```
ASSIGN M:CI,(FILE,EDIT,:DUOCT)
METASYM CI,CN,LB
    .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 5496/M:READ 5520/M:PRECORD 6073/M:SETDCB 6078/M:WRITE 6106/M:SETDCB
6112/M:WRITE

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 1357/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/COM 124/COM 124/COM 124/COM 131/DO 132/DO
133/PSW 135/LCI 135/LCI 136/PSM 146/DO 147/SET 148/DO
149/PLW 151/LCI 151/LCI 152/PLM 162/DO 163/DO 166/SET
166/SET 197/SET 199/GEN 201/ERROR 203/GEN 203/GEN 205/GEN
205/GEN

AFA
199/GEN 202/DO

ALLFLAG
304*DATA 649/STW 699/STW 1764/STW 1872/MTW 3488/MTW 4306/MTW
4308/STW 4389/LW 4444/STW 4458/STW

ALLOK
305*DATA 1830/STW 3516/STW 3568/STW 3624/STW 4422/MTW

ALPH
184*EQU 742/NXTPRM 910/NXTPRM 917/NXTPRM 931/NXTPRM 1068/NXTPRM 1069/LI
1158/NXTPRM 1165/LI 1202/NXTPRM 1216/NXTPRM 1217/LI 4095/LI

ANLZRIGHT
4323*EQU 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
BADI0
5084-RES 5476/BNE 5514/BNE 6088/BNE 6122/BNE
BADI01
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

```
C0M
        185-EQU         1088/NXTPRM     1097/NXTPRM     1111/NXTPRM     1182/NXTPRM     1201/NXTPRM
       1229/NXTPRM     1307/NXTPRM     1367/NXTPRM     1375/NXTPRM     1412/NXTPRM     1423/NXTPRM
       1431/NXTPRM     1485/NXTPRM     1521/NXTPRM     3826/NXTNAM     3839/NXTNAM     3854/NXTNAM
       3856/NXTNAM     3871/NXTNAM
C0PYFL
        312-DATA        1765/MTW        1875/STW        2133/STW        2234/MTW        2268/MTW        2331/MTW
CPY1
       2145-AI         2152/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
CPY4C
       2171/BCS        2319/B          2322-BAL        2476/BCS
CPY5
       2192-BAL        2203/B
```

```
        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$TYPEPERR
        446*EQU         3955/STW        3956/B
DMYSTKDW
        275*DATA         702/LD
DO
         116*EQU        2674/LW         2693/STW        2705/LB         2706/STB        3023/LW         3025/LLM
        3027/CLM        3029/LW         3031/CLM        3033/CLM        3340/STW        3367/CW         3374/CW
        3781/LW         3782/STW        3798/LW         3799/CI         4907/LI         4908/DW         4909/AI
        4910/STB        5138/LI         5139/CB         5148/CB         5155/LB         5156/STB        5161/LI
        5162/STB        5169/LB         5170/STB        5179/LB         5181/STB        5899/LB         5913/LW
        5914/AI         5915/SLS        5925/BR         5926/STB        5958/LB         5959/AI         5961/STB
        6029/LW         6031/SLD        6035/STW        6041/LI         6044/CB         6048/CB
EDIT$DCBT
        475/DATA        487*EQU
EDIT$TCB
        248/DATA        470*EQU
```

```
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
ERR0RCNT
                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/EQU        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 3813/NXTNAM 3818/LI 3837/NXTNAM 3853/NXTNAM 3861/NXTNAM
3869/NXTNAM
ERRP4
399-TEXTC 968/NXTPRM* 979/NXTPRM* 1009/NXTPRM* 1067/NXTPRM 1071/NXTPRM*
1087/NXTPRM* 1110/NXTPRM* 1126/NXTPRM* 1181/NXTPRM* 1200/NXTPRM* 1263/NXTPRM*
1306/NXTPRM* 1314/NXTPRM* 1360/NXTPRM* 1366/NXTPRM* 1374/NXTPRM* 1381/NXTPRM*
1411/NXTPRM* 1422/NXTPRM* 1430/NXTPRM* 1437/NXTPRM* 1484/NXTPRM* 1520/NXTPRM*
1532/NXTPRM* 1578/NXTPRM* 3822/NXTNAM
ERRP5
400-TEXTC 1095/NXTPRM 1205/NXTPRM 1233/NXTPRM 1296/NXTPRM 1339/NXTPRM
1397/NXTPRM 1470/NXTPRM 1486/NXTPRM 1510/NXTPRM 1524/NXTPRM 1559/NXTPRM
1606/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-BAL
EXC40
1839/BLZ 1914-BAL
EXC5
1837/BL 1841/BLE 1860-MTW

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

```
        1945/BAL        2368=EQU
F:TA
        1937/BAL        2616=EQU
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=EQU        4253/DATA
FID1ADR
         317=DATA       2474/STW        2541/LW
FID2ADR
         318=DATA       2511/STW        2543/LW
FIELDCNT
         319=RES        4743/STW        4758/MTW        4776/MTW        4806/MTW        4809/MTW        4852/MTW
FILE
         601/M:DCB       616/M:DCB
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
FINDCOLUMN
        3499/BAL        3517/BAL        3546/BAL        3569/BAL        3585/BAL        3603/BAL        3625/BAL
        3640/BAL        4386=EQU
FINDMATCH
        2817/BAL        4434/BAL        4487=EQU
```

```
FINISH$STEP$LOOP
        3192/B          3222=EQU        3688/B
FIRST$F:CMND
         220=EQU        1834/CI
FIRST$I:CMND
         221=EQU        1358/CI         1567/CI         1840/CI
FIRST$R:CMND
         222=EQU        1836/CI
FIRSTFROM
         322=DATA       2480/STW        2493/STW        2499/LW         2548/LW
FIRSTSET
         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=EQU
FNDTBL2
        2852/EXU        2886=EQU
FNDTYP
        2817=BAL        4532/CI
FND20
        2811=CW         2848/B
FND30
```

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

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

GP66

4203/BNE 4237-LI

HEXCHAR

277-TEXT 5088/LB 5091/LB 5097/LB 5101/LB

I:DELETE

1969/BAL 3484-EQU

I:DELETE01

3489/BLZ 3493/BNE 3498-EQU

I:DELETE02

3491-LB 3494/BDR

I:FOLLOW$BY

1971/BAL 3545-EQU

I:JUMP

1977/BAL 3677-EQU

I:LNK

111-EQU 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-EQU

I:OVERWR$EXTEND

1970/BAL 3515-EQU

I:OVERWRITE

1973/BAL 3584-EQU

I:PRECEDE$BY

1974/BAL 3602-EQU

I:REVERSE$BPFLAG

1979/BAL 3727-EQU

I:SET

1968/BAL 3387-EQU

I:SHIFT$LEFT

1972/BAL 3567-EQU

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$SEMICOLON
         761-EQU        1072/NXTPRM     1089/NXTPRM     1112/NXTPRM     1127/NXTPRM     1183/NXTPRM
        1203/NXTPRM     1231/NXTPRM     1264/NXTPRM     1279/NXTPRM     1315/NXTPRM     1361/NXTPRM
        1424/NXTPRM     1432/NXTPRM     1438/NXTPRM     1533/NXTPRM     1579/NXTPRM     3825/NXTNAM
ILGL$SEQ2
        1099/NXTPRM     1124/BEZ        1132-EQU        1299/NXTPRM     1513/NXTPRM     1562/NXTPRM
INOUT
         603/M:DCB
```

130-EQU 145-EQU 169-LI 198-BAL 235-EQU 417/GEN4 417/GEN4
4018/DATA 4251/DATA 5935/LI 5988/LI

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

```
        5179-LB         5184/BDR
MRG10
        2488/BNE        2499-LW
MRG13
        2530-BAL
MRG14
        2531-LI         2584/B
MRG15
        2534/BEZ        2541-LW
MRG17
        2548-LW
MRG20
        2551-CW         2575/B
MRG25
        2568/BLE        2572-CW
MRG30
        2513/BCS        2578-EQU
MRG35
        2583-LW
MRG55
        2552/BGE        2554/BG         2571/B          2587-SW
MRG56
        2565/BEZ        2589-BAL
MRG65
        2573/BGE        2593-STW
MRG70
        2502/BGE        2504/BG         2598-BAL
MRG80
        2477/BCS        2600-BAL
MRG82
        2514/BCS        2605-BAL
MSG0
         417-GEN4        656/DATA        1677/DATA       1752/DATA       5868/DATA       5876/DATA       6012/DATA
MSG1
         418-TEXTC      2185/DATA
MSG2
         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=BAL
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
MVE1C
       3002=LI
MVE20
       3015/BEZ         3023=LW
MVE30
       3053=CW          3073/B
MVE35
       3056/BLE         3077=BL
```

```
@$FPT
        562-GEN         5212/STW        5214/STW        5219/CAL1       5370/STW        5372/STW        5377/CAL1
       5389/STW        5390/CAL1       5636/CAL1
@$NAME
        571-RES         5215/LI         5373/LI
@$PASS
        575-RES         5217/LI         5375/LI
@EX10
       3529-CW         3533/B
@EX20
       3530/BGE        3537-BAL
@N$ABN
       5369/LI         5384-RES
@NEWKEY
       6140/M;WRITE
@PEN
       2058/BAL        2422/BAL        2512/BAL        5200-EQU
@PENINIT
       5065/BAL        5218/BAL        5268/BAL        5297-EQU        5376/BAL
@PENNEW
       2029/BAL        5367-EQU
@PEN1
       2170/BAL        2259/BAL        2303/BAL        2475/BAL        2542/BAL        5206-EQU
@PEN2
       2177/BAL        2253/BAL        2544/BAL        5258-EQU
@PEN3
       2180/BAL        2298/BAL        5253-PUSH
@UT
        618/M;DCB
@2$ABN
        581/DATA        5262/LI         5276-RES
@2$ACCT
        585-RES         5063/LI         5266/LI
@2$FPT
        579-GEN         5059/STW        5061/STW        5066/CAL1       5261/STW        5263/STW        5269/CAL1
       5281/STW        5282/CAL1
@2$NAME
```

583-RES 5062/LI 5265/LI
82$PASS
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:NB
        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
```

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

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

P2

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$SEQ
        1953/BAL        2896=EQU
R:LNK
         110=EQU        1950/BAL        1951/BAL        1952/BAL        1953/BAL        1954/BAL        1955/BAL
        1956/BAL        1957/BAL        1958/BAL        1959/BAL        1960/BAL        1961/BAL        1962/BAL
        1963/BAL        2698/BEZ*       2774/B*         2863/B*         2868/BNE*       2871/B*         2974/B*
        3100/B*         3106/B*         3130/B*         3134/B*         3154/B*         3160/B*         3166/B*
        3380/B*
R:MOVE$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
```

```
RP
        238*EQU         1744/GEN4       3125/GEN4       4018/DATA
RP$FLAG
        354*DATA        2375/STW        4635/MTW        5814/MTW
RPAR
        188*EQU         3855/NXTNAM     3856/NXTNAM     3857/NXTNAM     3870/NXTNAM     3871/NXTNAM
       3872/NXTNAM
RP3
       2375*STW         2379/BE
RP5
       2373/BNE         2378*CW
RR$ERR
       5460/M:SETDCB    5462/M:READ     5473*RFS
RS$ABN
       5492/M:SETDCB    5496/M:READ     5511*RFS
RS$ABNABN
       5520/M:PRECORD   5521/M:READ     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
```

	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		

339-RES 4493/STW 4494/CW 4506/CW

ST0PLASTCMD

1755/BE 1787-PULL

STP10

3197-LI 3210/BEZ 3239/B

STRG

183-EQU 741/NXTPRM 907/LI 918/NXTPRM 928/LI 965/NXTPRM 966/LI

998/CI 1416/NXTPRM 1417/LI 4122/LI 4400/CI

SVBPFLAG

341-DATA 1814/LW 1996/STW 2004/STW 3469/LW

SV1STSET

340-RES 1900/LW 3279/STW 3403/STW

TAB

2635/M:DEVICE 2636/M:DEVICE 2637/M:DEVICE 2638/M:DEVICE

TABCFLAG

505-DATA 1822/STW 1865/STW 5696/STW 5729/MTW 5762/MTW

TABC0MPRESS

5402/BAL 5759-EQU

TABC10

5770-LB 5774/BL

TABC13

5767/BNEZ 5772-AI

TABC15

5771/BEZ 5775-AI 5779/BG 5783/BNE 5802/BE 5812/B

TABC17

5786/BLZ 5789-LW

TABC18

5790-CB 5794/BG

TABC20

5791/BNE 5796-AI

TABC25

5804-LB 5809/BL

TABC30

5776/BLZ 5814-MTW

TABC35

5818-STB 5821/BL

TABC5

5768*PULL 5815/BNEZ 5822/B

TABERRFLAG
504*DATA 2063/STW 2429/STW 5712/MTW 5714/MTW

TABEXPAND
5674/BAL 5686*EGU

TABSET
2632/EXU 2635*MIDEVICE

TABXFLAG
506*DATA 1823/STW 1866/STW 2772/MTW 2915/MTW 3275/MTW 5689/MTW

TABX10
5699/BE 5707*LB 5724/BL

TABX12
5712*MTW

TABX15
5708/BNEZ 5719*AI

TABX17
5721/BG 5727*LI

TABX19
5739*LB 5745/BGE

TABX4
5697*LI 5735/BEZ 5749/B

TABX5
5698*CB 5702/BL

TABX7
5704*PULL 5711/BNEZ 5713/BNEZ 5717/B 5725/B

TA5
2625/BE 2632*EXU

TC10
5860*LB 5865/BDR 5870/B

TC15
5864/BLEZ 5875*BAL

TC25
5855/BGE 5881*BAL

TC5
5859*LI 5873*CAL1 5885/B

TEMPBLCK
342*RES 5132/STW 5133/LI 5135/LW 5139/CB 5148/CB 5155/LB

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

