEQ #
 BCS,8 RNM13 DID THIS SEQ # ALREADY EXIST
 BAL,LNK DELETERECORD NO = DELETE OLD RECORD
 B *R:LNK EXIT
 *
 * ERROR: OLD RECORD DOESN'T EXIST
 *
 RNM10 BAL,LNK TYPMSG TYPE: !=P1!NO SUCH REC!
 DATA ERRP1
 B *R:LNK EXIT
 *
 * ERROR: NEW RECORD ALREADY EXISTS
 *
 RNM13 BAL,LNK TYPMSG TYPE: !=P2!REC EXISTS!
 DATA ERRP2
 B *R:LNK EXIT

PAGE

```

3167
3168
3169
3170
3171
3172
3173      05 00699
3174 05 00699 22300001 A
3175 05 0069A 6800069C
3176
3177
3178      05 0069B
3179 05 0069B 223FFFFFF A
3180 05 0069C 35300167 02
3181 05 0069D 353000E9 02
3182 05 0069E 22300005 A
3183 05 0069F F24600AC 02
3184 05 006A0 B25800AC 02
3185 05 006A1 22300006 A
3186 05 006A2 6A700A34
3187 05 006A3 6A700C61
3188 05 006A4 35B00005 A
3189 05 006A5 31B00E28
3190 05 006A6 683006C0
3191 05 006A7 35B000B9 02
3192 05 006A8 680006B8
3193
3194
3195
3196
3197 05 006A9 22800000 A
3198 05 006AA 358000E9 02
3199 05 006AB 35800167 02
3200 05 006AC 68000004
3201
3202
3203

```

```

*****
* RECRD COMMANDS: SET AND STEP (AND TYPE) *
*****
*
*
R:SET$STEP      EQU $
                LI,X1 1      USE STEPFLAG=1 FOR 'SSI'
                B      R:SET$STEP$TYPE+1
*
*
R:SET$STEP$TYPE EQU $
                LI,X1 =1     USE STEPFLAG=-1 FOR 'ST'
                STW,X1 STEPFLAG TURN ON 'SET AND STEP MODE' FLAGS
                STW,X1 SETFLAG
                LI,X1 5      GET STARTING SEQ # FROM CDT
                LB,X2 *CDTADR,X1
                LW,P1 *CDTADR,X2
                LI,X1 6
                BAL,LNK PROCESSCOL#PAIR PROCESS COL # PARAMS
                BAL,LNK READNXTRAND0M
                STW,R1 P1     PUT FIRST REC. NO. IN P1.
                CW,R1 L(E0F)
                BE      SPL20
                STW,R1 FIRSTSET NO , S0 USE THE FIRST RECORD
                B      FINISH$STEP$LOOP NUMBER FOUND THAT IS HIGHER
                                THAN THE INPUT RECORD NO.
*
*
* NULL COMMAND OR ERROR: TURN OFF 'SET MODE' AND 'STEP MODE' FLAGS
*
STP10 LI,T1 0      TURN OFF MODE FLAGS
      STW,T1 SETFLAG
      STW,T1 STEPFLAG
      B      MASTERPARSER EXIT TO PARSER
*
*
*

```

H01 20144 SEP 08, 175

3204		05 006AU		STEP*LOOP	EGU *	(EXC ENTERS HERE AT IEND OF CDT)	181
3205	05	006AD	33F000C0 02	MTW,-1	NOCHGFLG	WAS INPUT A 'NO' COMMAND	
3206	05	006AE	683006B3	BEZ	SPL10	YES = SKIP WRITE	
3207	05	006AF	325000B9 02	LW,P1	FIRSTSET	WRITE CURRENT RECORD	
3208	05	006B0	6A700E1F	BAL,LNK	WRITERANDOM		
3209	05	006B1	3300019A 02	MTW,0	TTYIMGSZ	WAS INPUT A NULL COMMAND	
3210	05	006B2	683006A9	BEZ	STP10	YES = GO EXIT	
3211				*			
3212				* READ NEXT INPUT RECORD AND TYPE AS REQUIRED			
3213				*			
3214	05	006B3	6A700C7F	SPL10	BAL,LNK READSEQUEN	READ NEXT RECORD	
3215	05	006B4	31B00E28	CW,R1	L(EOF)	WAS IT AN EOF	
3216	05	006B5	683006C0	BE	SPL20	YES = ERROR	
3217	05	006B6	35B000B9 02	STW,R1	FIRSTSET	NO = SAVE NEW SEQ #	
3218	05	006B7	325000B9 02	LW,P1	FIRSTSET		
3219				*			
3220				*			
3221				*			
3222		05 006B8		FINISH*STEP*LOOP	EGU *	('JU' ENTERS HERE TO FINISH)	
3223	05	006B8	6A700A56	BAL,LNK	SETE0D	SET E0D MARKER	
3224	05	006B9	33000167 02	MTW,0	STEPFLAG		
3225	05	006BA	692006BD	BGZ	SPL15	WAS 'ST' CMND USED	
3226	05	006BB	6A700D82	BAL,LNK	TYPECARD	YES = TYPE CARD IMAGE	
3227	05	006BC	68000004	B	MASTERPARSER	EXIT TO PARSER	
3228				*			
3229				* 'SS' COMMAND USED: JUST TYPE SEQ #			
3230				*			
3231	05	006BD	6A700DD8	SPL15	BAL,LNK TYPESEQ	TYPE: 'DDDD.DDD;'	
3232	05	006BE	08000000 A	GEN4	E0M,0,0,0		
3233	05	006BF	68000004	B	MASTERPARSER	EXIT TO PARSER	
3234				*			
3235				* ERROR: EOF HIT			
3236				*			
3237	05	006C0	6A700DC4	SPL20	BAL,LNK TYPMSG	TYPE: '==EOF HIT'	
3238	05	006C1	000001D5 02	DATA	ERRM1		
3239	05	006C2	680006A9	B	STP10	GO EXIT	

PAGE

```

3240
3241
3242
3243
3244
3245      05 006C3
3246
3247      05 006C3
3248      05 006C3      22200001 A
3249      05 006C4      680006C6
3250
3251
3252      05 006C5
3253      05 006C5      22200000 A
3254      05 006C6      22300001 A
3255      05 006C7      353000E9 02
3256
3257
3258      05 006C8      22C00000 A
3259      05 006C9      22300005 A
3260      05 006CA      F24600AC 02
3261      05 006CB      B25800AC 02
3262      05 006CC      20400001 A
3263      05 006CD      B26800AC 02
3264      05 006CE      356000BE 02
3265      05 006CF      20300001 A
3266      05 006D0      6A700A34
3267      00000001
3268      05 006D1      323000B7 02
3269      05 006D2      693006D9
3270      05 006D3      323000BB 02
3271      05 006D4      2130008C A
3272      05 006D5      691006D9
3273      05 006D6      21200001 A
3274      05 006D7      683006D9
3275      05 006D8      331002C7 02
3276      05 006D9
    
```

```

*****
* RECORD COMMANDS: TYPE(SUPPRESSING SEQ. NUMBERS) *
*****
*
R:TYPE$COMPRESSED EQU $
*
R:TYPE      EQU      $
          LI,X4      1      USE X4=1 FOR 'ITY'
          B          R:TYPE$SUP$SEQ+1
*
*
R:TYPE$SUP$SEQ      EQU $
          LI,X4      0      USE X4=0 FOR 'ITS'
          LI,X1      1
          STW,X1      SETFLAG      SET THE SETFLAG TO ONE
*
*
          LI,R2      0      THE RANGE FROM TY IS USED FOR
          LI,X1      5      AN SE COMMAND.
          LB,X2      *CDTADR,X1      START COUNT OF RECORDS OUTPUT.
          LW,P1      *CDTADR,X2      GET ADDR OF FIRST SEQ # IN CDT
          AI,X2      1      SET P1=FIRST SEQ #
          LW,P2      *CDTADR,X2      P2=LAST SEQ #
          STW,P2      LASTSET      SAVE ENDING SEQ #
          AI,X1      1      SET UP COL. NUMBERS
          BAL,LNK      PROCESSCOL#PAIR
          D0          MODE=2
          LW,X1      FRSTCLMN      MUST EXPAND TABS, IF
          BNEZ       TYP5
          LW,X1      LASTCLMN      COL. NO. SPECIFIED, OR
          CI,X1      MAXCLMN
          BL         TYP5
          CI,X4      1      'ITY' AND 'ITC'
          BE         TYP5
          MTW,1      TABXFLAG
          RES        0
TYP5
    
```

3277				FIN		
3278	05	006D9	6A700C61	BAL, LNK	READNXTRAND0M	READ FIRST SEQ # OR NEXT HIGHEST
3279	05	006DA	35B00169 02	STW, R1	SV1STSET	SET UP FIRST RECORD NO.
3280	05	006DB	35B000B9 02	STW, R1	FIRSTSET	AS IF A SET COMMAND WERE GIVIN.
3281				*		
3282				*	READ AND TYPE UNTIL LAST SEQ # READ OR PASSED	
3283				*		
3284	05	006DC	31B00E28	TYP10	CW, R1	L(EOF)
3285	05	006DD	683006F4		BE	TYP20
3286	05	006DE	3160000B A		CW, P2	R1
3287	05	006DF	682006E9		BLE	TYP15
3288	05	006E0	3250000B A		LW, P1	R1
3289	05	006E1	680406E2		B	\$+1, X4
3290	05	006E2	225FFFFFF A		LI, P1	=1
3291	05	006E3	6A7006FE		BAL, LNK	TYP40
3292	05	006E4	6A700A56		BAL, LNK	SETEOB
3293	05	006E5	6A700D82		BAL, LNK	TYPECARD
3294	05	006E6	2UC00001 A		AI, R2	1
3295	05	006E7	6A700C7F		BAL, LNK	READSEQUEN
3296	05	006E8	680006DC		B	TYP10
3297				*		
3298				*	LAST SEQ # HIT OR PASSED: FINISH UP	
3299				*		
3300	05	006E9	691006F1	TYP15	BL	TYP17
3301	05	006EA	3250000B A		LW, P1	R1
3302	05	006EB	680406EC		B	\$+1, X4
3303	05	006EC	225FFFFFF A		LI, P1	=1
3304	05	006ED	6A7006FE		BAL, LNK	TYP40
3305	05	006EE	6A700A56		BAL, LNK	SETEOB
3306	05	006EF	6A700D82		BAL, LNK	TYPECARD
3307	05	006F0	2UC00001 A		AI, R2	1
3308	05	006F1	2UC00000 A	TYP17	AI, R2	0
3309	05	006F2	682006F9		BLEZ	TYP25
3310	05	006F3	6800072C		B	TYP90
3311				*		
3312				*	ERROR: EOF HIT	
3313				*		

H01 20144 SEP 08, 175

184

3314	05	006F4	6A700DC4	TYP20	BAL, LNK	TYPMSG	TYPE; !=EOF HIT!
3315	05	006F5	000001D5	02	DATA	ERRM1	
3316	05	006F6	20C00000	A	AI, R2	0	WERE ANY RECORDS GOTTEN
3317	05	006F7	682006FB		BLEZ	TYP25A	NO, RESET SET FLAG
3318	05	006F8	6800072C		B	TYP90	
3319				*			
3320	05	006F9	6A700DC4	TYP25	BAL, LNK	TYPMSG	TYPE; !=NONE!
3321	05	006FA	000001E6	02	DATA	ERRM6	
3322	05	006FB	22800000	A	LI, T1	0	TURN OFF SET FLAG IF !=NONE!
3323	05	006FC	358000E9	02	STW, T1	SETFLAG	
3324	05	006FD	6800072C		B	TYP90	
3325				*			
3326				*			
3327	05	006FE	323000B7	02	TYP40	LW, X1	ADJUST THE IMAGE FOR COLUMN BOUNDS
3328	05	006FF	6830070A		BEZ	TYP50	OR COMPRESSION,
3329	05	00700	22400000	A	LI, X2	0	MOVE (FIRSTCLMN, LASTCLMN=1), DOWN TO
3330	05	00701	72060024	02	TYP42	LB, R0	ZERO.
3331	05	00702	75080024	02	STB, R0	CARDIMG, X1	
3332	05	00703	20400001	A	AI, X2	1	INCREMENT DEST. COL. #
3333	05	00704	20300001	A	AI, X1	1	INCREMENT TO NEXT BYTE.
3334	05	00705	313000BB	02	CW, X1	LASTCLMN	CHECK IF DONE.
3335	05	00706	68100708		BGE	TYP45	YES
3336	05	00707	68000701		B	TYP42	
3337	05	00708	35400003	A	TYP45	STW, X2	
3338	05	00709	6800070B		B	\$+2	SET FINISH COLUMN FOR NEXT ROUTINE
3339	05	0070A	323000BB	02	TYP50	LW, X1	IF LESS THAN FULL IMAGE DESIRED,
3340	05	0070B	3530000E	A	STW, X1	DO	SAVE TERMINAL POSITION FOR COMPRESS
3341	05	0070C	2130008C	A	CI, X1	MAXCLMN	
3342	05	0070D	68100713		BGE	TYP60	
3343	05	0070E	22400040	A	LI, X2	1 1	BLANK OUT REGION
3344	05	0070F	75460024	02	TYP55	STB, X2	(LASTCLMN, MAXCLMN=1)
3345	05	00710	20300001	A	AI, X1	1	
3346	05	00711	2130008B	A	CI, X1	MAXCLMN=1	
3347	05	00712	6820070F		BLE	TYP55	
3348				*			
3349	05	00713	22300001	A	TYP60	LI, X1	1
3350	05	00714	F24600AC	02	LB, X2	*CDTADR, X1	FINALLY CHECK FOR COMPRESSION .

3351	05 00715	21400016	A		CI,X2	R;TY\$CMND\$NMR	
3352	05 00716	69200718			BG	TYP70	YES. OTHERWISE,
3353				*			
3354	05 00717	680E0000	A	TYP65	B	0,LNK	EXIT
3355				*			
3356				*			
3357	05 00718	22300000	A	TYP70	LI,X1	0	IN RANGE (0, LAST CLMN) COMPRESS
3358	05 00719	22000040	A		LI,R0	1 1	BLANK STRINGS TO LENGTH ONE.
3359	05 0071A	22400000	A		LI,X2	0	
3360	05 0071B	71080024	02	TYP72	CB,R0	CARDIMG,X2	CHECK FOR BLANK IN CURRENT POSITION.
3361	05 0071C	68300725			BE	TYP80	IF NOT,
3362	05 0071D	72F80024	02	TYP75	LB,D1	CARDIMG,X2	MOVE NON-BLANK STRING DOWN.
3363	05 0071E	75080024	02		STB,R0	CARDIMG,X2	BLANKING VACATED POSITIONS.
3364	05 0071F	75F60024	02		STB,D1	CARDIMG,X1	
3365	05 00720	20300001	A		AI,X1	1	INCREMENT TO AND
3366	05 00721	20400001	A		AI,X2	1	FROM BYTE POINTERS.
3367	05 00722	3140000E	A		CW,X2	D0	IF AT UPRR LIMIT.
3368	05 00723	6910071B			BL	TYP72	
3369	05 00724	680E0000	A		B	0,LNK	THEN RETURN
3370				*			
3371				*			
3372	05 00725	20300001	A	TYP80	AI,X1	1	INCREMENT 'TO' POINTER TO LEAVE THIS
3373	05 00726	20400001	A	TYP82	AI,X2	1	BLANK. SKIP TO NON-BLANK.
3374	05 00727	3140000E	A		CW,X2	D0	
3375	05 00728	681E0000	A		BGE	0,LNK	
3376	05 00729	71080024	02		CB,R0	CARDIMG,X2	
3377	05 0072A	68300726			BE	TYP82	
3378	05 0072B	6800071D			B	TYP75	MOVE NEXT STRING DOWN.
3379	05 0072C			TYP90	EGU	\$	
3380	05 0072C	E800000D	A		B	*R;LNK	

PAGE

```

3381
3382
3383
3384
3385
3386
3387      05 00720
3388      05 0072D 22800001 A
3389      05 0072E 358000E9 02
3390      05 0072F 22300005 A
3391      05 00730 F24600AC 02
3392      05 00731 B25800AC 02
3393      05 00732 20400001 A
3394      05 00733 B26800AC 02
3395      05 00734 356000BE 02
3396      05 00735 22300006 A
3397      05 00736 6A700A34
3398      05 00737 323000AC 02
3399      05 00738 F2B000AC 02
3400      05 00739 3030000B A
3401      05 0073A 353000E8 02
3402      05 0073B 6A700C61
3403      05 0073C 35B00169 02
3404      05 0073D 35B000B9 02
3405      05 0073E 35B00005 A
3406      05 0073F 31B000BE 02
3407      05 00740 68200746
3408      05 00741 6A700DC4
3409      05 00742 000001E6 02
3410      05 00743 22800000 A
3411      05 00744 358000E9 02
3412      05 00745 68000004
3413
3414
3415
3416      05 00746 6A700A56
3417      05 00747 E800000D A

*****
* INTRALINE COMMAND: SET *
*****
*
*
I:SET      EGU      #
LI,T1      1      TURN 'ISET MODE' FLAG ON
STW,T1     SETFLAG
LI,X1      5
LB,X2      *CDTADR,X1      GET ADDR OF FIRST SEQ # IN CDT
LW,P1      *CDTADR,X2     SET P1=FIRST SEQ #
AI,X2      1      P2=LAST SEQ #
LW,P2      *CDTADR,X2
STW,P2     LASTSET
LI,X1      6
BAL,LNK    PROCESSCOL#PAIR  PROCESS COL # PARAMS
LW,X1      CDTADR      CALC X1=ADDR IN CDT OF NEXT COMMAND
LB,R1      *CDTADR      AFTER 'ISET'
AW,X1      R1
STW,X1     SETADR      PUT THIS IN SETADR FOR I:CMND LOOP
BAL,LNK    READNXRANDOM READ FIRST RECORD IN RANGE.
STW,R1     SV1STSET    SET FIRST SEQ NO.
STW,R1     FIRSTSET    SET LOOP CONTROL
STW,R1     P1
CW,R1      LASTSET    MAKE SURE THAT THE FIRST RECORD IS
BLE        SET10      IN TH P1=P2 RANGE.
BAL,LNK    TYPMSG     NO = TYPE! !=P1;NO SUCH REC.
DATA      ERRM6      INONE!
LI,T1      0      TURN OFF 'ISET MODE' FLAG
STW,T1     SETFLAG
B          MASTERPARSER  EXIT TO PARSER
*
* SET EOD MARKER AND EXIT
*
SET10     BAL,LNK  SETEOD      SET EOD MARKER
B         B       *I:LNK     EXIT
    
```

```

3418
3419
3420
3421      05 00748      SET$LOOP EQU          $          (EXC ENTERS HERE AT 'END OF CDT')
3422      05 00748      330000E9 02      MTW,0      SETFLAG          HAS ANY INTRALINE CMND BUT 'SE'
3423      05 00749      69200004      BGZ          MASTERPARSER      BEEN EXECUTED
3424      05 0074A      325000B9 02      LW,P1      FIRSTSET          YES * HAS LAST RECORD IN RANGE OF
3425      05 0074B      315000BE 02      CW,P1      LASTSET           I:SET BEEN PROCESSED
3426      05 0074C      69300760      BNE          STL10          NO * GO PROCESS MORE
3427      05 0074D      6A700E1F      BAL,LNK     WRITERANDOM      YES * WRITE LAST RECORD
3428      05 0074E      DB1          MODE=2
3429      05 0074E      355002C3 02      STW,P1      INTFLAG1
3430
3431      * AT END OF SET LOOP: MARK SETFLAG SO LOOP WILL BE RESTARTED IF
3432      * ANOTHER I:CMND IS GIVEN
3433
3434      05 0074F      22800001 A      STL5      LI,T1          1          MARK SETFLAG TO RESTART RANGE ON
3435      05 00750      3250019F 02      LW,P1      CHG:STG:CNT     GET THE NO. OF STRINGS CHANGED
3436      05 00751      21500001 A      CI,P1          1          CHECK FOR ONLY 1 HIT.
3437      05 00752      6830075A      BE          STL30          GO MAKE FURTHER CHECKS.
3438      05 00753      69200756      BG          $+3          CHECK FOR ZERO STRINGS
3439      05 00754      330001A4 02      MTW,0      ZERO:STG:FLG    AND WHETHER TO ANNOUNCE IT
3440      05 00755      6930075A      BNEZ       STL30          NO, DON'T BOTHER
3441      05 00756      22600A65 02      LI,P2      BA(MSG8)+1
3442      05 00757      6A700B11      BAL,LNK     BINTODEC
3443      05 00758      6A700DC4      BAL,LNK     TYPEMSG
3444      05 00759      00000299 02      DATA      MSG8
3445      05 0075A      STL30      EQU          $
3446      05 0075A      22500000 A      LI,P1          0          CLEAR THE CHANGED STRING COUNT
3447      05 0075B      3550019F 02      STW,P1      CHG:STG:CNT
3448      05 0075C      355001A4 02      STW,P1      ZERO:STG:FLG    THE ZERO STRING FLAG,
3449      05 0075D      355001A2 02      STW,P1      TXFLAG          AND THE ITX: FLAG
3450      05 0075E      358000E9 02      STW,T1      SETFLAG         NEXT I:CMND
3451      05 0075F      68000004      B          MASTERPARSER      EXIT TO PARSER
3452
3453      * MORE RECORDS ARE LEFT IN RANGE OF LAST I:SET TO BE PROCESSED
3454

```

H01 20:44 SEP 08, '75

188

3455	05	00760	6A700E1F	STL10	BAL, LNK	WRITERANDOM	WRITE CURRENT RECORD
3456	05	00761			DB1	MODE=2	
3457	05	00761	355002C3	02	STW, P1	INTFLAG1	
3458	05	00762	6A700C7F		BAL, LNK	READSEQUEN	READ NEXT RECORD
3459	05	00763	31B00E28		CW, R1	L(EOF)	WAS IT AN EOF
3460	05	00764	68300770		BE	STL20	YES = ERROR
3461	05	00765	31B000BE	02	CW, R1	LASTSET	IS INPUT SEQ # > SEQ # TO STOP AT
3462	05	00766	6920074F		BG	STL5	YES = GO EXIT
3463	05	00767	35B000B9	02	STW, R1	FIRSTSET	NO = SAVE NEW SEQ #
3464	05	00768	228FFFFFFE	A	LI, T1	=2	SUPPRESS CERRS FOR LOOPS
3465	05	00769	358000B2	02	STW, T1	ERRORCNT	TWO AND FF.
3466	05	0076A	328000E8	02	LW, T1	SETADR	SET CDTADR BACK TO BEGINNING OF LOOP
3467	05	0076B	358000AC	02	STW, T1	CDTADR	
3468	05	0076C	6A700A56		BAL, LNK	SETE0D	SET E0D MARKER
3469	05	0076D	3280016A	02	LW, T1	SVBPFLAG	RESTORE LAST DFLT VALUE
3470	05	0076E	35800023	02	STW, T1	BPFLAG	OF BPFLAG FOR NEXT ITERATION
3471	05	0076F	6800030C		B	RESTART\$EXECUTIVE	GO RESTART IICMND LOOP
3472					*		
3473					*	ERROR: EOF HIT	
3474					*		
3475	05	00770	6A700DC4	STL20	BAL, LNK	TYPEMSG	TYPE: !=EOF HIT!
3476	05	00771	00000105	02	DATA	ERRM1	
3477	05	00772	6800074F		B	STL5	GO EXIT

PAGE

* INTRALINE COMMAND: 'DELETE' X *

*
*
*
*
*
*

05 00773

I:DELETE EQU *
* IT IS AN ERROR TO HAVE STRING SUB 1 ALL BLANKS.*

3478
3479
3480
3481
3482
3483
3484
3485
3486
3487
3488 05 00773 33000020 02
3489 05 00774 6910077D
3490 05 00775 F2300175 02
3491 05 00776 F2660175 02
3492 05 00777 21600040 A
3493 05 00778 6930077D
3494 05 00779 64300776
3495 05 0077A 6A700DC4
3496 05 0077B 00000223 02
3497 05 0077C 68000004
3498 05 0077U
3499 05 0077D 6A7009A4
3500 05 0077E E980000D A
3501 05 0077F 30500006 A
3502 05 00780 32A00006 A
3503 05 00781 22600000 A
3504 05 00782 6A700A70
3505 05 00783 3850000A A
3506 05 00784 6A700978
3507 05 00785 6A700A56
3508 05 00786 E800000D A

MTW,0 ALLFLAG SEE IF ALL FLAG IS SET.
BLZ I:DELETE01
LB,X1 *TEXTCADR GET THE CHARACTER COUNT.
I:DELETE02 LB,P2 *TEXTCADR,X1 SEARCH THE STRING FOR ANY
CI,P2 X'40' NON BLANK CHARACTER.
BNE I:DELETE01 CONTINUE
BDR,X1 I:DELETE02
BAL,LNK TYPEMSG ALL BLANKS MESSAGE
DATA ERRM21
B MASTERPARSER
I:DELETE01 EQU \$
BAL,LNK FINDCOLUMN FIND COLUMN CORRES TO FIRST PARAM
BCS,8 *I:LNK NONE FOUND = EXIT
AW,P1 P2 SET P1,CHAR AFTER PARAM STRING
LW,P3 P2 P2=0 (FIELD WIDTH)
LI,P2 0 P3=# TO SHIFT (=LENGTH OF STRG)
BAL,LNK SHIFLEFT SHIFT LEFT TO DELETE STRING
SW,P1 P3 IF ALLFLAG IS ON, SET TO RESUME
BAL,LNK ADJUSTALLFLAG MATCHING AFTER X AS DELETED
BAL,LNK SETE0D RESET E0D MARKER
B *I:LNK EXIT

PAGE

 * INTRALINE COMMAND: 'OVERWRITE AND EXTEND' X BY Y *

3509
 3510
 3511
 3512
 3513
 3514
 3515 05 00787
 3516 05 00787 35D00021 02
 3517 05 00788 6A7009A4
 3518 05 00789 E980000D A
 3519 05 0078A 20300001 A
 3520 05 0078B F26600AC 02
 3521 05 0078C 306000AC 02
 3522 05 0078D 6A700A19
 3523 05 0078E F2800006 A
 3524 05 0078F 30500008 A
 3525 05 00790 22800040 A
 3526
 3527
 3528
 3529 05 00791 3150008B 02
 3530 05 00792 68100796
 3531 05 00793 758A0024 02
 3532 05 00794 20500001 A
 3533 05 00795 68000791
 3534
 3535
 3536
 3537 05 00796 6A700A56
 3538 05 00797 E800000D A

I:OVERWR\$EXTEND EQU \$
 STW,I;LNK ALL0K
 BAL,LNK FINDCOLUMN FIND COLUMN CORRES TO FIRST PARAM
 BCS,B *I;LNK
 AI,X1 1
 LB,P2 *CDTADR,X1 GET ADDR OF 2ND STRING IN CDT
 AW,P2 CDTADR SET P2,ABSOLUTE ADDR OF STRING
 BAL,LNK MOVESTRING OVERWRITE WITH NEW STRING
 LB,T1 *P2 SET P1=COL. AFTER LAST NEW CHAR
 AW,P1 T1
 LI,T1 ' '
 *
 * BLANK OUT REST OF CARD IMAGE
 *
 0EX10 CW,P1 LASTCLMN BLANK OUT BUFFER FROM CHAR AFTER
 BGE 0EX20 LAST NEW CHAR TO COL. TO STOP AT
 STB,T1 CARDIMG,P1
 AI,P1 1
 B 0EX10
 *
 * SET E0D AND EXIT
 *
 0EX20 BAL,LNK SETE0D RESET E0D MARKER
 B *I;LNK EXIT