```
        3300/BL         3308=AI
TYP20
        3285/BE         3314=BAL
TYP25
        3309/BLEZ       3320=BAL
TYP25A
        3317/BLEZ       3322=LI
TYP40
        3291/BAL        3304/BAL        3327=LW
TYP42
        3330=LB         3336/B
TYP45
        3335/BGE        3337=STW
TYP5
        3269/BNEZ       3272/BL         3274/BE         3276=RES
TYP50
        3328/BEZ        3339=LW
TYP55
        3344=STB        3347/BLE
TYP60
        3342/BGE        3349=LI
TYP65
        3354=B
TYP70
        3352/BG         3357=LI
TYP72
        3360=CB         3368/BL
TYP75
        3362=LB         3378/B
TYP80
        3361/BE         3372=AI
TYP82
        3373=AI         3377/BE
TYP90
        3310/B          3318/B          3324/B          3379=EQU
T1
         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/EBR 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/BDR
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/BDR
4815/LW 4819/BDR 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         5015/AI         5016/LW         5024/PULL       5030/LW         5038/PULL       5062/LI
        5215/LI         5265/LI         5310/STB*       5312/STB*       5314/LB*        5373/LI
T2
         105=EQU         922/LI          923/XW          927/STW        1788/LI         1794/LI         2528/STW
        2555/STW        2556/SW         2564/AI         2685/LW         2686/AI         2690/BDR        2701/LI
        2713/CB         2722/STB        2929/LW         2965/CW         3006/LW         3024/LW         3033/CLM
        3040/CW         3055/CW         4488/PUSH       4496/PURGE      4508/PULL       4524/LB         4525/CB
        4535/PULL       4801/LB         4802/STB        4816/LI         4817/STB        4848/LH         4849/AW
        4850/AND        4851/AWM        5063/LI         5216/LI         5266/LI         5299/STW*       5302/STW*
        5306/STW*       5322/STB*       5335/STB*       5374/LI
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;INTB
         280-GEN4       1221/CW
X;M
         284-TEXTC
X;BN
         278-TEXTC      1162/LW         1168/CW
X;BVER
         279-GEN4       1170/CW         2161/CW
X;S
         285-TEXTC
X;TB
         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
XF0
         266-DATA       5925/BR
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	2047/AI	2048/LB	2081/LI	2083/CB	2085/BDR	2134/LI
2135/LB	2137/LI	2138/LB	2148/LB	2150/CB	2156/CB	2158/LI
2159/LB	2166/LI	2167/LB	2174/LI	2175/LB	2186/LI	2187/LB
2207/LI	2208/LB	2211/LI	2212/LB	2214/AI	2215/LB	2250/LI
2251/LB	2255/LI	2256/LB	2267/LI	2269/LB	2271/LB	2272/SLS
2273/AI	2274/AW	2275/LB	2276/SLS	2277/AI	2278/AW	2279/LW
2281/LI	2282/LB	2284/LB	2285/SLS	2286/AI	2287/AW	2288/LB
2289/SLS	2290/AI	2291/AW	2292/LW	2295/LI	2296/LB	2299/LI
2300/LB	2334/LI	2335/LB	2351/LI	2352/LB	2369/LI	2370/LB
2389/LI	2390/LB	2419/LI	2420/LB	2469/LI	2470/STW	2471/LI
2472/LB	2484/AI	2485/LB	2486/AI	2490/LB	2497/AI	2508/AI
2509/LB	2516/AI	2517/LB	2532/AI	2533/LB	2535/AI	2536/LB
2579/AI	2580/LB	2618/LI	2619/LB	2623/LI	2624/CW	2626/BDR
2632/EXU	2647/LI	2648/LB	2650/LI	2651/LB	2677/LW	2678/LB
2680/STB	2692/STB	2699/LI	2705/LB	2707/AI	2713/CB	2715/AI
2766/LI	2767/LB	2794/LI	2795/LB	2802/LI	2803/LB	2805/LI
2910/LI	2911/LB	2916/LI	2917/LB	2919/AI	2920/LB	2945/LI
2947/CB	2949/BDR	3000/LI	3001/STW	3002/LI	3003/LB	3007/LI
3008/LB	3013/LI	3014/LB	3016/AI	3017/LB	3142/LI	3143/LB
3145/LI	3146/LB	3174/LI	3179/LI	3180/STW	3181/STW	3182/LI
3183/LB	3185/LI	3254/LI	3255/STW	3259/LI	3260/LB	3265/AI
3268/LW	3270/LW	3271/CI	3327/LW	3330/LB	3333/AI	3334/CW
3337/STW	3339/LW	3340/STW	3341/CI	3344/STB	3345/AI	3346/CI
3349/LI	3350/LB	3357/LI	3364/STB	3365/AI	3372/AI	3390/LI
3391/LB	3396/LI	3398/LW	3400/AW	3401/STW	3490/LB	3491/LB
3494/BDR	3519/AI	3520/LB	3550/AI	3551/LB	3571/AI	3572/LB
3587/AI	3588/LB	3591/LB	3592/AW	3605/AI	3606/LB	3627/AI
3628/LB	3642/AI	3643/LB	3682/LI	3683/LB	3768/PUSH	3769/LW
3770/STB	3771/AI	3773/STB	3779/LW	3784/BDR	3785/LW	3786/AND
3787/AI	3788/STW	3789/PULL	3811/PUSH	3812/LI	3832/AI	3842/AI
3878/AI	3882/STW	3886/STW	3887/BDR	3888/PULL	3897/AW	3915/PUSH
3921/LI	3922/CB	3924/BDR	3938/LI	3939/CB	3941/BDR	3974/LI
3978/BDR	3990/LB	3995/BDR	4003/PULL	4008/LB	4059/PUSH	4065/LI
4066/CB	4068/BDR	4100/LB	4127/LI	4131/BDR	4154/LI	4200/CI
4208/LI	4392/LI	4393/LB	4396/LI	4398/LB	4399/LI	4402/LB
4410/LI	4417/LI	4418/LB	4431/LI	4432/LB	4447/AI	4488/PUSH
4496/PURGE	4501/LI	4502/LB	4508/PULL	4517/LI	4522/AI	4524/LB

4535/PULL 4547/PUSH 4549/LB 4561/BDR 4565/PULL 4570/AI 4576/BDR
4589/PUSH 4592/LB 4594/AI 4595/LB 4598/AI 4602/AI 4603/LB
4605/AI 4606/LB 4619/PULL 4637/PUSH 4638/LI 4640/CW 4642/BDR
4645/LI 4646/CB 4648/BDR 4652/LI 4654/SLS 4655/AI 4658/STW
4659/AI 4660/STW 4661/PULL 4741/PUSH 4772/LW 4794/LW 4801/LB
4803/AI 4808/PULL 4820/PULL 4882/LI 4883/SW 4903/PUSH 4916/LI
4924/BDR 4926/PULL 5057/PUSH 5068/PULL 5075/LB 5076/CI 5078/PULL
5085/LW 5087/SCS 5088/LB 5090/SLS 5091/LB 5097/LB 5099/STB
5101/LB 5102/STB 5136/LI 5139/CB 5141/AI 5155/LB 5158/AI
5159/CI 5167/CW 5169/LB 5172/AI 5177/LI 5179/LB 5183/AI
5201/PUSH 5207/PUSH 5220/LW 5221/SLS 5222/AND 5223/CI 5225/PULL
5231/PULL 5238/LB 5239/CI 5241/PULL 5253/PUSH 5259/PUSH 5270/PULL
5277/LB 5278/CI 5280/LI 5281/STW 5283/PULL 5298/LW 5299/STW
5300/STW 5302/STW 5303/STW 5305/LW 5306/STW 5307/LW 5308/STW
5311/LB 5312/STB 5320/LB 5322/STB 5334/LI 5335/STB 5344/LB
5346/STB 5353/LI 5354/STB 5368/PUSH 5378/PULL 5385/LB 5386/CI
5388/LI 5389/STW 5391/PULL 5551/PUSH 5557/PUSH 5560/LI 5562/BDR
5566/AI 5569/LW 5575/LB 5576/LB 5578/BDR 5595/LI 5597/CW
5599/CB 5601/AI 5606/LW 5612/PULL 5617/STB 5618/STB 5621/STW
5622/STW 5707/LB 5719/AI 5720/CW 5734/SW 5736/AW 5737/STW
5741/STB 5742/AI 5747/LW 5766/LB 5770/LB 5777/LB 5778/CW
5780/AI 5782/CB 5789/LW 5797/AI 5801/CW 5804/LB 5807/AI
5808/CW 5818/STB 5819/AI 5820/CW 5849/PUSH 5859/LI 5860/LB
5862/AI 5877/PULL 5896/PUSH 5897/LI 5899/LB 5905/PUSH 5920/LW
5921/AI 5922/CB 5924/AI 5926/STB 5940/BAL 5942/PULL 5949/LB
5950/LW 5953/LB 5956/BDR 5957/LI 5958/LB 5960/LI 5961/STB
5975/PUSH 5977/LW 5978/SLS 5979/LB 5980/AI 5981/LB 5992/LI
5996/CB 6005/LI 6008/AI 6014/PULL

X1FFFF

269=DATA 3952/AND

X2

100=EQU 525/PZE* 946/LI 948/STB 1698/LB 1700/LW 1701/LI
1702/CW 1704/BDR 1707/CI 1731/LI 1732/LB 1735/AI 1832/LB
1834/CI 1836/CI 1840/CI 1871/EXU 1885/LB 1991/LB 1992/LW
2039/LB 2040/LW 2048/LB 2049/LW 2078/LW 2079/AI 2080/LB
2088/STB 2159/LB 2160/LW 2208/LB 2209/LW 2215/LB 2216/LW
2352/LB 2353/LW 2370/LB 2371/LW 2648/LB 2649/LW 2651/LB

2652/LW	2679/LI	2680/STB	2684/LI	2687/LB	2689/AI	2700/LW
2706/STB	2708/AI	2711/CI	2720/CI	2722/STB	2723/AI	2767/LB
2768/LW	2769/AI	2770/LW	2795/LB	2796/LW	2799/AI	2800/LW
2911/LB	2912/LW	2920/LB	2921/LW	2942/LW	2943/AI	2944/LB
2952/STB	3003/LB	3004/LW	3005/AI	3006/LW	3008/LB	3009/LW
3010/AI	3011/LW	3017/LB	3018/LW	3143/LB	3144/LW	3146/LB
3147/LW	3183/LB	3184/LW	3260/LB	3261/LW	3262/AI	3263/LW
3329/LI	3331/STB	3332/AI	3337/STW	3343/LI	3344/STB	3350/LB
3351/CI	3359/LI	3360/CB	3362/LB	3363/STB	3366/AI	3367/CW
3373/AI	3374/CW	3376/CB	3391/LB	3392/LW	3393/AI	3394/LW
3551/LB	3552/AW	3553/LB*	3555/LW	3572/LB	3573/LW	3606/LB
3607/AW	3608/LB*	3610/XW	3612/AW	3628/LB	3629/LW	3643/LB
3644/AW	3645/LB*	3664/LW	3683/LB	3684/LW	3780/LI	3781/LW
3782/STW	3783/AI	3788/STW	3898/LI	3899/LW	3901/AI	3925/LI
3942/CI	3965/STB	3966/AI	3976/STB	3977/AI	3979/AI	3980/STB
3981/AI	3982/SLS	3989/STW	3991/LI	3992/CB	3994/AI	4000/SLS
4001/LW	4073/LI	4085/STB	4086/AI	4101/LI	4112/STB	4113/AI
4129/STB	4130/AI	4132/AI	4134/STB	4135/AI	4136/SLS	4155/LI
4165/LI	4185/AI	4193/CI	4196/BDR	4204/CI	4209/LI	4217/LI
4222/CI	4227/LI	4238/CI	4242/LI	4387/PUSH	4398/LB	4400/CI
4402/LB	4403/LW	4418/LB	4419/LW	4451/PULL	4462/PULL	4518/LW
4523/AI	4525/CB	4556/LB	4557/STB	4572/LI	4574/CB	4592/LB
4595/LB	4596/LW	4603/LB	4606/LB	4607/LW	4637/PUSH	4639/LW
4640/CW	4646/CB	4650/CB	4661/PULL	4773/LW	4779/LH	4784/AH
4786/STH	4790/LH	4792/AND	4793/AWM	4794/LW	4795/AW	4802/STB
4804/AI	4808/PULL	4817/STB	4818/AI	4871/LW	4872/AI	4884/LI
4892/LI	5088/LB	5089/STB	5091/LB	5093/STB	5094/LB	5095/SLS
5096/SCS	5097/LB	5100/SLS	5101/LB	5137/LI	5142/BDR	5147/LI
5151/BDR	5178/LI	5184/BDR	5301/LI	5302/STW	5303/STW	5304/LI
5306/STW	5308/STW	5309/LB	5310/STB	5311/LB	5312/STB	5313/BDR
5314/LB	5315/AI	5316/SLS	5317/AW	5318/LB	5320/LB	5321/AI
5322/STB	5323/AI	5325/LB	5326/AI	5327/SLS	5328/AW	5335/STB
5339/LB	5344/LB	5345/AI	5346/STB	5347/AI	5354/STB	5551/PUSH
5552/LI	5557/PUSH	5558/LI	5561/EXU	5565/EXU	5577/EXU	5581/EXU
5606/LW	5612/PULL	5731/LW	5732/AI	5736/AW	5739/LB	5740/STB
5743/AI	5744/CW	5789/LW	5790/CB	5792/AI	5793/CW	5796/AI
5799/STB	5800/AI	5801/CW	5805/STB	5806/AI	5811/STW	5816/LW

5817/STW 5820/CW 5851/LW 5852/AI 5863/AI 5898/LI 5912/LI
5922/CB 5928/LB 5932/BDR 5950/LW 5951/AI 5952/STB 5954/STB
5955/AI 5975/PUSH 5979/LB 5985/BDR 5993/LB 5996/CB 5998/BDR
6001/CB 6004/AI 6014/PULL 6025/PUSH 6026/LW 6030/LW 6038/LW
6043/LI 6044/CB 6046/STB 6047/AI 6048/CB 6050/STB 6058/PULL

X3

97=EQU 674/LI 675/STB 1674/PUSH 1756/PULL 1787/PULL 1790/STS
1793/PULL 1796/STS 2132/LI 2133/STW 2526/LW 2572/CW 2583/LW
2678/LB 2692/STB 3045/LW 3058/CW 3079/CW 5092/LI 5093/STB
5094/LB 5098/AI 5099/STB 5691/PUSH 5692/LW 5693/STW 5694/LI
5704/PULL 5707/LB 5710/AI 5722/AI 5723/CI 5748/AI 5764/PUSH
5765/LI 5766/LB 5768/PULL 5770/LB 5772/AI 5773/CI 5775/AI
5777/LB 5784/LW

X4

98=EQU 986/LI 991/LI 1007/B 2169/LI 2173/LI 2195/CI
2258/LI 2262/LI 2302/LI 2306/LI 2782/LI 2787/LI 2792/LI
2820/EXU 2852/EXU 2897/LI 2909/LI 2933/B 2937/EXU 2990/LI
2995/LI 3061/B 3081/B 3248/LI 3253/LI 3273/CI 3289/B
3302/B 5130/PUSH 5131/LW 5179/LB* 5186/PULL 5695/LI 5696/STW
5698/CB 5700/AI 5701/CW 5720/CW 5728/STB 5730/AI 5734/SW
5744/CW 5747/LW 5784/LW 5785/AI 5787/LB 5787/LB 5788/AI
5793/CW

X800000

271=DATA 3949/CW

ZERO!STG!FLG

358=DATA 3439/MTW 3448/STW 3739/MTW 3746/MTW 4411/MTW

4BLNKS

273=DATA 4639/LW 4936/LW 5298/LW 5559/LW

```
**************************************************************************
*M* EDIT IS A FILE BUILDING/MAINTENANCE UTILITY FOR ON-LINE CP-V USERS
**************************************************************************
*P*      NAME      EDIT
*P*
*P*      PURPOSF   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*
*P*      DESCRIPTION: EDIT IS ORGINIZED IN A HIGHLY MODULAR FASHON.
*P*                UPON ENTRY, 'BEGINEDITOR' PERFORMS INITIALIZATION
*P*                AFTER WHICH 'MASTERPARSER' CONTROLS INPUT COMMAND
```

```
38                          *P*             SCAN OF A LINE OF USER COMMANDS.  FROM A LINE OF
39                          *P*             INPUT COMMAND(S) THE COMMAND DESCRIPTION TABLE (CDT)
40                          *P*             IS BUILT.  ERROR CHECKS ARE MADE AND WARNINGS GIVEN
41                          *P*             TO THE USER IF NECESSARY. 'MASTERPARSER' USES A
42                          *P*             NUMBER OF SUBROUTINES TO BUILD THE CDT: 'GETNAME'
43                          *P*             AND 'GETNEXTPARAM' TO BREAK DOWN TEXT STRINGS;
44                          *P*             'PARSE;I;CMND$INTG' TO PROCESS INTEGER STRINGS;
45                          *P*             'PARSE;I;CMND$STRG' TO PROCESS ALPHABETIC STRINGS
46                          *P*             IN SLASHES; AND ROUTINES OF THE FORM 'PARSE;CMND'
47                          *P*             FOR COMMAND PROCESSING.
48                          *P*
49                          *P*             ON ENCOUNTERING A CARRIAGE RETURN CHARACTER,
50                          *P*             CONTROL IS PASSED TO THE 'MASTEREXECUTIVE' ROUTINE
51                          *P*             TO PERFORM THE COMMANDS WHICH THEN RESIDE IN THE CDT.
52                          *P*             'MASTEREXECUTIVE' SERVES AS A DRIVER FOR COMMAND
53                          *P*             PROCESSING USING 'F;' ROUTINES FOR FILE COMMANDS,
54                          *P*             'R;' ROUTINES FOR RECORD COMMANDS AND 'I;' ROUTINES
55                          *P*             FOR INTRA-RECORD COMMAND PROCESSING.
56                          *P*
57                          *               MODE = 1 FOR BTM VERSION
58                          *                    = 2 FOR UTS VERSION
59                          *
60          00000002        MODE    EQU     2
61                          *
62  02 00000                EDITBASE CSECT  0
63                      *S*         SYSTEM  SIG7FD
64          00000C01                SYSTEM  BPM
65                          *
66          00000000        S       FNAME                   THIS FUNCTION SIMPLY SELECTS
67                                  PROC                    THE FIRST OR SECOND PARAMETER
68                                  LOCAL   A               DEPENDING ON THE ASSEMBLY MODE
69                          A       SET     AF(MODE)
70                                  PEND    A
71                          *
72                          *
73                                  DEF     EDITBASE        DATA AREA FOR EDIT
74                                  DEF     BEGINEDITOR     EDITOR START ADDRESS
```

```
75                                      DEF     SECT1           DATA AREA ADDRESS (FOR GENMDS)
76                                      DEF     SECT5           PURE PROCEDURE ADDR (FOR GENMDS)
77          00000001                    DB      MODE=2
78                                      DEF     PATCH           PATCH AREA (FOR GENMDS)
79                                      REF     J:CCBUF         TTY BUFFER PASSED FROM TEL
80                                      REF     M:UC            DCB FOR USER TERMINAL
81                                      REF     M:EI            EDIT INPUT DCB
82                                      REF     M:EO            EDIT OUTPUT DCB
83                                      REF     JB:CCARS        BYTE COUNT OF STRING IN J:CCBUF
84                                      ELSE
85                          *S*         DEF     F:EI            BTM DCB (INPUT)
86                          *S*         DEF     F:EO            BTM DCB (OUTPUT)
87                                      FIN
88                              *
```

```
                              PAGE
                   *************************
                   *  REGISTER ALLOCATION  *
                   *************************
                   *
                   *
                   *  REGISTERS 1-13 MUST BE PRESERVED BY ANY SUBR WHICH USES THEM
                   *
     00000001      X3        EQU     1
     00000002      X4        EQU     2
     00000003      X1        EQU     3
     00000004      X2        EQU     4
     00000005      P1        EQU     5
     00000006      P2        EQU     6
     00000007      LNK       EQU     7
     00000008      T1        EQU     8
     00000009      T2        EQU     9
     0000000A      P3        EQU     10
     0000000B      R1        EQU     11
     0000000C      R2        EQU     12
     0000000D      F:LNK     EQU     13
     0000000D      R:LNK     EQU     13
     0000000D      I:LNK     EQU     13
                   *
                   *  REGISTERS 0,14-15 ARE NEVER SAVED BY SUBRS
                   *
     00000000      R0        EQU     0
     0000000E      D0        EQU     14
     0000000F      D1        EQU     15
```

```
                                PAGE
                        ************************
                        *  SYSTEM PROCEDURES   *
                        ************************
                        *
                        *
GEN4        COM,8,8,8,8     AF(1),AF(2),AF(3),AF(4)
                        *
                        *
00000000    PUSH        CNAME
                        PROC
                        LOCAL       I
                        LF          EQU         $
                        I           DO          NUM(AF)
                        DO          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
                        *
                        *
00000000    PULL        CNAME
                        PROC
                        LOCAL       I,K
                        LF          EQU         $
                        I           DO          NUM(AF)
                        K           SET         NUM(AF)=I+1
                        DO          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
```

```
155                                     PEND
156                         *
157                         *
158         00000C0U        PURGE       CNAME
159                                     PR0C
160                                     L0CAL     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                                 PAGE
173                         ************************
174                         *  PARSER PROCEDURES   *
175                         ************************
176                         *
177                         *
178          00000000       END      EQU      0
179          00000001       NAME     EQU      1
180          00000002       SEQ      EQU      2
181          00000003       SEQ2     EQU      3
182          00000004       INTG     EQU      4
183          00000005       STRG     EQU      5
184          00000006       ALPH     EQU      6
185          00000007       COM      EQU      7
186          00000008       SCOL     EQU      8
187          00000009       LPAR     EQU      9
188          0000000A       RPAR     EQU      10
189          0000000B       PERIOD   EQU      11                  UTS FILE SEPARATOR
190          0000000C       BLANK    EQU      12
191                         *
192                         *
193          FR             NXTNAM   CNAME    GETNEXTNAME
194          FR             NXTPRM   CNAME    GETNEXTPARAM
195                                   PROC
196                                   LOCAL    I,N
197                         N         SET      NUM(AF)-1
198                         LF        BAL,LNK  NAME(1)
199                                   GEN,8,1,23 N,AFA(1),AF(1)
200                         I         DO       N
201                                   ERROR,1,NUM(AF(I+1)) -2 !ILGL SYNTAX!
202                                   DO       AFA(I+1,2)-1
203                                   GEN,8,24 AF(I+1,1),$+N-I+AF(I+1,2)+1
204                                   ELSE
205                                   GEN,8,24 AF(I+1,1),AF(I+1,2)
206                                   FIN
207                                   FIN
208                                   PEND
```

```
                                PAGE
                        ****************************
                        *  ADJUSTABLE PARAMETERS   *
                        ****************************
                        *
                        *
000003E8                DFLTSEQ  EQU     1000            DEFAULT STARTING SEQ. #
00000008                MAXCLMN  EQU     140
00989677                SEQLIM   EQU     9999999         FOR MAX. SEQ. NO.
0000007D                STACKSZ  EQU     125             SIZE OF TEMP STACK
                        *
00000004                FIRST$F:CMND     EQU 4
0000001E                FIRST$I:CMND     EQU 30
00000000                FIRST$R:CMND     EQU 11
0000002A                I:TS$CMND$NMR    EQU 42
0000002B                I:TY$CMND$NMR    EQU 43
00000015                R:TS$CMND$NMR    EQU 21
00000016                R:TY$CMND$NMR    EQU 22
                        *
                        *
TEXT                    BL       EQU     ' '
TEXT                    CM       EQU     ','
00000000                CR       EQU     S(X'15',X'0D')
00989680                E8F      EQU     10000000
00000008                E8M      EQU     X'08'
0000000C                FF       EQU     X'0C'
00000015                LF       EQU     S(X'25',X'15')
TEXT                    PR       EQU     '.'
TEXT                    LP       EQU     '('
TEXT                    RP       EQU     ')'
TEXT                    SC       EQU     ';'
```

```
240                              PAGE
241                     ****************************************
242                     *  EDIT/BTM INTERFACE CONTROL BLOCK    *
243                     ****************************************
244                     *
245                     *
246          00000000                   DC         MODE=1
247                     *S*             ORG        EDITBASE
248                     *S*             DATA       EDIT$TCB
249                     *S*             DATA,16    0,0
250                     *S*             DATA       BEGINEDITOR
251                     *S*             DC1        X'40'-10
252                     *S*             DATA       0
253                                     FIN
```

```
254                                     PAGE
255                                     ********************
256                                     *  CONSTANT DATA   *
257                                     ********************
258                                     *
259                                     *
260          02 00000                   SECT1     EQU     $
261    02 00000     00000001 A          K1        DATA    1
262    02 00001     0000000A A          K10       DATA    10
263    02 00002     0000004B A          KPE       DATA    '.'
264                                     *
265    02 00003     0000000F A          XF        DATA    X'F'
266    02 00004     000000F0 A          XF0       DATA    X'F0'
267    02 00005     0000FF00 A          XFF00     DATA    X'FF00'
268    02 00006     0000FFFF A          XFFFF     DATA    X'FFFF'
269    02 00007     0001FFFF A          X1FFFF    DATA    X'1FFFF'
270    02 00008     00FFFFFF A          XFFFFFF   DATA    X'FFFFFF'
271    02 00009     00800000 A          X800000   DATA    X'800000'
272                                     *
273    02 0000A     40404040 A          4BLNKS    DATA    '    '
274                                               BOUND   8
275    02 0000C     000000EA            DMYSTKDW  DATA    STACK
276    02 0000D     007D     A                    DATA,2  STACKSZ,0
       02 0000D 2   0000     A
277    02 0000E     F0F1F2F3 A          HEXCHAR   TEXT    '0123456789ABCDEF'
       02 0000F     F4F5F6F7 A
       02 00010     F8F9C1C2 A
       02 00011     C3C4C5C6 A
278    02 00012     02D6D540 A          X'ON      TEXTC   'ON'
279    02 00013     04D6E5C5 A          X'OVER    GEN4    4,'O','V','E'
280    02 00014     04C9D5E3 A          X'INTO    GEN4    4,'I','N','T'
281    02 00015     02E3D640 A          X'TO      TEXTC    'TO'
282          00000001                             DO      MODE=2
283    02 00016     01C64040 A          X'F       TEXTC   'F'
284    02 00017     01D44040 A          X'M       TEXTC   'M'
285    02 00018     01E24040 A          X'S       TEXTC   'S'
286    02 00019     01C34040 A          X'C       TEXTC   'C'
```

```
287                                 *
288                                         FIN
289                                 *
290                                 *  SPECIAL LIMITS
291                                 *
292                                         BOUND     8
293   02 0001A    000000F0 A    DIGITS   DATA      '0','9'
      02 0001B    000000F9 A
294   02 0001C    000000C1 A    LETTERS  DATA      'A','Z'
      02 0001D    000000E9 A
295         00000001                    DB        MODE=2
296   02 0001E    00000081 A    LCLETTERS DATA     X'81',X'A9'
      02 0001F    000000A9 A
297                                         FIN
```

```
298                                    PAGE
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               EODCLMN  RES      1           GLOBAL: COL.# OF LAST NON-BLANK
316   02 000B2               ERRORCNT 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               FIELDCNT 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 'FROM' 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   NOCHGFLG 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.
```

```
 335   02 000E8                   SETADR    RES       1                     GLOBAL; ADR OF LAST SET CMND IN CDT
 336   02 000E9   00000000 A      SETFLAG   DATA      0                     GLOBAL: ON(1) IF SET CMND ACTIVE
 337   02 000EA                   STACK     RES       STACKSZ               GLOBAL: STACK USED FOR PUSH/PULL
 338   02 00167   00000000 A      STEPFLAG  DATA      0                     GLOBAL; ON(1) IF STEP CMND ACTIVE
 339   02 00168                   STOPCLMN  RES       1                     FINDMTCH; COL. # TO STOP MATCHING AT
 340   02 00169                   SV1STSET  RES       1                     GLOBAL: INITIAL 1ST SEQ # FOR SET
 341   02 0016A   00000000 A      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                     FINDMTCH: 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 A      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       A              DATA,2    STACKSZ,0
       02 0019D 2 0000       A
 352   02 0019E   00000000 A      MVD:REC:CNT DATA    0                     COUNT OF RECORDS MOVED
 353   02 0019F   00000000 A      CHG:STG:CNT DATA    0                     COUNT OF STRINGS CHANGED
 354   02 001A0   00000001 A      RP$FLAG   DATA      1                     RECORD SIZE PRESERVATION (OFF)
 355   02 001A1   00000000 A      CT$FLAG   DATA      0                     TYPE BEFORE 'CM' INPUT
 356   02 001A2   00000000 A      TXFLAG    DATA      0                     TX RECORD CHANGED FLAG
 357   02 001A3   00000000 A      TRECSIZE  DATA      0                     TAB RECORD SIZE (FOR RP=ON)
 358   02 001A4   00000000 A      ZERO:STG:FLG DATA   0                     FLAG TO INDICATE ZERO STRINGS
```

```
                                          PAGE
                                          ********************
                                          *  ERROR MESSAGES  *
                                          ********************
                                          *
                                          *
02 001A5     0D6060C3 A   ERRC1     TEXTC     '==C1;OVERFLOW'
02 001A6     F17AD6E5 A
02 001A7     C5D9C6D3 A
02 001A8     D6E64040 A
02 001A9     0E6060C3 A   ERRC2     TEXTC     '==C1;UNDERFLOW'
02 001AA     F17AE4D5 A
02 001AB     C4C5D9C6 A
02 001AC     D3D6E640 A
02 001AD     0F60C3F1 A   ERRC3     TEXTC     '=C1:NO SUCH REC'
02 001AE     7AD5D640 A
02 001AF     E2E4C3C8 A
02 001B0     40D9C5C3 A
02 001B1     1260C3F1 A   ERRC4     TEXTC     '=C1:CMND ILGL HERE'
02 001B2     7AC3D4D5 A
02 001B3     C440C9D3 A
02 001B4     C7D340C8 A
02 001B5     C5D9C540 A
02 001B6     116060C3 A   ERRC5     TEXTC     '==C1:NO SUCH STRG'
02 001B7     F17AD5D6 A
02 001B8     40E2E4C3 A
02 001B9     C840E2E3 A
02 001BA     D9C74040 A
02 001BB     0E6060C3 A   ERRC6     TEXTC     '==C1:COL>LIMIT'
02 001BC     F17AC3D6 A
02 001BD     D36ED3C9 A
02 001BE     D4C9E340 A
02 001BF     126060C3 A   ERRC7     TEXTC     '==C1:''ALL'' IGNORED'
02 001C0     F17A7DC1 A
02 001C1     D3D37D40 A
02 001C2     C9C7D5D6 A
02 001C3     D9C5C440 A
```

```
372  02 001C4  0D60C3F1 A  ERRC8   TEXTC  !-C1:UNKN CMND!
     02 001C5  7AE4D5D2 A
     02 001C6  D540C3D4 A
     02 001C7  D5C44040 A
373  02 001C8  0F60C3F1 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:C0L<LIMIT!