PAGE

 * INTRALINE COMMAND: 'FOLLOW' X BY Y *

*
 *

3539
 3540
 3541
 3542
 3543
 3544
 3545 05 00798
 3546 05 00798 6A7009A4
 3547 05 00799 E980000D A
 3548 05 0079A 30500006 A
 3549 05 0079B 22600000 A
 3550 05 0079C 20300001 A
 3551 05 0079D F24600AC 02
 3552 05 0079E 304000AC 02
 3553 05 0079F F2A00004 A
 3554 05 007A0 6A700A99
 3555 05 007A1 32600004 A
 3556 05 007A2 6A700A19
 3557 05 007A3 3050000A A
 3558 05 007A4 6A700978
 3559 05 007A5 6A700A56
 3560 05 007A6 E800000D A

I:FOLLOW\$BY EGU \$
 BAL, LNK FINDCOLUMN FIND COLUMN CORRES TO FIRST PARAM
 BCS, 8 *I:LNK NONE FOUND - EXIT
 AW, P1 P2 SET P1=CHAR AFTER PARAM STRING
 LI, P2 0 P2=0 (FIELD WIDTH)
 AI, X1 1 GET ADDR OF 2ND STRING IN CDT
 LB, X2 *CDTADR, X1 SET X2=ABSOLUTE ADDR OF STRING
 AW, X2 CDTADR P3=LENGTH OF STRING
 LB, P3 *X2
 BAL, LNK SHIFTRIGHT SHIFT RIGHT TO MAKE ROOM FOR 2ND
 LW, P2 X2 STRING
 BAL, LNK MOVESTRING MOVE STRING INTO HOLE
 AW, P1 P3 IF ALLFLAG IS ON, SET TO RESUME
 BAL, LNK ADJUSTALLFLAG MATCHING AFTER Y AS ADDED
 BAL, LNK SETEBD RESET EBD MARKER
 B *I:LNK EXIT

PAGE

```
*****
* INTRALINE COMMAND: SHIFT X 'LEFT' BY N *
*****
*
*
```

```
3561
3562
3563
3564
3565
3566
3567      05 007A7
3568      05 007A7      35D00021 02
3569      05 007A8      6A7009A4
3570      05 007A9      E980000D A
3571      05 007AA     20300001 A
3572      05 007AB     F24600AC 02
3573      05 007AC     B2A800AC 02
3574      05 007AD     E830000D A
3575      05 007AE     6A700A70
3576      05 007AF     6A700A56
3577      05 007B0     E800000D A
```

```
I:SHIFT$LEFT      EQU *
      STW,I;LNK ALLBK
      BAL,LNK FINDCOLUMN      FIND COLUMN CORRES TO FIRST PARAM
      BCS,8 *I;LNK      NONE FOUND = EXIT
      AI,X1      1
      LB,X2 *CDTADR,X1      GET ADDR OF N IN CDT
      LW,P3 *CDTADR,X2      SET P3,NUMBER TO SHIFT (N)
      BEZ *I;LNK      IF N=0 = EXIT
      BAL,LNK SHIFLEFT      SHIFT LEFT N SPACES
      BAL,LNK SETE0D      RESET E0D MARKER
      B *I;LNK      EXIT
```

PAGE

 * INTRALINE COMMAND: 'OVERWRITE' X BY Y *

 *
 *

3578
 3579
 3580
 3581
 3582
 3583
 3584 05 007B1
 3585 05 007B1 6A7009A4
 3586 05 007B2 E980000D A
 3587 05 007B3 20300001 A
 3588 05 007B4 F26600AC 02
 3589 05 007B5 306000AC 02
 3590 05 007B6 6A700A19
 3591 05 007B7 F2300006 A
 3592 05 007B8 30500003 A
 3593 05 007B9 6A700978
 3594 05 007BA 6A700A56
 3595 05 007BB E800000D A

I:OVERWRITE EQU *

BAL, LNK	FINDCOLUMN	FIND COLUMN CORRES TO FIRST PARAM
BCS, 8	*I:LNK	NONE FOUND = EXIT
AI, X1	1	
LB, P2	*CDTADR, X1	GET ADR OF 2ND STRING IN CDT
AW, P2	CDTADR	CALC P2=ABSOLUTE ADDR OF STRING
BAL, LNK	MOVESTRING	OVERWRITE WITH NEW STRING
LB, X1	*P2	IF ALLFLAG IS ON, SET TO RESUME
AW, P1	X1	MATCHING AFTER Y AS OVERWRITTEN
BAL, LNK	ADJUSTALLFLAG	
BAL, LNK	SETE0D	RESET E0D MARKER
B	*I:LNK	EXIT

PAGE

 * INTRALINE COMMAND: 'PRECEDE' X BY Y *

 *
 *

3596
 3597
 3598
 3599
 3600
 3601
 3602 05 007B0
 3603 05 007BC 6A7009A4
 3604 05 007BD E980000D A
 3605 05 007BE 20300001 A
 3606 05 007BF F24600AC 02
 3607 05 007C0 304000AC 02
 3608 05 007C1 F2A00004 A
 3609 05 007C2 6A700A99
 3610 05 007C3 46600004 A
 3611 05 007C4 6A700A19
 3612 05 007C5 30500004 A
 3613 05 007C6 3050000A A
 3614 05 007C7 6A700978
 3615 05 007C8 6A700A56
 3616 05 007C9 E800000D A

I:PRECEDE BY EQU \$
 BAL, LNK FINDCOLUMN FIND COLUMN CORRES TO FIRST PARAM
 BCS, 8 *I:LNK NONE FOUND = EXIT
 AI, X1 1
 LB, X2 *CDTADR, X1 GET ADDR OF 2ND STRING IN CDT
 AW, X2 CDTADR SET X2=ABSOLUTE ADDR OF STRING
 LB, P3 *X2 P3=LENGTH OF STRING
 BAL, LNK SHIFTRIGHT SHIFT RIGHT TO MAKE ROOM FOR 2ND
 XW, P2 X2 STRING
 BAL, LNK MOVESTRING MOVE STRING INTO HOLE
 AW, P1 X2 IF ALLFLAG IS ON, SET TO RESUME
 AW, P1 P3 MATCHING AFTER X AS PRECEDED BY Y
 BAL, LNK ADJUSTALLFLAG
 BAL, LNK SETEOD RESET EOD MARKER
 B *I:LNK EXIT

PAGE

```
*****
* INTRALINE COMMAND: SHIFT X 'RIGHT' BY N *
*****
*
*
```

```
3617
3618
3619
3620
3621
3622
3623      05 007CA
3624 05 007CA 35D00021 02
3625 05 007CB 6A7009A4
3626 05 007CC E980000D A
3627 05 007CD 20300001 A
3628 05 007CE F24600AC 02
3629 05 007CF B2A800AC 02
3630 05 007D0 E830000D A
3631 05 007D1 6A700A99
3632 05 007D2 6A700A56
3633 05 007D3 E800000D A
```

```
I:SHIFT$RIGHT      EGU $
STW,I;LNK ALLBK
BAL,LNK FINDCOLUMN      FIND COLUMN CORRES TO FIRST PARAM
BCS,8 *I;LNK          NONE = FOUND ERROR
AI,X1 1
LB,X2 *CDTADR,X1      GET ADDR OF N IN CDT
LW,P3 *CDTADR,X2      SET P3,NUMBER TO SHIFT (N)
BEZ *I;LNK          IF N=0 = EXIT
BAL,LNK SHIFTRIGHT      SHIFT RIGHT N SPACES
BAL,LNK SETE0D          RESET E0D MARKER
B *I;LNK          EXIT
```

PAGE

 * INTRALINE COMMAND: FOR X 'SUBSTITUTE' Y *

3634
 3635
 3636
 3637
 3638
 3639 05 007D4
 3640 05 007D4 6A7009A4
 3641 05 007D5 E980000D A
 3642 05 007D6 20300001 A
 3643 05 007D7 F24600AC 02
 3644 05 007D8 304000AC 02
 3645 05 007D9 F2A00004 A
 3646 05 007DA 32800005 A
 3647 05 007DB 30500006 A
 3648 05 007DC 38A00006 A
 3649 05 007DD 682007E1
 3650 05 007DE 22600000 A
 3651 05 007DF 6A700A99
 3652 05 007E0 680007E5
 3653
 3654
 3655
 3656 05 007E1 683007E5
 3657 05 007E2 3AA0000A A
 3658 05 007E3 22600000 A
 3659 05 007E4 6A700A70
 3660
 3661
 3662
 3663 05 007E5 32500008 A
 3664 05 007E6 32600004 A
 3665 05 007E7 6A700A19
 3666 05 007E8 F2800006 A
 3667 05 007E9 30500008 A
 3668 05 007EA 6A700978
 3669 05 007EB 6A700A56
 3670 05 007EC E800000D A

*
 I:SUBSTITUTE EQU \$
 BAL, LNK FINDCOLUMN FIND COLUMN CORRES TO FIRST PARAM
 BCS, 8 *I:LNK NONE FOUND = EXIT
 AI, X1 1
 LB, X2 *CDTADR, X1 GET ADDR OF 2ND STRING IN CDT
 AW, X2 CDTADR SET X2=ABSOLUTE ADDR OF STRING
 LB, P3 *X2 P3=LENGTH OF STRING
 LW, T1 P1 SAVE P1
 AW, P1 P2 SET P1=CHAR AFTER PARAM1 STRING
 SW, P3 P2 CALC NUMBER TO SHIFT IN P3
 BLEZ SBS10 IS NEW STRING LONGER THAN OLD STRING
 LI, P2 0 YES = SET P2=0 (FIELD WIDTH)
 BAL, LNK SHIFTRIGHT SHIFT RIGHT AMOUNT OF DIFFERENCE
 B SBS15 GO TO MOVE IN NEW STRING
 *
 * NEW STRING SHORTER OR EQUAL THAN OLD ONE
 *
 SBS10 BE SBS15 ARE NEW AND OLD STRINGS OF = LENGTH
 LCW, P3 P3 NO = NEW SHORTER
 LI, P2 0 SET P2=0 (FIELD WIDTH)
 BAL, LNK SHIFLEFT SHIFT LEFT AMOUNT OF DIFFERENCE
 *
 * MOVE NEW STRING INTO POSITION
 *
 SBS15 LW, P1 T1 SET P1=CBL. OF PARAM1 STRING
 LW, P2 X2 P2=ADDR OF NEW STRING
 BAL, LNK MOVESTRING MOVE NEW STRING IN PLACE
 LB, T1 *P2 IF ALLFLAG IS ON, SET TO RESUME
 AW, P1 T1 MATCHING AFTER Y AS SUBSTITUTED
 BAL, LNK ADJUSTALLFLAG
 BAL, LNK SETE0D RESET E0D MARKER
 B *I:LNK EXIT

PAGE

 * INTRALINE COMMAND: JUMP *

3671
 3672
 3673
 3674
 3675
 3676
 3677 05 007E0
 3678 05 007ED 39000167 02
 3679 05 007EE 683007F8
 3680 05 007EF 325000B9 02
 3681 05 007F0 6A700E1F
 3682 05 007F1 22300005 A
 3683 05 007F2 F24600AC 02
 3684 05 007F3 B25800AC 02
 3685 05 007F4 6A700C6D
 3686 05 007F5 698007FB
 3687 05 007F6 3b5000B9 02
 3688 05 007F7 680006B8
 3689
 3690
 3691
 3692 05 007F8 6A700D8F
 3693 05 007F9 000001B1 02
 3694 05 007FA E800000D A
 3695
 3696
 3697
 3698 05 007FB 6A700D8F
 3699 05 007FC 000001AD 02
 3700 05 007FD 325000B9 02
 3701 05 007FE 6A700C6D
 3702 05 007FF E800000D A

I:JUMP FQU \$
 MTW,0 STEPFLAG IS SYSTEM IN 1STEP MODE!
 BEZ JMP10 NO = ERROR
 LW,P1 FIRSTSET
 BAL,LNK WRITERANDBM WRITE CURRENT RECORD
 LI,X1 5 GET SEQ # FOR JUMP FROM CDT
 LB,X2 *CDTADR,X1
 LW,P1 *CDTADR,X2
 BAL,LNK READRANDOM READ THIS RECORD
 BCS,8 JMP15 DID IT EXIST
 STW,P1 FIRSTSET SAVE NEW SEQ #
 B FINISH\$STEP\$LOOP YES = GO FINISH JUMP

*
 * ERROR: 1JUI ILLEGAL AT THIS POINT
 *

JMP10 BAL,LNK TYPECERR TYPE: 1-CN;CMND ILGL HERE!
 DATA ERRC4
 B *I:LNK EXIT

*
 * ERROR: RECORD TO JUMP TO DOESN'T EXIST
 *

JMP15 BAL,LNK TYPECERR TYPE: 1-CN;NO SUCH REC!
 DATA ERRC3
 LW,P1 FIRSTSET
 BAL,LNK READRANDOM RESTORE OLD RECORD
 B *I:LNK EXIT

PAGE

* INTRALINE COMMAND: NO CHANGE *

*
*

3703
3704
3705
3706
3707
3708
3709 05 00800
3710 05 00800 33000167 02
3711 05 00801 68300805
3712 05 00802 22800001 A
3713 05 00803 358000C0 02
3714 05 00804 E800000D A
3715
3716
3717
3718 05 00805 6A700D8F
3719 05 00806 000001B1 02
3720 05 00807 E800000D A

I:NO\$CHANGE EQU \$
MTW,0 STEPFLAG IS SYSTEM IN STEP MODE
BEZ NCG10 NO = ERROR
LI,T1 1
STW,T1 NOCHGFLG TURN ON 'NO CHANGE' FLAG
B *I;LNK EXIT
*
* ERROR: 'NO' ILLEGAL AT THIS POINT
*
NCG10 BAL,LNK TYPECERR TYPE: '-CN:CMND ILGL HERE'
DATA ERRC4
B *I;LNK EXIT

PAGE

```

*****
* INTRALINE COMMAND: REVERSE BLANK PRESERVATION FLAG *
*****
*
*

```

```

3721
3722
3723
3724
3725
3726
3727      05 00808
3728      05 00808      32800023 02
3729      05 00809      48800000 02
3730      05 0080A      35800023 02
3731      05 0080B      E800000D A

```

```

I:REVERSE$BPFLAG EQU $
          LW,T1    BPFLAG      REVERSE BPFLAG
          EOR,T1   K1
          STW,T1   BPFLAG
          B        *I:LNK      EXIT

```

PAGE

* INTRALINE COMMANDS: TYPE(SUPPRESSING SEQ. NUMBERS) *

3732
3733
3734
3735
3736
3737
3738 05 0080C
3739 05 0080C 331001A4 02
3740 05 0080D 325000B9 02
3741 05 0080E 6A700D82
3742 05 0080F E800000D A
3743
3744
3745 05 00810
3746 05 00810 331001A4 02
3747 05 00811 225FFFFFF A
3748 05 00812 6A700D82
3749 05 00813 E800000D A
3750
3751 05 00814
3752 05 00814 330001A2 02
3753 05 00815 E830000D A
3754 05 00816 325000B9 02
3755 05 00817 6A700D82
3756 05 00818 22500000 A
3757 05 00819 355001A2 02
3758 05 0081A E800000D A
3759

I:TYPE EQU \$
MTW,1 ZERO:STG:FLG DON'T OUTPUT ON TY
LW,P1 FIRSTSET GET SEQ #
BAL,LNK TYPECARD TYPE CARD IMAGE WITH SEQ #
B *I:LNK EXIT
*
*
I:TYPE*SUP*SEQ EQU \$
MTW,1 ZERO:STG:FLG DON'T OUTPUT ON TS
LI,P1 =1
BAL,LNK TYPECARD TYPE CARD IMAGE WITHOUT SEQ #
B *I:LNK EXIT
*
I:TYPEX EQU \$
MTW,0 TXFLAG ANYTHING CHANGED ON THIS LINE
BEZ *I:LNK NO
LW,P1 FIRSTSET GET SEQUENCE NO.
BAL,LNK TYPECARD TYPE CARD IMAGE WITH SEQ NO.
LI,P1 0 ZERO ITX! FLAG
STW,P1 TXFLAG
B *I:LNK RETURN
*

PAGE

 * ADD NEW PARAMETER TO CDT *
 * P1 = TYPE OF PARAMETER *

3760
 3761
 3762
 3763
 3764
 3765
 3766
 3767
 05 0081B 02200030 A
 05 0081C 0B30019C 02
 3769 05 0081D 323000E5 02
 3770 05 0081E F55600AC 02
 3771 05 0081F 20300001 A
 3772 05 00820 F25000AC 02
 3773 05 00821 F55600AC 02
 3774 05 00822 332000E5 02
 3775 05 00823 305000E6 02
 3776 05 00824 F55000AC 02
 3777 05 00825 385000E6 02
 3778 05 00826 305000AC 02
 3779 05 00827 323000E6 02
 3780 05 00828 22400000 A
 3781 05 00829 32E800C1 02
 3782 05 0082A B5E80005 A
 3783 05 0082B 20400001 A
 3784 05 0082C 64300829
 3785 05 0082D B23000AC 02
 3786 05 0082E 4B300005 02
 3787 05 0082F 20300100 A
 3788 05 00830 B5380005 A
 3789 05 00831 02200030 A
 05 00832 0A30019C 02
 3790 05 00833 680E0000 A

ADDCDTPARAM EQU \$
 PUSH (X1,P1)
 LW,X1 PARAMPSN
 STB,P1 *CDTADR,X1
 AI,X1 1
 LB,P1 *CDTADR
 STB,P1 *CDTADR,X1
 MTW,2 PARAMPSN
 AW,P1 PRMBUFSZ
 STB,P1 *CDTADR
 SW,P1 PRMBUFSZ
 AW,P1 CDTADR
 LW,X1 PRMBUFSZ
 LI,X2 0
 LW,D0 PARAMBUF,X2
 STW,D0 *P1,X2
 AI,X2 1
 BDR,X1 #-3
 LW,X1 *CDTADR
 AND,X1 XFF00
 AI,X1 X'0100'
 STW,X1 *P1,X2
 PULL (X1,P1)
 B 0,LNK

SAVE REGS
 BUILD CONTROL HW FOR PARAM IN CDT:
 BYTE 0: PARAM TYPE
 BYTE 1: LOC OF PARAM VALUE RELATIVE TO CURRENT CDTADR
 INCR TO NEXT HW
 ADJUST COUNT OF # OF WORDS IN ENTRY BY SIZE OF PARAM
 SET P1=ABSOLUTE ADR TO PUT VALUE AT
 MOVE PARAM VALUE TO CDT ENTRY
 BUILD IEND OF CDT' MARKER USING NUMBER OF NEXT CMND IN CDT
 SET IEND OF CDT' MARKER
 RESTORE REGS
 EXIT

PAGE

* CHECK IF ONLY ONE ENTRY IN CDT *

*
*

3791				
3792				
3793				
3794				
3795				
3796				
3797		05 00834		
3798	05 00834	32E00048	02	
3799	05 00835	21E00001	A	
3800	05 00836	683E0000	A	
3801	05 00837	6A700D8F		
3802	05 00838	000001B1	02	
3803	05 00839	68000004		

CHECK1CDTENTRY	EQU \$	
LW,DO	CDT	CHECK IF ONLY ONE ENTRY IN CDT
CI,DO	1	
BE	0,LNK	YES = EXIT
BAL,LNK	TYPECERR	NO = TYPE: !=CN: CMND ILGL HERE!
DATA	ERRC4	
B	MASTERPARSER	EXIT TO PARSER

PAGE

```

3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831

```

05	0083A	02200060	A			
05	0083B	0B30019C	02			
05	0083C	22300000	A			
05	0083D	6A700889				
05	0083E	01000236	02			
05	0083F	01000840				
05	00840	728000C1	02			
05	00841	2180001F	A			
05	00842	68200845				
05	00843	22700E29		GF5		
05	00844	680008A1				
05	00845	6A70087F				
05	00846	328000AD	02			
05	00847	6A700889				
05	00848	0B000239	02			
05	00849	0100084E				
05	0084A	0B000854				
05	0084B	0B000035				
05	0084C	0700084E				
05	0084D	0U00084E				
05	0084E	358000AD	02			
05	0084F	22600000	A			
05	00850	0960019C	02			
05	00851	0960019C	02			

```

*****
* GET FILE IDENTIFICATION *
*****
*
*
GETFILEID      EQU $
                (X1,T1)          SAVE REGS
                PUSH
                LI,X1            0          USE X1 AS COUNT OF # OF WDS PUSHED
                NXTNAM          ERRP3,,
                                (NAME,*)
                LB,T1           PARAMBUF   ALLOW ONLY <= 31 BYTES IN FILE
                CI,T1           31        NAME.
                BLE             $+3
                LI,LNK          L(ERRP3)
                B               GETNEXT$ERROR
                BAL,LNK         GF$PUSH$SUBR  PUSH 'FILE NAME' PARAM
                LW,T1           CHARPSN    SAVE NEXT SCAN PSN
                NXTNAM          ERRP4,,
                                (NAME,*),,
                                (S(LPAR,PERIOD),GF10),,
                                (SCOL,ILGL$SEMICOLON),,
                                (COM,*),,
                                (END,*)
                STW,T1          CHARPSN    RESTORE TO SCAN , OR C/R AGAIN
                LI,P2           0
                PUSH            P2        SET 'ACCT #' & 'PASSWORD' PARAMS =0
                PUSH            P2

```

3832	05	00852	20300002	A	AI,X1	2	ADJ PUSH COUNT
3833	05	00853	68000877		B	GF30	GO FINISH UP
3834					*		
3835					*		LEFT PARENTHESIS FOUND: GET ACCOUNT NUMBER AND PASSWORD
3836					*		
3837					GF10	NXTNAM	ERRP3,,
3838							(NAME,GF15),,
3839	05	00854	6A700889				(S(COM,PERIOD),*)
	05	00855	02000236	02			
	05	00856	0100085C				
	05	00857	08000858				
3840	05	00858	22600000	A	LI,P2	0	
3841	05	00859	0960019C	02	PUSH	P2	SET IACCT #1 PARAM = 0
3842	05	0085A	20300001	A	AI,X1	1	
3843	05	0085B	68000867		B	GF18	GO PROCESS 'PASSWORD'
3844					*		
3845					*		ACCOUNT NUMBER FOUND: PROCESS IT
3846					*		
3847	05	0085C	728000C1	02	GF15	LB,T1	PARAMBUF
3848	05	0085D	21800008	A		CI,T1	8
3849	05	0085E	69200843			BG	GF5
3850	05	0085F	6A70087F			BAL,LNK	GF\$PUSH\$SUBR
3851	05	00860				DB1	MODE=2
3852	05	00860	328000AD	02		LW,T1	CHARPSN
3853						NXTNAM	ERRP3,,
3854							(S(COM,PERIOD),GF18),,
3855							(S(RPAR,NAME),GF20),,
3856							(S(RPAR,COM),GF20),,
3857	05	00861	6A700889				(S(RPAR,END),GF20)
	05	00862	04000236	02			
	05	00863	08000867				
	05	00864	01000874				
	05	00865	07000874				
	05	00866	00000874				
3858					*		
3859					*		PASSWORD PRESENT: GET AND PROCESS IT
3860					*		

3861				GF18	NXTNAM	ERRP3,,	
3862	05	00867	6A700889			(NAME,*)	
	05	00868	01000236	02			
	05	00869	0100086A				
3863	05	0086A	728000C1	02	LB,T1	PARAMBUF	& CHARACTERS MAX.
3864	05	0086B	21800008	A	CI,T1	8	
3865	05	0086C	69200843		BG	GF5	
3866	05	0086D	6A70087F		BAL, LNK	GF,PUSH\$SUBR	PUSH (PASSWORD, PARAM
3867	05	0086E			DB1	MODE=2	
3868	05	0086E	32800CAD	02	LW,T1	CHARPSN	
3869					NXTNAM	ERRP3,,	
3870						(S(RPAR,NAME),GF30),,	
3871						(S(RPAR,COM),GF30),,	
3872	05	0086F	6A700889			(S(RPAR,END),GF30)	
	05	00870	09000236	02			
	05	00871	01000877				
	05	00872	07000877				
	05	00873	00000877				
3873					*		
3874					* NB PASSWORD PRESENT		
3875					*		
3876	05	00874	22600000	A	GF20	LI,P2	0
3877	05	00875	0960019C	02		PUSH	P2
3878	05	00876	20300001	A		AI,X1	1
3879					*		
3880					* RECONSTRUCT FILE ID IN 'PARAMBUF'		
3881					*		
3882	05	00877	353000E6	02	GF30	STW,X1	PRMBUFSZ
3883	05	00878				DB1	MODE=2
3884	05	00878	358000AD	02		STW,T1	CHARPSN
3885	05	00879	0860019C	02		PULL	P2
3886	05	0087A	356600C0	02		STW,P2	PARAMBUF=1,X1
3887	05	0087B	64300879			RDR,X1	*=2
3888	05	0087C	02200060	A		PULL	(X1,T1)
	05	0087D	0A30019C	02			RESTORE REGS
3889	05	0087E	680E0000	A		B	0, LNK
3890					*		EXIT

NO1 20:44 SEP 08, '75

208

3940	05	0089D	683008AC	BE	GN30	YES = GO PUT CHAR IN PARAMBUF
3941	05	0089E	6430089C	BDR,X1	*-2	LOOP
3942	05	0089F	21400001 A	CI,X2	1	NOT A 'NAME' CHAR = WERE ANY SUCH
3943	05	008A0	692008B2	BG	GN35	CHARS FOUND (IF NO, ERROR)
3944				*		
3945				*		
3946				*		
3947	05	008A1		GETNEXT\$ERR0R	EGU *	(ENTER HERE IF NO LEGAL TYPE FOUND)
3948	05	008A1	325E0000 A	LW,P1	0,LNK	GET ADDR OF ERROR MSG
3949	05	008A2	31500009 02	CW,P1	X800000	TEST IF 'DECR PARAMPSN' BIT SET
3950	05	008A3	684008A5	BAZ	*+2	
3951	05	008A4	39E000E5 02	MTW,-2	PARAMPSN	YES = DECR PARAM PSN BY 1
3952	05	008A5	48500007 02	AND,P1	X1FFFF	
3953	05	008A6	2150022E 02	CI,P1	ERRP1	IS IT A 'P' ERROR
3954	05	008A7	691008AA	BL	GN25	NO = IT IS A 'C' ERROR
3955	05	008A8	355002BA 02	STW,P1	DMY\$TYPEPERR+1	PUT ERROR MSG ADDR IN DUMMY CALL
3956	05	008A9	680002B9 02	B	DMY\$TYPEPERR	GO PRINT ERROR MSG
3957				*		
3958				*		
3959				*		
3960	05	008AA	355002B7 02	GN25	STW,P1	DMY\$TYPECERR+1
3961	05	008AB	680002B6 02	B	DMY\$TYPECERR	PUT ERROR MSG ADDR IN DUMMY CALL
3962				*		
3963				*		
3964				*		
3965	05	008AC	755800C1 02	GN30	STB,P1	PARAMBUF,X2
3966	05	008AD	20400001 A	AI,X2	1	PUT CHAR IN PARAMBUF
3967	05	008AE	725C0176 02	LB,P1	TTYIMG,P2	INCR PARAMBUF INDEX
3968	05	008AF	20600001 A	AI,P2	1	GET NEXT INPUT CHAR
3969	05	008B0	21500040 A	CI,P1	' '	INCR CHAR PSN
3970	05	008B1	69300895	BNE	GN10	IS CHAR=BLANK
3971				*		NO = GO GET NEXT CHAR
3972				*		
3973				*		
3974	05	008B2	22300003 A	GN35	LI,X1	3
3975	05	008B3	22500040 A	LI,P1	' '	
3976	05	008B4	755800C1 02	STB,P1	PARAMBUF,X2	PUT 3 TRAILING BLANKS ON 'NAME'