     02 001CD  F17AC3D6 A
     02 001CE  D34CD3C9 A
     02 001CF  D4C9E340 A
375  02 001D0  1260C2C1 A  ERRC11  TEXTC  !-BAD C0L. N0. PAIR!
     02 001D1  C440C3D6 A
     02 001D2  D34B40D5 A
     02 001D3  D64B40D7 A
     02 001D4  C1C9D940 A
376                        *
377                        *
378  02 001D5  1860C0C5 A  ERRM1   TEXTC  !--E0F HIT AFTER YYYY.YYY!
     02 001D6  D6C640C8 A
     02 001D7  C9E340C1 A
     02 001D8  C6E3C5D9 A
     02 001D9  40E8E8E8 A
     02 001DA  E84BE8E8 A
     02 001DB  E8404040 A
379  02 001DC  0A60C0D6 A  ERRM3   TEXTC  !--0VERFL0W!
     02 001DD  E5C5D9C6 A
     02 001DE  D3D6E640 A
380  02 001DF  0660D9D5 A  ERRM4   TEXTC  !-RNG 0VERLAP!
     02 001E0  C740D6E5 A
     02 001E1  C5D9D3C1 A
     02 001E2  D7404040 A
381  02 001E3  0B60D5D6 A  ERRM5   TEXTC  !-N0T 0N/0FF!
     02 001E4  E340D6D5 A
     02 001E5  61D6C6C6 A
382  02 001E6  0660C0D5 A  ERRM6   TEXTC  !--N0NE!
```

```
                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    5B40D4E4 A
                02 001F0    E2E340C3 A
                02 001F1    D6D7E840 A
385             02 001F2    0B60D5D6 A   ERRM13    TEXTC     '=NO FILE NAMED'
                02 001F3    40C6C9D3 A
                02 001F4    C540D5C1 A
                02 001F5    D4C5C440 A
386             02 001F6    0960D5D6 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    1960D5D6 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
```

```
       02 0020B    C5E8C5C4 A
391    02 0020C    1660D4C5 A   ERRM18   TEXTC  '-MERGE DESTINATION NOT KEYED'
       02 0020D    D9C7C540 A
       02 0020E    C4C5E2E3 A
       02 0020F    C9D5C1E3 A
       02 00210    C9D6D540 A
       02 00211    D5D6E340 A
       02 00212    D2C5E8C5 A
       02 00213    C4404040 A
392    02 00214    2360E2D6 A   ERRM19   TEXTC    '-SORRY... NO PASSWORD ALLOWED HERE.'
       02 00215    D9D9E84B 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    2960C3C1 A   ERRM21   TEXTC     '-CAN NOT DELETE ALL OCCRRENCES OF BLANKS.'
       02 00224    D540D5D6 A
       02 00225    E340C4C5 A
       02 00226    D3C5E3C5 A
       02 00227    40C1D3D3 A
       02 00228    40D6C3C3 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    0660D7F2 A    ERRP2     TEXTC     !-P2;REC EXISTS!
                02 00233    7AD9C5C3 A
                02 00234    40C5F7C9 A
                02 00235    E2E3F240 A
     398        02 00236    0960D7F1 A    ERRP3     TEXTC     !-P1;BAD FID!
                02 00237    7AC2C1C4 A
                02 00238    40C6C9C4 A
     399        02 00239    0F60D7F1 A    ERRP4     TEXTC     !-P1;ILGL SYNTAX!
                02 0023A    7AC9D3C7 A
                02 0023B    D340F2E8 A
                02 0023C    D5E3C1E7 A
     400        02 0023D    0760D7F1 A    ERRP5     TEXTC     !-P1;NOT SEQ #!
                02 0023E    7AD5D6E3 A
                02 0023F    40E2C5D8 A
                02 00240    407B4040 A
     401        02 00241    0660D7F1 A    ERRP6     TEXTC     !-P1;NOT INCR!
                02 00242    7AD5D6E3 A
                02 00243    40C9D5C3 A
                02 00244    D9404040 A
     402        02 00245    0760D7F1 A    ERRP7     TEXTC     !-P1;NOT COL #!
                02 00246    7AD5D6E3 A
                02 00247    40C3D6D3 A
                02 00248    407B4040 A
     403        02 00249    0660D7F1 A    ERRP8     TEXTC     !-P1;NOT STRG!
                02 0024A    7AD5D6E3 A
                02 0024B    40E2F3D9 A
                02 0024C    C7404040 A
     404        02 0024D    0960D7F1 A    ERRP9     TEXTC     !-P1;NOT CNT!
                02 0024E    7AD5D6E3 A
                02 0024F    40C3D5E3 A
     405        02 00250    0660D7F1 A    ERRP10    TEXTC     !-P1;ILGL SEQ #!
                02 00251    7AC9D3C7 A
                02 00252    D340E2C5 A
                02 00253    D8407B40 A
```

```
406   02 00254   0U60D7F1 A   ERRP11   TEXTC   !-P1;SEQ2<SEQ1!
      02 00255   7AE2C5D8 A
      02 00256   F24CF2C5 A
      02 00257   D8F14040 A
407   02 00258   1U60D7F1 A   ERRP12   TEXTC   !-P1;NO SUCH FILE!
      02 00259   7AD5D640 A
      02 0025A   E2E4C3C8 A
      02 0025B   40C6C9D3 A
      02 0025C   C5404040 A
408   02 0025D   0P60D7F2 A   ERRP13   TEXTC   !-P2;FILE EXISTS!
      02 0025E   7AC6C9D3 A
      02 0025F   C540C5E7 A
      02 00260   C9E2F3E2 A
409   02 00261   0U60D7F2 A   ERRP14   TEXTC   !-P2;COL ERROR!
      02 00262   7AC3D6D3 A
      02 00263   40C5D9D9 A
      02 00264   D6D94040 A
410         02 00261         ERRP14A  EQU     ERRP14
411   02 00265   0U60D7F1 A   ERRP15   TEXTC   !-P1;ILGL STRG!
      02 00266   7AC9D3C7 A
      02 00267   D340F2E3 A
      02 00268   D9C74040 A
412   02 00269   1U60D7F1 A   ERRP16   TEXTC   !-P1;FILE NOT KEYED & P3 NULL!
      02 0026A   7AC6C9D3 A
      02 0026B   C540D5D6 A
      02 0026C   E340D2C5 A
      02 0026D   E8C5C440 A
      02 0026E   5040D7F3 A
      02 0026F   40D5F4D3 A
      02 00270   D3404040 A
413   02 00271   1160D7F1 A   ERRP17   TEXTC   !-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   !-P1;NULL STRG!
      02 00277   7AD5F4D3 A
```

```
                02 00278    D340F2E3 A
                02 00279    D9C74040 A
     415                             *
     416                             *
     417        02 0027A    03150800 A   MSG0    GEN4    3,S(CR,LF),S(LF,E0M),S(E0M,0)
     418        02 0027B    094B4BC3 A   MSG1    TEXTC   !..C0PYING!
                02 0027C    D6D7E8C9 A
                02 0027D    D5C74040 A
     419        02 0027E    0B4B4BC3 A   MSG2    TEXTC   !..C0PY D0NE!
                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 ST0PPED!
                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    184040 40 A  MSG6    TEXTC   !          REC0RDS 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 REC0RDS M0VED!
                02 00294    F0F0F0F0 A
                02 00295    F040D9C5 A
                02 00296    C3D6D9C4 A
                02 00297    E240D4D6 A
                02 00298    E5C5C440 A
     425        02 00299    184040 40 A  MSG8    TEXTC   !          STRINGS CHANGED!
                02 0029A    40404040 A
```

```
      02 0029B    40404E2E3 A
      02 0029C    D9C9D5C7 A
      02 0029D    E2404C3C8 A
      02 0029E    C1D5C7C5 A
      02 0029F    C4404040 A
426                           *
427                           *
428   02 002A0    1B60C2C1 A  I8ERRMSG DATA     X'0060C2C1'+((I8ERRC8D+2*I8ERRMSG)*4+1)**24
429   02 002A1    C440C961 A            TEXT     'D I/8; ABN C8DE'
      02 002A2    D65E40C1 A
      02 002A3    C2D540C3 A
      02 002A4    D6C4C540 A
430   02 002A5    40404040 A  I8ERRC8D TEXT     '         '         ABN C8DE=SUBC8DE PUT HERE
      02 002A6    40404040 A
431                           *
432                           *
433   02 002A7    0A6060C4 A  MVEMSG1  TEXTC    '=DONE AT '
      02 002A8    D6D5C540 A
      02 002A9    C1E34040 A
434   02 002AA                          RES      3
435   02 002AD    0A6060C3 A  MVEMSG2  TEXTC    '=CUT8FF AT '
      02 002AE    E4E3D6C6 A
      02 002AF    C64CC1E3 A
      02 002B0    40404040 A
436   02 002B1                          RES      5
437                           *
438                           *  DUMMY CALLS F8R TYPECERR AND TYPEPERR
439                           *
440         02 002B6          DMY$TYPECERR       EQU $
441   02 002B6    6A7008D8F 05          BAL,LNK  TYPECERR          TYPE ERRCN
442   02 002B7    00000000 A            DATA     0
443   02 002B8    68000004 05           B        MASTERPARSER      G8 T8 PARSER
444                           *
445                           *
446         02 002B9          DMY$TYPEPERR       EQU $
447   02 002B9    6A7008D99 05          BAL,LNK  TYPEPERR          TYPE ERRPN
448   02 002BA    00000000 A            DATA     0
```

```
             00000001                   DO         MODE=2
   02 002BB     330002C1                   MTW,0      BUILDFLAG          WAS ENTRY FROM TEL ( B)
   02 002BC     69300004 05                BNEZ       MASTERPARSER       NO
   02 002BD     680004BA 05                B          F:END              YES, EXIT
                                           ELSE
                             *S*           B          MASTERPARSER       CONTINUE
                                           FIN
                                *
                                *
   02 002BE     6A700DC4 05  DMY$TPM       BAL,LNK    TYPEMSG            TYPE ERRP(C)N
   02 002BF     00000000 A                 DATA       0
   02 002C0     68060000 A                 B          0,X1               RETURN
                                *
```

```
                                   PAGE
                            ************************************
                            *  EDIT TASK CONTROL BLOCK, ETC.   *
                            ************************************
                            *
                            *
          00000000                 DO         MODE=1
                        *S*        BOUND      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*        DO1        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
                                   ELSE
                            *
                            ****************************************************
                            *        UTS INTERFACE PARAMETERS AND MESSAGES.    *
                            ****************************************************
```

```
                                 *
02 002C1   00000000 A   BUILDFLAG DATA     0                  IF ZERO, ENTERED BY  BUILD
02 002C2   00000000 A   CFLAG     DATA
02 002C3   FFFFFFFF A   INTFLAG1  DATA     -1                 /INTERRUPT SEQ INDICATORS FOR
02 002C4   FFFFFFFF A   INTFLAG2  DATA     -1                 /THOSE COMMANDS WHICH DISPLAY.
02 002C5   00000000 A   TABERRFLAG         DATA
02 002C6   00000001 A   TABCFLAG           DATA   1
02 002C7   00000001 A   TABXFLAG          DATA 1
02 002C8   00000000 A   TSADDR    DATA     0                  TEMP STACK ADDRESS
02 002C9   FFFFFFFF A   XEQFLAG   DATA     -1                 MINUS ONE IF NOT IN EXECUTION.
      02 002CA          PROMPT$FPT         EQU $
02 002CA   2C00005C A             GEN,8,24 X'2C','*'
      02 002CB          PROMPT2$FPT         EQU   $
02 002CB   2C00004B A             GEN,8,24  X'2C','.'
      02 002CC          NOPROMPT$FPT       EQU $
02 002CC   2C000000 A             GEN,8,24 X'2C',0
02 002CD   10000000 N   BR$FPT    GEN,8,24 X'10',M:UC
02 002CE   34000010 A             DATA     X'34000010',CFLAG,1,0
02 002CF   000002C2
02 002D0   00000001 A
02 002D1   00000000 A
02 002D2   11000000 N   TPC$FPT   GEN,8,24 X'11',M:UC
02 002D3   34000010 A             DATA     X'34000010'
02 002D4   00000024               DATA     CARDIMG
02 002D5   800000E7               PZE      *RECSIZE
02 002D6   00000000 A             DATA     0
02 002D7   11000000 N   TYPM$FPT  GEN,8,24 X'11',M:UC
02 002D8   34000010 A             DATA     X'34000010'
02 002D9   80000007 A             PZE      *LNK
02 002DA   80000004 A             PZE      *X2
02 002DB   00000001 A             DATA      1
02 002DC   10000000 N   RT$FPT    GEN,8,24 X'10',M:UC
02 002DD   34000010 A             DATA     X'34000010'
02 002DE   80000003 A             PZE      *X1
02 002DF   0000008C A             DATA     MAXCLMN
02 002E0   00000000 A             DATA     0
                                 *
```

```
533                                     *
534   02 002E1   09C5C4C9 A   UTSM1   TEXTC   'EDIT HERE'
      02 002E2   E34CC8C5 A
      02 002E3   D9C54040 A
535   02 002E4   025C0840 A   UTSM2   TEXTC   '*H'                 * + E0M
536   02 002E5   0C60D5D6 A   UTSM3   TEXTC   '-N0T F/M/S/C'
      02 002E6   E34CC661 A
      02 002E7   D461F261 A
      02 002E8   C3404040 A
537   02 002E9   2460A0C9 A   UTSM4   TEXTC   '--INTRA-REC0RD C0MMAND 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   1760A0C3 A   UTSM5   TEXTC   '--C0MMAND INTERRUPT AT '
      02 002F7   D6D4D4C1 A
      02 002F8   D5C440C9 A
      02 002F9   D5E3C5D9 A
      02 002FA   D9E4D7E3 A
      02 002FB   40C1F340 A
540   02 002FC                        RES     7
541   02 00303   0C606040 A   UTSM6   TEXTC   '-- X T0 ABORT.'
      02 00304   E740F3D6 A
      02 00305   40C1C2D6 A
      02 00306   D9E34B40 A
542   02 00307   0CE6C8C9 A   UTSM7   TEXTC   'WHILE DELETING)'
      02 00308   D3C540C4 A
      02 00309   C5D3C5E3 A
      02 0030A   C9D5C75D A
543   02 0030B   3260A0E3 A   UTSM8   TEXTC   '--TAB CHAR. F0UND; ''TA'' NEEDED F0R C0L. SIMULATI0N'
      02 0030C   C1C240C3 A
```

```
      02 0030D      C8C1D94B A
      02 0030E      40C6D6E4 A
      02 0030F      D5C45E40 A
      02 00310      7UE3C17D 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
                                   *                                                   *****
              02 00317             BDISPTBL EQU      $-1                   THESE COMMANDS REQUIRE DISPLAY
      02 00318      04C3D6D7 A              GEN,8,24 4,'COP'               OF SEQ. NUMBERS SET UP IN
      02 00319      05D4C5D9 A              GEN,8,24 5,'MER'               INTFLAG1 AND INTFLAG2, WHEN
      02 0031A      02D4D240 A              TEXTC    'MK'                  INTERRUPTED BY THE BREAK KEY
      02 0031B      02D4C440 A              TEXTC    'MD'
      02 0031C      02C4C540 A              TEXTC    'DE'
      02 0031D      02C6C440 A              TEXTC    'FD'
      02 0031E      02C6F340 A              TEXTC    'FT'                                  *****
      02 0031F                     PATCH    RES      50
                                   *
                                            FIN
```

```
556                                          PAGE
557                                 ***********************************************
558                                 *  OPEN FPT'S (HAND-CODED TO AVOID PROBLEMS)  *
559                                 ***********************************************
560                                 *
561                                 *
562   02 00351   14000000 N   O$FPT     GEN,8,24  X'14',F:EI
563   02 00352   65480001 A             GEN,32    X'65480001'
564   02 00353   00000BE4 05            DATA      O$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                O$NAME    RES       8
572   02 00362   02000202 A             DATA      X'02000202'
573   02 00363                O$ACCT    RES       2                    ACCOUNT
574   02 00365   03010202 A             DATA      X'03010202'
575   02 00366                O$PASS    RES       2
576                                 *
577                                 *  OPEN FPT FOR COPY FILE
578                                 *
579   02 00368   14000000 N   O2$FPT    GEN,8,24  X'14',F:EO
580   02 00369   65480001 A             GEN,32    X'65480001'          SAME PARAMETERS AS ABOVE
581   02 0036A   00000BFE 05            DATA      O2$ABN,CARDIMG,2,4,2,3
      02 0036B   00000024
      02 0036C   00000002 A
      02 0036D   00000004 A
      02 0036E   00000002 A
      02 0036F   00000003 A
582   02 00370   01000808 A             DATA      X'01000808'
583   02 00371                O2$NAME   RES       8
584   02 00379   02000202 A             DATA      X'02000202'
585   02 0037A                O2$ACCT   RES       2
586   02 0037C   03010202 A             DATA      X'03010202'
587   02 0037D                O2$PASS   RES       2
```

```
588          00000001                      D8        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
```

```
                                        PAGE
                                ***********************************
                                *  DCB'S FOR UPDATE AND COPY FILES  *
                                ***********************************
                                *
                                *
          00000000                      DO        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)
                                        ELSE
          EXT                   F:EI    EQU       M:EI
                                        FIN
                                *
                                *  COPY FILE DCB
                                *
          00000000                      DO        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')
                                        ELSE
          EXT                   F:EO    EQU       M:EO
                                        FIN
```

```
                                       PAGE
                               ***********************************
                               *                                 *
                               *     B E G I N   E D I T O R     *
                               *                                 *
                               ***********************************
                               *
                               *
                               *
 05 00000                              CSECT     S(0,1)
       05 00000                BEGINEDITOR       EQU $
       05 00000                SECT5   EQU       $
 05 00000    68000001                  B         BGD10           :::ENTER HERE AT NORMAL START
       00000000                        DO        MODE=1
                          *S*          MTW,0     FILETYPE        :::ENTER HERE AT BREAK
                          *S*          BLZ       $+2               RE-OPEN FILE IF ONE WAS OPEN