3977	05	008B5	20400001	A	AI,X2	1		
3978	05	008B6	643008B4		BDR,X1	\$=2		
3979	05	008B7	204FFFFC	A	AI,X2	=4	PUT COUNT IN PARAMBUF TO FORM	
3980	05	008B8	754000C1	02	STB,X2	PARAMBUF	TEXTC=STRING	
3981	05	008B9	20400004	A	AI,X2	4	SET PARAMBUF SIZE = # WDS OF TEXT	
3982	05	008BA	2540007E	A	SLS,X2	=2		
3983	05	008BB	22500001	A	LI,P1	NAME	SET TYPE=NAME!	
3984	05	008BC	206FFFFF	A	AI,P2	=1	SET CHAR PSN TO RESCAN LAST CHAR	
3985					*			
3986					*			
3987					*			
3988		05 008BU			GETNEXT\$FINISH	EQU \$	(ENTER HERE IF LEGAL TYPE FOUND)	
3989	05	008BD	354000E6	02	STW,X2	PRMBUFSZ	SET PARAMBUF SIZE	
3990	05	008BE	F2300007	A	LB,X1	*LNK	SET X1=# OF BRANCHES	
3991	05	008BF	22400004	A	LI,X2	4	SET X2=INDEX INTO PARAM LIST	
3992	05	008C0	F1580007	A	CB,P1	*LNK,X2	SEARCH FOR CORRES TYPE IN LIST	
3993	05	008C1	683008C5		BE	GN45		
3994	05	008C2	20400004	A	AI,X2	4	INCR INDEX	
3995	05	008C3	643008C0		BDR,X1	\$=3	LOOP	
3996	05	008C4	680008A1		B	GETNEXT\$ERROR	NONE FOUND = ERROR	
3997					*			
3998					*	MATCHING BRANCH FOUND: GO EXECUTE IT		
3999					*			
4000	05	008C5	2540007E	A	GN45	SLS,X2	=2	SET D1=BRANCH ADDR
4001	05	008C6	82F80007	A	LW,D1	*LNK,X2		
4002	05	008C7	356000AD	02	STW,P2	CHARPSN	RESET CHAR PSN	
4003	05	008C8	02200040	A	PULL	(X1,P2)	RESTORE REGS	
	05	008C9	0A30019C	02				
4004	05	008CA	E800000F	A	B	*D1	GO TO BRANCH ADDR	
4005					*			
4006					*	A LEGAL 'GETNEXTNAME' TYPE FOUND		
4007					*			
4008	05	008CB	725608CF		GN50	LB,P1	GNTYTB1,X1	SET P1=TYPE OF MATCH FOUND
4009	05	008CC	680008BD		B	GETNEXT\$FINISH	GO FINISH UP	
4010					*			
4011					*	TABLE OF LEGAL 'GETNEXTNAME' MATCH CHARS		
4012					*			

H01 20:44 SEP 08, '75

4013 05 008CD 00
 4014 05 008CD 1 00
 4015 05 008CD 2 65
 4016 05 008CD 3 45
 4017 05 008CE 15
 4018 00000004
 4019
 4020
 4021
 4022
 4023
 4024 05 008CF
 4025 05 008CF 1 00
 4026 05 008CF 2 07
 4027 05 008CF 3 05
 4028 05 008DO 00
 4029
 4030
 4031
 4032
 4033
 4034 05 008D1
 4035 05 008D1 1 60
 4036 05 008D1 2 55
 4037 05 008D1 3 50
 4038 05 008D2 1 60
 4039 05 008D2 2 7A
 4040 05 008D2 3 75
 4041 05 008D3 7C
 4042 05 008D3 1
 4043 00000008
 4044
 4045
 4046
 4047

GNTBL1 EGU \$
 DATA,1 0 (FILLER)
 DATA,1 CR 0: C/R HIT
 DATA,1 CM 7: COMMA
 DATA,1 S(LP,PR) 9,11: LEFT PAREN,PERIOD
 DATA,1 S(RP,LF) 10,0: RIGHT PAREN, LINE FEED.
 GNTBL1SZ EGU BA(\$)-BA(GNTBL1)=1
 BOUND 4
 *
 * TABLE OF TYPES CORRESPONDING TO LEGAL CHARS
 *
 GNTYTBL1 EGU \$
 DATA,1 0 (FILLER)
 DATA,1 0 0: C/R HIT
 DATA,1 7 7: COMMA
 DATA,1 S(9,11) 9,11: LEFT PAREN,PERIOD
 DATA,1 S(10,0) 10,0: RIGHT PAREN, LINE FEED.
 BOUND 4
 *
 * TABLE OF LEGAL SPECIAL CHARS IN A 'NAME'
 *
 GNTBL2 EGU \$
 DATA,1 0
 DATA,1 ' '
 DATA,1 '\$'
 DATA,1 '*'
 DATA,1 '-'
 DATA,1 'X'
 DATA,1 '!'
 DATA,1 '#'
 DATA,1 '@'
 D01 MODE=1
 DATA,1 ' '
 S
 GNTBL2SZ EGU BA(\$)-BA(GNTBL2)=1
 BOUND 4


```

4084
4085 05 008EA 755800C1 02 GP10 STB,P1 PARAMBUF,X2 PUT CHAR IN PARAMBUF
4086 05 008EB 20400001 A AI,X2 1 INCR PARAMBUF INDEX
4087 05 008EC 725C0176 02 LB,P1 TTYIMG,P2 GET NEXT CHAR
4088 05 008ED 20600001 A AI,P2 1 INCR CHAR PSN
4089 00000001 DB MODE=2
4090 05 008EE 1950001E 02 CLM,P1 LCLETTERS
4091 05 008EF 689008EA BIL GP10
4092 FIN
4093 05 008F0 1950001C 02 CLM,P1 LETTERS IS CHAR A LETTER
4094 05 008F1 689008EA BIL GP10 YES = LOOP
4095 05 008F2 22500006 A LI,P1 ALPH NO = SET TYPE=ALPH
4096 05 008F3 68000905 B GP40 GO FINISH UP
4097
4098 *
4099 * A LEGAL 'GETNEXTPARAM' TYPE FOUND
4100 05 008F4 72560961 GP20 LB,P1 GPTYTBL,X1 SET P1=TYPE OF MATCH FOUND
4101 05 008F5 22400001 A LI,X2 1 SET INDEX IN CASE OF DEFAULT
4102 05 008F6 680008BD B GETNEXT$FINISH GO FINISH UP
4103
4104 *
4105 * STRING FOUND: BUILD TEXTC=STRING IN PARAMBUF
4106 05 008F7 725C0176 02 GP30 LB,P1 TTYIMG,P2 GET NEXT INPUT CHAR
4107 05 008F8 20600001 A AI,P2 1 INCR CHAR PSN
4108 05 008F9 3160019A 02 CW,P2 TTYIMG$Z CHECK IF END OF CMND HIT
4109 05 008FA 69200914 BG GP45 YES = ERROR
4110 05 008FB 21500061 A CI,P1 '/ ' IS CHAR='/'
4111 05 008FC 68300900 BE GP35
4112 05 008FD 755800C1 02 GP30A STB,P1 PARAMBUF,X2 NO = PUT CHAR IN PARAMBUF
4113 05 008FE 20400001 A AI,X2 1 INCR PARAMBUF INDEX
4114 05 008FF 680008F7 B GP30 LOOP
4115
4116 *
4117 * '/' FOUND: DETERMINE IF IT IS END OF STRING OR '/'
4118 05 00900 725C0176 02 GP35 LB,P1 TTYIMG,P2 GET NEXT INPUT CHAR
4119 05 00901 20600001 A AI,P2 1 INCR CHAR PSN
4120 05 00902 21500061 A CI,P1 '/' IS IT A '/' ALSO

```

4121	05 00903	683008FD	BE	GP30A	YES = PUT ONE '/' IN PARAMBUF
4122	05 00904	22500005 A	LI,P1	STRG	NO = SET TYPE='STRG'
4123			*		
4124			*	END OF ALPHA TEST OR STRING FOUND: ADD TRAILING BLANKS AND FINISH	
4125			*	BUILDING PARAMBUF	
4126			*		
4127	05 00905	22300003 A	GP40	LI,X1 3	
4128	05 00906	22F00040 A		LI,D1 1 1	
4129	05 00907	75F800C1 02		STB,D1 PARAMBUF,X2	PUT 3 TRAILING BLANKS ON TEXT OR STRING
4130	05 00908	20400001 A		AI,X2 1	
4131	05 00909	64300907		BDR,X1 3-2	
4132	05 0090A	204FFFFC A		AI,X2 4	CALC LENGTH OF STRING
4133	05 0090B	68300911		BEZ GP43	IS LENGTH=0
4134	05 0090C	754000C1 02		STB,X2 PARAMBUF	NO = BUILD TEXTC=STRING WITH LENGTH
4135	05 0090D	20400004 A		AI,X2 4	SET PARAMBUF SIZE = # OF WDS OF TEXT
4136	05 0090E	2540007E A		SLS,X2 2	
4137	05 0090F	206FFFFF A		AI,P2 1	SET CHAR PSN TO RESCAN LAST CHAR
4138	05 00910	680008BD		B GETNEXT*FINISH	GO FINISH UP
4139			*		
4140			*	ERROR: STRING IS NULL	
4141			*		
4142	05 00911	6A700D99	GP43	BAL,LNK TYPEPERR	TYPE: 1=PN;NULL STRNG!
4143	05 00912	00000276 02		DATA ERRP18	
4144	05 00913	68000004		B MASTERPARSER	GO TO PARSER
4145			*		
4146			*	ERROR: STRING TOO LONG TO FIT IN BUFFER	
4147			*		
4148	05 00914	6A700D99	GP45	BAL,LNK TYPEPERR	TYPE: 1=PN;ILGL STRG!
4149	05 00915	00000265 02		DATA ERRP15	
4150	05 00916	68000004		B MASTERPARSER	EXIT TO PARSER
4151			*		
4152			*	DIGIT OR DECIMAL POINT FOUND: INITIALIZE	
4153			*		
4154	05 00917	22300000 A	GP50	LI,X1 0	USE X1 TO INDICATE 1ST OR 2ND SEQ #
4155	05 00918	224FFFFFF A		LI,X2 1	USE X2 TO SHOW INTG(=1) OR SEQ(>=0)
4156	05 00919	22F00000 A		LI,D1 0	USE D1 AS ACCUMULATOR
4157			*		

```

4158 * DETERMINE WHAT WAS FOUND; IF DIGIT, ACCUMULATE DIGITS AS A BINARY
4159 * NUMBER
4160 *
4161 05 0091A 1950001A 02 GP52 CLM,P1 DIGITS IS CHAR A DIGIT
4162 05 0091B 68900920 BIL GP52A YES = GO ACCUMULATE IT
4163 05 0091C 2150004B A CI,P1 1,1 IS CHAR A 1,1
4164 05 0091D 69300938 BNE GP60
4165 05 0091E 22400003 A LI,X2 3 YES = USE X3 TO CNT DIGITS AFTER 1,1
4166 05 0091F 68000928 B GP53 GO PRCESS 1,1
4167 05 00920 23F0000A A GP52A MI,D1 10 ACCUMULATE DIGIT
4168 05 00921 205FFF10 A AI,P1 =10'
4169 05 00922 30F00005 A AW,D1 P1
4170 05 00923 31F00E2A CW,D1 L(10000)
4171 05 00924 68100931 BGE GP53A
4172 05 00925 725C0176 02 LB,P1 TTYIMG,P2 GET NEXT INPUT CHAR
4173 05 00926 20600001 A AI,P2 1 INCR CHAR PSN
4174 05 00927 6800091A B GP52 LOOP
4175 *
4176 * DECIMAL POINT FOUND; ACCUMULATE DIGITS AFTER IT
4177 *
4178 05 00928 725C0176 02 GP53 LB,P1 TTYIMG,P2 GET NEXT INPUT CHAR
4179 05 00929 20600001 A AI,P2 1 INCR CHAR PSN
4180 05 0092A 1950001A 02 CLM,P1 DIGITS IS CHAR A DIGIT
4181 05 0092B 69900934 BBL GP55
4182 05 0092C 23F0000A A MI,D1 10 YES = ACCUMULATE IT
4183 05 0092D 205FFF10 A AI,P1 =10'
4184 05 0092E 30F00005 A AW,D1 P1
4185 05 0092F 204FFFFFF A AI,X2 =1 CHECK IF >3 DIGITS FOUND
4186 05 00930 68100928 BGEZ GP53 NO = LOOP
4187 05 00931 6A700D99 GP53A BAL,LNK TYPEPERR YES = TYPE; 1-PN:ILGL SEQ #!
4188 05 00932 00000250 02 DATA ERRP10
4189 05 00933 68000004 B MASTERPARSER GO TO PARSER
4190 *
4191 * END OF DIGITS AFTER DECIMAL POINT
4192 *
4193 05 00934 21400000 A GP55 CI,X2 0 WERE EXACTLY 3 DIGITS FOUND
4194 05 00935 68300938 BE GP60

```

H01 20:44 SEP 08, 175

4195	05 00936	23F0000A	A	MI,D1	10		NO = ADJ SEQ # FOR MISSING DIGITS
4196	05 00937	64400936		BDR,X2	*-1		
4197				*			
4198				*		END OF INTEGER OR SEQ #: SEE IF SEQ # PAIR PRESENT	
4199				*			
4200	05 00938	21300001	A	GP60	CI,X1	1	WAS THIS 2ND SEQ # OF PAIR
4201	05 00939	6830094B			BE	GP63	
4202	05 0093A	21500060	A		CI,P1	1-1	NO = DOES A 1-1 FOLLOW FIRST
4203	05 0093B	69300957			BNE	GP66	
4204	05 0093C	214FFFFFF	A		CI,X2	=1	YES = WAS FIRST AN INTEGER
4205	05 0093D	6930093F			BNE	*+2	
4206	05 0093E	23F003E8	A		MI,D1	1000	YES = CONVERT TO A SEQ #
4207	05 0093F	35F000C1	02		STW,D1	PARAMBUF	PUT VALUE IN PARAMBUF
4208	05 00940	22300001	A		LI,X1	1	SET X1=2ND SEQ #
4209	05 00941	224FFFFFF	A		LI,X2	=1	RESET X2 & D1
4210	05 00942	22F00000	A		LI,D1	0	
4211	05 00943	725C0176	02		LB,P1	TTYIMG,P2	GET NEXT INPUT CHAR
4212	05 00944	20600001	A		AI,P2	1	INCR CHAR PSN
4213	05 00945	1950001A	02		CLM,P1	DIGITS	IS CHAR A DIGIT
4214	05 00946	68900920			BIL	GP52A	YES = GO ACCUMULATE IT
4215	05 00947	21500048	A		CI,P1	1.1	IS CHAR A '1.1'
4216	05 00948	69300931			BNE	GP53A	NO = ERROR
4217	05 00949	22400003	A		LI,X2	3	YES = USER X3 TO CNT DIGITS AFTER 1.
4218	05 0094A	68000928			B	GP53	GO PROCESS 1.1
4219				*			
4220				*		DONE WITH SECOND SEQ # OF PAIR: FINISH UP	
4221				*			
4222	05 0094B	214FFFFFF	A	GP63	CI,X2	=1	WAS SECOND AN INTEGER
4223	05 0094C	6930094E			BNE	*+2	
4224	05 0094D	23F003E8	A		MI,D1	1000	YES = CONVERT TO A SEQ #
4225	05 0094E	35F000C2	02		STW,D1	PARAMBUF+1	PUT VALUE IN PARAMBUF
4226	05 0094F	22500003	A		LI,P1	SEQ2	SET TYPE=1SEQ2!
4227	05 00950	22400002	A		LI,X2	2	SET PARAMBUF SIZE = 2
4228	05 00951	206FFFFFF	A		AI,P2	=1	SET CHAR PSN TO RESCAN LAST CHAR
4229	05 00952	31F000C1	02		CW,D1	PARAMBUF	IS SEQ # 2 >= SEQ # 1
4230	05 00953	681008BD			BGE	GETNEXT*FINISH	YES = GO FINISH UP
4231	05 00954	6A700D99			BAL,LNK	TYPEPERR	NO = TYPE: 1=PN;SEQ2<SEQ1!

```

4232 05 00955 00000254 02
4233 05 00956 68000004
4234
4235
4236
4237 05 00957 22500002 A
4238 05 00958 214FFFFFF A
4239 05 00959 6930095B
4240 05 0095A 22500004 A
4241 05 0095B 35F000C1 02
4242 05 0095C 22400001 A
4243 05 0095D 206FFFFFF A
4244 05 0095E 6800088D
4245
4246
4247
4248      05 0095F
4249 05 0095F 00 A
4250 05 0095F 1 00 A
4251 05 0095F 2 15 A
4252 05 0095F 3
4253 05 0095F 3 00 A
4254 05 00960 65 A
4255 05 00960 1 5E A
4256      00000C05
4257
4258
4259
4260
4261      05 00961
4262 05 00961 00 A
4263 05 00961 1 00 A
4264 05 00961 2 00 A
4265 05 00961 3
4266 05 00961 3 00 A
4267 05 00962 07 A
4268 05 00962 1 08 A
    
```

```

DATA ERRP11
B MASTERPARSER GO TO PARSE

*
* NB '!' FOLLOWS FIRST; FINISH UP
*
GP66 LI,P1 SEG SET TYPE='INTG' OR 'SEQ' AS APPR
CI,X2 =1
BNE $+2
LI,P1 INTG
STW,D1 PARAMBUF PUT VALUE IN PARAMBUF
LI,X2 1 SET PARAMBUF SIZE = 1
AI,P2 =1 SET CHAR PSN TO RESCAN LAST CHAR
B GETNEXT$FINISH GO FINISH UP

*
* TABLE OF LEGAL 'GETNEXTPARAM' MATCH CHARS
*
GPTBL EQU $
DATA,1 0 (FILLER)
DATA,1 CR 0: C/R HIT
DATA,1 LF 0: LINE FEED HIT.
DB1 MODE=2
DATA,1 FF 0: FORM FEED HIT
DATA,1 CM 7: COMMA
DATA,1 SC 8: SEMI-COLON
GPTBLSZ EQU BA($)-BA(GPTBL)-1
BOUND 4

*
* TABLE OF TYPES CORRESPONDING TO LEGAL CHARS
*
GPTYTBL EQU $
DATA,1 0 (FILLER)
DATA,1 0 0: C/R HIT
DATA,1 0 0: LINE FEED HIT.
DB1 MODE=2
DATA,1 0 0: FORM FEED HIT
DATA,1 7 7: COMMA
DATA,1 8 8: SEMI-COLON
    
```

H01 20144 SEP 08, '75
4269

BUND 4

217

PAGE

```
*****
* CREATE NEW ENTRY IN CDT *
* P1 = NUMBER OF COMMAND TO ADD *
* WORD AFTER BAL = NUMBER OF PARAMETERS *
*****
```

4270						
4271						
4272						
4273						
4274						
4275						
4276						
4277						
4278		05 00963		NEWCDTENTRY	EGU \$	
4279	05 00963	02200020	A	PUSH	(P1,P2)	SAVE REGS
	05 00964	0B50019C	02			
4280	05 00965	2B500008	A	SLS,P1	8	BUILD CONTROL WORD OF ENTRY:
4281	05 00966	49500048	02	BR,P1	CDT	BYTE 0: LENGTH OF ENTRY (=0)
4282	05 00967	2B500008	A	SLS,P1	8	BYTE 1: COMMAND #
4283	05 00968	495E0000	A	BR,P1	0,LNK	BYTE 2: # OF ENTRY IN CDT
4284	05 00969	B55000AC	02	STW,P1	*CDTADR	BYTE 3: # OF PARAMETERS
4285	05 0096A	326E0000	A	LW,P2	0,LNK	COMPUTE LENGTH OF ENTRY =
4286	05 0096B	20600003	A	AI,P2	3	(# OF PARAMETERS)/2+1
4287	05 0096C	2B60007F	A	SLS,P2	=1	
4288	05 0096D	F56000AC	02	STB,P2	*CDTADR	PUT THIS IN BYTE 0
4289	05 0096E	4B500005	02	AND,P1	XFF00	BUILD IEND OF CDT: MARKER USING
4290	05 0096F	20500100	A	AI,P1	X'0100'	NUMBER OF NEXT CMND IN CDT
4291	05 00970	B55C00AC	02	STW,P1	*CDTADR,P2	PUT IT AFTER PARAM CONTROL HW'S
4292	05 00971	22500000	A	LI,P1	0	
4293	05 00972	6B000094	A	B	\$+2	
4294	05 00973	B55C00AC	02	STW,P1	*CDTADR,P2	SET ALL PARAM CONTROL HW'S TO ZERO
4295	05 00974	64600093	A	BCR,P2	\$-1	
4296	05 00975	02200020	A	PULL	(P1,P2)	RESTORE REGS
	05 00976	0A50019C	02			
4297	05 00977	6B0E0001	A	B	1,LNK	EXIT

PAGE

```
*****
* ADJUST ALL FLAG *
* P1 = COLUMN NUMBER TO RESUME MATCHING AT *
*****
*
*
```

```
4305          05 00978
4306 05 00978 33000020 02
4307 05 00979 691E0000 A
4308 05 0097A 39500020 02
4309 05 0097B 680E0000 A
```

```
ADJUSTALLFLAG EQU $
                MTW,0  ALLFLAG      IS ALLFLAG ON
                BLZ    0,LNK        NO = EXIT
                STW,P1 ALLFLAG      YES = SET IT TO COL. TO RESUME MATCH
                B      0,LNK        EXIT
```

PAGE

```

4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344

```

* ANALYZE COMPOSITION OF FIELD TO RIGHT *
* P1 = COLUMN AT WHICH TO START ANALYZE *
* R1 (BP OFF) = NUMBER OF NON-BLANKS TO 1ST BLANK *
* R1 (BP ON) = NUMBER OF CHARS TO LAST NON-BLANK ON CARD *
* R2 (BP OFF) = NUMBER OF BLANKS (-1) FROM 1ST BLANK TO NEXT *
* NON-BLANK *
* R2 (BP ON) = NUMBER OF TRAILING BLANKS ON CARD *
* CC1=1 IF INITIAL P1>END OF BUFFER, CC1=0 OTHERWISE *

05 00974 ANLZRRIGHT EQU \$
05 0097C 02200020 A PUSH (P1,P2) SAVE REGS
05 0097D 0P50019C 02
05 0097E 2150008C A CI,P1 MAXCLMN IS START OF FIELD PAST END OF BUFFER
05 0097F 69100986 BL AR10 NO = GO ON
05 00980 22B00000 A LI,R1 0 SET R1=R2=0
05 00981 22C00000 A LI,R2 0
05 00982 220FFFFE A PURGE (P1,P2) YES = CLEAR STACK
05 00983 1900019C 02
05 00984 02200080 A LCI 8 SET CC1=1
05 00985 680E0000 A B 0,LNK EXIT

*
* TEST BP FLAG, IF OFF CALC R1=NUMBER OF NON-BLANKS *
*

05 00986 22B00000 A AR10 LI,R1 0 SET R1=0
05 00987 315000B1 02 CW,P1 E0DCLMN IS START OF FIELD PAST LAST NON-BLNK
05 00988 69200993 BG AR12
05 00989 33000023 02 MTW,0 BPFLAG NO = IS BLANK PRES. ON
05 0098A 6930099E BNEZ AR20
05 0098B 22C00000 A LI,R2 0 NO = SET R2=0
05 0098C 22600040 A LI,P2 , ,
05 0098D 716A0024 02 AR10A CB,P2 CARDIMG,P1 IS CHAR AT P1=BLANK
05 0098E 68300997 BE AR15+1 YES = GO COUNT BLANKS
05 0098F 20B00001 A AI,R1 1 NO = INCR R1 & P1

4345	05 00990	20500001 A	AI,P1	1	
4346	05 00991	315000B1 02	CW,P1	E0DCLMN	IS P1 PAST LAST NON-BLANK
4347	05 00992	6820098D	BLE	AR10A	N0 = CONTINUE SCAN
4348			*		
4349			*	NOW PAST LAST NON-BLANK, CALC R2=NUMBER OF BLANKS TO END	
4350			*		
4351	05 00993	22C0008C A	AR12	LI,R2	MAXCLMN
4352	05 00994	38C00005 A		SW,R2	P1
4353	05 00995	6800099A		B	AR18
4354			*		
4355			*	AT END OF NON-BLANKS, COUNT BLANK FIELD	
4356			*		
4357	05 00996	20C00001 A	AR15	AI,R2	1
4358	05 00997	20500001 A		AI,P1	1
4359	05 00998	716A0024 02		CB,P2	CARDIMG,P1
4360	05 00999	68300996		BE	AR15
4361			*		
4362			*	EXIT WITH CC1=0	
4363			*		
4364	05 0099A	02200020 A	AR18	PULL	(P1,P2)
	05 0099B	0A50019C 02			
4365	05 0099C	02200000 A		LCI	0
4366	05 0099D	680E0000 A		B	0,LNK
4367			*		
4368			*	BP FLAG 0N, CALC R1 & R2	
4369			*		
4370	05 0099E	32B000B1 02	AR20	LW,R1	E0DCLMN
4371	05 0099F	38B00005 A		SW,R1	P1
4372	05 009A0	20B00001 A		AI,R1	1
4373	05 009A1	22C0008B A		LI,R2	MAXCLMN=1
4374	05 009A2	38C000B1 02		SW,R2	E0DCLMN
4375	05 009A3	6800099A		B	AR18