                          *S*          BAL,LNK   REOPEN
                          *S*          LI,T1     0                 RESET ASSORTED FLAGS, ETC.
                          *S*          STW,T1    LASTKEY
                          *S*          STW,T1    NOCHGFLG
                          *S*          STW,T1    SETFLAG
                          *S*          STW,T1    STEPFLAG
                          *S*          LI,T1     -1
                          *S*          STW,T1    ALLFLAG
                          *S* *
                          *S* *  FINISH INITIALIZATION
                          *S* *
                          *S* BGD10    LI,R0     4                 SET ACTIVATION TYPE = 4
                          *S*          CAL3,2    0
                          *S*          BAL,LNK   TYPEMSG           TYPE: L/F + C/R
                          *S*          DATA      MSG0
                          *S* *
                          *S*          LI,T1     EDITBASE          CONVERT BASE TO
                          *S*          SLS,T1    -9                PAGES.
                          *S* *
                          *S*          LI,LNK    BEGINEDITOR       CONVERT PROGRAM TO NEXT
                          *S*          AI,LNK    X'1FF'            HIGHER PAGE, PUT
```

```
663                         *S*             SLS,LNK   -9
664                         *S*             STW,LNK   R0              IN R0.
665                         *S* *
666                         *S*             SW,R0     T1              CONVERT R0 TO PROGRAM DATA
667                         *S*             STB,LNK   R0              PAGE COUNT.
668                         *S* *
669                         *S*             LI,T1     ENDEDITOR       COMPUTE PURE PROCEDURE PAGE COUNT
670                         *S*             AI,T1     X'1FF'
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          05 00001                BGD10  EQU       $
679    05 00001    35 0002C8 02             STW,R0    TSADDR
680                                 *
681    05 00002    04 80037F 02             CAL1,8       RELATIVE     TABING
682    05 00003    04 800000 03             MIINT     BRK$KEY
       03 00000    0E 00028F 05
683                                         FIN
```

```
                                          PAGE
                                 ************************************
                                 *                                  *
                                 *    M A S T E R   P A R S E R     *
                                 *                                  *
                                 ************************************
                                 *
                                 *
                                 *
           05 00004              MASTERPARSER      EQU $
           00000001                       D8       M8DE=2
   05 00004     0410002CA 02              CAL1,1    PR8MPT$FPT
   05 00005     228FFFFF A                LI,T1    -1
   05 00006     358002C3 02               STW,T1   INTFLAG1
   05 00007     358002C4 02               STW,T1   INTFLAG2
   05 00008     35800020 02               STW,T1    ALLFLAG          RESET ALL FLAG
   05 00009     358002C9 02               STW,T1   XEQFLAG
                                          FIN
   05 0000A     1280000C 02               LD,T1    DMYSTKDW          PURGE STACK
   05 0000B     1580019C 02               STD,T1   STACKDW
   05 0000C     22800000 A                LI,T1    0
   05 0000D     35800048 02               STW,T1   CDT               SET # 8F CMNDS = 0
   05 0000E     358000AD 02               STW,T1   CHARPSN           SET NEXT CHAR T8 SCAN = 0
   05 0000F     3580019E 02               STW,T1    MVD:REC:CNT      SET MVD:REC:CNT = 0
   05 00010     3580019F 02               STW,T1    CHG;STG;CNT      SET CHG;STG;CNT = 0
   05 00011     358001A1 02               STW,T1   CT$FLAG           SET TYPE 'CM' FLAG 8FF
   05 00012     22800100 A                LI,T1    X'0100'           PUT 'END 8F CDT' MARKER IN CDT
   05 00013     35800049 02               STW,T1   CDT+1
   05 00014     22800049 02               LI,T1    CDT+1             INIT CDTADR=1ST CMND ADDR
   05 00015     358000AC 02               STW,T1   CDTADR
   05 00016     2287A120 A                LI,T1    500000            SET T8 PRINT ALL ERR8R MSGS
   05 00017     358000B2 02               STW,T1   ERR8RCNT
   05 00018     33000167 02               MTW,0    STEPFLAG          IS SYSTEM IN STEP M8DE
   05 00019     6830001C                  BEZ      $+3
           00000000                       D8       M8DE=1
                                *S*       LI,R0    '*'               YES - TYPE; '*'
                                *S*       CAL3,1   0
```

```
                                    *S*        LI,R0       '*'                TYPE PROMPT: '*'
                                    *S*        CAL3,1      0
                                               ELSE
  05 0001A    6A700DC4                         BAL,LNK     TYPEMSG            YES - TYPE '*'
  05 0001B    000002E4 02                      DATA        UTSM2
                                               FIN
  05 0001C    6A700CAE                         BAL,LNK     READTELETYPE2      READ IN COMMANDS
  05 0001D    20BFFFFF A                       AI,R1       -1                 SAVE CNT OF # OF CHARS INPUT,
  05 0001E    35B0019A 02                      STW,R1      TTYIMGSZ            LESS C/R
                                    *
                                    *
                                    *
        05 0001F                    RESUME#PARSING         EQU $              (ENTER HERE AFTER SEMI-COLON FOUND)
  05 0001F    F28000AC 02                      LB,T1       *CDTADR            INCR CDTADR TO NEXT ENTRY
  05 00020    668000AC 02                      AWM,T1      CDTADR
  05 00021    22800004 A                       LI,T1       4                  SET PSN OF NEXT PARAM = 1
  05 00022    358000E5 02                      STW,T1      PARAMPSN
  05 00023    33100048 02                      MTW,1       CDT                INCR OCUNT OF # OF ENTRIES
                                               NXTPRM      ERRC9,;
                                                           (INTG,PARSE:I:CMND#INTG),;
                                                           (STRG,PARSE:I:CMND#STRG),;
                                                           (ALPH,*),;
  05 00024    6A7008D4                                     (END,MASTEREXECUTIVE)
  05 00025    040001C8 02
  05 00026    04000096
  05 00027    0500008E
  05 00028    0600002A
  05 00029    00000301
  05 0002A    22300025 A                       LI,X1       CTBLSZ
  05 0002B    328000C1 02                      LW,T1       PARAMBUF           SEARCH FOR COMMAND NAME IN TABLE
  05 0002C    49800E25                         OR,T1       =X'00404040'       CONVERT CMND TO UPPER CASE
  05 0002D    31860039                         CW,T1       CNAMETBL,X1
  05 0002E    68300033                         BE          PRS10              FOUND - GO PROCESS
  05 0002F    6430002D                         BDR,X1      $-2                LOOP
  05 00030    6A700D8F                         BAL,LNK     TYPECERR           NOT IN TBL - TYPE: '-CN:UNKN CMND'
  05 00031    000001C4 02                      DATA        ERRC8
  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   2280 0100 A              LI,T1     X'0100'           INCR TO TYPE # OF NEXT CMND
763   05 00036   6680 0048 02             AWM,T1    CDT
764   05 00037   6A70 0D8F                BAL,LNK   TYPECERR          TYPE: '=CN;CMND ILGL HERE'
765   05 00038   0000 01B1 02             DATA      ERRC4
766   05 00039   6800 0004                B         MASTERPARSER
```

```
767                                     PAGE
768                             *
769                             *  COMMAND NAME TABLE
770                             *
771          05 00039           CNAMETBL EQU     $-1
772     05 0003A    02C2D740 A            TEXTC   'BP'              1: BP
773     05 0003B                          DC1     MODE=2
774     05 0003B    02E3C140 A            TEXTC   'TA'              2: TAB
775     05 0003C    02C3D940 A            TEXTC   'CR'              3: CR
776     05 0003D    05C2E4C9 A            GEN,8,24 5,'BUI'          4: BUILD
777     05 0003E    04C3D6D7 A            GEN,8,24 4,'COP'          5: COPY
778     05 0003F    06C4C5D3 A            GEN,8,24 6,'DEL'          6: DELETE
779     05 00040    04C5C4C9 A            GEN,8,24 4,'EDI'          7: EDIT
780     05 00041    03C5D5C4 A            TEXTC   'END'             8: END
781     05 00042    05D4C5D9 A            GEN,8,24 5,'MER'          9: MERGE
782     05 00043    02D9D740 A            TEXTC   'RP'              10: RP
783     05 00044    02C4C540 A            TEXTC   'DE'              11: DE
784     05 00045    02C6C440 A            TEXTC   'FD'              12: FD
785     05 00046    02C6E340 A            TEXTC   'FT'              13: FT
786     05 00047    02C9D540 A            TEXTC   'IN'              14: IN
787     05 00048    02C9E240 A            TEXTC   'IS'              15: IS
788     05 00049    02D4C440 A            TEXTC   'MD'              16: MD
789     05 0004A    02D4D240 A            TEXTC   'MK'              17: MK
790     05 0004B    02D9D540 A            TEXTC   'RN'              18: RN
791     05 0004C    02E2E240 A            TEXTC   'SS'              19: SS
792     05 0004D    02E2E340 A            TEXTC   'ST'              20: ST
793     05 0004E    02E3E240 A            TEXTC   'TS'              21: TS
794     05 0004F    02E3E840 A            TEXTC   'TY'              22: TY
795     05 00050    02E3C340 A            TEXTC   'TC'              23: TC
796     05 00051    02C6E240 A            TEXTC   'FS'              24: FS
797     05 00052    02C3D440 A            TEXTC   'CM'              25: CM
798     05 00053    02C3E340 A            TEXTC   'CT'              25: CT
799     05 00054    02E2C540 A            TEXTC   'SE'              30: SE
800     05 00055    02D1E440 A            TEXTC   'JU'              39: JU
801     05 00056    02D5D640 A            TEXTC   'NO'              40: NO
802     05 00057    02D9C640 A            TEXTC   'RF'              41: RF
803     05 00058    01C24040 A            TEXTC   'B'               4: BUILD (SHORT FORM)
```

```
05 00059   01C34040 A                TEXTC    'C'             5: COPY   (SHORT FORM)
05 0005A   01C44040 A                TEXTC    'D'             6: DELETE(SHORT FORM)
05 0005B   01C54040 A                TEXTC    'E'             7: EDIT   (SHORT FORM)
05 0005C   01E74040 A                TEXTC    'X'             8: END    (SHORT FORM)
05 0005D   01D44040 A                TEXTC    'M'             9: MERGE  (SHORT FORM)
05 0005E   02E3E740 A                TEXTC    'TX'           44: TX
        00000025          CTBLSZ     EQU      $-CNAMETBL-1
                          *
                          *  COMMAND BRANCH TABLE
                          *
        05 0005E          CBRCHTBL   EQU      $-1
05 0005F   680000F7                  B        PARSE:BP        1: BP
05 00060                             DO1      MODE=2
05 00060   680000F7                  B        PARSE:TA        2: TAB
05 00061   680000F7                  B        PARSE:CR        3: CR
05 00062   68000103                  B        PARSE:BUILD     4: BUILD
05 00063   68000131                  B        PARSE:COPY      5: COPY
05 00064   68000181                  B        PARSE:DELETE    6: DELETE
05 00065   68000181                  B        PARSE:EDIT      7: EDIT
05 00066   6800018B                  B        PARSE:END       8: END
05 00067   68000150                  B        PARSE:MERGE     9: MERGE
05 00068   680000F7                  B        PARSE:RP       10: RP
05 00069   680001AF                  B        PARSE:DE       11: DE
05 0006A   680001DC                  B        PARSE:FD       12: FD
05 0006B   680001DC                  B        PARSE:FT       13: FT
05 0006C   6800020C                  B        PARSE:IN       14: IN
05 0006D   6800020C                  B        PARSE:IS       15: IS
05 0006E   68000210                  B        PARSE:MD       16: MD
05 0006F   68000210                  B        PARSE:MK       17: MK
05 00070   68000228                  B        PARSE:RN       18: RN
05 00071   68000245                  B        PARSE:SS       19: SS
05 00072   68000245                  B        PARSE:ST       20: ST
05 00073   68000269                  B        PARSE:TS       21: TS
05 00074   68000269                  B        PARSE:TY       22: TY
05 00075   68000262                  B        PARSE:TC       23: TC
05 00076   680001DC                  B        PARSE:FS       24: FS
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:NB          40: NB
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
853                            *
854                            *  COMMAND NUMBER TABLE
855                            *
856        05 00084            CNMRTBL EQU     $
857  05 00084    00        A           DATA,1  0             (FILLER)
858  05 00084 1  01        A           DATA,1  1             1: BP
859  05 00084 2                        DBL     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
```

```
05 00089      14      A      DATA,1   20      20: ST
05 00089 1    15      A      DATA,1   21      21: TS
05 00089 2    16      A      DATA,1   22      22: TY
05 00089 3    17      A      DATA,1   23      23: TC
05 0008A      18      A      DATA,1   24      24: FS
05 0008A 1    19      A      DATA,1   25      25: CM
05 0008A 2    19      A      DATA,1   25      25: CT
05 0008A 3    1E      A      DATA,1   30      30: SE
05 0008B      27      A      DATA,1   39      39: JU
05 0008B 1    28      A      DATA,1   40      40: NO
05 0008B 2    29      A      DATA,1   41      41: RF
05 0008B 3    04      A      DATA,1   4       4: BUILD (SHORT FORM)
05 0008C      05      A      DATA,1   5       5: COPY  (SHORT FORM)
05 0008C 1    06      A      DATA,1   6       6: DELETE(SHORT FORM)
05 0008C 2    07      A      DATA,1   7       7: EDIT  (SHORT FORM)
05 0008C 3    08      A      DATA,1   8       8: END   (SHORT FORM)
05 0008D      09      A      DATA,1   9       9: MERGE (SHORT FORM)
05 0008D 1    2C      A      DATA,1   44      44: TX
                             BOUND    4
```

```
                                     PAGE
                              ********************************
                              *  PROCESS INTRALINE COMMANDS  *
                              ********************************
                              *
                              *
          05 0008E            PARSE:I:CMND$STRG EQU $
05 0008E     22500000 A                LI,P1     0
05 0008F     6A700963                  BAL,LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY WITH CMND=0
05 00090     00000002 A                DATA      2
05 00091     22500005 A                LI,P1     STRG
05 00092     6A70081B                  BAL,LNK   ADDCDTPARAM       PUT STRING IN CDT
                                       NXTPRM    ERRC9,;
05 00093     6A7008D4                            (ALPH,ICS10)
05 00094     010001C8 02
05 00095     060000AA
                              *
                              *
          05 00096            PARSE:I:CMND$INTG EQU $
05 00096     22500000 A                LI,P1     0
05 00097     328000C1 02               LW,T1     PARAMBUF          SAVE INTEGER
                                       NXTPRM    ERRC9,;
                                                 (ALPH,ICS50),;
05 00098     6A7008D4                            (STRG,*)
05 00099     020001C8 02
05 0009A     060000B6
05 0009B     0500009C
05 0009C     6A700963                  BAL,LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY WITH CMND=0
05 0009D     00000003 A                DATA      3
05 0009E     468000C1 02               XW,T1     PARAMBUF          SAVE STRING AND PUT INTG IN PARAMBUF
05 0009F     22900001 A                LI,T2     1                 SAVE PARAMBUF SIZE FOR STRING AND
05 000A0     469000E6 02               XW,T2     PRMBUFSZ           SET IT = 1
05 000A1     22500004 A                LI,P1     INTG              PUT INTG IN CDT
05 000A2     6A70081B                  BAL,LNK   ADDCDTPARAM
05 000A3     358000C1 02               STW,T1    PARAMBUF          RESTORE STRING
05 000A4     359000E6 02               STW,T2    PRMBUFSZ          RESTORE PARAMBUF SIZE
05 000A5     22500005 A                LI,P1     STRG              PUT STRING IN CDT
```

```
929   05 000A6   6A700081B              BAL,LNK  ADDCDTPARAM
930                                     NXTPRM   ERRC9,1
931   05 000A7   6A7008D4                       (ALPH,*)
      05 000A8   010001C8 02
      05 000A9   060000AA
932                              *
933                              *  COMMAND NAME FOUND: IDENTIFY IT
934                              *
935   05 000AA   22300008 A      ICS10    LI,X1    ICTBLSZ
936   05 000AB   328000C1 02             LW,T1    PARAMBUF          SEARCH TABLE FOR CMND NAME
937   05 000AC   318600E3                CW,T1    ICNAMETBL,X1
938   05 000AD   683000B2                BE       ICS20             FOUND - GO PROCESS
939   05 000AE   643000AC                BDR,X1   *-2               LOOP
940   05 000AF   6A700D8F                BAL,LNK  TYPECERR          TYPE: '-CN:UNKN CMND'