CALC R1=DISTANCE FROM P1 TO LAST NON-BLANK

CALC R2=NUMBER OF TRAILING BLANKS

GO EXIT

PAGE

```
*****
* EVALUATE FIRST PARAMETERS FOR INTRALINE COMMANDS *
* CDTADR = ADDR OF CURRENT COMMAND IN CDT *
* RESULTS: P1 = COLUMN COMPUTED FROM PARAMETERS *
*          P2 = WIDTH OF FIELD AT THIS COLUMN *
*          X1 = POSITION OF NEXT CDT CONTROL BYTE *
*          CC1=1 IF NO COLUMN FOUND; CC1=0 OTHERWISE *
*****
```

```
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387      05 009A4
          05 009A4 0940019C 02
          05 009A5 02200020 A
          05 009A6 0B70019C 02
4388      05 009A7 22800000 A
4389      05 009A8 32500020 02
4390      05 009A9 681009CC
4391      05 009AA 325000B7 02
4392      05 009AB 22300003 A
4393      05 009AC F28600AC 02
4394      05 009AD 21800002 A
4395      05 009AE 692009C1
4396      05 009AF 22300004 A
4397      05 009B0 22800001 A
4398      05 009B1 F24600AC 02
4399      05 009B2 22300005 A
4400      05 009B3 21400005 A
4401      05 009B4 683009CD
4402      05 009B5 F24600AC 02
4403      05 009B6 B25800AC 02
4404      05 009B7 205FFFFFF A
4405      05 009B8 315000B7 02
4406      05 009B9 691009EC
4407      05 009BA 315000BB 02
4408      05 009BB 681009E9
4409      05 009BC 22600001 A
4410      05 009BD 22300006 A
```

```
FINDCOLUMN      EQU *
                  PUSH      X2,(LNK,T1)      SAVE REGS

                  LI,T1      0              SET T1=ALL OCCURRENCES
                  LW,P1      ALLFLAG        P1=CBL. TO START MATCHING AT
                  BGEZ       FC15          IS SYSTEM IN IALLI MODE
                  LW,P1      FRSTCLMN      NO = SET P1=CBL. TO START AT
                  LI,X1      3              GET NUMBER OF PARAMS IN CDT
                  LB,T1      *CDTADR,X1
                  CI,T1      2              ARE THERE > 2 PARAMS
                  BG         FC10
                  LI,X1      4
                  LI,T1      1
                  LB,X2      *CDTADR,X1    NO = GET PARAM1 TYPE
                  LI,X1      5
                  CI,X2      STRG          IS IT A STRING
                  BE         FC15A        YES = FORM IS: /ST/ X =
                  LB,X2      *CDTADR,X1    NO = FORM IS: C X =
                  LW,P1      *CDTADR,X2    GET CBL. # FROM CDT
                  AI,P1      =1           ADJUST TO INTERNAL CBL. #
                  CW,P1      FRSTCLMN     IS IT BELOW CBL. TO START AT
                  BL         FC45        YES = ERROR
                  CW,P1      LASTCLMN     IS TO BEYOND CBL. TO STOP AT
                  BGE        FC40        YES = ERROR
                  LI,P2      1           NO = SET FIELD WIDTH = 1
                  LI,X1      6           SET NEXT CDT CTRL BYTE = 6
```



```

4448 *
4449 * EXIT WITH CC1=0
4450 *
4451 05 009DD 02200020 A FC20 PULL X2,(LNK,T1) RESTORE REGS
      05 009DE 0A70019C 02
      05 009DF 0B40019C 02
4452 05 009E0 02200000 A LCI 0
4453 05 009E1 680E0000 A B 0,LNK EXIT WITH CC1=0
4454 *
4455 * NO MATCH FOUND: IF IN 'ALL' MODE, EXIT 'ALL' MODE; OTHERWISE, ERROR
4456 *
4457 05 009E2 228FFFFFF A FC30 LI,T1 =1 TURN OFF ALL MODE.
4458 05 009E3 35800020 02 STW,T1 ALLFLAG
4459 *
4460 * EXIT WITH CC1=1
4461 *
4462 05 009E4 02200020 A FC35 PULL X2,(LNK,T1) RESTORE REGS
      05 009E5 0A70019C 02
      05 009E6 0B40019C 02
4463 05 009E7 02200080 A LCI 8
4464 05 009E8 680E0000 A B 0,LNK EXIT WITH CC1=1
4465 *
4466 * ERROR: COLUMN NUMBER BEYOND COLUMN TO STOP AT
4467 *
4468 05 009E9 6A700D8F FC40 BAL,LNK TYPECERR TYPE: !=CN:COL>LIMIT!
4469 05 009EA 000001BB 02 DATA ERRC6
4470 05 009EB 680009E4 B FC35 GO TO EXIT
4471 *
4472 * ERROR: COLUMN NUMBER BELOW COLUMN TO START AT
4473 *
4474 05 009EC 6A700D8F FC45 BAL,LNK TYPECERR TYPE: !=CN:COL<LIMIT!
4475 05 009ED 000001CC 02 DATA ERRC10
4476 05 009EE 680009E4 B FC35 GO TO EXIT

```

PAGE

```

4477
4478 *****
4479 * FIND MATCHING STRING ON CARD *
4480 * P1 = COLUMN AT WHICH TO START SEARCH *
4481 * P2 = ADDR OF TEXTC=STRING TO MATCH *
4482 * R1 = COLUMN AT WHICH MATCH OCCURRED *
4483 * CC1=0 IF MATCH FOUND, CC1=1 IF NO MATCH *
4484 *****
4485 *
4486 *
4487 05 009EF FINDMATCH EQU $
4488 05 009EF 02200070 A PUSH (X1,T2) SAVE REGS
05 009F0 0B30019C 02
4489 05 009F1 35600175 02 STW,P2 TEXTCADR SAVE ADDR OF TEXTC=STRING
4490 05 009F2 32B000BB 02 LW,R1 LASTCLMN CALC: STOPCLMN=LAST COL. # AT WHICH
4491 05 009F3 F2600175 02 LB,P2 *TEXTCADR MATCH CAN TAKE PLACE
4492 05 009F4 38B00006 A SW,R1 P2
4493 05 009F5 35B00168 02 STW,R1 STOPCLMN
4494 05 009F6 31500168 02 CW,P1 STOPCLMN IS INITIAL COL.=STOPCLMN
4495 05 009F7 682009FB BLE FM10
4496 05 009F8 220FFFF9 A PURGE (X1,T2) YES = CLEAR STACK
05 009F9 1300019C 02
4497 05 009FA 68000A04 B FM15 GO EXIT WITH CC1=1
4498 *
4499 * GET 1ST CHAR OF TEXTC=STRING AND SEARCH FOR IT IN CARD
4500 *
4501 05 009FB 22300001 A FM10 LI,X1 1 SET T1=1ST CHAR OF TEXTC=STRING
4502 05 009FC F2860175 02 LB,T1 *TEXTCADR,X1
4503 05 009FD 718A0024 02 FM10A CB,T1 CARDIMG,P1 DOES 1ST CHAR MATCH CHAR ON CARD
4504 05 009FE 68300A06 BE FM20 YES = GO COMPARE REST
4505 05 009FF 20500001 A FM10B AI,P1 1 NO = INCR TO NEXT COLUMN
4506 05 00A00 31500168 02 CW,P1 STOPCLMN IS NEW COLUMN>STOPCLMN
4507 05 00A01 682009FD BLE FM10A NO = GO COMPARE MORE
4508 05 00A02 02200070 A PULL (X1,T2) YES = RESTORE REGS
05 00A03 0A30019C 02
4509 *
4510 * EXIT WITH NO MATCH FOUND (CC1=1)

```

```

4511
4512 05 00A04 02200080 A FM15 LCI 8
4513 05 00A05 680E0000 A B 0,LNK EXIT WITH CC1=1
4514
4515 *
* 1ST CHAR MATCH FOUND, NOW COMPARE CARD WITH REST OF TEXTC=STRING
4516 *
4517 05 00A06 22300001 A FM20 LI,X1 1 SET X1=POSITION IN TEXTC=STRING
4518 05 00A07 32400005 A LW,X2 P1 X2=CBL, # ON CARD
4519 05 00A08 F2600175 02 LB,P2 *TEXTCADR P2=# OF CHARS TO COMPARE
4520 05 00A09 206FFFFFF A AI,P2 =1
4521 05 00A0A 68300A11 BEZ FM30 IF STRING IS 1 CHAR LONG = EXIT
4522 05 00A0B 20300001 A FM20A AI,X1 1 INCR X1 & X2
4523 05 00A0C 20400001 A AI,X2 1
4524 05 00A0D F2960175 02 LB,T2 *TEXTCADR,X1 DO 2 CHARS MATCH
4525 05 00A0E 71980024 02 CB,T2 CARDIMG,X2
4526 05 00A0F 693009FF BNE FM10B NO = GO START 1ST CHAR SEARCH AGAIN
4527 05 00A10 64600A0B BDR,P2 FM20A YES = LOOP UNTIL CORRECT # MATCH
4528
4529 *
* EXIT WITH MATCH FOUND (CC1=0)
4530 *
4531 05 00A11 32800005 A FM30 LW,R1 P1 MATCH FOUND = SET R1=CBL, # OF MATCH
4532 05 00A12 217005AF CI,LNK FNDTYP+1 IF I CAME FROM A FT,FD OR FS TYPE OF
4533 05 00A13 68300A15 BE *+2 COMMAND DON'T INCREMENT CHG:STG:CNT.
4534 05 00A14 3310019F 02 MTW,1 CHG:STG:CNT COUNT THE NO. OF HITS.
4535 05 00A15 02200070 A PULL (X1,T2) RESTORE REGS
4536 05 00A16 0A30019C 02
4537 05 00A17 02200000 A LCI 0
4537 05 00A18 680E0000 A B 0,LNK EXIT WITH CC1=0
    
```

PAGE

```

4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572

```

05	00A19	02200050	A	MOVESTRING	EGU	*	
05	00A1A	0B30019C	02		(X1, LNK)		SAVE REGS
05	00A1B	25600002	A	SLS, P2	2		CONVERT P2 TO A BYTE ADDR
05	00A1C	723C0000	A	LB, X1	0, P2		SET X1=# OF CHARS TO MOVE
05	00A1D	2150008C	A	CI, P1	MAXCLMN		IS STARTING COL. BEYOND END OF CARD
05	00A1E	68100A2B		BGE	MS20A-1		YES = GO CHECK
* MOVE CHAR FROM TEXTC=STRING TO CARD							
05	00A1F	20600001	A	MS5	AI, P2	1	INCR TO NEXT TEXTC=STRING CHAR
05	00A20	724C0000	A		LB, X2	0, P2	MOVE CHAR TO CARD
05	00A21	754A0024	02		STB, X2	CARDIMG, P1	
05	00A22	20500001	A		AI, P1	1	INCR COLUMN
05	00A23	2150008C	A		CI, P1	MAXCLMN	
05	00A24	68100A29			BGE	MS20	HAS END OF BUFFER BEEN PASSED
05	00A25	64300A1F			BDR, X1	MS5	NO = LOOP UNTIL ALL CHARS MOVED
* EXIT							
05	00A26	02200050	A	MS10	PULL	(X1, LNK)	RESTORE REGS
05	00A27	0A30019C	02				
05	00A28	680E0000	A		B	0, LNK	EXIT
* AT END OF BUFFER: IF MORE NON-BLANKS TO MOVE, TYPE ERROR MESSAGE							
05	00A29	203FFFFFF	A	MS20	AI, X1	=1	END OF BUFFER: ARE THERE MORE CHARS
05	00A2A	68300A26			BEZ	MS10	TO MOVE
05	00A2B	22400040	A		LI, X2	' '	

HC1 20:44 SEP 08, '75

4573	05	00A2C	20600001	A	MS20A
4574	05	00A2D	714C0000	A	
4575	05	00A2E	69300A31		
4576	05	00A2F	64300A2C		
4577	05	00A30	68000A26		
4578	05	00A31	6A700D8F		MS20B
4579	05	00A32	000001A5	02	
4580	05	00A33	68000A26		

A1,P2	1
CB,X2	0,P2
BNE	MS20B
BDR,X1	MS20A
B	MS10
BAL,LNK	TYPECERR
DATA	ERRC1
B	MS10

YES = IS NEXT CHAR OF TEXTC-STRING
 A BLANK
 NO = TYPE ERROR MSG
 YES = LOOP UNTIL ALL CHARS CHECKED
 ALL BLANKS = GO EXIT
 TYPE: !=-CN:OVERFLOW!
 GO EXIT

PAGE

```

4581
4582 *****
4583 * PROCESS COLUMN NUMBER PAIR *
4584 * X1 = LOC OF NEXT PARAMETER CONTROL BYTE IN CDT *
4585 *****
4586 *
4587 *
4588          05 00A34      PROCESSCOL#PAIR      EQU $
4589 05 00A34      02200040 A      PUSH          (X1,P2)      SAVE REGS
          05 00A35      0B30019C 02
4590 05 00A36      22500000 A      LI,P1          0      SET P1=DFLT STARTING COL #
4591 05 00A37      2260008C A      LI,P2          MAXCLMN      P2=DFLT STOPPING COL #
4592 05 00A38      F24600AC 02      LB,X2          *CDTADR,X1      GET NEXT PARAM TYPE
4593 05 00A39      68300A3F      BEZ           PP10      IS PARAM PRESENT
4594 05 00A3A      20300001 A      AI,X1          1
4595 05 00A3B      F24600AC 02      LB,X2          *CDTADR,X1      YES = SET P1=STARTING COL #
4596 05 00A3C      B25800AC 02      LW,P1          *CDTADR,X2
4597 05 00A3D      205FFFFFF A      AI,P1          =1      ADJUST TO INTERNAL COL #
4598 05 00A3E      203FFFFFF A      AI,X1          =1
4599
4600 *
4601 * PROCESS SECOND COLUMN NUMBER PARAMETER
4602 *
4602 05 00A3F      20300002 A      PP10      AI,X1          2
4603 05 00A40      F24600AC 02      LB,X2          *CDTADR,X1      GET NEX PARAM TYPE
4604 05 00A41      68300A45      BEZ           PP20      IS PARAM PRESENT
4605 05 00A42      20300001 A      AI,X1          1
4606 05 00A43      F24600AC 02      LB,X2          *CDTADR,X1      YES = SET P2=STOPPING COL # + 1
4607 05 00A44      B26800AC 02      LW,P2          *CDTADR,X2
4608
4609 *
4610 * FINISH INITIALIZATION AND EXIT
4611 *
4611 05 00A45      355000B7 02      PP20      STW,P1         FRSTCLMN      SET STARTING AND STOPPING COL #IS
4612 05 00A46      356000BB 02      STW,P2         LASTCLMN
4613 05 00A47      31500006 A      CW,P1          P2
4614 05 00A48      68100A50      BGE           PP25
4615 05 00A49      21500000 A      CI,P1          0
4616 05 00A4A      69100A50      BL            PP25
    
```

HC1 20:44 SEP 08, '75

4617 05 00A4B 2160008C A
 4618 05 00A4C 69200A50
 4619 05 00A4D 02200040 A
 05 00A4E 0A30019C 02
 4620 05 00A4F 680E0000 A
 4621
 4622 05 00A50 6A700DC4
 4623 05 00A51 000001D0 02
 4624 05 00A52 22700000 A
 4625 05 00A53 3b7000E9 02
 4626 05 00A54 3b700167 02
 4627 05 00A55 68000004

*
PP25

CI,P2 MAXCLMN
 BG PP25
 PULL (X1,P2)

 B O,LNK
 BAL,LNK TYPEMSG
 DATA ERRCL1
 LI,LNK 0
 STW,LNK SETFLAG
 STW,LNK STEPFLAG
 B MASTERPARSER

RESTORE REGS

EXIT

TYPE: 'BAD COL. NO. PAIR'

PAGE

```

4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662

```

05	00A56	390001A0	02	SETEBD	FGU	\$	
05	00A57	683E0000	A		MTW,0	RP\$FLAG	CHECK RECORD PRESERVATION
05	00A58	02200020	A		BEZ	0,LNK	10N1, DONT CHANGE RECSIZE
05	00A59	0B30019C	02		PUSH	(X1,X2)	SAVE REGS
05	00A5A	22300022	A		LI,X1	MAXCLMN/4-1	
05	00A5B	3240000A	02		LW,X2	4BLNKS	
05	00A5C	31460024	02		CW,X2	CARDIMG,X1	MAKE GROSS COMPARISON FOR ALL
05	00A5D	69300A67			BNE	SRS10	BLANK WORDS.
05	00A5E	64300A5C			BDR,X1	\$-2	
05	00A5F	22300003	A		LI,X1	3	CHECK FIRST WORD BY BYTE.
05	00A60	71460024	02	SRS5	CB,X2	CARDIMG,X1	ITERATE THROUGH BYTES OF
05	00A61	69300A6A			BNE	SRS15	TARGET WORD.
05	00A62	64300A60			BDR,X1	\$-2	
05	00A63	71400024	02		CB,X2	CARDIMG	CHECK FIRST BYTE OF FIRST WORD,
05	00A64	69300A6A			BNE	SRS15	FOR BLANK.
05	00A65	223FFFFFF	A		LI,X1	=1	IF BLANK, RECORD SIZE =0.
05	00A66	68000A6A			B	SRS15	
05	00A67	2B300002	A	SRS10	SLS,X1	2	REVERT TO BYTE INDEXING, TO GET
05	00A68	2U300003	A		AI,X1	3	BYTE WITHIN WORD.
05	00A69	68000A60			B	SRS5	
05	00A6A	3B3000B1	02	SRS15	STW,X1	E0DCLMN	SAVE ENDING COLUMN (BYTE INDEX)
05	00A6B	2U300001	A		AI,X1	1	
05	00A6C	3B3000E7	02		STW,X1	RECSIZE	AND RECORD SIZE (TRUE BYTE COUNT)
05	00A6D	02200020	A		PULL	(X1,X2)	
05	00A6E	0A30019C	02				
05	00A6F	680E0000	A		B	0,LNK	EXIT

H01 20:44 SEP 08, 175

4699 05 00A84 20500001 A
 4700 05 00A85 20600001 A
 4701 05 00A86 64B00A82
 4702
 4703
 4704
 4705 05 00A87 22800040 A
 4706 05 00A88 758C0024 02
 4707 05 00A89 20600001 A
 4708 05 00A8A 64A00A88
 4709 05 00A8B 02200080 A
 05 00A8C 0A50019C 02
 4710 05 00A8D 680E0000 A
 4711
 4712
 4713
 4714
 4715 05 00A8E 30A00006 A
 4716 05 00A8F 38B00006 A
 4717 05 00A90 30A0000B A
 4718 05 00A91 22600000 A
 4719 05 00A92 68000A87
 4720
 4721
 4722
 4723 05 00A93 38500006 A
 4724 05 00A94 2150008C A
 4725 05 00A95 69100A76
 4726 05 00A96 32600005 A
 4727 05 00A97 3860000A A
 4728 05 00A98 68000A87

AI,P1 1
 AI,P2 1
 BDR,R1 SL5A
 *
 * BLANK OUT CLEARED CHARS ON RIGHT
 *
 SL10 LI,T1 ' ' BLANK OUT
 STB,T1 CARDING,P2
 AI,P2 1
 BDR,P3 *-2
 PULL (P1,R2) RESTORE REGS
 B O,LNK EXIT
 *
 * SHIFT PUSHES EVERYTHING, INCLUDING FIELD AT P1, OFF CARD, SO BLANK
 * OUT AND EXIT
 *
 SL20 AW,P3 P2 CALC P3=# OF COLUMNS WIPED OUT
 SW,R1 P2
 AW,P3 R1
 LI,P2 0 SET 'TB' FOR BLANKING=0
 B SL10 GO BLANK OUT
 *
 * FIELD TO SHIFT IS BEYOND END OF CARD: SET UP TO SHIFT IN BLANKS
 *
 SL30 SW,P1 P2 RESTORE R1
 CI,P1 MAXCLMN IS FIELD BEYOND END OF CARD
 BL SL3 NO = CONTINUE NORMALLY
 LW,P2 P1 SET P2=CBL. AT WHICH TO START
 SW,P2 P3 BLANKING OUT
 B SL10 GO BLANK OUT

PAGE

```
*****
* SHIFT STRING RIGHT *
* P1 = COLUMN AT WHICH TO START SHIFT *
* P2 = WIDTH OF FIELD STARTING AT THIS COLUMN *
* P3 = NUMBER TO SHIFT RIGHT *
*****
```

```
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738      05 00A99
4739      05 00A99 2150008C A
4740      05 00A9A 681E0000 A
4741      05 00A9B 022000A0 A
          05 00A9C 0530019C 02
4742      05 00A9D 228000C0 A
4743      05 00A9E 358000B5 02
4744      05 00A9F 35800022 02
4745      05 00AA0 30500006 A
4746      05 00AA1 2150008C A
4747      05 00AA2 683000E03
4748
4749
4750
4751
4752      05 00AA3 6A700097C
4753      05 00AA4 698000ADD
4754      05 00AA5 66C00022 02
4755      05 00AA6 55B0000C A
4756      05 00AA7 30B00005 A
4757      05 00AA8 20BFFFFFF A
4758      05 00AA9 331000B5 02
4759      05 00AAA 31A00022 02
4760      05 00AAB 682000AB3
4761      05 00AAC 02200020 A
          05 00AAD 05B0019C 02
4762      05 00AAE 3250000B A
4763      05 00AAF 45C00006 02
```

```
*
* SHIFTRIGHT      EGU $
* CI,P1      MAXCLMN      IS FIELD BEYOND END OF CARD
* BGE      0,LNK      YES = EXIT
* PUSH      (X1,R2)      SAVE REGS
*
* LI,T1      0      SET CNTS=0
* STW,T1      FIELDCNT
* STW,T1      BLANKCNT
* AW,P1      P2      START ANLZ AFTER ORIG FIELD
* CI,P1      MAXCLMN      DOES FIELD ABUTT END OF CARD
* BE      SR70      YES = GO PROCESS
*
* BUILD 2-WD DATA BLOCK FOR EACH FIELD TO BE COMPRESSED AND PUSH
* ON STACK
*
* SR5      BAL,LNK      ANLZRIGHT      ANLZ FIELD AT P1
          BCS,8      SR50      00PS = END OF CARD
          AWM,R2      BLANKCNT      CNT BLNKS TO COMPRESS
          STH,R1      R2
          AW,R1      P1      BUILD: R1=COLUMN AT END OF NON-BLNKS
          AI,R1      =1      R2=(# OF NON-BLNKS,# TO SHFT)
          MTW,1      FIELDCNT      CNT FIELDS COMPRESSED
          CW,P3      BLANKCNT      ARE ENOUGH BLNKS COMPRESSED YET
          BLE      SR8      YES
          PUSH      (R1,R2)      NO = SAVE FIELD DATA BLOCK
*
* LW,P1      R1      INCR P1 TO NEXT FIELD
* AND,R2      XFFFF
```

4764	05	00AB0	3050000C	A		AW,P1	R2	
4765	05	00AB1	20500002	A		AI,P1	2	
4766	05	00AB2	68000AA3			B	SR5	ANLZ NEXT FIELD
4767					*			
4768					*			INITIALIZE TO DB ACTUAL SHIFTS (I.E., COMPRESSING)
4769					*			
4770	05	00AB3	38A00022	02	SR8	SW,P3	BLANKCNT	ADJUST (# TO SHIFT) SPEC IN R2 TO
4771	05	00AB4	30C0000A	A		AW,R2	P3	PRESERVE EXCESS BLNKS IN LAST FIELD
4772	05	00AB5	3230000B	A	SR8A	LW,X1	R1	AVOID: PUSH R1,R2
4773	05	00AB6	3240000C	A		LW,X2	R2	PULL R1,R2
4774	05	00AB7	22800000	A		LI,T1	0	
4775	05	00AB8	35800022	02		STW,T1	BLANKCNT	CLEAR BLNK CNT
4776	05	00AB9	33F000B5	02		MTW,-1	FIELD CNT	DECR FIELD CNT
4777	05	00ABA	69200AC1			BGZ	SR12	>0 = 1 OR MORE FIELDS ON STK
4778	05	00ABB	68300ABE			BEZ	SR10	=0 = AT 1ST FIELD (STK EMPTY)
4779	05	00ABC	52800004	A		LH,T1	X2	<0 = SHIFT WIPES ALL BUT ORIG FIELD
4780	05	00ABD	68000AC3			B	SR12A	AT P1
4781					*			
4782					*			READY TO SHIFT 1ST FIELD, BUT FIRST ADD ON ORIG FIELD AT P1
4783					*			
4784	05	00ABE	50600004	A	SR10	AH,P2	X2	ADD LENGTH OF ORIG FIELD TO (# OF
4785	05	00ABF	206FFFFF	A		AI,P2	=1	NON-BLNKS) SPEC IN R2
4786	05	00AC0	59600004	A		STH,P2	X2	
4787					*			
4788					*			SET UP PARAMETERS FOR CURRENT SHIFT
4789					*			
4790	05	00AC1	52800004	A	SR12	LH,T1	X2	SET T1=# OF CHARS IN FIELD TO SHIFT
4791	05	00AC2	20800001	A		AI,T1	1	(INCLUDING PRECEDING BLANK)
4792	05	00AC3	48400006	02	SR12A	AND,X2	XFFFF	KEEP CUMULATIVE CNT OF BLNKS
4793	05	00AC4	66400022	02		AWM,X2	BLANKCNT	COMPRESSED BUT
4794	05	00AC5	32400003	A		LW,X2	X1	CALC: X1=END OF 'FROM' FIELD
4795	05	00AC6	30400022	02		AW,X2	BLANKCNT	X2=END OF 'TO' FIELD
4796	05	00AC7	21800000	A		CI,T1	0	IS # OF CHARS TO SHIFT = 0
4797	05	00AC8	68300ACE			BE	SR15A	YES = SKIP SHIFT
4798					*			
4799					*			DB CURRENT SHIFT, THEN CHECK NUMBER LEFT TO DB
4800					*			

4801	05	00AC9	72960024	02	SR15	LB,T2	CARDIMG,X1	COMPRESS FIELDS
4802	05	00ACA	75980024	02		STB,T2	CARDIMG,X2	
4803	05	00ACB	203FFFFFF	A		AI,X1	=1	
4804	05	00ACC	204FFFFFF	A		AI,X2	=1	
4805	05	00ACD	64800AC9			BDR,T1	SR15	
4806	05	00ACE	33F000B5	02	SR15A	MTW,-1	FIELD CNT	DECR FIELD CNT
4807	05	00ACF	69100AD5			BLZ	SR20	<0 = ALL SHIFTS DONE
4808	05	00AD0	02200020	A		PULL	(X1,X2)	>=0 = GET NEXT FIELD DATA BLOCK
	05	00AD1	0A30019C	02				
4809	05	00AD2	330000B5	02		MTW,0	FIELD CNT	TEST FIELD CNT
4810	05	00AD3	69200AC1			BGZ	SR12	>0 = 1 OR MORE FIELDS LEFT
4811	05	00AD4	68000ABE			B	SR10	=0 = AT 1ST FIELD
4812					*			
4813					*	ALL SHIFTS DONE, SO BLANK BUT CLEARED CHARS ON LEFT		
4814					*			
4815	05	00AD5	32800022	02	SR20	LW,T1	BLANKCNT	
4816	05	00AD6	22900040	A	SR20A	LI,T2	' '	BLANK BUT
4817	05	00AD7	75980024	02		STB,T2	CARDIMG,X2	
4818	05	00AD8	204FFFFFF	A		AI,X2	=1	
4819	05	00AD9	64800AD7			BDR,T1	=2	
4820	05	00ADA	022000A0	A		PULL	(X1,R2)	RESTORE REGS
	05	00ADB	0A30019C	02				
4821	05	00ADC	680E0000	A		B	0,LNK	EXIT
4822					*			
4823					*	END-OF-BUFFER HIT: NOT ENOUGH BLANKS TO ABSORB SHIFT		
4824					*			
4825	05	00ADD	6A700D8F		SR50	BAL,LNK	TYPECERR	TYPE: !=CN:OVERFLOW!
4826	05	00ADE	000001A5	02		DATA	ERRC1	
4827	05	00ADF	22800000	A		LI,T1	0	CLEAR BLNK CNT
4828	05	00AEO	46800022	02		XW,T1	BLANKCNT	SET T1=(# OF NON-BLANKS TO DESTROY)
4829	05	00AE1	3880000A	A		SW,T1	P3	
4830	05	00AE2	02200020	A		PULL	(R1,R2)	START ON LAST FIELD
	05	00AE3	0AB0019C	02				
4831					*			
4832					*	PULL FIELD DATA BLOCKS FROM STACK AND DESTROY NON-BLANKS UNTIL		
4833					*	ENOUGH ROOM FOUND, WHEN FOUND BUILD APPROPRIATE DATA BLOCK		
4834					*			

H01 20144 SEP 08, '75

237

4835 05 00AE4 5080000C A
 4836 05 00AE5 682000AE
 4837 05 00AE6 58B0000C A
 4838 05 00AE7 30B00008 A
 4839 05 00AE8 50C0000C A
 4840 05 00AE9 48C00006 02
 4841 05 00AEA 38C00008 A
 4842 05 00AEB 30C00022 02
 4843 05 00AEC 5880000C A
 4844 05 00AED 680000AB5

SR52 AH,T1 R2
 BLEZ SR55
 SR52A SH,R1 R2
 AW,R1 T1
 AH,R2 R2
 AND,R2 XFFFF
 SW,R2 T1
 AW,R2 BLANKCNT
 STH,T1 R2
 B SR8A

IS CURRENT FIELD (+OTHERS ALREADY
 WIPED OUT) LONG ENOUGH FOR OVERFLOW
 YES =BUILD R1 & R2 AS BEFORE;
 R1=COLUMN AT END OF NON-BLNKS
 NOT DESTROYED
 R2=(# OF NON-BLNKS NOT DESTROYED,
 # TO SHIFT)

GO SHIFT

*
 * NOT ENOUGH ROOM FOUND YET, GET NEXT FIELD DOWN AND DESTROY PART OF IT
 *

4848 05 00AEE 5290000C A
 4849 05 00AEF 3090000C A
 4850 05 00AF0 48900006 02
 4851 05 00AF1 68900022 02
 4852 05 00AF2 38F00085 02
 4853 05 00AF3 68300AF9
 4854 05 00AF4 02200020 A
 05 00AF5 0AB0019C 02
 4855 05 00AF6 20B00001 A
 4856 05 00AF7 20C10000 A
 4857 05 00AF8 680000AE4

SR55 LH,T2 R2
 AW,T2 R2
 AND,T2 XFFFF
 AWM,T2 BLANKCNT
 MTW,-1 FIELDCNT
 BEZ SR58
 PULL (R1,R2)
 AI,R1 1
 AI,R2 X'10000'
 B SR52

KEEP CUMULATIVE CNT OF CHARS
 DESTROYED
 DECR FIELD CNT
 =0 = AT 1ST FIELD
 >0 = GET NEXT FIELD DATA BLOCK
 INC. FOLLOWING BLANK IN FIELD

*
 * AT 1ST FIELD AND STILL NOT ENOUGH ROOM
 *

4861 05 00AF9 30800006 A
 4862 05 00AFA 682000AFF
 4863 05 00AFB 58B0000C A
 4864 05 00AFC 22C00000 A
 4865 05 00AFD 5960000C A
 4866 05 00AFE 680000AE6

SR58 AW,T1 P2
 BLEZ SR60
 SH,R1 R2
 LI,R2 0
 STH,P2 R2
 B SR52A

ADD IN ORIG FIELD AT P1 AND CHK ROOM
 ENOUGH FOUND = FIX R1 & R2 TO
 DESTROY PART OF ORIG FIELD AT P1

*
 * SHIFT PUSHES ALL FIELDS OFF CARD, SO BLANK OUT AND EXIT
 *

4870 05 00AFF 66600022 02

SR60 AWM,P2 BLANKCNT

SET UP TO BLANK FROM ORIG P1

4871 05 00800 32400005 A
 4872 05 00801 204FFFFE A
 4873 05 00802 68000AD5
 4874
 4875
 4876
 4877 05 00803 6A700D8F
 4878 05 00804 000001A5 02
 4879 05 00805 3860000A A
 4880 05 00806 68200B0D
 4881 05 00807 35A00022 02
 4882 05 00808 2230008B A
 4883 05 00809 3830000A A
 4884 05 0080A 2240008B A
 4885 05 0080B 32800006 A
 4886 05 0080C 68000AC9
 4887
 4888
 4889
 4890 05 0080D 3060000A A
 4891 05 0080E 32800006 A
 4892 05 0080F 2240008B A
 4893 05 00810 68000AD6

LW,X2 P1
 AI,X2 =2
 B SR20 GO BLANK OUT

*
 * FIELD TO SHIFT ABUTTS END OF CARD; SET UP TO PERFORM THIS SHIFT
 *

SR70 BAL, LNK TYPECERR TYPE: 1=CN;OVERFLOW
 DATA ERRC1
 SW,P2 P3 DOES SHIFT PUSH ORIG FIELD OFF CARD
 BLEZ SR72
 STW,P3 BLANKCNT NO = SET BLANKCNT=# OF CHARS TO
 LI,X1 MAXCLMN=1 BLANK OUT
 SW,X1 P3 SET X1=END OF (FROM) FIELD
 LI,X2 MAXCLMN=1 X2=LAST COLUMN ON CARD
 LW,T1 P2 T1=# OF CHARS TO SHIFT
 B SR15 GO SHIFT THIS FIELD

*
 * ABUTTING FIELD IS SHIFTED OFF CARD, SO SET UP TO BLANK OUT
 *

SR72 AW,P2 P3 SET T1=# OF CHARS TO BLANK OUT
 LW,T1 P2 (=ORIG FIELD WIDTH)
 LI,X2 MAXCLMN=1 X2=LAST COLUMN ON CARD
 B SR20A GO BLANK OUT

PAGE

```
*****
* CONVERT BINARY TO DECIMAL STRING *
* P1 = BINARY NUMBER *
* P2 = BYTE ADDR TO PUT STRING IN *
*****
*
```

4902	05	00B11	BINTDEC	EGU	\$	
4903	05	00B11		PUSH	(P1,P2),X1	SAVE RGS
	05	00B12				
	05	00B13				
4904	05	00B14		AI,P2	7	SET P2=LAST BYTE ADDR OF STRING
4905	05	00B15		LW,D1	P1	
4906	05	00B16		LI,P1	7	SET TO LOOP 7 TIMES
4907	05	00B17	BD10	LI,D0	0	
4908	05	00B18		DW,D0	K10	EXTRACT RIGHTMOST DIGIT
4909	05	00B19		AI,D0	'0'	CONVERT TO EBCDIC AND PUT IN STRING
4910	05	00B1A		STB,D0	0,P2	
4911	05	00B1B		AI,P2	=1	
4912	05	00B1C		BDR,P1	BD10	LOOP
4913	05	00B1D		LI,D1	' '	SET 1ST BYTE = BLANK
4914	05	00B1E		STB,D1	0,P2	
4915	05	00B1F		AI,P2	1	GET PAST THE BLANK
4916	05	00B20		LI,X1	6	NUMBER OF FIELDS TO CHECK.
4917	05	00B21		LI,D1	X'40'	NULL CHARACTER
4918	05	00B22		LI,P1	X'F0'	
4919	05	00B23	BD30	EGU	\$	
4920	05	00B23		CB,P1	0,P2	CHECK FOR ZERO
4921	05	00B24		BNE	BD20	NOT EQU IS NOT A LEADING ZERO
4922	05	00B25		STB,D1	0,P2	MAKE IT NULL
4923	05	00B26		AI,P2	1	INCREMENT BYTE ADR.
4924	05	00B27		BDR,X1	BD30	GO FIND ANOTHER
4925	05	00B28	BD20	EGU	\$	
4926	05	00B28		PULL	(P1,P2),X1	
	05	00B29				
	05	00B2A				

H01 20:44 SEP 08, '75
4927 05 00828 680E0000 A

B

O, LNK

EXIT

240

PAGE

* BLANK INPUT BUFFER *

4928					
4929					
4930					
4931					
4932					
4933					
4934	05	00B2C	0970019C 02	BLANKBUF PUSH	LNK
4935	05	00B2D	22700023 A	LI, LNK	MAXCLMN/4
4936	05	00B2E	32F0000A 02	LW, D1	4BLNKS
4937				*	
4938	05	00B2F	35FE0023 02	STW, D1	CARDIMG=1, LNK
4939	05	00B30	64700B2F	BDR, LNK	\$=1
4940				*	
4941	05	00B31	0870019C 02	PULL	LNK
4942	05	00B32	680E0000 A	B	0, LNK

PAGE

4943
 4944
 4945
 4946
 4947
 4948
 4949
 05 00B33
 4950 05 00B33 04100011 03
 03 00011 15000000 N
 80000000
 03 00013 00000002 A
 4951 05 00B34 680E0000 A
 4952
 4953
 4954
 4955
 4956
 4957 05 00B35
 4958 05 00B35 04100014 03
 03 00014 15000000 N
 80000000
 03 00016 00000002 A
 4959 05 00B36 680E0000 A
 4960
 4961
 4962 05 00B37 04100017 03
 03 00017 15000000 N
 80000000
 03 00019 00000001 A
 4963 05 00B38 680E0000 A

 * CLOSE UPDATE FILE *

 *
 *
 CLOSE EQU \$
 M:CLOSE F:EI,(SAVE)

 B O,LNK

 * CLOSE COPY FILE *

 *
 *
 CLOSE2 EQU \$
 M:CLOSE F:EB,(SAVE)

 B O,LNK
 *
 *
 CLOSE3 M:CLOSE F:EB,(REL)

 B O,LNK

PAGE

```

4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999

```

```

*****
* DELETE SPECIFIED RECORDS *
* P1 = FIRST SEQ. NUMBER TO DELETE *
* P2 = LAST SEQ. NUMBER TO DELETE *
* R1 = SEQ. NUMBER OF LAST RECORD READ *
* R2 = NUMBER OF RECORDS DELETED *
* CC1=1 IF LAST SEQ # PASSED; CC1=0 OTHERWISE *
*****
*
*
DELETED EQU $
PUSH (P1,T1) SAVE REGS
LI,T1 0 USE T1 TO COUNT # OF RECS DELETED
*
* DELETE RECORDS VIA:
* READ N
* READ N+1
* DELETE N
* READ N+2
* DELETE N+1
* ETC.
05 00B3C 6A700C61 BAL,LNK READNXTRANDM READ 1ST SEQ # OR NEXT HIGHEST #
*
* READ AND DELETE UNTIL LAST SEQ # READ OR PASSED
*
DL10 CW,R1 L(EOF) WAS AN EOF READ
BE DL30 YES = GO TYPE ERROR MESSAGE
CW,P2 R1 NO = WAS INPUT SEQ # >= LAST SEQ #
BLE DL15 YES = GO FINISH UP
STW,R1 DELNXT N TO DELETE BUFFER
LW,P1 DELNXT
AI,P1 1
BAL,LNK READNXTRANDM READ N+1
LI,LNK 3 SET KEY LENGTH
STB,LNK DELNXT

```

```

05 00B39 02200040 A
05 00B3A 0B50019C 02
05 00B3B 22800000 A

```

```

5000 05 00B47 0410001A 03
      03 0001A 00000000 N
      80000000
      03 0001C 0000019B 02
5001 05 00B48 20800001 A
5002      00000001
5003 05 00B49 3250019B 02
5004 05 00B4A 40500E2B
5005 05 00B4B 355002C3 02
5006
5007 05 00B4C 60000B3D
5008
5009
5010
5011 05 00B4D 69100B5C
5012 05 00B4E 6A700B95
5013 05 00B4F
5014 05 00B4F 35B002C3 02
5015 05 00B50 20800001 A
5016 05 00B51 32500008 A
5017 05 00B52 21500001 A
5018 05 00B53 68300B58
5019 05 00B54 22600A31 02
5020 05 00B55 6A700B11
5021 05 00B56 6A700DC4
5022 05 00B57 0000028C 02
5023 05 00B58
5024 05 00B58 02200040 A
      05 00B59 0A50019C 02
5025 05 00B5A 02200000 A
5026 05 00B5B 680E0000 A
5027
5028
5029
5030 05 00B5C 32500008 A
5031 05 00B5D 21500001 A
5032 05 00B5E 68200B63
    
```

MIDELREC F:EI,(KEY,DELNXT) DELETE N

```

      AI,T1      1      BUMP DELETED RECORD COUNTER
      DB      MODE#2
      LW,P1      DELNXT
      AND,P1      #XFFFFFFF GET RID OF THE BYTE COUNT
      STW,P1      INTFLAG1
      FIN
      B      DL10      N+1 IS BK, SET TO DELETE IT
*
* LAST SEG # HIT OR PASSED: IF HIT, FINISH UP AND EXIT WITH CC1#0
*
DL15      BL      DL20      WAS LAST SEG # PASSED
      BAL,LNK      DELETERECORD      NO, WAS HIT = DELETE IT
      DB1      MODE#2
      STW,R1      INTFLAG1
      AI,T1      1      INCR DELETE COUNT
      LW,P1      T1
      CI,P1      1      DONT SAY ANYTHING IF ITS ONLY 1
      BE      DL17
      LI,P2      BA(MSG6)+1
      BAL,LNK      BINTODEC
      BAL,LNK      TYPEMSG
      DATA      MSG6
DL17      EGU      $
      PULL      (P1,T1)      RESTORE REGS
      LCI      0
      B      0,LNK      EXIT WITH CC1#0
*
* LAST SEG # WAS PASSED: EXIT WITH CC1#1
*
DL20      LW,P1      T1
      CI,P1      1      DONT SAY ANYTHING IF ITS ONLY 1
      BLE      DL25      OR LESS
    
```

H01 20144 SEP 08, 175

5033	05	00B5F	22600A31	02		LI,P2	BA(MSG6)+1	
5034	05	00B60	6A700B11			BAL,LNK	BINT0DEC	
5035	05	00B61	6A700DC4			BAL,LNK	TYPEMSG	
5036	05	00B62	0000028C	02		DATA	MSG6	
5037		05	00B63		DL25	EGU	6	
5038	05	00B63	02200040	A		PULL	(P1,T1)	RESTORE REGS
	05	00B64	0A50019C	02				
5039	05	00B65	02200080	A		LCI	8	
5040	05	00B66	680E0000	A		B	0,LNK	EXIT WITH CC1=1
5041								
5042								
5043								
5044	05	00B67	6A700DC4		DL30	BAL,LNK	TYPEMSG	TYPE: !=EOF HIT!
5045	05	00B68	000001D5	02		DATA	ERRM1	
5046	05	00B69	68000B5C			B	DL20	GO EXIT WITH CC1=1

*
* ERROR: EOF HIT
*

PAGE

```
*****
* DELETE FILE *
* P1 = ADDR OF FILE ID IN CDT *
* CC1=1 IF FILE DOES NOT EXIST, CC1=0 OTHERWISE *
*****
*
*
```

```
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056      05 00B6A
5057      05 00B6A 02200080 A
          05 00B6B 0A30019C 02
5058      05 00B6C 22600B7A
5059      05 00B6D 3560036A 02
5060      05 00B6E 22600004 A
5061      05 00B6F 3560036D 02
5062      05 00B70 22800371 02
5063      05 00B71 2290037A 02
5064      05 00B72 22A0037D 02
5065      05 00B73 6A700C08
5066      05 00B74 04100368 02
5067      05 00B75 0410001D 03
          03 0001D 15000000 N
          80000000
          03 0001F 00000001 A
5068      05 00B76 02200080 A
          05 00B77 0A30019C 02
5069      05 00B78 02200000 A
5070      05 00B79 680E0000 A
5071
5072
5073
5074      05 00B7A
5075      05 00B7A 7230000A A
5076      05 00B7B 21300003 A
5077      05 00B7C 69300B82
5078      05 00B7D 02200080 A
```

```
LOCAL      $20,$50
DELETEFILE EQU $
PUSH       (X1,P3)

LI,P2      DF$ABN
STW,P2     02$FPT+2
LI,P2      4 INOUT
STW,P2     02$FPT+5
LI,T1      02$NAME
LI,T2      02$ACCT
LI,P3      02$PASS
BAL,LNK    0PENINIT
CAL,1      02$FPT
MICLOSEF  FILE0,(REL) FILE EXISTS, SO CLOSE AND RELEASE

PULL       (X1,P3)

LCI        0
B          0,LNK

*
*
*
DF$ABN     RES      0
           LB,X1    P3
           CI,X1    3
           BNE     BADI01
           PULL    (X1,P3)
```

```

5079 05 00B7E 0A30019C 02
5080 05 00B7F 02200080 A
5081 05 00B80 680E0000 A
5082
5083
5084 05 00B81
5085 05 00B81 3230000F A
5086 05 00B82
5087 05 00B82 2530027C A
5088 05 00B83 7246000E 02
5089 05 00B84 754002A5 02
5090 05 00B85 25300064 A
5091 05 00B86 7246000E 02
5092 05 00B87 22100001 A
5093 05 00B88 754202A5 02
5094 05 00B89 7242000A A
5095 05 00B8A 2540007F A
5096 05 00B8B 2540027C A
5097 05 00B8C 7238000E 02
5098 05 00B8D 20100002 A
5099 05 00B8E 753202A5 02
5100 05 00B8F 25400064 A
5101 05 00B90 7238000E 02
5102 05 00B91 753002A6 02
5103 05 00B92 6A700DC4
5104 05 00B93 000002A0 02
5105 00000000
5106
5107
5108 05 00B94 04900002 A
5109

```

```

LCI 8
B 0, LNK
*
* ERROR: BAD I/O
*
BADI0 RES 0
LW, X1 D1
BADI01 RES 0
SCS, X1 =4
LB, X2 HEXCHAR, X1
STB, X2 I0ERRC0D
SLS, X1 =28
LB, X2 HEXCHAR, X1
LI, X3 1
STB, X2 I0ERRC0D, X3
LB, X2 P3, X3
SLS, X2 =1
SCS, X2 =4
LB, X1 HEXCHAR, X2
AI, X3 2
STB, X1 I0ERRC0D, X3
SLS, X2 =28
LB, X1 HEXCHAR, X2
STB, X1 I0ERRC0D+1
BAL, LNK TYPEMSG
DATA I0ERRMSG
DB MODE=1
CAL3, 6 0
ELSE
M;ERR
FIN

```

MOVE CODE TO X1.
 ENTER HERE IF CODE IN X1.
 BUILD ERROR CODE

GET SUBCODE
 AND RIGHT JUSTIFY
 BUILD ERROR SUBCODE

ERROR TO UTS:

PAGE

* DELETE LAST RECORD READ *

*
*
DELETERECORD EQU \$
M:DELREC F:EI,(KEY, LASTKEY)

5110
5111
5112
5113
5114
5115
5116 05 00B95
5117 05 00B95 04100020 03
03 00020 00000000 N
80000000
03 00022 000000BD 02
5118 05 00B96 680E0000 A

B O, LNK

PAGE

```

5119
5120
5121
5122
5123
5124
5125
5126
5127
5128
5129      05 00B97
5130      05 00B97 02200060 A
          05 00B98 0B20019C 02
5131      05 00B99 32200007 A
5132      05 00B9A 3560016E 02
5133      05 00B9B 226005AC 02
5134      05 00B9C 6A700B11
5135      05 00B9D 3260016E 02
5136      05 00B9E 22300001 A
5137      05 00B9F 22400003 A
5138      05 00BA0 22E000F0 A
5139      05 00BA1 71E6016B 02
5140      05 00BA2 69300BA5
5141      05 00BA3 20300001 A
5142      05 00BA4 64400BA1
5143
5144
5145
5146      05 00BA5 22500007 A
5147      05 00BA6 22400003 A
5148      05 00BA7 71EA016B 02
5149      05 00BA8 69300BAB
5150      05 00BA9 205FFFFFF A
5151      05 00BAA 64400BA7
5152
5153
5154
    
```

```

*****
* MOVE SEQUENCE NUMBER *
* P1 = SEQ. NUMBER TO CONVERT *
* P2 = BYTE ADDR AT WHICH TO PUT STRING *
* WORD AFTER BAL = 4 CHARS TO APPEND TO STRING *
* R1 = NUMBER OF CHARS IN RESULTANT STRING *
*****
*
*
MOVESEQ  FGU      $
          PUSH     (X4, LNK)      SAVE REGS
          LW, X4    LNK           SAVE LINK
          STW, P2   TEMPBLCK+3    SAVE P2
          LI, P2    BA(TEMPBLCK)
          BAL, LNK  BINTODEC      CONVERT SEQ # TO EBCDIC: 'DDDDDD'
          LW, P2    TEMPBLCK+3    RESTORE P2
          LI, X1    1
          LI, X2    3
          LI, D0    '0'
          CB, D0    TEMPBLCK, X1  CALC X1=POSITION OF 1ST NON-ZERO
          BNE       MQ10          CHAR OR 4TH DIGIT
          AI, X1    1
          BDR, X2   $=3
*
* SUPPRESS TRAILING ZEROS
*
MQ10     LI, P1    7
          LI, X2    3
          CB, D0    TEMPBLCK, P1  CALC P1=POSITION OF 1ST NON-ZERO
          BNEZ      MQ20          DIGIT FROM RIGHT OF 4TH DIGIT
          AI, P1    =1
          BDR, X2   $=3
*
* BUILD STRING TO LEFT OF DECIMAL POINT
*
    
```

NO1 20:44 SEP 08, 1975

250

5155	05	00BAB	72E6016B	02	MQ20	LB,D0	TEMPBLCK,X1		MOVE NON-ZERO DIGITS TO LEFT OF
5156	05	00BAC	75EC0000	A		STB,D0	0,P2		DEC. PT. TO ADDR IN P2 (AT LEAST
5157	05	00BAD	20600001	A		AI,P2	1		1 DIGIT MOVED)
5158	05	00BAE	20300001	A		AI,X1	1		
5159	05	00BAF	21300004	A		CI,X1	4		
5160	05	00BB0	68200BAB			BLE	MQ20		
5161	05	00BB1	22E0004B	A		LI,D0	1,1		MOVE 1,1 TO ADDR IN P2
5162	05	00BB2	75EC0000	A		STB,D0	0,P2		
5163	05	00BB3	20600001	A		AI,P2	1		
5164									
5165									
5166									
5167	05	00BB4	31500003	A	MQ25	CW,P1	X1		MOVE (IF ANY) DIGITS TO RIGHT OF
5168	05	00BB5	69100BBB			BL	MQ30		DEC. PT. TO ADDR IN P2
5169	05	00BB6	72E6016B	02		LB,D0	TEMPBLCK,X1		
5170	05	00BB7	75EC0000	A		STB,D0	0,P2		
5171	05	00BB8	20600001	A		AI,P2	1		
5172	05	00BB9	20300001	A		AI,X1	1		
5173	05	00BBA	68000BB4			B	MQ25		
5174									
5175									
5176									
5177	05	00BBB	22300000	A	MQ30	LI,X1	0		
5178	05	00BBC	22400004	A		LI,X2	4		
5179	05	00BBD	F2E60002	A	MQ30A	LB,D0	*X4,X1		MOVE 4 CHARS SPECIFIED TO END OF
5180	05	00BBE	68300BC1			BEZ	*+3		THIS STRING, SKIPPING 0 CHARS
5181	05	00BBF	75EC0000	A		STB,D0	0,P2		
5182	05	00BC0	20600001	A		AI,P2	1		
5183	05	00BC1	20300001	A		AI,X1	1		
5184	05	00BC2	64400BBD			BDR,X2	MQ30A		
5185	05	00BC3	32B00006	A		LW,R1	P2		
5186	05	00BC4	02200060	A		PULL	(X4,LNK)		RESTORE REGS
5187	05	00BC5	0A20019C	02					
5188	05	00BC6	38B00006	A		SW,R1	P2		CALC R1=NUMBER OF CHARS IN STRING
						B	1,LNK		EXIT

* BUILD STRING TO RIGHT OF DECIMAL POINT

* APPEND 4 SPECIFIED CHARS

PAGE

```

5189
5190 *****
5191 * OPEN UPDATE FILE *
5192 * OPEN UPDATE FILE (OPEN1 OPENS COPY INPUT FILE) *
5193 * P1 = ADDR OF FILE ID IN CDT *
5194 * CC1=1 IF FILE DOES NOT EXIST; CC1=0 OTHERWISE *
5195 * CC2=1 IF FILE IS NOT KEYED; CC2=0 OTHERWISE *
5196 *****
5197 *
5198 *
5199 LOCAL $20,$90
5200 OPEN EQU $
5201 05 00BC8 02200080 A PUSH (X1,P3)
05 00BC9 0B30019C 02
5202 05 00BCA 22600004 A LI,P2 4 INPUT
5203 05 00BCB 68000000 F B $20
5204 *
5205 *
5206 05 00BCL 05 00BCL OPEN1 EQU $
5207 05 00BCC 02200080 A PUSH (X1,P3)
05 00BCD 0B30019C 02
5208 05 00BCE 22600001 A LI,P2 1 INPUT
5209 *
5210 *
5211 $20 RES 0
5212 05 00BCF 35600356 02 STW,P2 0$FPT+5
5213 05 00BD0 22600BE4 LI,P2 0$ABN
5214 05 00BD1 35600353 02 STW,P2 0$FPT+2
5215 05 00BD2 2280035A 02 LI,T1 0$NAME SET ADDRESS REGISTERS FOR
5216 05 00BD3 22900363 02 LI,T2 0$ACCT STORING PARAMETERS INTO
5217 05 00BD4 22A00366 02 LI,P3 0$PASS FPT.
5218 05 00BD5 6A700C08 BAL,LNK OPENINIT
5219 05 00BD6 04100351 02 CAL,1 0$FPT OPEN FILE
5220 05 00BD7 32300005 N LW,X1 F:EI+5 FILE EXISTS.
5221 05 00BD8 2530007C A SLS,X1 =4 ORGANIZATION SHOULD BE KEYED.
5222 05 00BD9 4B300003 02 AND,X1 XF
5223 05 00BDA 21300002 A CI,X1 2

```

H01 20:44 SEP 08, '75

5224	05	00BDB	69300000	F	BNE	\$90	
5225	05	00BDC	02200080	A	PULL	(X1,P3)	IT IS.
	05	00BDD	0A30019C	02			
5226	05	00BDE	02200000	A	LCI	0	
5227	05	00BDF	680E0000	A	B	0, LNK	
5228					*		
5229					*		
5230	05	00BEO			\$90	RES	0
5231	05	00BEO	02200080	A	PULL	(X1,P3)	
	05	00BE1	0A30019C	02			
5232	05	00BE2	02200040	A	LCI	4	
5233	05	00BE3	680E0000	A	B	0, LNK	
5234					*		
5235					*		
5236					*		
5237	05	00BE4			0\$ABN	RES	0
5238	05	00BE4	7230000A	A	LB,X1	P3	
5239	05	00BE5	21300003	A	CI,X1	3	
5240	05	00BE6	69300B82		BNE	BADI01	
5241	05	00BE7	02200080	A	PULL	(X1,P3)	NO FILE.
	05	00BE8	0A30019C	02			
5242	05	00BE9	02200080	A	LCI	8	
5243	05	00BEA	680E0000	A	B	0, LNK	

PAGE

```

5244
5245 *****
5246 * OPEN (OUTPUT) FILE FOR COPYING *
5247 * P1 = ADDR OF FILE ID IN CDT *
5248 * CC1=1 IF FILE DOES NOT EXIST; CC1=0 OTHERWISE *
5249 *****
5250 *
5251 *
5252 LOCAL $20
5253 05 00BE8 02200080 A OPEN3 PUSH (X1,P3)
5254 05 00BE9 0A30019C 02 LI,P2 2 OUTPUT
5255 05 00BEA 22600002 A B $20
5256 *
5257 *
5258 05 00BE7 05 00BE7 OPEN2 EQU $
5259 05 00BE8 02200080 A PUSH (X1,P3)
5260 05 00BE9 0A30019C 02
5261 05 00BEA 22600004 A LI,P2 4 INPUT
5262 05 00BEB 3B60036D 02 $20 STW,P2 @2$FPT+5
5263 05 00BEC 22600BFE LI,P2 @2$ABN
5264 05 00BED 3B60036A 02 STW,P2 @2$FPT+2
5265 *
5266 05 00BEF 22800371 02 LI,T1 @2$NAME SAME.
5267 05 00BEE 2290037A 02 LI,T2 @2$ACCT
5268 05 00BEF 22A0037D 02 LI,P3 @2$PASS
5269 05 00BE8 6A700C08 BAL,LNK @PENINIT
5270 05 00BE9 04100368 02 CAL1,1 @2$FPT
5271 05 00BEA 02200080 A PULL (X1,P3)
5272 05 00BEB 0A30019C 02
5273 05 00BEC 02200000 A LCI 0
5274 05 00BED 680E0000 A B 0,LNK
5275 *
5276 *
5277 05 00BEF 02$ABN RES 0
05 00BEF 7230000A A LB,X1 P3

```

HC1 20:44 SEP 08, '75

254

5278 05 008FF 21300003 A
5279 05 00C00 69300B82
5280 05 00C01 22300002 A
5281 05 00C02 3530036D 02
5282 05 00C03 04100368 02
5283 05 00C04 02200080 A
05 00C05 0A30019C 02
5284 05 00C06 02200080 A
5285 05 00C07 680E0000 A

CI,X1 3
BNE BADI01
LI,X1 2
STW,X1 02\$FPT+5
CAL1,1 02\$FPT
PULL (X1,P3)

LCI 8
B 0,LNK

NO PREVIOUS FILE, OPEN FOR OUTPUT.