941   05 000B0   000001C4 02             DATA     ERRC8
942   05 000B1   68000004                B        MASTERPARSER      GO TO PARSER
943                              *
944                              *  COMMAND IDENTIFIED: GO PROCESS LAST PARAMETER
945                              *
946   05 000B2   22400001 A      ICS20    LI,X2    1                 PUT CMND NUMBER IN CDT
947   05 000B3   7256000EC               LB,P1    ICNMRTBL,X1
948   05 000B4   F558000AC 02            STB,P1   *CDTADR,X2
949   05 000B5   670600EE                FXU      ICBRCHTBL,X1      GO PROCESS LAST PARAM
950                              *
951                              *  FORM FOUND IS:  C X -  , PROCESS THIS
952                              *
953   05 000B6   6A700963        ICS50    BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY WITH CMND=0
954   05 000B7   000000002 A             DATA     2
955   05 000B8   468000C1 02             XW,T1    PARAMBUF          PUT INTG IN PARAMBUF AND SAVE NAME
956   05 000B9   22500004 A              LI,P1    INTG              PUT INTG IN CDT
957   05 000BA   6A70081B                BAL,LNK  ADDCDTPARAM
958   05 000BB   358000C1 02             STW,T1   PARAMBUF          RESTORE CMND NAME
959   05 000BC   680000AA                B        ICS10             GO IDENTIFY CMND
960                              *
961                              *  FINISH TYPE ALPHA:  - X /STR2/
962                              *
963         05 000BD             TYPE$ALPHA       EQU $
```

```
964                                          NXTPRM   ERRP8,;
965   05 000BD   6A7008D4                             (STRG,*)
      05 000BE   01000249 02
      05 000BF   05000000
966   05 000C0   22500005 A                  LI,P1    STRG                PUT STRING IN CDT
967   05 000C1   6A70081B                    BAL,LNK  ADDCDTPARAM
968                                          NXTPRM   *ERRP4,;
969                                                   (SCBL,RESUME$PARSING),;
970   05 000C2   6A7008D4                             (END,MASTEREXECUTIVE)
      05 000C3   02800239 02
      05 000C4   0800001F
      05 000C5   00000301
971                                  *
972                                  *  FINISH TYPE BETA:  = X N
973                                  *
974         05 000C6                 TYPE$BETA          EQU $
975                                          NXTPRM   ERRP9,;
976   05 000C6   6A7008D4                             (INTG,*)
      05 000C7   01000240 02
      05 000C8   040000C9
977   05 000C9   22500004 A                  LI,P1    INTG                PUT COUNT IN CDT
978   05 000CA   6A70081B                    BAL,LNK  ADDCDTPARAM
979                                          NXTPRM   *ERRP4,;
980                                                   (SCBL,RESUME$PARSING),;
981   05 000CB   6A7008D4                             (END,MASTEREXECUTIVE)
      05 000CC   02800239 02
      05 000CD   0800001F
      05 000CE   00000301
982                                  *
983                                  *  INTRALINE COMMANDS 'D' OR 'S' FOUND: CHECK THAT FORM IS: /STR1/ D(S)
984                                  *
985         05 000CF                 TYPE$I:CMND$D      EQU $
986   05 000CF   22200001 A                  LI,X4    1                   USE X4=1 FOR 'D'
987   05 000D0   680000D2                    B        TYPE$I:CMND$S+1
988                                  *
989                                  *
990         05 000D1                 TYPE$I:CMND$S      EQU $
```

```
 991   05 000D1   22200000 A          LI,X4    0                  USE X4=0 FOR 'S'
 992   05 000D2   22300003 A          LI,X1    3
 993   05 000D3   F2860CAC 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   6830000E            BE       ICS90              YES - FORM MUST BE: N /ST1/ D(S) -
 996   05 000D6   22300004 A          LI,X1    4
 997   05 000D7   F2860CAC 02        LB,T1    *CDTADR,X1         NO - GET TYPE OF PARAM1
 998   05 000D8   21800005 A          CI,T1    STRG               IS TYPE='STRING'
 999   05 000D9   6830000E            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   6804000F  ICS90    B        $+1,X4
1008   05 000DF   6800008D            B        TYPE$ALPHA
1009                                NXTPRM   *ERRP4,1
1010                                         (SCBL,RESUME$PARSING),1
1011   05 000E0   6A7008D4                   (END,MASTEREXECUTIVE)
       05 000E1   02800239 02
       05 000E2   08000001F
       05 000E3   00000301
1012                                *
1013                                *  INTRALINE COMMAND NAME TABLE
1014                                *
1015        05 000E3          ICNAMETBL        EQU $-1
1016   05 000E4   01C44040 A              TEXTC    'D'                31: D
1017   05 000E5   01C54040 A              TEXTC    'E'                32: E
1018   05 000E6   01C64040 A              TEXTC    'F'                33: F
1019   05 000E7   01D34040 A              TEXTC    'L'                34: L
1020   05 000E8   01D64040 A              TEXTC    'O'                35: O
1021   05 000E9   01D74040 A              TEXTC    'P'                36: P
1022   05 000EA   01D94040 A              TEXTC    'R'                37: R
1023   05 000EB   01E24040 A              TEXTC    'S'                38: S
1024        00000008          ICTBLSZ  EQU      $-ICNAMETBL-1
```

```
1025                                    *
1026                                    *  INTRALINE COMMAND NUMBER TABLE
1027                                    *
1028          05 000EC                  ICNMRTBL EQU      $
1029    05 000EC      00          A               DATA,1   0                  (FILLER)
1030    05 000EC 1    1F          A               DATA,1   31                 31: D
1031    05 000EC 2    20          A               DATA,1   32                 32: E
1032    05 000EC 3    21          A               DATA,1   33                 33: F
1033    05 000ED      22          A               DATA,1   34                 34: L
1034    05 000ED 1    23          A               DATA,1   35                 35: O
1035    05 000ED 2    24          A               DATA,1   36                 36: P
1036    05 000ED 3    25          A               DATA,1   37                 37: R
1037    05 000EE      26          A               DATA,1   38                 38: S
1038                                            BOUND    4
1039                                    *
1040                                    *  INTRALINE COMMAND BRANCH TABLE
1041                                    *
1042          05 000EE                  ICBRCHTBL         EQU $-1
1043    05 000EF      68000CF                     B         TYPE$I:CMND$D      31: D
1044    05 000F0      68000BD                     B         TYPE$ALPHA         32: E
1045    05 000F1      68000BD                     B         TYPE$ALPHA         33: F
1046    05 000F2      68000C6                     B         TYPE$BETA          34: L
1047    05 000F3      68000BD                     B         TYPE$ALPHA         35: O
1048    05 000F4      68000BD                     B         TYPE$ALPHA         36: P
1049    05 000F5      68000C6                     B         TYPE$BETA          37: R
1050    05 000F6      68000D1                     B         TYPE$I:CMND$S      38: S
```

```
1051                                    PAGE
1052                                ********************************
1053                                *  PARSE FORM:  BP BN(BFF)     *
1054                                *  PARSE FORM:  TA F(M,S)      *
1055                                *  PARSE FORM:  RP BN(BFF)     *
1056                                *  PARSE FORM:  CR BN(BFF)     *
1057                                ********************************
1058                                *
1059                                *
1060          05 000F7              PARSE:RP EQU     $
1061          05 000F7              PARSE:BP EQU     $
1062          05 000F7              PARSE:CR EQU     $
1063          05 000F7              PARSE:TA EQU     $
1064    05 000F7    6A700963                BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1065    05 000F8    00000001 A              DATA     1
1066    05 000F9    6A700834                BAL,LNK  CHECK1CDTENTRY    MAKE SURE 'BP' IS FIRST CMND
1067                                        NXTPRM   ERRP4,,
1068    05 000FA    6A7008D4                         (ALPH,*)
        05 000FB    01000239 02
        05 000FC    06000FD
1069    05 000FD    22500006 A              LI,P1    ALPH              PUT ALPHA TEXT IN CDT
1070    05 000FE    6A70081B                BAL,LNK  ADDCDTPARAM
1071                                        NXTPRM   *ERRP4,,
1072                                                 (SCOL,ILGL$SEMICOLON),,
1073    05 000FF    6A7008D4                         (END,MASTEREXECUTIVE)
        05 00100    02800239 02
        05 00101    08000035
        05 00102    00000301
```

```
1074                                    PAGE
1075                                ************************************
1076                                *  PARSE FORM:  BUILD FID(,N(,I))  *
1077                                ************************************
1078                                *
1079                                *
1080          05 00103              PARSE:BUILD       EQU $
1081    05 00103    6A700963                  BAL,LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY
1082    05 00104    00000003 A                DATA      3
1083    05 00105    6A700834                  BAL,LNK   CHECK1CDTENTRY    MAKE SURE 'BUILD' IS FIRST CMND
1084    05 00106    6A70083A                  BAL,LNK   GETFILEID         GET FILE ID
1085    05 00107    22500001 A                LI,P1     NAME              PUT IT IN CDT
1086    05 00108    6A70081B                  BAL,LNK   ADDCDTPARAM
1087                                          NXTPRM    *ERRP4,,
1088                                                    (COM,*),,
1089                                                    (SCOL,ILGL$SEMICOLON),,
1090    05 00109    6A7008D4                            (END,MASTEREXECUTIVE)
        05 0010A    03800239 02
        05 0010B    07000010E
        05 0010C    08000035
        05 0010D    00000301
1091                                *
1092                                *
1093                                *
1094          05 0010E              GET$SEQ$INCR      EQU $             (ENTER HERE FOR FORM:  N(,I) )
1095                                          NXTPRM    ERRP5,,
1096                                                    (INTG,*),,
1097                                                    (COM,PBU05),,
1098                                                    (SEQ,PBU10),,
1099    05 0010E    6A7008D4                            (SEQ2,ILGL$SEQ2)
        05 0010F    04000023D 02
        05 00110    04000114
        05 00111    07000012C
        05 00112    02000115
        05 00113    03000129
1100    05 00114    6A700287                  BAL,LNK   ADJINT
1101                                *
```

```
1102                                    *  PUT SEQ # IN CDT
1103                                    *
1104   05 00115   225000002 A   PBU10   LI,P1     SEQ              PUT SEQ # IN CDT
1105   05 00116   6A700081B            BAL,LNK   ADDCDTPARAM
1106                                    *
1107                                    *
1108                                    *
1109        05 00117            GET$INCREMENT   EQU $              (ENTER HERE FOR FORM: (,I) )
1110                                    NXTPRM    *ERRP4,;
1111                                              (COM,*),;
1112                                              (SCOL,ILGL$SEMICOLON),;
1113   05 00117   6A7008D4                       (END,MASTEREXECUTIVE)
       05 00118   038000239 02
       05 00119   070000110
       05 0011A   080000035
       05 0011B   000000301
1114        05 0011C            PBU30   EQU       $
1115                                    NXTPRM    ERRP6,;
1116                                              (INTG,*),;
1117   05 0011C   6A7008D4                       (SEQ,PBU20)
       05 0011D   020000241 02
       05 0011E   040000120
       05 0011F   020000121
1118   05 00120   6A700287             BAL,LNK   ADJINT
1119                                    *
1120                                    *  PUT INCREMENT IN CDT
1121                                    *
1122   05 00121   225000002 A   PBU20   LI,P1     SEQ              PUT INCR IN CDT
1123   05 00122   330000C1 02          MTW,0     PARAMBUF         MAY NOT BE ZERO.
1124   05 00123   683000129            BEZ       ILGL$SEQ2
1125   05 00124   6A700081B            BAL,LNK   ADDCDTPARAM
1126                                    NXTPRM    *ERRP4,;
1127                                              (SCOL,ILGL$SEMICOLON),;
1128   05 00125   6A7008D4                       (END,MASTEREXECUTIVE)
       05 00126   028000239 02
       05 00127   080000035
       05 00128   000000301
```

```
1129                                *
1130                                *
1131                                *
1132          05 00129              ILGL&SEQ2         EQU $
1133    05 00129    6A700D99                BAL,LNK   TYPEPERR          TYPE: !PN:ILGL SEQ #'
1134    05 0012A    00000250 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    6A700081B               BAL,LNK   ADDCDTPARAM
1143    05 00130    68000011C               B         PBU30              GET INCREMENT
```

```
1144                                    PAGE
1145                                ***********************************************
1146                                *  PARSE FORM:  COPY FID1 TO FID2(,N(,I))  *
1147                                ***********************************************
1148                                *
1149                                *
1150          05 00131              PARSE:COPY        EQU $
1151    05 00131    6A700963                        BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1152    05 00132    00000005 A                      DATA     5
1153    05 00133    6A700834                        BAL,LNK  CHECK1CDTENTRY    MAKE SURE 'COPY' IS FIRST CMND
1154    05 00134    6A70083A                        BAL,LNK  GETFILEID         GET FILE ID 1
1155    05 00135    22500001 A                      LI,P1    NAME              PUT IT IN CDT
1156    05 00136    6A70081B                        BAL,LNK  ADDCDTPARAM
1157                                                NXTPRM   ERRC9,,
1158    05 00137    6A7008D4                                 (ALPH,*)
        05 00138    01000108 02
        05 00139    06000013A
1159    05 0013A    32800001 02                     LW,T1     PARAMBUF
1160    05 0013B    31800015 02                     CW,T1     X:T0
1161    05 0013C    69300013F                       BNE       PC03
1162    05 0013D    32800012 02                     LW,T1     X:ON
1163    05 0013E    35800001 02                     STW,T1    PARAMBUF
1164          05 0013F              PC03      EQU       $
1165    05 0013F    22500006 A                      LI,P1     ALPH              PUT 'ON(OVER)' IN CDT
1166    05 00140    6A70081B                        BAL,LNK   ADDCDTPARAM
1167    05 00141    32800001 02                     LW,T1     PARAMBUF
1168    05 00142    31800012 02                     CW,T1     X:ON              DOES PARAM2='ON' OR 'OVER'
1169    05 00143    68300149                        BE        PC010
1170    05 00144    31800013 02                     CW,T1     X:OVER
1171    05 00145    68300149                        BE        PC010
1172          05 00146              PC05      EQU       $
1173    05 00146    6A700D8F                        BAL,LNK   TYPECERR          NO = TYPE! !=CN:ILGL SYNTAX!
1174    05 00147    00000108 02                     DATA      ERRC9
1175    05 00148    68000004                        B         MASTERPARSER      EXIT TO PARSER
1176                                *
1177                                *  GET 2ND FID AND THEN GO PROCESS FORM: (,N(,I))
1178                                *
```

```
1179   05 00149   6A7008 3A      PC010    BAL,LNK  GETFILEID          GET FILE ID 2
1180   05 0014A   6A7008 1B               BAL,LNK  ADDCDTPARAM        PUT IT IN CDT
1181                                      NXTPRM   *ERRP4,;
1182                                               (COM,GET$SEQ$INCR),;
1183                                               (SCOL,ILGL$SEMICOLON),;
1184   05 0014B   6A7008 D4                        (END,MASTEREXECUTIVE)
       05 0014C   03800239 02
       05 0014D   07000010E
       05 0014E   08000035
       05 0014F   00000301
1185                      *
```

```
1186                                    PAGE
1187                                ****************************************************************
1188                                *  PARSE FORM:    MERGE FID1(,N1(-N2)) INTO FID2,N3(-N4)(,I)    *
1189                                ****************************************************************
1190                                *
1191                                *
1192          05 00150              PARSE:MERGE     EQU $
1193    05 00150    6A700963                      BAL,LNK  NEWCDTENTRY       SET UP NEW ENTRY.
1194    05 00151    000000006 A                   DATA     6