PAGE

```
*****
* INITIALIZE OPEN FPT *
* P1 = ADDR OF FILE ID IN CDT *
* T1 = FPT ENTRY TO PUT FILE NAME IN *
* T2 = FPT ENTRY TO PUT ACCOUNT NUMBER IN *
* P3 = FPT ENTRY TO PUT PASSWORD IN *
*****
```

```
5286
5287
5288
5289
5290
5291
5292
5293
5294
5295
5296
5297      05 00C08
5298      05 00C08      3230000A 02
5299      05 00C09      B5300009 A
5300      05 00C0A      B530000A A
5301      05 00C0B      22400001 A
5302      05 00C0C      B5380009 A
5303      05 00C0D      B538000A A
5304      05 00C0E      224FFFFFF A
5305      05 00C0F      32300E2C
5306      05 00C10      B5380009 A
5307      05 00C11      32300E2D
5308      05 00C12      B538000A A
5309      05 00C13      F2400005 A
5310      05 00C14      F5400008 A
5311      05 00C15      F2380005 A
5312      05 00C16      F5380008 A
5313      05 00C17      64400C15
5314      05 00C18      F2400008 A
5315      05 00C19      20400004 A
5316      05 00C1A      2540007E A
5317      05 00C1B      30500004 A
5318      05 00C1C      F2400005 A
5319      05 00C1D      68300000 F
5320      05 00C1E      F2380005 A
5321      05 00C1F      204FFFFFF A
5322      05 00C20      F5380009 A
```

```
OPENINIT LOCAL          $50,$60,$65,$70,$80
          EQU            $
          LW,X1          4BLNKS
          STW,X1         *T2
          STW,X1         *P3
          LI,X2          1
          STW,X1         *T2,X2
          STW,X1         *P3,X2
          LI,X2          =1
          LW,X1          L(X'02000202')
          STW,X1         *T2,X2
          LW,X1          L(X'03010202')
          STW,X1         *P3,X2
          LB,X2          *P1
          STB,X2         *T1
          LB,X1          *P1,X2
          STB,X1         *T1,X2
          BDR,X2         $=2
          LB,X2          *T1
          AI,X2          4
          SLS,X2         =2
          AW,P1          X2
          LB,X2          *P1
          BEZ            $50
          LB,X1          *P1,X2
          AI,X2          =1
          STB,X1         *T2,X2
```

INITIALIZE ACCOUNT AND PASS CONTROLS

MOVE FILE NAME TO BUFFER.
P1 POINTS TO IT.

SKIP TO ACCOUNT. BYTE COUNT FROM FPT

P1 NOW AT ACCOUNT

NO ACCOUNT
MOVE ACCOUNT TO BUFFER
THIS LOOP PUTS NO BYTE COUNT INTO
FPT.

5323	05	00C21	20400000	A	AI,X2	0	
5324	05	00C22	69200C1E		BGZ	\$=4	
5325	05	00C23	F2400005	A	LB,X2	*P1	SKIP TO PASS
5326	05	00C24	20400004	A	AI,X2	4	&
5327	05	00C25	2540007E	A	SLS,X2	=2	
5328	05	00C26	30500004	A	AW,P1	X2	P1 NOW POINTS TO PASS
5329	05	00C27	68000000	F	B	\$60	
5330					*		
5331					*		
5332	05	00C28			\$50	RES	0
5333	05	00C28	20500001	A	AI,P1	1	STEP TO PASS
5334	05	00C29	223FFFFE	A	LI,X1	=2	SET FPT FOR NO ACCOUNT, BY SAYING
5335	05	00C2A	F5460009	A	STB,X2	*T2,X1	NO USABLE WORDS.
5336					*		
5337					*		
5338	05	00C2B			\$60	RES	0
5339	05	00C2B	F2400005	A	LB,X2	*P1	
5340	05	00C2C	68300000	F	BEZ	\$70	NO PASS
5341					*		
5342					*		
5343	05	00C2D			\$65	RES	0
5344	05	00C2D	F2380005	A	LB,X1	*P1,X2	MOVE PASSWORD WITH BYTE COUNT
5345	05	00C2E	204FFFFFF	A	AI,X2	=1	
5346	05	00C2F	F538000A	A	STB,X1	*P3,X2	
5347	05	00C30	20400000	A	AI,X2	0	
5348	05	00C31	69200C2D		BGZ	\$65	
5349	05	00C32	68000000	F	B	\$80	
5350					*		&
5351					*		
5352	05	00C33			\$70	RES	0
5353	05	00C33	223FFFFE	A	LI,X1	=2	SET FPT FOR NO PASS, BY SAYING
5354	05	00C34	F546000A	A	STB,X2	*P3,X1	NO USABLE WORDS.
5355					*		
5356					*		
5357	05	00C35			\$80	RES	0
5358	05	00C35	680E0000	A	B	0,LNK	

PAGE

 * OPEN NEW (OUTPUT ONLY) FILE *
 * P1 = ADDR OF FILE ID IN CDT *
 * CC1=1 IF FILE DOES NOT EXIST; CC1=0 OTHERWISE *

5359
 5360
 5361
 5362
 5363
 5364
 5365
 5366
 5367 05 00C36
 5368 05 00C36 02200080 A
 05 00C37 0A30019C 02
 5369 05 00C38 22600C45
 5370 05 00C39 35600353 02
 5371 05 00C3A 22600004 A
 5372 05 00C3B 35600356 02
 5373 05 00C3C 2280035A 02
 5374 05 00C3D 22900363 02
 5375 05 00C3E 22A00366 02
 5376 05 00C3F 6A700C08
 5377 05 00C40 04100351 02
 5378 05 00C41 02200080 A
 05 00C42 0A30019C 02
 5379 05 00C43 02200000 A
 5380 05 00C44 680E0000 A
 5381
 5382
 5383
 5384 05 00C45
 5385 05 00C45 7230000A A
 5386 05 00C46 21300003 A
 5387 05 00C47 69300B82
 5388 05 00C48 22300002 A
 5389 05 00C49 35300356 02
 5390 05 00C4A 04100351 02
 5391 05 00C4B 02200080 A
 05 00C4C 0A30019C 02
 5392 05 00C4D 02200080 A

OPENNEW EQU \$
 PUSH (X1,P3)
 LI,P2 @N\$ABN
 STW,P2 @SFPT+2
 LI,P2 4 INOUT
 STW,P2 @SFPT+5
 LI,T1 @NAME
 LI,T2 @ACCT
 LI,P3 @PASS
 BAL,LNK @PENINIT
 CAL,1 @SFPT
 PULL (X1,P3) FILE EXISTS.
 LCI 0 NOTE.
 B 0,LNK
 *
 *
 *
 @N\$ABN RES 0
 LB,X1 P3
 CI,X1 3
 BNE BADI@1
 LI,X1 2 OPEN FOR OUTPUT.
 STW,X1 @SFPT+5
 CAL,1 @SFPT
 PULL (X1,P3)
 LCI 8

```

5393 05 00C4E 680E0000 A B 0,LNK
5394 *
5395 *
5396 *****
5397 * VERIFY CARRIAGE RETURN EXISTS ON OUTPUT RECORD. *
5398 *****
5399 *
5400 05 00C4F 0970019C 02 PUTCR PUSH LNK
5401 05 00C50 D01 MODE=2
5402 05 00C50 6A70003B BAL,LNK TABCOMPRESS
5403 05 00C51 327000AF 02 LW,LNK CRFLAG DO NOT INSERT CR WHEN FLAG IS
5404 05 00C52 6930005F BNEZ PUTCR2 NON-ZERO
5405 05 00C53 327000E7 02 LW,LNK RECSIZE
5406 05 00C54 207FFFFFF A AI,LNK =1
5407 05 00C55 22F00015 A LI,D1 X,151
5408 05 00C56 71FE0024 02 CB,D1 CARDIMG,LNK
5409 05 00C57 6830005F BE PUTCR2
5410 *
5411 05 00C58 20700001 A AI,LNK 1 IF NO CR
5412 05 00C59 2170008C A CI,LNK MAXCLMN
5413 05 00C5A 6910005D BL $+3
5414 05 00C5B 22700088 A LI,LNK MAXCLMN=1 (DO NOT GO BEYOND COL. 140)
5415 05 00C5C 357000E7 02 STW,LNK RECSIZE
5416 05 00C5D 75FE0024 02 STB,D1 CARDIMG,LNK INSERT ONE
5417 05 00C5E 331000E7 02 MTW,1 RECSIZE
5418 *
5419 05 00C5F 0870019C 02 PUTCR2 PULL LNK
5420 05 00C60 680E0000 A B 0,LNK
    
```

PAGE

```
*****
* READ RANDOM RECORD OR NEXT HIGHEST ONE *
* P1 = SEQ. NUMBER TO READ *
* R1 = SEQ. NUMBER ACTUALLY READ *
* CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE *
*****
```

```
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431      05 00C61
5432      05 00C61 0970019C 02
5433      05 00C62 6A700C6D
5434      05 00C63 69800000 F
5435      05 00C64 32B000BD 02
5436      05 00C65 45B00008 02
5437      05 00C66 0870019C 02
5438      05 00C67 02200000 A
5439      05 00C68 680E0000 A
5440
5441
5442      05 00C69
5443      05 00C69 6A700C7F
5444      05 00C6A 0870019C 02
5445      05 00C6B 02200080 A
5446      05 00C6C 680E0000 A
```

```
*
*
* LOCAL $20
READNXRANDOM EQU $
* PUSH LNK
BAL,LNK READRANDOM
* BCS,8 $20
LW,R1 LASTKEY GOT IT, RETURN KEY.
* AND,R1 XFFFFFFF
* PULL LNK
* LCI 0
* B 0,LNK
*
*
* $20 RES 0
BAL,LNK READSEQUEN NOW GET NEXT KEY, IN R1.
* PULL LNK
* LCI 8
* B 0,LNK
```

PAGE

```

5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473

```

05	00C6D	02200040	A	READRANDOM	EQU *
05	00C6E	0B70019C	02	PUSH	(LNK,P3)
05	00C6F	6A700B2C		BAL,LNK	BLANKBUF
05	00C70	6A700CE6		BAL,LNK	SETKEY
05	00C71			DB1	MODE=2
05	00C71	04100023	03	M:SETDCB	F:EI,(ERR,RR\$ERR)
03	00023	06000000	N		
		80000000			
03	00025	00000C78	05		
				M:READ	F:EI,,
					(ERR,RR\$ERR),,
					(WAIT),,
					(SIZE,MAXCLMN),,
05	00C72	04100026	03		(KEY,KBUF)
03	00026	10000000	N		
		98000010			
03	00028	00000C78	05		
03	00029	0000008C	A		
03	0002A	000000BA	02		
05	00C73	6A700CEA		BAL,LNK	SETLASTKEY
05	00C74	02200040	A	PULL	(LNK,P3)
05	00C75	0A70019C	02		
05	00C76	02200000	A	LCI	0
05	00C77	680E0000	A	B	0,LNK
				*	
				*	
				*	
05	00C78			RR\$ERR	RES 0

```

*****
* READ RANDOM RECORD *
* P1 = SEQ. NUMBER TO READ *
* CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE *
*****

```

(P1) ARE KEY, I.E. SEQUENCE

H01 20:44 SEP 08, '75

5474 05 00C78 72F0000A A
5475 05 00C79 21F00043 A
5476 05 00C7A 69300E81
5477 05 00C7B 02200040 A
05 00C7C 0A70019C 02
5478 05 00C7D 02200080 A
5479 05 00C7E 680E0000 A

LB,D1 P3
CI,D1 X1431
BNE BAD10
PULL (LNK,P3)

LCI 8
B 0,LNK

PAGE

5480
 5481
 5482
 5483
 5484
 5485
 5486
 5487
 5488
 5489 05 00C7F 02200040 A
 05 00C80 0A70019C 02
 5490 05 00C81 6A700B2C
 5491 05 00C82
 5492 05 00C82 0410002B 03
 03 0002B 06000000 N
 40000000
 03 0002D 00000C90 05
 5493
 5494
 5495
 5496 05 00C83 0410002E 03
 03 0002E 10000000 N
 50000010
 03 00030 00000C90 05
 03 00031 0000008C A
 5497 05 00C84 6A700CEA
 5498 05 00C85 32F00005 N
 5499 05 00C86 25F0007C A
 5500 05 00C87 4BF00003 02
 5501 05 00C88 B2B0000A N
 5502 05 00C89 4BF00008 02
 5503 05 00C8A 21F00002 A
 5504 05 00C8B 68300C8D
 5505 05 00C8C 22B00000 A
 5506 05 00C8D 02200040 A
 05 00C8E 0A70019C 02
 5507 05 00C8F 680E0000 A

 * READ SEQUENTIAL RECORD *
 * R1 = SEQ. NUMBER READ IN *

 *
 LOCAL \$10,\$20
 *
 READSEQUEN EQU \$
 PUSH (LNK,P3)
 BAL,LNK BLANKBUF
 DB1 MBDE=2
 M:SETDCB F:EI,(ABN,RS,ABN)
 M:READ F:EI,,
 (WAIT),,
 (SIZE,MAXCLMN),,
 (ABN,RS,ABN)
 BAL,LNK SETLASTKEY
 LW,D1 F:EI+5
 SLS,D1 =4
 AND,D1 XF
 LW,R1 *F:EI+10
 AND,R1 XFFFFFFF
 CI,D1 2
 BE *+2
 LI,R1 0
 PULL (LNK,P3)
 B 0,LNK

CHECK ORGANIZATION

RETURN SEQUENCE

ZERO IF NOT KEYED.

5508				*			
5509				*			
5510				*			
5511	05	00C90		RS#ABN	RES	0	
5512	05	00C90	72F0000A A		LB,D1	P3	
5513	05	00C91	21F00006 A		CI,D1	6	
5514	05	00C92	69300B81		BNE	BAD10	
5515	05	00C93	32B00E28		LW,R1	L(EOF)	
5516				*			PUT LAST SEQ # IN EOF MESH
5517	05	00C94	02200070 A		PUSH	(P1,R1)	
	05	00C95	0550019C 02				
5518	05	00C96	22500000 A		LI,P1	0	INITIALIZE LASTKEY IN CASE SEQ
5519	05	00C97	355000BD 02		STW,P1	LASTKEY	BELOW TAKES ABN EXIT
5520	05	00C98	04100032 03		M:PRECOR	F:EI,(ABN,RS#ABNABN),(REV)	POSN BEFORE LAST REC
	03	00032	10000000 N				
			40000010				
	03	00034	00000C9B 05				
5521	05	00C99	04100035 03		M:READ	F:EI,(ERR,RS#ABNABN),(SIZE,MAXCLMN)	AND GET KEY
	03	00035	10000000 N				
			90000000				
	03	00037	00000C9B 05				
	03	00038	0000008C A				
5522	05	00C9A	6A700CEA		BAL,LNK	SETLASTKEY	IN CORE LOC LASTKEY
5523		05 00C9B		RS#ABNABN	EQU *		OR BYPASS SETTING IF TROUBLES
5524	05	00C9B	22B00015 A		LI,R1	21	NUMBER OF TEXTC BYTES
5525	05	00C9C	325000BD 02		LW,P1	LASTKEY	LAST SEQ # READ
5526	05	00C9D	45500008 02		AND,P1	FFFFFFF	ZAP TEXTC BYTE IN KEY
5527	05	00C9E	22600765 02		LI,P2	BA(ERRM1)+17	
5528	05	00C9F	6A700B97		BAL,LNK	MOVESEQ	
5529	05	00CA0	08000000 A		GEN4	E0M,0,0,0	
5530	05	00CA1	22500001 A		LI,P1	1	START PAST TEXTC COUNT
5531		05 00CA2		RS#ABNE0M	EQU *		
5532	05	00CA2	726A01D5 02		LB,P2	ERRM1,P1	GET BYTE LOOKING FOR E0M
5533	05	00CA3	21600008 A		CI,P2	E0M	
5534	05	00CA4	68300CA7		BE	RS#ABN0UT	EXIT WHEN FOUND
5535	05	00CA5	20500001 A		AI,P1	1	
5536	05	00CA6	68000CA2		B	RS#ABNE0M	KEEP GOING TILL FOUND

H01 20:44 SEP 08, 175

5537 05 00CA7 205FFFFF A
 5538 05 00CA7 205FFFFF A
 5539 05 00CA8 755001D5 02
 5540 05 00CA9 02200070 A
 05 00CAA 0A50019C 02
 5541 05 00CAB 02200040 A
 05 00CAC 0A70019C 02
 5542 05 00CAD 680E0000 A

RS\$ABNOUT

EQU \$

AI,P1 #1
 STB,P1 ERRM1
 PULL (P1,R1)

 PULL (LNK,P3)

 B 0,LNK

POINT TO BYTE BEFORE EOM
 AND USE IT AS TEXTC COUNT

PAGE

```

5543
5544 *****
5545 * READ TELETYPE *
5546 * R1 = NUMBER OF CHARS READ *
5547 *****
5548 *
5549 *
5550 05 00CAE READTELETYPE2 EQU $
5551 05 00CAE 02200020 A PUSH (X1,X2) SAVE REGS
5552 05 00CAF 0530019C 02 LI,X2 1 USE X2=1 FOR (READTELETYPE2)
5553 05 00CB0 22400001 A B RT5
5554 *
5555 *
5556 05 00CB2 READTELETYPE EQU $
5557 05 00CB2 02200020 A PUSH (X1,X2) SAVE REGS
5558 05 00CB3 0530019C 02 LI,X2 0 USE X2=0 FOR (READTELETYPE)
5559 05 00CB4 22400000 A RT5 LW,R0 4BLNKS
5560 05 00CB5 3200000A 02 RT5 LI,X1 MAXCLMN/4
5561 05 00CB7 67080CE0 EXU RTSTWTBL,X2
5562 05 00CB8 64300CB7 BDR,X1 $=1
5563 0000000U D0 MODE=1
5564 *S* CAL3,0 0 READ A CHAR
5565 *S* EXU RTSTBTBL,X2 PUT CHAR IN BUFFER
5566 *S* AI,X1 1 INCR CHAR COUNT
5567 *S* CI,R0 CR
5568 *S* BNE $=4 LOOP UNTIL C/R
5569 *S* LW,R1 X1 SET R1=COL# OF C/R
5570 ELSE
5571 05 00CB9 330002C2 02 MTW,0 CFLAG IF FIRST READ, MOVE COMMAND IN
5572 05 00CBA 69300CD7 BNEZ RT10 FROM J:CCBUF
5573 05 00CBB 331002C2 02 MTW,1 CFLAG
5574 *
5575 05 00CBC 72300000 X LB,X1 J:CCARS GET BYTE CNT FROM J:PUF
5576 05 00CBD 72060000 X LB,R0 J:CCBUF,X1 MOVE RECORD INTO APPROPRIATE
5577 05 00CBE 67080CDE EXU RTSTBTBL,X2 BUFFER

```

5578	05	00CBF	64300CBD		BDR,X1	*-2	
5579				*			
5580	05	00CC0	72000000 X		LB,R0	J:CCBUF	
5581	05	00CC1	67080CDE		EXU	RTSTBTBL,X2	
5582	05	00CC2	72800000 X		LB,R1	JB:CCARS	
5583	05	00CC3	210000C2 A		CI,R0	'B'	IS COMMAND 'BUILD' FROM TEL
5584	05	00CC4	68300CDB		BE	RT17	YES. IT STARTS WITH 'B'.
5585				*			
5586	05	00CC5	357002C1 02		STW,LNK	BUILDFLAG	
5587	05	00CC6	6A700DC4		BAL,LNK	TYPEMSG	TYPE
5588	05	00CC7	000002E1 02		DATA	UTSM1	'EDIT HERE'
5589	05	00CC8	20BFFFFF A		AI,R1	=1	
5590	05	00CC9	327002C1 02		LW,LNK	BUILDFLAG	
5591	05	00CCA	72000000 X		LB,R0	J:CCBUF	
5592	05	00CCB	210000C5 A		CI,R0	'E'	IS THIS A FORM OF EDIT
5593	05	00CCC	69300CB5		BNE	RT5	IF NOT, MUST BE A START, RUN,
5594				*			OR GJOB SO READ FIRST COMMAND.
5595	05	00CCD	22300004 A		LI,X1	4	START LOOKING IN BYTE 5.
5596	05	00CCE	22000040 A		LI,R0	1 1	
5597	05	00CCF	3130000B A	RT8	CW,X1	R1	CHECK FOR END OF RECORD
5598	05	00CD0	68100CB5		BGE	RT5	IF SO, GET NEXT COMMAND. OTHERWISE,
5599	05	00CD1	71060176 02		CB,R0	TTYIMG,X1	IF NON-BLANK ENCOUNTERED, ACCEPT
5600	05	00CD2	69300CD5		BNE	RT9	
5601	05	00CD3	20300001 A		AI,X1	1	INCREMENT TO NEXT BYTE.
5602	05	00CD4	68000CCF		B	RT8	
5603	05	00CD5	20800001 A	RT9	AI,R1	1	UPDATE TOTAL BYTE COUNT TO INCLUDE
5604	05	00CD6	68000CDB		B	RT17	CR, THEN EXECUTE COMMAND.
5605				*			
5606	05	00CD7	32380CE2	RT10	LW,X1	RTADDTBL,X2	
5607	05	00CD8	041002DC 02		CAL1,1	RT#FPT	
5608				*			
5609	05	00CD9	32800004 N	RT15	LW,R1	M:UC+4	
5610	05	00CDA	2580006F A		SLS,R1	=17	
5611					FIN		
5612	05	00CDB	02200020 A	RT17	PULL	(X1,X2)	RESTORE REGS.
	05	00CDC	0A30019C 02				
5613	05	00CDD	680E0000 A		B	0,LNK	EXIT

H01 20144 SEP 08, '75

267

5614			*		
5615			*		
5616	05	00CDE	RTSTBTBL	EGU	*
5617	05	00CDE		STB,R0	CARDIMG,X1
5618	05	00CDF		STB,R0	TTYIMG,X1
5619			*		
5620	05	00CEU	RTSTWTBL	EGU	*
5621	05	00CE0		STW,R0	CARDIMG=1,X1
5622	05	00CE1		STW,R0	TTYIMG=1,X1
5623			*		
5624		00000001		D8	MODE=2
5625	05	00CE2	RTADDTBL	FGU	*
5626	05	00CE2		DATA	CARDIMG
5627	05	00CE3		DATA	TTYIMG
5628				FIN	

PAGE

* RE-OPEN LAST UPDATE FILE *

5629
5630
5631
5632
5633
5634
5635 05 00CE4
5636 05 00CE4 0*100351 02
5637 05 00CE5 6*0E0000 A

*
*
REOPEN EQU *
 CAL1,1 6SFPT
 B 0,LNK

PAGE

5638
5639
5640
5641
5642
5643
5644 05 00CE6
5645 05 00CE6 355000BA 02
5646 05 00CE7 22F00003 A
5647 05 00CE8 75F000BA 02
5648 05 00CE9 680E0000 A
5649
5650
5651
5652 05 00CEA
5653 05 00CEA 0970019C 02
5654 05 00CEB 82F0000A N
5655 05 00CEC 35F000BD 02
5656 05 00CED 32700004 N
5657 05 00CEE 2570006F A
5658 05 00CEF 357000E7 02
5659 05 00CF0 207FFFFFFF A
5660 05 00CF1 357000B1 02
5661 05 00CF2 22F00015 A
5662 05 00CF3 71FE0024 02
5663 05 00CF4 68300CFA
5664 05 00CF5 22F0000D A
5665 05 00CF6 71FE0024 02
5666 05 00CF7 68300CFA
5667 05 00CF8 6A700A56
5668 05 00CF9 68000CFE
5669 05 00CFA 22F00040 A
5670 05 00CFB 75FE0024 02
5671 05 00CFC 357000E7 02
5672 05 00CFD 33F000B1 02
5673 05 00CFE
5674 05 00CFE 6A700D01

```

*****
* SET KEY FOR READ OR WRITE *
* P1 = SEQ. NUMBER TO PUT IN KEY *
*****
*
SETKEY EQU #
      STW,P1 KBUF
      LI,D1 3
      STB,D1 KBUF
      B 0,LNK
*
* SAVE KEY FROM LAST READ
*
SETLASTKEY EQU #
      PUSH LNK
      LW,D1 *F:EI+10
      STW,D1 LASTKEY
      LW,LNK F:EI+4
      SLS,LNK =17
      STW,LNK RECSIZE
      AI,LNK =1
      STW,LNK EBDCLMN
      LI,D1 X:15:
      CB,D1 CARDIMG,LNK
      BE SETK2
      LI,D1 X:0D:
      CB,D1 CARDIMG,LNK
      BE SETK2
      BAL,LNK SETE0D
      B SETK6
      SETK2 LI,D1 1:
      STB,D1 CARDIMG,LNK
      STW,LNK RECSIZE
      MTW,-1 EBDCLMN
      SETK6 DB1 MODE=2
      BAL,LNK TABEXPAND

```

SET RECORD SIZE RECEIVED

DELETE CR FROM IMAGE. CHECKBOTH
SAVE ENDING COLUMN
BTM

AND UTS CR'S

BLANK WILL NOT INTERFERE
WITH STRING EDITING.

DECREMENT ENDING COLUMN

H01 20:44 SEP 08, '75

5675 05 00CFF 0870019C 02
5676 05 00D00 680E0000 A

PULL
B

LNK
0, LNK

270

PAGE

 * IN UTS VERSION, EACH RECORD SUBJECT TO *
 * EDITING WILL HAVE EMBEDDED TAB CHARACTERS *
 * EXPANDED ACCORDING TO THE CURRENT TAB *
 * STOPS CONTAINED IN THE M:UC DCB. *

5677								
5678								
5679								
5680								
5681								
5682								
5683								
5684								
5685		00000001			DB	MODE=2		
5686		05 00D01		TABEXPAND		EQU \$		
5687	05	00D01	390000B6 02		MTW,0	FILETYPE		IF NOT EDITING,
5688	05	00D02	691E0000 A		BLZ	0,LNK		EXIT.
5689	05	00D03	390002C7 02		MTW,0	TABXFLAG		
5690	05	00D04	693E0000 A		BNEZ	0,LNK		
5691	05	00D05	02200070 A		PUSH	(X3,LNK)		
	05	00D06	0B10019C 02					
5692	05	00D07	321000E7 02		LW,X3	RECSIZE		SAVE RECSIZE
5693	05	00D08	3B1001A3 02		STW,X3	TRECSIZE		
5694	05	00D09	22100000 A		LI,X3	0		START AT FIRST TAB IN DCB.
5695	05	00D0A	22200000 A		LI,X4	0		START AT FIRST CHAR. IN CARDIMG
5696	05	00D0B	3B2002C6 02		STW,X4	TABCFLAG		INDICATE DONT COMPRESS
5697	05	00D0C	22500005 A	TABX4	LI,P1	X'05'		
5698	05	00D0D	71540024 02	TABX5	CB,P1	CARDIMG,X4		
5699	05	00D0E	6B300D15		BE	TABX10		
5700	05	00D0F	20200001 A		AI,X4	1		
5701	05	00D10	312000E7 02		CW,X4	RECSIZE		
5702	05	00D11	69100D0D		BL	TABX5		WHEN OUT OF CHARACTERS,
5703				*				
5704	05	00D12	02200070 A	TABX7	PULL	(X3,LNK)		EXIT
	05	00D13	0A10019C 02					
5705	05	00D14	6B0E0000 A		B	0,LNK		
5706				*				
5707	05	00D15	7232000F N	TABX10	LB,X1	M:UC+15,X3		
5708	05	00D16	69300D1F		BNEZ	TABX15		
5709				*				
5710	05	00D17	20100000 A		AI,X3	0		IF NO MORE TABS IN DCB, WE CAN
5711	05	00D18	69300D12		BNEZ	TABX7		EXIT, UNLESS THERE WERE NO TABS

5712	05	00D19	330002C5	02	TABX12	MTW,0	TABERRFLAG	AT ALL.
5713	05	00D1A	69300D12			BNEZ	TABX7	
5714	05	00D1B	331002C5	02		MTW,1	TABERRFLAG	
5715	05	00D1C	6A700DC4			BAL, LNK	TYPMSG	IN THAT CASE, ERROR.
5716	05	00D1D	0000030B	02		DATA	UTSMS	
5717	05	00D1E	68000D12			B	TABX7	
5718					*			
5719	05	00D1F	203FFFFFF	A	TABX15	AI, X1	=1	IS THIS TAB POSITION GREATER THAN
5720	05	00D20	31300002	A		CW, X1	X4	POSITION OF TAB CODE.
5721	05	00D21	69200D26			BG	TABX17	
5722	05	00D22	20100001	A		AI, X3	1	IF NOT, TRY NEXT TAB POSITION,
5723	05	00D23	21100010	A		CI, X3	16	IF NOT AT MAX NBR OF TABS.
5724	05	00D24	69100D15			BL	TABX10	
5725	05	00D25	68000D12			B	TABX7	
5726					*			
5727	05	00D26	22500040	A	TABX17	LI, P1	1 1	PUT A BLANK OVER ACTUAL TAB CODE.
5728	05	00D27	75540024	02		STB, P1	CARDIMG, X4	
5729	05	00D28	331002C6	02		MTW, 1	TABCFLAG	TO INDICATE COMPRESS
5730	05	00D29	20200001	A		AI, X4	1	INCREMENT TO NEXT BYTE.
5731	05	00D2A	324000E7	02		LW, X2	RECSIZE	
5732	05	00D2B	204FFFFFF	A		AI, X2	=1	DETERMINE LAST BYTE POSITION.
5733					*			
5734	05	00D2C	38300002	A		SW, X1	X4	COMPUTE NUMBER OF BLANKS TO INSERT.
5735	05	00D2D	68300D0C			BEZ	TABX4	IF ZERO, ITERATE.
5736	05	00D2E	30300004	A		AW, X1	X2	INCREMENT TO NEW LAST BYTE.
5737	05	00D2F	353000E7	02		STW, X1	RECSIZE	SET NEW RECORD SIZE.
5738	05	00D30	331000E7	02		MTW, 1	RECSIZE	
5739	05	00D31	72680024	02	TABX19	LB, P2	CARDIMG, X2	MOVE BYTES UP, STARTING AT TOP,
5740	05	00D32	75580024	02		STB, P1	CARDIMG, X2	BLANKING AS WE GO.
5741	05	00D33	75660024	02		STB, P2	CARDIMG, X1	
5742	05	00D34	203FFFFFF	A		AI, X1	=1	
5743	05	00D35	204FFFFFF	A		AI, X2	=1	
5744	05	00D36	31400002	A		CW, X2	X4	GO DOWN ONLY TO BYTE JUST ABOVE
5745	05	00D37	68100D31			BGE	TABX19	TAB BLANK.
5746					*			
5747	05	00D38	32200003	A		LW, X4	X1	INCREMENT BYTE POSITION TO LAST
5748	05	00D39	20100001	A		AI, X3	1	

```

5749 05 00D3A 68000DOC
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759 05 00D3B
5760 05 00D3B 39000086 02
5761 05 00D3C 691E0000 A
5762 05 00D3D 390002C6 02
5763 05 00D3E 683E0000 A
5764 05 00D3F 02200060 A
5765 05 00D40 0B10019C 02
5766 05 00D41 22100000 A
5767 05 00D42 7232000F N
5768 05 00D44 02200060 A
5769 05 00D46 680E0000 A
5770 05 00D47 7232000F N
5771 05 00D48 6830004C
5772 05 00D49 20100001 A
5773 05 00D4A 21100010 A
5774 05 00D4B 69100047
5775 05 00D4C 201FFFFFF A
5776 05 00D4D 6910006F
5777 05 00D4E 7232000F N
5778 05 00D4F 313000E7 02
5779 05 00D50 6920004C
5780 05 00D51 203FFFFFFE A
5781 05 00D52 22500040 A
5782 05 00D53 71560024 02
5783 05 00D54 6930004C

```

```

B TABX4 MOVED, AND LOOK FOR MORE TAB CODES.
*
*
*****
* ACCORDINGLY, EACH RECORD WRITTEN MUST BE *
* RE-COMPRESSED IN ORDER TO MINIMIZE RAD *
* STORAGE PER RECORD. *
*****
*
*
TABC0MPRESS EQU $
MTW,0 FILETYPE
BLZ 0,LNK
MTW,0 TABCFLAG IF NO COMPRESSION NEEDED, EXIT.
BEZ 0,LNK
PUSH (X3,P2)
LI,X3 0
LB,X1 M:UC+15,X3
BNEZ TABC13
PULL (X3,P2) EXIT.
B 0,LNK
LB,X1 M:UC+15,X3 SKIP TO LAST TAB POSITION+1
BEZ TABC15 IN DCB.
AI,X3 1
CI,X3 16
BL TABC10
AI,X3 =1 MOVE DOWN TO NEXT LOWER TAB
BLZ TABC30 POSITION. IF ALL GONE, EXIT
LB,X1 M:UC+15,X3
CW,X1 RECSIZE DONT PUT ANY TAB CHARACTERS
RG TABC15 PAST END OF RECORD
AI,X1 =2 MAKE INDEX TO NEXT LOWER BYTE.
LI,P1 ' ' IS NEXT LOWER BYTE A BLANK.
CB,P1 CARDIMG,X1
BNE TABC15 IF NOT, WE CAN'T COMPRESS IMAGE.

```

H01 20:44 SEP 08, '75

5784	05	00055	32200001	A		LW,X4	X3		IF BLANK, WE CAN COMPRESS DOWN
5785	05	00056	202FFFFF	A		AI,X4	=1		
5786	05	00057	69100D5A			BLZ	TABC17		TO NEXT LOWER TAB POSITION.
5787	05	00058	7224000F	N		LB,X4	M:UC+15,X4		
5788	05	00059	202FFFFE	A		AI,X4	=2		TAB POSITION,
5789	05	0005A	32400003	A	TABC17	LW,X2	X1		CREATE NEW INDEX,
5790	05	0005B	71580024	02	TABC18	CB,P1	CARDIMG,X2		MOVE IT DOWN TO
5791	05	0005C	69300D60			BNE	TABC20		A NON-BLANK,
5792	05	0005D	204FFFFF	A		AI,X2	=1		
5793	05	0005E	31400002	A		CW,X2	X4		OR TAB BOUNDARY.
5794	05	0005F	69200D5B			BG	TABC18		
5795					*				
5796	05	00060	20400001	A	TABC20	AI,X2	1		MOVE BACK UP TO BLANK.
5797	05	00061	20300001	A		AI,X1	1		MOVE BACK UP TO TAB COLUMN.
5798	05	00062	22600005	A		LI,P2	X'05'		PUT TAB CHARACTER OVER BLANK,
5799	05	00063	75680024	02		STB,P2	CARDIMG,X2		
5800	05	00064	20400001	A		AI,X2	1		INCREMENT, AND CHECK IF MORE SPACE
5801	05	00065	31400003	A		CW,X2	X1		EXISTS BETWEEN INDICES.
5802	05	00066	68300D4C			BE	TABC15		IF NOT, TRY NEXT LOWER TAB.
5803					*				
5804	05	00067	72660024	02	TABC25	LB,P2	CARDIMG,X1		MOVE BYTES DOWN, STARTING AT TAB
5805	05	00068	75680024	02		STB,P2	CARDIMG,X2		COLUMN, AND CONTINUING UP TO END
5806	05	00069	20400001	A		AI,X2	1		OF RECORD.
5807	05	0006A	20300001	A		AI,X1	1		
5808	05	0006B	313000E7	02		CW,X1	RECSIZE		
5809	05	0006C	69100D67			BL	TABC25		
5810					*				
5811	05	0006D	354000E7	02		STW,X2	RECSIZE		SET NEW, SMALLER RECORD SIZE,
5812	05	0006E	68000D4C			B	TABC15		AND GET NEXT TAB.
5813					*				
5814	05	0006F	330001A0	02	TABC30	MTW,0	RPFLAG		PRESERVE RECORD SIZE
5815	05	00070	69300D44			BNEZ	TABC5		NO
5816	05	00071	324001A3	02		LW,X2	TRECSIZE		YES, RESET RECORD SIZE
5817	05	00072	354000E7	02		STW,X2	RECSIZE		
5818	05	00073	75560024	02	TABC35	STB,P1	CARDIMG,X1		BLANK REST OF RECORD
5819	05	00074	20300001	A		AI,X1	1		
5820	05	00075	31300004	A		CW,X1	X2		END OF RECORD

H01 20:44 SEP 08, '75

5821 05 00D76 69100D73
5822 05 00D77 68000D44
5823

BL
B
FIN

TABC35
TABC5

N6
YES, RETURN

PAGE

 * TEST IF EDIT FILE IS ACTIVE *

 *
 *

5830		05 00D78		TESTEDITACTIVE	EQU \$	
5831	05	00D78	330000B6 02	MTW,0	FILETYPE	TEST IF EDIT FILE ACTIVE
5832	05	00D79	691E0000 A	BLZ	0, LNK	NO - EXIT
5833	05	00D7A	0970019C 02	PUSH	LNK	SAVE REG
5834	05	00D7B	6A700B33	BAL, LNK	CLOSE	CLOSE IT
5835	05	00D7C	6A700DC4	BAL, LNK	TYPMSG	TYPE: !..EDIT STOPPED!
5836	05	00D7D	00000284 02	DATA	MSG4	
5837	05	00D7E	22FFFFFF A	LI, D1	=1	SET FILETYPE=1 (NOT OPEN)
5838	05	00D7F	35F000B6 02	STW, D1	FILETYPE	
5839	05	00D80	0870019C 02	PULL	LNK	RESTORE REG
5840	05	00D81	680E0000 A	B	0, LNK	EXIT

PAGE

```

5841
5842 *****
5843 * TYPE CARD IMAGE *
5844 * P1 = SEQ. NUMBER TO TYPE *
5845 *****
5846 *
5847 *
5848 05 00D82 TYPECARD EQU $
5849 05 00D82 02200050 A PUSH (X1, LNK) SAVE REGS
05 00D83 0530019C 02
5850 00000000 DB MODE=1
5851 *S* LW, X2 EBDCLMN SET X2=NUMBER OF SIGNIFICANT CHARS
5852 *S* AI, X2 1
5853 FIN
5854 05 00D84 21500000 A CI, P1 0 IS SEQ # < 0 (MEANING DON'T TYPE IT)
5855 05 00D85 68100D8C BGE TC25
5856 *
5857 00000000 DB MODE=1
5858 *S* LI, P1 72 72 CHARACTERS IF NO SEQ #
5859 *S* TC5 LI, X1 0 INITIALIZE CHARACTER POSITION.
5860 *S* TC10 LB, R0 CARDIMG, X1 SEND CHARACTER
5861 *S* CAL3, 1 0
5862 *S* AI, X1 1 UPDATE CHARACTER POSITION.
5863 *S* AI, X2 =1 IF ALL CHARACTERS GONE, GET OUT.
5864 *S* BLEZ TC15
5865 *S* BDR, P1 TC10
5866 *S* *
5867 *S* BAL, LNK TYPMSG INTERSPERSE WITH CR/LF.
5868 *S* DATA MSGO
5869 *S* LI, P1 72 NOW ITERATE ON 72.
5870 *S* B TC10
5871 *S* *
5872 ELSE
5873 05 00D86 041002D2 02 TC5 CAL1, 1 TPC$FPT
5874 FIN
5875 05 00D87 6A700DC4 TC15 BAL, LNK TYPMSG
5876 05 00D88 0000027A 02 DATA MSGO
    
```

H01 20:44 SEP 08, 175

5877 05 00D89 02200050 A
 05 00D8A 0A30019C 02
 5878 05 00D8B 680E0000 A
 5879
 5880
 5881 05 00D8C 6A7000DB
 5882 05 00D8D 40080000 A
 5883 05 00D8E
 5884
 5885 05 00D8E 68000086

*
*

TC25

S

PULL (X1, LNK)

B 0, LNK

BAL, LNK TYPESEQ
 GEN4 BL, EOM, 0, 0
 D01 MODE=1
 LI, P1 62
 B TC5

RESTORE REGS

EXIT

TYPE SEQ #

62 CHARACTERS ALOWWS FOR SEQ #

H01 20144 SEP 08, '75

280

5919	05	00DA3	25500002	A	TP10	SLS,P1	2		SET P1=X1*BYTE ADDR OF STRING
5920	05	00DA4	32300005	A		LW,X1	P1		
5921	05	00DA5	20300001	A		AI,X1	1		SEARCH DOWN STRING TO FIRST 'C(P)'
5922	05	00DA6	71460000	A		CB,X2	0,X1		
5923	05	00DA7	69300DA5	A		BNE	*-2		
5924	05	00DA8	20300001	A		AI,X1	1		SET X1=ADDR OF CHAR AFTER 'C(P)'
5925	05	00DA9	49E00004	02		BR,D0	XFO		CONVERT COUNT TO EBCDIC (MOD 10)
5926	05	00DAA	75E60000	A		STB,D0	0,X1		AND PUT IN STRING
5927			00000000			D0	MODE=1		
5928				*S*		LB,X2	0,P1		SET X2=LENGTH OF STRING
5929				*S*		AI,P1	1		
5930				*S*		LB,R0	0,P1		GET CHAR FROM STRING
5931				*S*		CAL3,1	0		TYPE IT
5932				*S*		BDR,X2	*-3		LOOP
5933				*S*		LI,R0	CR		TYPE: L/F + C/R
5934				*S*		CAL3,1	0		
5935				*S*		LI,R0	LF		
5936				*S*		CAL3,1	0		
5937						ELSE			
5938	05	00DAB	2550007E	A		SLS,P1	*2		GO BACK TO WORD ADDRESS.
5939	05	00DAC	355002BF	02		STW,P1	DMY\$TPM+1		SET UP ADDRESS FOR TYPMSG.
5940	05	00DAD	6A3002BE	02		BAL,X1	DMY\$TPM		
5941						FIN			
5942	05	00DAE	0870019C	02		PULL	(X1,P1),LNK		
	05	00DAF	02200030	A					
	05	00DB0	0A30019C	02					
5943	05	00DB1	680E0001	A		B	1,LNK		EXIT
5944									
5945									
5946									
5947	05	00DB2	32F00DC3	A	TP20	LW,D1	TPMSG		PUT '-C1' IF TEMPBLCK
5948	05	00DB3	35F0016B	02		STW,D1	TEMPBLCK		
5949	05	00DB4	F2300005	A		LB,X1	*P1		GET LENGTH OF ERROR MSG
5950	05	00DB5	32400003	A		LW,X2	X1		
5951	05	00DB6	20400002	A		AI,X2	2		
5952	05	00DB7	7540016B	02		STB,X2	TEMPBLCK		PUT LENGTH+2 IN TEMPBLCK
5953	05	00DB8	F2F60005	A		LB,D1	*P1,X1		MOVE ERROR MSG TO TEMPBLCK AFTER

*
 * THERE IS MORE THAN ONE COMMAND IN CDT: ADD 'CN' TO ERROR MSG
 *

5954	05	00DB9	75F8016B	02	STB,D1	TEMPBLCK,X2	'=C1'
5955	05	00DBA	204FFFFFF	A	AI,X2	=1	
5956	05	00DBB	64300DB8		BDR,X1	=-3	LOOP
5957	05	00DBC	22300002	A	LI,X1	2	
5958	05	00DBD	F2E600AC	02	LB,D0	*CDTADR,X1	GET # 8F CMND IN CDT
5959	05	00DBE	20E000F0	A	AI,D0	'0'	CONVERT TO EBCDIC
5960	05	00DBF	22300003	A	LI,X1	3	
5961	05	00DC0	75E6016B	02	STB,D0	TEMPBLCK,X1	PUT IT AFTER IC1 TO YIELD FORM;
5962	05	00DC1	2250016B	02	LI,P1	TEMPBLCK	'=CNP1:ERROR MSG'
5963	05	00DC2	68000D9F		B	TP5	GO PROCESS 'P'
5964					*		
5965					*		
5966	05	00DC3	0360C3F1	A	TPMSG	TEXTC	'=C1'

PAGE

```

5967
5968 *****
5969 * TYPE MESSAGE
5970 * WORD AFTER BAL = WORD ADDR OF TEXTC=STRING *
5971 *****
5972 *
5973 *
5974 05 00DC4 TYPMSG EQU *
5975 05 00DC4 02200020 A PUSH (X1,X2),LNK SAVE REGS
05 00DC5 0530019C 02
05 00DC6 0970019C 02
5976 00000000 DB MODE=1
5977 *S* LW,X1 0,LNK SET X1=BYTE ADDR OF STRING
5978 *S* SLS,X1 2 X2=NUMBER OF CHARS TO TYPE
5979 *S* LB,X2 0,X1
5980 *S* AI,X1 1
5981 *S* LB,R0 0,X1 GET CHAR FROM STRING
5982 *S* CI,R0 EOM IS CHAR=EOM
5983 *S* BE TM5 YES = STOP TYPING
5984 *S* CAL3,1 0 TYPE IT
5985 *S* BDR,X2 #=5 LOOP
5986 *S* LI,R0 CR TYPE: L/F + C/R
5987 *S* CAL3,1 0
5988 *S* LI,R0 LF
5989 *S* CAL3,1 0
5990 ELSE
5991 05 00DC7 327E0000 A LW,LNK 0,LNK GET ADDRESS OF MESSAGE AND BYTE
5992 05 00DC8 22300000 A LI,X1 0
5993 05 00DC9 F2400007 A LB,X2 *LNK
5994 *
5995 *
5996 05 00DCA F1380007 A CB,X1 *LNK,X2
5997 05 00DCB 69300DCD BNE #+2
5998 05 00DCC 64400DCA BDR,X2 #-2
5999 *
6000 05 00DCD 22000008 A LI,R0 EOM IF EOM, DO NOT PRINT IT.
6001 05 00DCE F1080007 A CB,R0 *LNK,X2 BUT MARK NOT TO RETURN CARRIAGE.

```

H01 20:44 SEP 08, '75

283

6002	05	00DCF	69300DD2		BNE	TM4	
6003				*			
6004	05	00DD0	204FFFFFF A		AI,X2	=1	EOM FOUND.
6005	05	00DD1	223FFFFFF A		LI,X1	=1	
6006				*			
6007	05	00DD2	041002D7 02	TM4	CAL1,1	TYPMSFFT	
6008	05	00DD3	20300000 A		AI,X1	0	
6009	05	00DD4	69100DD7		BLZ	TM5	
6010				*			
6011	05	00DD5	6A700DC4		BAL, LNK	TYPMSG	YES. CALL RECURSIVELY TO
6012	05	00DD6	0000027A 02		DATA	MSG0	SEND IT OUT.
6013					FIN		
6014	05	00DD7	0870019C 02	TM5	PULL	(X1,X2), LNK	RESTORE REGS.
	05	00DD8	02200020 A				
	05	00DD9	0A30019C 02				
6015	05	00DDA	680E0001 A		B	1, LNK	EXIT

PAGE

```
*****
* TYPE SEQUENCE NUMBER *
* P1 = SEQ. NUMBER TO TYPE *
* WORD AFTER BAL = 4 CHARS TO APPEND TO SEQ # *
*****
*
```

```
6016
6017
6018
6019
6020
6021
6022
6023
6024 05 00DD8 02200040 A
6025 05 00DDC 0540019C 02
6026 05 00DDD 32400007 A
6027 05 00DDE 226005AC 02
6028 05 00DDF 6A700B11
6029 05 00DE0 32E0016C 02
6030 05 00DE1 32F80000 A
6031 05 00DE2 25E00178 A
6032 05 00DE3 7250016C 02
6033 05 00DE4 25500008 A
6034 05 00DE5 49500002 02
6035 05 00DE6 35E0016C 02
6036 05 00DE7 5550016C 02
6037 05 00DE8 35F0016D 02
6038 05 00DE9 32580000 A
6039 05 00DEA 25500018 A
6040 05 00DEB 3550016E 02
6041 05 00DEC 22E00040 A
6042 05 00DED 22F000F0 A
6043 05 00DEE 224005B2 02
6044 05 00DEF 71E80000 A
6045 05 00DF0 69300DF6
6046 05 00DF1 75F80000 A
6047 05 00DF2 20400001 A
6048 05 00DF3 71E80000 A
6049 05 00DF4 69300DF6
6050 05 00DF5 75F80000 A
6051
```

```
TYPESEQ FGU $
PUSH (X2, LNK) SAVE REGS

LW, X2 LNK
LI, P2 BA(TEMPBLCK)
BAL, LNK BINTODEC CONVERT SEQ # TO EBCDIC: ' DDDDDDD'
LW, D0 TEMPBLCK+1 PUT A '.' BETWEEN 4TH AND 5TH
LW, D1 0, X2 DIGITS AND APPEND 4 SPECIFIED
SLD, D0 =8 CHARS TO END
LB, P1 TEMPBLCK+1
SLS, P1 8
BR, P1 KPE
STW, D0 TEMPBLCK+1 PUT THIS BACK IN TEMP BLOCK
STH, P1 TEMPBLCK+1
STW, D1 TEMPBLCK+2
LW, P1 0, X2 GET 4TH SPECIFIED CHAR AND PUT
SLS, P1 24 IN TEMP BLOCK
STW, P1 TEMPBLCK+3
LI, D0 ' ' ADD ZEROS BETWEEN DECIMAL
LI, D1 '0' POINT AND ANY NUMBER
LI, X2 BA(TEMPBLCK)+6
CB, D0 0, X2 IF NECESSARY,
BNE TS10 ZEROS COULD ONLY BE NEEDED
STB, D1 0, X2 IN TEMPBLCK +6 AND +7.
AI, X2 1 SEE IF SECOND ONE IS NECESSARY
CB, D0 0, X2
BNE TS10
STB, D1 0, X2
```

6052									
6053									
6054	05	00DF6	2250000C	A	TS10	LI,P1	12		ATTACH COUNT TO MAKE A TEXTC=STRING
6055	05	00DF7	7550016B	02		STB,P1	TEMPBLCK		
6056	05	00DF8	6A700DC4			BAL,LNK	TYPMSG		TYPE: ,DDD,DDDXXX, WITH LEADING
6057	05	00DF9	0000016B	02		DATA	TEMPBLCK		0'S SUPPRESSED
6058	05	00DFA	02200040	A		PULL	(X2,LNK)		RESTORE REGS
6059	05	00DFB	0A40019C	02					
	05	00DFC	680E0001	A		B	1,LNK		EXIT

* MAKE STRING INTO A TEXTC=STRING AND TYPE

*

PAGE

6060
 6061
 6062
 6063
 6064
 6065
 6066
 6067
 6068 05 UODFU
 6069 05 00DFD 02200040 A
 05 00DFE 0B70019C 02
 6070 05 00DFE 6A700CE6
 6071 05 00E00 6A700C4F
 6072 05 00E01
 6073 05 00E01 04100039 03
 03 00039 06000000 N
 40000000
 03 0003B 00000E07 05
 6074
 6075
 6076
 6077
 6078
 6079 05 00E02 0410003C 03
 03 0003C 11000000 N
 58000030
 03 0003E 00000E07 05
 03 0003F 800000E7 02
 03 00040 00000CBA 02
 6080 05 00E03 02200040 A
 05 00E04 0A70019C 02
 6081 05 00E05 02200000 A
 6082 05 00E06 680E0000 A
 6083
 6084
 6085 05 00E07
 6086 05 00E07 72F0000A A

```

*****
* WRITE RECORD IN COPY FILE *
* P1 = SEQ. NUMBER TO WRITE *
* CC1=1 IF RECORD EXISTS; CC1=0 OTHERWISE *
*****
*
*
WRITE2 FQU $
      PUSH (LNK,P3)
      BAL,LNK SETKEY
      BAL,LNK PUTCR
      DB1 MODE=2
      M;SETDCB F;EB,(ABN,W2$ABN)

MIWRITE FIEB,,
      (WAIT),,
      (KEY,KBUF),,
      (NEWKEY),,
      (ABN,W2$ABN),,
      (SIZE,*RECSIZE)

      PULL (LNK,P3)

      LCI 0 NON-EXISTENT
      B 0,LNK

*
*
W2$ABN RES 0
      LB,D1 P3
    
```