1195    05 00152    6A700834                      BAL,LNK  CHECK1CDTENTRY    MUST BE FIRST.
1196    05 00153    6A70083A                      BAL,LNK  GETFILEID         GET THE FID,
1197    05 00154    225000001 A                   LI,P1    NAME              AND ADD IT TO THE CDT.
1198    05 00155    6A70081B                      BAL,LNK  ADDCDTPARAM
1199                                *
1200                                              NXTPRM   *ERRP4,;          CHECK FOR SPECIFIC RECORD RANGE.
1201                                                       (COM,*),;
1202                                                       (ALPH,PME20),;
1203    05 00156    6A7008D4                               (SCOL,ILGL$SEMICOLON)
        05 00157    038000239 02
        05 00158    07000015B
        05 00159    06000167
        05 0015A    08000035
1204                                *
1205                                              NXTPRM   ERRP5,;           CONVERT SEQUENCE SPECIFICATION.
1206                                                       (INTG,*),;
1207                                                       (SEQ,PME5),;
1208    05 00158    6A7008D4                               (SEQ2,PME15)
        05 0015C    03000023D 02
        05 0015D    04000160
        05 0015E    02000161
        05 0015F    03000162
1209                                *
1210    05 00160    6A700287                      BAL,LNK  ADJINT            ADJUST INTEGER,
1211    05 00161    6A70028B          PME5        BAL,LNK  REPSEQ            DUPLICATE SINGLE VALUE.
1212    05 00162    225000003 A       PME15       LI,P1    SEQ2
1213    05 00163    6A70081B                      BAL,LNK  ADDCDTPARAM       PUT SEQ # PAIR IN CDT.
1214                                *
```

```
1215                                          NXTPRM   ERRC9,;          VERIFY 'INTO' NEXT.
1216   05 00164   6A7008D4                             (ALPH,*)
       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   32800001 02           LW,T1    PARAMBUF          MAKE SURE OF PARAMETER.
1221   05 0016A   31800014 02           CW,T1    X'INTO
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,;          VERIFY PRESENCE OF DESTINATION
1229                                                   (COM,*),;        SEQ #.
1230                                                   (END,PME40),;
1231   05 0016F   6A7008D4                             (SCOL,ILGL*SEMICOLON)
       05 00170   03000108 02
       05 00171   07000174
       05 00172   0A00017E
       05 00173   08000035
1232                                 *
1233                                          NXTPRM   ERRP5,;          CONVERT SPECIFICATION.
1234                                                   (INTG,*),;
1235                                                   (SEQ,PME30),;
1236   05 00174   6A7008D4                             (SEQ2,PME35)
       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
```

```
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     ERRP17
1247  05 00180   68000004                  B        MASTERPARSER
```

```
1248                                        PAGE
1249                                ********************************
1250                                *  PARSE FORMS:  DELETE FID    *
1251                                *                EDIT   FID    *
1252                                ********************************
1253                                *
1254                                *
1255          05 00181              PARSE:DELETE      EQU $
1256          05 00181              PARSE:EDIT        EQU $
1257    05 00181    6A700963                BAL,LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY
1258    05 00182    00000001 A              DATA      1
1259    05 00183    6A700834                BAL,LNK   CHECK1CDTENTRY    MAKE SURE 'DELETE(EDIT)' IS 1ST CMND
1260    05 00184    6A70083A                BAL,LNK   GETFILEID         GET FILE ID
1261    05 00185    22500001 A              LI,P1     NAME              PUT IT IN CDT
1262    05 00186    6A70081B                BAL,LNK   ADDCDTPARAM
1263                                        NXTPRM    *ERRP4,;
1264                                                  (SCBL,ILGL*SEMICBLBN),;
1265    05 00187    6A7008D4                          (END,MASTEREXECUTIVE)
        05 00188    02800239 02
        05 00189    08000035
        05 0018A    00000301
```

```
1266                                        PAGE
1267                                ************************
1268                                *  PARSE FORMS:  END   *
1269                                *                N0    *
1270                                ************************
1271                                *
1272                                *
1273          05 0018B              PARSE:END         EQU $
1274          05 0018B              PARSE:N0          EQU $
1275    05 0018B    6A700963                  BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1276    05 0018C    00000000 A                DATA     0
1277    05 0018D    6A700834                  BAL,LNK  CHECK1CDTENTRY    MAKE SURE 'END(N0)' IS FIRST CMND
1278                                          NXTPRM   ERRC9,,
1279                                                   (SC0L,ILGL$SEMIC0LON),,
1280    05 0018E    6A7008D4                           (END,MASTEREXECUTIVE)
        05 0018F    02C001C8 02
        05 00190    08000035
        05 00191    00000301
1281                                *
```

```
1282                                        PAGE
1283                                ************************
1284                                *  PARSE FORM:  CM N,C  *
1285                                *  PARSE FORM:  CT N,C  *
1286                                ************************
1287                                *
1288                                *
1289          05 00192              PARSE:CT FQU      $
1290    05 00192    331001A1 02               MTW,1    CT$FLAG          SET TYPE FLAG ON
1291                                *                                     AND TURN CMND INTO 'CM'
1292          05 00193              PARSE:CM FQU      $
1293    05 00193    6A700963                  BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1294    05 00194    00000002 A                DATA     2
1295    05 00195    6A700834                  BAL,LNK  CHECK1CDTENTRY   MAKE SURE 'CM' IS FIRST CMND
1296                                          NXTPRM   ERRP5,,
1297                                                   (INTG,*),,
1298                                                   (SEQ,PCM10),,
1299    05 00196    6A7008D4                       (SEQ2,ILGL$SEQ2)
        05 00197    03000230 02
        05 00198    04000198
        05 00199    02000190
        05 0019A    03000129
1300    05 0019B    6A700287                  BAL,LNK  ADJINT
1301                                *
1302                                *  SEQ # GIVEN: PUT IT IN CDT AND PROCESS COLUMN NUMBER
1303                                *
1304    05 0019C    22500002 A    PCM10       LI,P1    SEQ              PUT SEQ # IN CDT
1305    05 0019D    6A70081B                  BAL,LNK  ADDCDTPARAM
1306                                          NXTPRM   *ERRP4,,
1307                                                   (COM,*),,
1308                                                   (SCOL,PCM20),,
1309    05 0019E    6A7008D4                       (END,PCM20)
        05 0019F    03800239 02
        05 001A0    07000183
        05 001A1    08000196
        05 001A2    00000196
1310                                          NXTPRM   ERRP7,,
```

```
1311    05 001A3    6A7008D4                        (INTG,*)
        05 001A4    01000245 02
        05 001A5    040001A6
1312    05 001A6    22500004 A          LI,P1       INTG            PUT C8L, # IN CDT
1313    05 001A7    6A70081B            BAL,LNK     ADDCDTPARAM
1314                                    NXTPRM      *ERRP4,;
1315                                                (SC8L,ILGL*SEMIC8L8N),;
1316    05 001A8    6A7008D4                        (END,MASTEREXECUTIVE)
        05 001A9    02800239 02
        05 001AA    08000035
        05 001AB    00000301
1317                        *
1318                        *   ERR8R: SEC8ND PARAMETER MISSING
1319                        *
1320    05 001AC    6A700D99        PCM20       BAL,LNK     TYPEPERR        TYPE: !=PN!PARAM MISSING!
1321    05 001AD    00000271 02                 DATA        ERRP17
1322    05 001AE    68000004                    B           MASTERPARSER    G8 T8 PARSER
```

```
1323                                    PAGE
1324                                ************************************
1325                                *  PARSE FORMS:  DE N(=M)          *
1326                                *                SE N(=M)(,C(,D))  *
1327                                ************************************
1328                                *
1329                                *
1330        05 001AF            PARSE:DE EQU     $
1331  05 001AF   6A700963               BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1332  05 001B0   00000001 A             DATA     1
1333  05 001B1   68001B4               B        PARSE:SE+2
1334                                *
1335                                *
1336        05 001B2            PARSE:SE EQU     $
1337  05 001B2   6A700963               BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1338  05 001B3   00000003 A             DATA     3
1339                                    NXTPRM   ERRP5,,
1340                                             (INTG,PDE5),,
1341                                             (SEQ,PDE10),,
1342  05 001B4   6A7008D4               (SEQ2,PDE15)
      05 001B5   03000230 02
      05 001B6   04001B9
      05 001B7   02001BA
      05 001B8   03001BB
1343                                *
1344                                *  SEQ, # IS AN INTEGER: ADJUST IT
1345                                *
1346  05 001B9   6A700287           PDE5     BAL,LNK  ADJINT
1347                                *
1348                                *  ONLY ONE SEQ. # GIVEN: DUPLICATE IT
1349                                *
1350  05 001BA   6A70028B           PDE10    BAL,LNK  REPSEQ
1351                                *
1352                                *  PUT SEQ. # PAIR IN CDT AND CHECK IF COMMAND IS FIRST FOR 'DE'
1353                                *
1354  05 001BB   6A700834           PDE15    BAL,LNK  CHECK1CDTENTRY    MAKE SURE 'DE(SE)' IF FIRST CMND
1355  05 001BC   32800005 A             LW,T1    P1                    SAVE CMND #
```

```
1356  05 001BD  225000003 A          LI,P1    SEQ2              PUT SEQ # PAIR IN CDT
1357  05 001BE  6A700081B            BAL,LNK  ADDCDTPARAM
1358  05 001BF  2180001E A           CI,T1    FIRST$I!CMND      IS CMND='DE'
1359  05 001C0  681001C5             BGE      PDE20             NO - CMND='SE'
1360                                 NXTPRM   *ERRP4,;
1361                                          (SCOL,ILGL$SEMICOLON),;
1362  05 001C1  6A7008D4                       (END,MASTEREXECUTIVE)
      05 001C2  028800239 02
      05 001C3  080000035
      05 001C4  000000301
1363                          *
1364                          *  FINISH UP 'SF'
1365                          *
1366                          PDE20    NXTPRM   *ERRP4,;
1367                                          (COM,*),;
1368                                          (SCOL,RESUME$PARSING),;
1369  05 001C5  6A7008D4                       (END,MASTEREXECUTIVE)
      05 001C6  038800239 02
      05 001C7  070001CA
      05 001C8  0800001F
      05 001C9  000000301
1370                                 NXTPRM   ERRP7,;
1371  05 001CA  6A7008D4                       (INTG,*)
      05 001CB  010000245 02
      05 001CC  040001CD
1372  05 001CD  225000004 A          LI,P1    INTG
1373  05 001CE  6A700081B            BAL,LNK  ADDCDTPARAM       PUT 1ST COL # IN CDT
1374                                 NXTPRM   *ERRP4,;
1375                                          (COM,*),;
1376                                          (SCOL,RESUME$PARSING),;
1377  05 001CF  6A7008D4                       (END,MASTEREXECUTIVE)
      05 001D0  038800239 02
      05 001D1  070001D4
      05 001D2  0800001F
      05 001D3  000000301
1378                                 NXTPRM   ERRP7,;
1379  05 001D4  6A7008D4                       (INTG,*)
```

```
          05 001D5    01000245 02
          05 001D6    04 0001D7
    1380  05 001D7    6A7 0081B          BAL,LNK  ADDCDTPARAM       PUT 2ND COL # IN CDT
    1381                                 NXTPRM   *ERRP4,;
    1382                                          (SCOL,RESUME*PARSING),;
    1383  05 001D8    6A7 008D4                   (END,MASTEREXECUTIVE)
          05 001D9    028 00239 02
          05 001DA    08 00001F
          05 001DB    00000301
```

```
1384                                    PAGE
1385                                ********************************************
1386                                *  PARSE FORMS:  FD N(=M),/STRG/(,C(,D))   *
1387                                *                FT N(=M),/STRG/(,C(,D))   *
1388                                ********************************************
1389                                *
1390                                *
1391          05 001DC              PARSE:FD EQU     *
1392          05 001DC              PARSE:FS EQU     *
1393          05 001DC              PARSE:FT EQU     *
1394    05 001DC    6A700963              BAL,LNK  NEWCDTENTRY       BUILD NEW CDT ENTRY
1395    05 001DD    00000004 A            DATA     4
1396    05 001DE    6A700834              BAL,LNK  CHECK1CDTENTRY    MAKE SURE 'FD(FT)' IS FIRST CMND
1397                                      NXTPRM   ERRP5,;
1398                                               (INTG,*),;
1399                                               (SEQ,PFD10),;
1400    05 001DF    6A7008D4                     (SEQ2,PFD15)
        05 001E0    03000023D 02
        05 001E1    040001E4
        05 001E2    020001E5
        05 001E3    030001E6
1401    05 001E4    6A700287              BAL,LNK  ADJINT
1402                                *
1403                                *  ONLY ONE SEQ # GIVEN: DUPLICATE IT
1404                                *
1405    05 001E5    6A70028B        PFD10 BAL,LNK  REPSEQ
1406                                *
1407                                *  PUT SEQ # PAIR IN CDT AND GET 2ND PARAMETER
1408                                *
1409    05 001E6    22500003 A      PFD15 LI,P1    SEQ2                PUT 'SEQ # PAIR' PARAM IN CDT
1410    05 001E7    6A70081B              BAL,LNK  ADDCDTPARAM
1411                                      NXTPRM   *ERRP4,;
1412                                               (COM,*),;
1413                                               (SCOL,PFD20),;
1414    05 001E8    6A7008D4                     (END,PFD20)
        05 001E9    03800239 02
        05 001EA    070001ED
```

```
        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              PUT 'STRING' PARAM IN CDT
1418    05 001F1    6A700818            BAL,LNK  ADDCDTPARAM
1419                                *
1420                                *
1421        05 001F2                GET$C0L#$PAIR    EQU $
1422                                    NXTPRM   *ERRP4,;
1423                                             (C0M,*),;
1424                                             (SC0L,ILGL$SEMIC0L0N),;
1425    05 001F2    6A7008D4                     (END,MASTEREXECUTIVE)
        05 001F3    03800239 02
        05 001F4    070001F7
        05 001F5    08000035
        05 001F6    00000301
1426                                    NXTPRM   ERRP7,;
1427    05 001F7    6A7008D4                     (INTG,*)
        05 001F8    01000245 02
        05 001F9    040001FA
1428    05 001FA    22500004 A          LI,P1    INTG              PUT 1ST 'C0L #' IN CDT
1429    05 001FB    6A700818            BAL,LNK  ADDCDTPARAM
1430                                    NXTPRM   *ERRP4,;
1431                                             (C0M,*),;
1432                                             (SC0L,ILGL$SEMIC0L0N),;
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
```

```
1436  05 00204  6A700818            BAL,LNK  ADDCDTPARAM       PUT 2ND 'COL #' IN CDT
1437                                 NXTPRM   *ERRP4,,
1438                                          (SCOL,ILGL$SEMICOLON),,
1439  05 00205  6A7008D4                      (END,MASTEREXECUTIVE)
      05 00206  02800239 02
      05 00207  08000035
      05 00208  00000301
1440                          *
1441                          *  ERROR: SECOND PARAMETER MISSING
1442                          *
1443  05 00209  6A700D99      PFD20    BAL,LNK  TYPEPERR         TYPE: '=PN;PARAM MISSING'
1444  05 0020A  00000271 02            DATA     ERRP17
1445  05 0020B  68000004               B        MASTERPARSER     GO TO PARSER
```

```
1446                                PAGE
1447                        ****************************
1448                        *  PARSE FORM:  IN N(,I)   *
1449                        ****************************
1450                        *
1451                        *
1452          05 00020C     PARSE:IN EQU     $
1453          05 00020C     PARSE:IS EQU     $
1454    05 00020C   6A700963          BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1455    05 00020D   00000002 A        DATA     2