H01 20144 SEP 08, '75

6087	05	00E08	21F00016	A
6088	05	00E09	69300B81	
6089				
6090	05	00E0A	02200040	A
	05	00E0B	0A70019C	02
6091	05	00E0C	02200080	A
6092	05	00E0D	680E0000	A

CI,D1	X:16'
BNE	BAD10
PULL	(LNK,P3)
LCI	8
B	0, LNK

RECORD EXISTED

PAGE

6093
6094
6095
6096
6097
6098
6099
6100
6101 05 00E0E
6102 05 00E0E 02200040 A
05 00E0F 0570019C 02
6103 05 00E10 6A700CE6
6104 05 00E11 6A700C4F
6105 05 00E12
6106 05 00E12 04100041 03
03 00041 06000000 N
40000000
03 00043 00000E18 05
6107
6108
6109
6110
6111
6112 05 00E13 04100044 03
03 00044 11000000 N
58000030
03 00046 00000E18 05
03 00047 800000E7 02
03 00048 000000BA 02
6113 05 00E14 02200040 A
05 00E15 0A70019C 02
6114 05 00E16 02200000 A
6115 05 00E17 680E0000 A
6116
6117
6118
6119 05 00E18

* WRITE NEW RANDOM RECORD *
* P1 = SEQ. NUMBER TO WRITE *
* CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE *

*
* WRITENEWRANDOM EGU \$
PUSH (LNK,P3)
BAL, LNK SETKEY
BAL, LNK PUTCR
DB1 MODE=2
M;SETDCB F:EI, (ABN, WNR\$ABN)
M;WRITE F:EI,,
(WAIT),,
(KEY, KBUF),,
(NEWKEY),,
(SIZE, *RECSIZE),,
(ABN, WNR\$ABN)
PULL (LNK,P3)
LCI 0
B 0, LNK
*
*
*
WNR\$ABN RES 0

HC1 20:44 SEP 08, '75

6120	05	00E18	72F0000A	A
6121	05	00E19	21F00016	A
6122	05	00E1A	69300B81	
6123	05	00E1B	02200040	A
	05	00E1C	0A70019C	02
6124	05	00E1D	02200080	A
6125	05	00E1E	680E0000	A

LB,D1	P3
CI,D1	X'16'
BNE	BADI0
PULL	(LNK,P3)
LCI	8
B	0,LNK

PAGE

6126
6127
6128
6129
6130
6131
6132
6133 05 00E1F
6134 05 00E1F 0970019C 02
6135 05 00E20 6A700CE6
6136 05 00E21 6A700C4F
6137
6138
6139
6140
6141 05 00E22 04100049 03
03 00049 11000000 N
18000050
03 00048 800000E7 02
03 0004C 000000BA 02
6142 05 00E23 0870019C 02
6143 05 00E24 680E0000 A
6144 05 00E2F
6145 05 0000U
05 00E25 00404040 A
05 00E26 06700002 A
05 00E27 00200000 A
05 00E28 00989680 A
05 00E29 00000236 02
05 00E2A 00000710 A
05 00E2B 00FFFFFF A
05 00E2C 02000202 A
05 00E2D 03010202 A

```

*****
* WRITE RANDOM RECORD *
* P1 = SEQ. NUMBER TO WRITE *
*****
*
*
WRITERANDOM EQU $
PUSH LNK
BAL, LNK SETKEY
BAL, LNK PUTCR
MWRITE FILE,,
(WAIT),,
(KEY, KBUF),,
(NEWKEY),,
(SIZE, *RECSIZE)

PULL LNK
B O, LNK
ENDEDITOR EQU $+10
END BEGINEDITOR

```

CONTROL SECTION SUMMARY: 01 00000 PT 0 02 00382 PT 0 03 0004D PT 0 04 00000 PT 1
05 00E2E PT 1

ADDCDTPARAM/05 0081B
ALLFLAG/02 0002U
AR10/05 00986
AR18/05 0099A
BDISPTBL/02 00317
BGD10/05 00001
BLANKBUF/05 00B2C
BLD12/05 00382
BLD5/05 003A0
BPV10/05 0038B
BRK30/05 002A3
BRK55/05 002C9
BRK91/05 002ED
CBRCHTBL/05 0005E
CHARPSN/02 000AV
CLOSE/05 00B33
CMNDTBL/05 00351
CMT30/05 00579
CMT70/05 00589
C8M/00000007
CPY1B/05 003ED
CPY20/05 0042B
CPY35/05 00466
CPY5/05 00406
CPY58/05 0047D
CR3/05 00487
DELETE/05 00B39
DELNXT/02 0019B
DIGITS/02 0001A
DL15/05 00B4D
DL30/05 00B67
DMY\$TYPEPERK/02 002B9
D1/0000000F
EDT5/05 004A6
E8F/0098968U
ERRC11/02 001D0

ALL8K/02 00021
AR10A/05 0098D
AR20/05 0099E
BD10/05 00B17
BINT8DEC/05 00B11
BLANKCNT/02 00022
BLD25/05 003C8
BPFLAG/02 00023
BPV5/05 00386
BRK40/05 002AC
BRK60/05 002CD
BRK99/05 002FC
CDT/02 00048
CHECK1CDTENTRY/05 00834
CLOSE2/05 00B35
CMT10/05 00554
CMT40/05 00580
CM10A/05 00562
C8PYFL/02 000AE
CPY10/05 00410
CPY3/05 00401
CPY36/05 0046A
CPY5A/05 0040B
CPY60/05 0047E
CR5/05 00489
DELETEREFILE/05 00B6A
DF\$ABN/05 00B7A
DLT10/05 004A0
DL17/05 00B58
DMY\$TPM/02 002BE
EDT10/05 004B1
END/00000000
E8M/00000008
ERRC2/02 001A9

ADJINT/05 00287
ALPH/00000006
AR12/05 00993
BADI8/05 00B81
BD20/05 00B28
BL/TEXT
BLD08/05 003A6
BLD30/05 003D2
BPV8FF/05 0038F
BR\$FPT/02 002CD
BRK50/05 002B8
BRK80/05 002D2
BUILDFLAG/02 002C1
CDTADR/02 000AC
CLOSE3/05 00B37
CMT15/05 00568
CMT50/05 00583
CNAMETBL/05 00039
CPY1/05 003E4
CPY15/05 0041A
CPY30/05 00430
CPY37/05 0046C
CPY50/05 00471
CR/0000000D
CT\$FLAG/02 001A1
DELETEREORD/05 00B95
DFLTINCR/02 000B0
DLT5/05 0049D
DL20/05 00B5C
DMY\$TYPECERR/02 002B6
DMYSTKDW/02 0000C
EDT15/05 004B3
ENEDIT8R/05 00E2F
ERRC1/02 001A5
ERRC3/02 001AD

ADJUSTALLFLAG/05 00978
ANLZRIGHT/05 0097C
AR15/05 00996
BADI81/05 00B82
BD30/05 00B23
BLANK/0000000C
BLD10/05 003A8
BLD40/05 003D5
BPV8N/05 0038E
BRK\$KEY/05 0028F
BRK53/05 002C4
BRK90/05 002DC
CARDIMG/02 00024
CFLAG/02 002C2
CHG1STG1CNT/02 0019F
CM/TEXT
CMT20/05 00574
CMT60/05 00586
CNMRTBL/05 00084
CPY1A/05 003EB
CPY2/05 003F2
CPY32/05 0043E
CPY40/05 0046E
CPY56/05 0047C
CRFLAG/02 000AF
CTBLSZ/00000025
DFLTSEQ/000003E8
DL10/05 00B3D
DL25/05 00B63
D0/0000000E
EDT20/05 004B6
E8DCLMN/02 000B1
ERRC10/02 001CC
ERRC4/02 001B1

ERRC5/02 001B6
 ERRC9/02 001C8
 ERRM14/02 001F6
 ERRM18/02 0020C
 ERRM3/02 001DC
 ERRM8/02 001E8
 ERRP11/02 00254
 ERRP14A/02 00261
 ERRP18/02 00276
 ERRP5/02 0023D
 ERRP9/02 0024D
 EXC30/05 00344
 EXC55/05 00351
 F;BUILD/05 00390
 F;EDIT/05 004A3
 F;LNK/000000D
 FC10/05 009C1
 FC30/05 009E2
 FF/000000C
 FILETYPE/02 000B6
 FINISH\$STEP\$LAB/05 006B8
 FIRST\$I:CMND/0000001E
 FIRSTFROM/02 000B8
 FM10B/05 009FF
 FM30/05 00A11
 FND20/05 005A8
 FND40/05 005B8
 FND65A/05 005CC
 GET\$INCREMENT/05 00117
 GETFILEID/05 0083A
 GETNEXTNAME/05 00889
 GF\$PUSH\$SUBR/05 0087F
 GF18/05 00867
 GNTBL1/05 008CD
 GNTYTBL1/05 008CF
 GN35/05 008B2
 GP\$BLSZ/00000005

ERRC6/02 001BB
 ERRM1/02 001D5
 ERRM15/02 001FA
 ERRM19/02 00214
 ERRM4/02 001DF
 ERRORCNT/02 000B2
 ERRP12/02 00258
 ERRP15/02 00265
 ERRP2/02 00232
 ERRP6/02 00241
 EXC10/05 0032B
 EXC40/05 00347
 EXC6/05 00322
 F;COPY/05 003D9
 F;EI/EXT
 F;MERGE/05 004BE
 FC15/05 009CC
 FC35/05 009E4
 FID1ADR/02 000B3
 FINDCOLUMN/05 009A4
 FIRSTSET/02 000B9
 FM15/05 00A04
 FNDTBL1/05 005D2
 FND30/05 005B2
 FND50/05 005C1
 FND70/05 005CF
 GETNEXT\$ERROR/05 008A1
 GF20/05 00874
 GNTBL1SZ/00000004
 GN10/05 00895
 GN45/05 008C5
 GPYTBL/05 00961

ERRC7/02 001BF
 ERRM12/02 001EB
 ERRM16/02 00201
 ERRM20/02 0021D
 ERRM5/02 001E3
 ERRP1/02 0022E
 ERRP13/02 0025D
 ERRP16/02 00269
 ERRP3/02 00236
 ERRP7/02 00245
 EXC15/05 00334
 EXC5/05 00325
 F;BLANK\$PRESERV/05 0037E
 F;CR/05 00481
 F;END/05 004BA
 F;IRP/05 0048C
 FC15A/05 009CD
 FC40/05 009E9
 FID2ADR/02 000B4
 FINDMATCH/05 009EF
 FIRST\$I:CMND/00000004
 FIRST\$R:CMND/0000000B
 FM10/05 009FB
 FM20/05 00A06
 FNDTBL2/05 005D5
 FND32/05 005B4
 FND60/05 005C2
 FRSTCLMN/02 000B7
 GET\$SEQ\$INCR/05 0010E
 GETNEXTPARAM/05 008D4
 GF10/05 00854
 GF30/05 00877
 GNTBL2/05 008D1
 GN25/05 008AA
 GN50/05 008CB
 GP10/05 008EA

ERRC8/02 001C4
 ERRM13/02 001F2
 ERRM17/02 00206
 ERRM21/02 00223
 ERRM6/02 001E6
 ERRP10/02 00250
 ERRP14/02 00261
 ERRP17/02 00271
 ERRP4/02 00239
 ERRP8/02 00249
 EXC20/05 00337
 EXC50/05 0034A
 F;DELETE/05 00497
 F;E8/EXT
 F;TA/05 00534
 FC20/05 009DD
 FC45/05 009EC
 FIELDCNT/02 000B5
 FM10A/05 009FD
 FM20A/05 00A0B
 FNDTYP/05 005AE
 FND35/05 005B8
 FND65/05 005CA
 GET\$COL#\$PAIR/05 001F2
 GETNEXT\$FINISH/05 008BD
 GF15/05 0085C
 GF5/05 00843
 GNTBL2SZ/00000008
 GN30/05 008AC
 GPTBL/05 0095F
 GP20/05 008F4

20:44 SEP 08, 175

GP30/05 008F7	GP30A/05 008FD	GP35/05 00900	GP40/05 00905
GP43/05 00911	GP45/05 00914	GP50/05 00917	GP52/05 0091A
GP52A/05 00920	GP53/05 00928	GP53A/05 00931	GP55/05 00934
GP60/05 00938	GP63/05 0094B	GP66/05 00957	HEXCHAR/02 0000E
I:DELETE/05 0C773	I:DELETE01/05 0077D	I:DELETE02/05 00776	I:FBLLBW\$BY/05 00798
I:JUMP/05 007ED	I:LNK/0000000D	I:NB\$CHANGE/05 00800	
I:OVERWR\$EXTEND/05 0C787		I:OVERWRITE/05 007B1	
I:PRECEDE\$BY/05 007BC	I:SHIFT\$LEFT/05 007A7	I:REVERSE\$BPFLAG/05 00808	I:SHIFT\$RIGHT/05 007CA
I:SET/05 0072D		I:TS\$CMND\$NMR/0000002A	
I:SUBSTITUTE/05 007D4		I:TYPE/05 0080C	I:TYPE\$SUP\$SEQ/05 00810
I:TY\$CMND\$NMR/0000002B		ICNAMETBL/05 000E3	ICNMRTBL/05 000EC
I:TYPEX/05 00814	ICBRCHTBL/05 000EF	ICS50/05 000B6	ICS90/05 000DE
ICS10/05 000AA	ICS20/05 000B2	INS20/05 005ED	ILGL\$SEQ2/05 00129
IC*BL\$Z/00000008	ILGL\$SEMICOLON/05 00035	INS50/05 00610	INS35/05 00604
INSMSG/05 00613	INS10/05 005E9	IBERRC0D/02 002A5	INTFLAG1/02 002C3
INS38/05 0060D	INS40/05 0060F	KBUF/02 000BA	IBERRMSG/02 002A0
INTFLAG2/02 00244	INTG/00000004	LASTCLMN/02 000BB	KPE/02 00002
JMP10/05 007F8	JMP15/05 007FB	LCLETTERS/02 0001E	LASTFR0M/02 000BC
K1/02 00000	K1C/02 00001	LP/TEXT	LETTERS/02 0001C
LASTKEY/02 000BU	LASTSET/02 000BE	M0DE/00000002	LPAR/00000009
LF/00000015	LNK/00000007	MQ20/05 008AB	MASTERPARSER/05 00004
M:TRTN/05 002DB	MASTEREXECUTIVE/05 00301	MRG10/05 004D9	MOVE\$EQ/05 00B97
MAXCLMN/00000084	MAXSEQ/02 000BF	MRG17/05 00503	MQ25/05 008B4
MOVESTRING/05 00A19	MQ10/05 008A5	MRG35/05 00520	MRG13/05 004F3
MQ30/05 008BB	MQ30A/05 008BD	MRG70/05 0052A	MRG20/05 00506
MRG14/05 004F4	MRG15/05 004FD	MSG1/02 0027B	MRG55/05 00522
MRG25/05 00518	MRG30/05 0051C	MSG5/02 00288	MRG80/05 0052C
MRG56/05 00523	MRG65/05 00526	MS10/05 00A26	MSG2/02 0027E
MRG82/05 00530	MSG0/02 0027A	MS5/05 00A1F	MSG6/02 0028C
MSG3/02 00281	MSG4/02 00284	MVE10/05 00619	MS20/05 00A29
MSG7/02 00293	MSG8/02 00299	MVE40/05 00661	MVD:REC:CNT/02 0019E
MS20A/05 00A2C	MS20B/05 00A31	MVE58/05 00683	MVE20/05 0062B
MVEMSG1/02 002A7	MVEMSG2/02 002AD		MVE50/05 0066E
MVE30/05 00646	MVE35/05 00659		NAME/00000001
MVE53/05 00671	MVE56/05 00674		NBCHGFLG/02 000C0
NCG10/05 00805	NEWCDTENTRY/05 00963		B\$ACCT/02 00363
NBPR0MPT\$FPT/02 002CC		B\$ABN/05 00BE4	

@FPT/02 00351
 @EX20/05 00796
 @PENNEW/05 00C36
 @2\$ABN/05 00BFE
 @2\$PASS/02 0037U
 PARSE:BUILD/05 00103
 PARSE:ICR/05 000P7
 PARSE:EDIT/05 00181
 PARSE:IFT/05 001UC
 PARSE:II:CMND\$STKG/05 0008E
 PARSE:IJU/05 00248
 PARSE:MK/05 00210
 PARSE:RP/05 000P7
 PARSE:ITA/05 000P7
 PARSE:TY/05 00269
 PBU30/05 0011C
 PC03/05 0013F
 PDE20/05 001C5
 PFD15/05 001E6
 PMD20/05 00225
 PME30/05 0017A
 PP10/05 00A3F
 PRMBUFSZ/02 000E6
 PR\$CESSCBL#PAIR/05 00A34
 PRS10/05 00033
 PTY15/05 00283
 P1/00000005
 R:DELETE/05 0058C
 R:IFIND\$SEQUENCE/05 00594
 R:INSERT/05 005UB
 R:MOVE\$DELETE/05 00614
 R:RENUMBER/05 00686
 R:ITS\$CMND\$NMR/00000015
 R:TYPE/05 006C3
 READNXTRAND0M/05 00C61
 READTELETYPE/05 00CB2
 RECSIZE/02 000E7

@NAME/02 0035A
 @N\$ABN/05 00C45
 @PEN1/05 00BCC
 @2\$ACCT/02 0037A
 PARAMBUF/02 000C1
 PARSE:CT/05 00192
 PARSE:END/05 0018B
 PARSE:II:CMND\$INTG/05 00096
 PARSE:MD/05 00210
 PARSE:N0/05 0018B
 PARSE:SE/05 001B2
 PARSE:TC/05 00262
 PBU05/05 0012C
 PCM10/05 0019C
 PC05/05 00146
 PDE5/05 001B9
 PFD20/05 00209
 PMD25/05 00226
 PME35/05 0017B
 PP20/05 00A45
 PRN10/05 00231
 PSS10/05 00250
 PTY5/05 00281
 P2/00000006
 R:IFIND\$DELETE/05 00596
 R:INSERT\$SUP\$SEQ/05 005D8
 R:SET\$STEP/05 00699
 R:TYPE\$COMPRESSED/05 006C3
 RELATIVE/02 0037F

@PASS/02 00366
 @PEN/05 00BC8
 @PEN2/05 00BEF
 @2\$FPT/02 00368
 PARAMPSN/02 000E5
 PARSE:CM/05 00193
 PARSE:DE/05 001AF
 PARSE:FD/05 001DC
 PARSE:IN/05 0020C
 PARSE:MERGE/05 00150
 PARSE:RF/05 0025C
 PARSE:SS/05 00245
 PARSE:TS/05 00269
 PBU10/05 00115
 PCM20/05 001AC
 PDE10/05 001BA
 PERIOD/0000000B
 PMD10/05 00219
 PME15/05 00162
 PME40/05 0017E
 PP25/05 00A50
 PRN20/05 0023D
 PR0MPT\$FPT/02 002CA
 PSS20/05 00256
 PUTCR/05 00C4F
 P3/0000000A
 R:IFIND\$TYPE/05 00598
 R:MOVE\$KEEP/05 00616
 R:SET\$STEP\$TYPE/05 0069B
 R:TYPE\$CMND\$NMR/00000016
 READRANDOM/05 00C6D
 READTELETYPE2/05 00CAE
 REOPEN/05 00CE4

@EX10/05 00791
 @PENINIT/05 00C08
 @PEN3/05 00BEB
 @2\$NAME/02 00371
 PARSE:BP/05 000F7
 PARSE:COPY/05 00131
 PARSE:DELETE/05 00181
 PARSE:FS/05 001DC
 PARSE:IS/05 0020C
 PARSE:RN/05 00228
 PARSE:ST/05 00245
 PARSE:TX/05 0025C
 PBU20/05 00121
 PC010/05 00149
 PDE15/05 001BB
 PFD10/05 001E5
 PMD15/05 0021A
 PME20/05 00167
 PME5/05 00161
 PR/TEXT
 PRN30/05 00242
 PR0MPT2\$FPT/02 002CB
 PTY10/05 00282
 PUTCR2/05 00C5F
 R:COMMENTARY/05 00545
 R:LNK/0000000D
 R:TYPE\$SUP\$SEQ/05 006C5
 READSEQUEN/05 00C7F
 REPSEQ/05 0028B

20:44 SEP 08, 1975

RESTART\$EXECUTIVE/05 0030C

RNM10/05 00693
 RPAR/000000UA
 RS\$ABN/05 00C90
 RT\$FPT/02 002DC
 RT10/05 00CD7
 RT3/05 00CCF
 R2/0000000C
 SC/TEXT
 SEG2/000000U3
 SETFLAG/02 000E9
 SETLASTKEY/05 00CEA
 SL10/05 00A87
 SL5/05 00A80
 SPL20/05 006CC
 SR10/05 00ABE
 SR15A/05 00ACE
 SR50/05 00AUD
 SR58/05 00AF9
 SR8/05 00AB3
 STACKSZ/0000007U
 STL20/05 00770
 ST\$PLASTCMD/05 002F7
 SVBPFLAG/02 0016A
 TABC10/05 00D47
 TABC18/05 00D5B
 TABC35/05 00D73
 TABSET/05 00541
 TABX15/05 00D1F
 TABX5/05 00U0D
 TC25/05 00D8C
 TEXTCADR/02 00175
 TPMSG/05 00UC3
 TRECSIZE/02 001A3
 TTYIMGSZ/02 0019A
 TYPE\$I:CMND\$D/05 00OCF
 TYPECARD/05 00D82

RNM13/05 00696
 RP3/05 00492
 RS\$ABNABN/05 00C9B
 RTADDTBL/05 00CE2
 RT15/05 00CD9
 RT9/05 00CD5
 S/FUNC
 SCBL/00000008
 SET\$LBPP/05 00748
 SETKEY/05 00CE6
 SET10/05 00746
 SL20/05 00A8E
 SL5A/05 00A82
 SRS10/05 00A67
 SR12/05 00AC1
 SR20/05 00AD5
 SR52/05 00AE4
 SR60/05 00AFF
 SR8A/05 00AB5
 STEP\$LBPP/05 006AD
 STL30/05 0075A

SV1STSET/02 00169
 TABC13/05 00D49
 TABC20/05 00D60
 TABC5/05 00D44
 TABXFLAG/02 002C7
 TABX17/05 00D26
 TABX7/05 00D12
 TC5/05 00D86
 TM4/05 00DD2
 TP10/05 00DA3
 TSADDR/02 002C8
 TXFLAG/02 001A2
 TYPECERR/05 00D8F

RESUME\$PARSING/05 0001F

RP/TEXT
 RP5/05 00494
 RS\$ABNEBM/05 00CA2
 RTSTBTBL/05 00CDE
 RT17/05 00CDB
 R0/00000000
 SBS10/05 007E1
 SEQ/00000002
 SETADR/02 000E8
 SETK2/05 00CFA
 SHIFTLLEFT/05 00A70
 SL3/05 00A76
 SPL10/05 006B3
 SRS15/05 00A6A
 SR12A/05 00AC3
 SR20A/05 00AD6
 SR52A/05 00AE6
 SR70/05 00B03
 STACK/02 000EA
 STEPFLAG/02 00167
 STL5/05 0074F
 STP10/05 006A9
 TABCFLAG/02 002C6
 TABC15/05 00D4C
 TABC25/05 00D67
 TABERRFLAG/02 002C5
 TABX10/05 00D15
 TABX19/05 00D31
 TA5/05 0053F
 TEMPBLCK/02 0016B
 TM5/05 00DD7
 TP20/05 00DB2
 TS10/05 00DF6
 TYPE\$ALPHA/05 000BD
 TYPE\$I:CMND\$S/05 000D1
 TYPMSG/05 00DC4

RP\$FLAG/02 001A0
 RR\$ERR/05 00C78
 RS\$ABNBUT/05 00CA7
 RTSTWTBL/05 00CE0
 RT5/05 00CB5
 R1/0000000B
 SBS15/05 007E5
 SEQLIM/0098967F
 SETED/05 00A56
 SETK6/05 00CFE
 SHIFTRIGHT/05 00A99
 SL30/05 00A93
 SPL15/05 006BD
 SRS5/05 00A60
 SR15/05 00AC9
 SR5/05 00AA3
 SR55/05 00AEE
 SR72/05 00B0D
 STACKDW/02 0019C
 STL10/05 00760
 STBPCLMN/02 00168
 STRG/00000005
 TABCOMPRESS/05 00D38
 TABC17/05 00D5A
 TABC30/05 00D6F
 TABEXPAND/05 00D01
 TABX12/05 00D19
 TABX4/05 00D0C
 TC15/05 00D87
 TESTEDITACTIVE/05 00D78
 TPC\$FPT/02 002D2
 TP5/05 00D9F
 TTYIMG/02 00176
 TYPE\$BETA/05 000C6
 TYPEPERR/05 00D99

TYPESEQ/05 000D5
 TYP17/05 006F1
 TYP40/05 006FE
 TYP50/05 0070A
 TYP70/05 00718
 TYP82/05 00726
 UTSM1/02 002E1
 UTSM5/02 002F6
 WNR\$ABN/05 00E18
 WRITE2/05 00DFD
 XI:INT0/02 00014
 XI8/02 00018
 XFFFF/02 00006
 X1/00000003
 X4/00000002
 4BLNKS/02 0000A

TYPM\$FPT/02 002D7
 TYP20/05 006F4
 TYP42/05 00701
 TYP55/05 0070F
 TYP72/05 0071B
 TYP90/05 0072C
 UTSM2/02 002E4
 UTSM6/02 00303
 WRITENEWRAND0M/05 00E0E
 W2\$ABN/05 00E07
 XI:M/02 00017
 XI:T0/02 00015
 XFFFFFF/02 00008
 X1FFFF/02 00007
 X800000/02 00009

TYP10/05 006DC
 TYP25/05 006F9
 TYP45/05 00708
 TYP60/05 00713
 TYP75/05 0071D
 T1/00000008
 UTSM3/02 002E5
 UTSM7/02 00307
 XI:C/02 00019
 XI8N/02 00012
 XEGFLAG/02 002C9
 XFF00/02 00005
 X2/00000004
 ZER0:STG:FLG/02 001A4

TYP15/05 006E9
 TYP25A/05 006FB
 TYP5/05 006D9
 TYP65/05 00717
 TYP80/05 00725
 T2/00000009
 UTSM4/02 002E9
 UTSM8/02 0030B
 WRITERAND0M/05 00E1F
 XI:F/02 00016
 XI:0VER/02 00013
 XF/02 00003
 XF0/02 00004
 X3/00000001

* EXTERNAL DEFINITIONS
 BEGINEDITOR/05 00000
 SECT1/02 0000C

SECT5/05 00000

EDITBASE/02 00000

PATCH/02 0031F

* PRIMARY REFERENCES
 JICCBUF JB:CCARS

M:EI

M:EB

M:UC

- * NO SECONDARY REFERENCES
- * NO UNDEFINED SYMBOLS
- * ERROR SEVERITY LEVEL: 0
- * NO ERROR LINES