1456    05 00020E   6A700834          BAL,LNK  CHECK1CDTENTRY   MAKE SURE 'IN' IS FIRST CMND
1457    05 00020F   68000010E         B        GET$SEQ$INCR     GO PROCESS FORM:  N(,I)
```

```
1458                                    PAGE
1459                                ****************************************
1460                                *  PARSE FORMS:  MD N(=M),K(=L)(,I)    *
1461                                *                MK N(=M),K(=L)(,I)    *
1462                                ****************************************
1463                                *
1464                                *
1465          05 00210              PARSE:MD EQU     $
1466          05 00210              PARSE:MK EQU     $
1467    05 00210    6A700963        BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1468    05 00211    00000003 A      DATA     3
1469    05 00212    6A700834        BAL,LNK  CHECK1CDTENTRY   MAKE SURE 'MD(MK)' IS FIRST CMND
1470                                NXTPRM   ERRP5,,
1471                                         (INTG,*),,
1472                                         (SEQ,PMD10),,
1473    05 00213    6A7008D4                 (SEQ2,PMD15)
        05 00214    03000023D 02
        05 00215    04000218
        05 00216    02000219
        05 00217    03000021A
1474    05 00218    6A700287        BAL,LNK  ADJINT
1475                                *
1476                                *  ONLY ONE SEQ. # GIVEN: DUPLICATE IT
1477                                *
1478    05 00219    6A70028B    PMD10    BAL,LNK  REPSEQ
1479                                *
1480                                *  PUT FIRST SEQ # PAIR IN CDT AND GET 2ND PARAMETER
1481                                *
1482    05 0021A    22500003 A  PMD15    LI,P1    SEQ2             PUT 'SEQ # PAIR' PARAM IN CDT
1483    05 0021B    6A70081B            BAL,LNK  ADDCDTPARAM
1484                                    NXTPRM   *ERRP4,,
1485    05 0021C    6A7008D4                     (COM,*)
        05 0021D    01800239 02
        05 0021E    07000021F
1486                                    NXTPRM   ERRP5,,
1487                                             (INTG,*),,
1488                                             (SEQ,PMD20),,
```

```
1489   05 0021F    6A7008D4                           (SEQ2,PMD25)
       05 00220    0300023D 02
       05 00221    04000224
       05 00222    02000225
       05 00223    03000226
1490   05 00224    6A700287               BAL,LNK  ADJINT
1491                             *
1492                             *  ØNLY ØNE SEQ. # GIVEN; DUPLICATE IT
1493                             *
1494   05 00225    6A70028B      PMD20    BAL,LNK  REPSEQ
1495                             *
1496                             *  PUT 2ND SEQ # PAIR IN CDT AND GØ PRØCESS INCREMENT
1497                             *
1498   05 00226    6A70081B      PMD25    BAL,LNK  ADDCDTPARAM       PUT 'SEQ # PAIR' IN CDT
1499   05 00227    68000117               B        GET$INCREMENT     GØ PRØCESS INCR
```

```
1500                                        PAGE
1501                                ************************
1502                                *  PARSE FORM:  RN N,K   *
1503                                ************************
1504                                *
1505                                *
1506          05 00228              PARSE:RN EQU      *
1507    05 00228    6A700963                BAL,LNK   NEWCDTENTRY       BUILD NEW CDT ENTRY
1508    05 00229    00000002 A              DATA      2
1509    05 0022A    6A700834                BAL,LNK   CHECK1CDTENTRY    MAKE SURE 'RN' IS FIRST CMND
1510                                        NXTPRM    ERRP5,,
1511                                                  (INTG,*),,
1512                                                  (SEQ,PRN10),,
1513    05 0022B    6A7008D4                          (SEQ2,ILGL$SEQ2)
        05 0022C    03000023D 02
        05 0022D    04000230
        05 0022E    02000231
        05 0022F    03000129
1514    05 00230    6A700287                BAL,LNK   ADJINT
1515                                *
1516                                *  PUT SEQ # IN CDT AND GET 2ND SEQ #
1517                                *
1518    05 00231    22500002 A    PRN10     LI,P1     SEQ               PUT SEQ # IN CDT
1519    05 00232    6A70081B                BAL,LNK   ADDCDTPARAM
1520                                        NXTPRM    *ERRP4,,
1521                                                  (COM,*),,
1522                                                  (SCOL,PRN30),,
1523    05 00233    6A7008D4                          (END,PRN30)
        05 00234    03800239 02
        05 00235    07000238
        05 00236    08000242
        05 00237    00000242
1524                                        NXTPRM    ERRP5,,
1525                                                  (INTG,*),,
1526    05 00238    6A7008D4                          (SEQ,PRN20)
        05 00239    02000023D 02
        05 0023A    04000023C
```

```
        05 0023B    02000023D
1527    05 0023C    6A700287            BAL,LNK  ADJINT
1528                                *
1529                                *  PUT 2ND SEQ # IN CDT AND FINISH UP
1530                                *
1531    05 0023D    6A700818    PRN20   BAL,LNK  ADDCDTPARAM      PUT 2ND SEQ # IN CDT
1532                                    NXTPRM   *ERRP4,,
1533                                             (SCOL,ILGL$SEMICOLON),,
1534    05 0023E    6A7008D4                    (END,MASTEREXECUTIVE)
        05 0023F    028000239 02
        05 00240    080000035
        05 00241    000000301
1535                                *
1536                                *  ERROR: SECOND PARAMETER MISSING
1537                                *
1538    05 00242    6A700D99    PRN30   BAL,LNK  TYPEPERR          TYPE: '=PN:PARAM MISSING'
1539    05 00243    000000271 02        DATA     ERRP17
1540    05 00244    680000004           B        MASTERPARSER      GO TO PARSER
```

```
1541                                    PAGE
1542                                ************************************
1543                                *  PARSE FORMS:  SS N(,C(,D))      *
1544                                *                ST N(,C(,D))      *
1545                                *                JU N              *
1546                                ************************************
1547                                *
1548                                *
1549          05 00245              PARSE:SS EQU     $
1550          05 00245              PARSE:ST EQU     $
1551    05 00245    6A700963                BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1552    05 00246    00000003 A              DATA     3
1553    05 00247    68000249                B        PARSE:JU+2
1554                                *
1555                                *
1556          05 00248              PARSE:JU EQU     $
1557    05 00248    6A700963                BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1558    05 00249    00000001 A              DATA     1
1559                                        NXTPRM   ERRP5,,
1560                                                 (INTG,*),,
1561                                                 (SEQ,PSS10),,
1562    05 0024A    6A7008D4                        (SEQ2,ILGL$SEQ2)
        05 0024B    03000230 02
        05 0024C    04000241
        05 0024D    02000250
        05 0024E    03000129
1563    05 0024F    6A700287                BAL,LNK  ADJINT
1564                                *
1565                                *  PUT SEQ # IN CDT AND MAKE SURE CMND IS FIRST FOR 'SS' AND 'ST'
1566                                *
1567    05 00250    2150001E A    PSS10     CI,P1    FIRST$I:CMND     IS CMND='JU'
1568    05 00251    68100256                BGE      PSS20
1569    05 00252    6A700834                BAL,LNK  CHECK1CDTENTRY   NO - MAKE SURE 'SS(ST)' IS 1ST CMND
1570    05 00253    22500002 A              LI,P1    SEQ              PUT SEQ # IN CDT
1571    05 00254    6A70081B                BAL,LNK  ADDCDTPARAM
1572    05 00255    680001F2                B        GET$CBL#$PAIR
1573                                *
```

```
1574                                         *  PUT SEQ # FOR 'JU' IN CDT, BUT IT NEED NOT BE FIRST
1575                                         *
1576   05 00256    22500002 A   PSS20        LI,P1     SEQ                   PUT SEQ # IN CDT
1577   05 00257    6A70081B                  BAL,LNK   ADDCDTPARAM
1578                                         NXTPRM    *ERRP4,;
1579                                                   (SCOL,ILGL$SEMICOLON),;
1580   05 00258    6A7008D4                            (END,MASTEREXECUTIVE)
       05 00259    02800239 02
       05 0025A    08000035
       05 0025B    00000301
```

```
1581                                        PAGE
1582                                  **********************
1583                                  *  PARSE FORM:  RF   *
1584                                  *  PARSE FORM:  TX   *
1585                                  **********************
1586                                  *
1587                                  *
1588          05 0025C                PARSE:TX EQU      *
1589          05 0025C                PARSE:RF EQU      *
1590    05 0025C    6A700963                   BAL,LNK   NEWCDTENTRY
1591    05 0025D    00000000 A                 DATA      0
1592                                           NXTPRM    ERRC9,,
1593                                                     (SCBL,RESUME*PARSING),,
1594    05 0025E    6A7008D4                              (END,MASTEREXECUTIVE)
        05 0025F    020001C8 02
        05 00260    08000001F
        05 00261    00000301
```

```
1595                                    PAGE
1596                                ************************************
1597                                *  PARSE FORMS:  TS N(=M)  &  TS   *
1598                                *                TY N(=M)  &  TN   *
1599                                *                TC N(=M)          *
1600                                ************************************
1601                                *
1602                                *
1603          05 00262              PARSE:TC EQU     $
1604    05 00262    6A700963                BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1605    05 00263    00000003 A              DATA     3
1606                                        NXTPRM   ERRP5,,          'TC' MUST SPECIFY RECORD.
1607                                                 (INTG,PTY5),,
1608                                                 (SEQ,PTY10),,
1609    05 00264    6A7008D4                        (SEQ2,PTY15)
        05 00265    03000023D 02
        05 00266    04000281
        05 00267    02000282
        05 00268    03000283
1610                                *
1611          05 00269              PARSE:TS EQU     $
1612          05 00269              PARSE:TY EQU     $
1613    05 00269    6A700963                BAL,LNK  NEWCDTENTRY      BUILD NEW CDT ENTRY
1614    05 0026A    00000003 A              DATA     3
1615                                        NXTPRM   ERRC9,,
1616                                                 (INTG,PTY5),,
1617                                                 (SEQ,PTY10),,
1618                                                 (SEQ2,PTY15),,
1619                                                 (SCOL,*),,
1620    05 0026B    6A7008D4                        (END,*)
        05 0026C    050001C8 02
        05 0026D    04000281
        05 0026E    02000282
        05 0026F    03000283
        05 00270    08000272
        05 00271    00000272
1621    05 00272    33F000AD 02             MTW,=1   CHARPSN          SET TO RESCAN LAST CHAR
```

```
1622  05 00273  B28000AC 02          LW,T1     *CDTADR            MUST BE INTRALINE 'TS' OR 'TY' SO
1623  05 00274  48800005 02          AND,T1    XFF00               WIPE OUT CDT ENTRY JUST BUILT
1624  05 00275  B8800CAC 02          STW,T1    *CDTADR
1625  05 00276  22800024 A           LI,T1     I;TS$CMND$NMR
1626  05 00277  21500015 A           CI,P1     R;TS$CMND$NMR      IS CMND 'TS'
1627  05 00278  68300279             BE        $+2
1628  05 00279  22800028 A           LI,T1     I;TY$CMND$NMR      NO - MUST BE 'TY'
1629  05 0027A  32500008 A           LW,P1     T1                 BUILD ENTRY IN CDT FOR THIS CMND
1630  05 0027B  6A700963             BAL,LNK   NEWCDTENTRY
1631  05 0027C  00000000 A           DATA      0
1632                                 NXTPRM    ERRC9,,
1633                                           (SCOL,RESUME$PARSING),,
1634  05 0027D  6A7008D4                       (END,MASTEREXECUTIVE)
      05 0027E  02000108 02
      05 0027F  08000018
      05 00280  00000301
1635                           *
1636                           *
1637  05 00281  6A700287       PTY5      BAL,LNK   ADJINT             SCALE INTEGER TO SEQ #.
1638  05 00282  6A70028B       PTY10     BAL,LNK   REPSEQ             REPLICATE SINGLE SEQ #.
1639  05 00283  6A700834       PTY15     BAL,LNK   CHECK1CDTENTRY     INSURE TY(TS) IS FIRST COMMAND.
1640                           *
1641  05 00284  22500003 A             LI,P1     SEQ2               ADD SEQ # PAIR TO COMMAND TABLE.
1642  05 00285  6A70081B               BAL,LNK   ADDCDTPARAM
1643                           *
1644  05 00286  68000172               B         GET$COL#$PAIR      NOW GET OPTIONAL COLUMN NUMBERS.
```

```
1645                                    PAGE
1646                                ********************************
1647                                *  PARSER UTILITY ROUTINES     *
1648                                ********************************
1649                                *
1650                                *                   FORM SEQUENCE NUMBER AS INTEGER*1000.
1651                                *
1652   05 00287   32F000C1 02   ADJINT   LW,D1     PARAMBUF
1653   05 00288   23F003E8 A             MI,D1     1000
1654   05 00289   35F000C1 02           STW,D1     PARAMBUF
1655   05 0028A   E8000007 A             B          *LNK
1656                               *
1657                               *                   REPLICATE SINGLE SEQUENCE NUMBER IN PARAMBUF+1.
1658                               *
1659   05 0028B   32800C1 02    REPSEQ   LW,T1      PARAMBUF
1660   05 0028C   35800C2 02             STW,T1     PARAMBUF+1
1661   05 0028D   33100E6 02             MTW,1      PRMBUFSZ
1662   05 0028E   E8000007 A             B          *LNK
1663                               *
1664                               *
```

```
1665                                    PAGE
1666                                ************************************
1667                                *                                  *
1668                                *  BREAK-KEY INTERRUPT HANDLER     *
1669                                *  UTS ONLY.                       *
1670                                *                                  *
1671                                ************************************
1672                                *
1673          00000001                          DO        MODE=2
1674   05 0028F   091C0019C 02      BRK$KEY   PUSH      X3              SAVE POINTER OF PSD IN STACK
1675   05 00290   048C00E26                    CAL1,8    =X'06700002'    PURGE TERMINAL OUTPUT BUFFERS
1676   05 00291   6A7C00DC4                    BAL,LNK   TYPEMSG         MOVE TO A CLEAN LINE ON USER
1677   05 00292   0C000027A 02                 DATA      MSG0            TERMINAL.
1678   05 00293   33C0002C9 02                 MTW,0     XEQFLAG         IF NOT EXECUTING, GET NEXT COMMAND.
1679   05 00294   691C002FC                    BLZ       BRK99
1680   05 00295   33C000167 02                 MTW,0     STEPFLAG        IF STEPPING, SKIP DISPLAY CHECK.
1681   05 00296   693C002D2                    BNEZ      BRK80
1682   05 00297   33C0000E9 02                 MTW,0     SETFLAG         IS SYSTEM IN SET MODE.
1683   05 00298   683C002A3                    BEZ       BRK30           ZERO SAYS NO.
1684                                *
1685   05 00299   325C002C3 02                 LW,P1     INTFLAG1        IF DISPLAY FLAG SET,
1686   05 0029A   691C002D2                    BLZ       BRK80
1687   05 0029B   226C00BC9 02                 LI,P2     BA(UTSM4)+37    BUILD SEQ # INTO MESSAGE AND
1688   05 0029C   6A7C00B97                    BAL,LNK   MOVESEQ         SEND IT OUT.
1689   05 0029D   0C0000000 A                  GEN4      0,0,0,0
1690   05 0029E   208C00024 A                  AI,R1     36              ADJUST COUNT OF FULL STRING
1691   05 0029F   75BC002E9 02                 STB,R1    UTSM4
1692   05 002A0   6A7C00DC4                    BAL,LNK   TYPEMSG
1693   05 002A1   0C0002E9 02                 DATA      UTSM4
1694   05 002A2   680C002D2                    B         BRK80           NOW ASK ABOUT CONTINUE.
1695                                *
1696                                *
1697   05 002A3   223C00001 A      BRK30     LI,X1     1               EXECUTING FILE OR EDIT COMMAND.
1698   05 002A4   F246C00AC 02                 LB,X2     *CDTADR,X1      GET COMMAND NUMBER AND RETRIEVE
1699   05 002A5   683C002FC                    BEZ       BRK99           ORIGINAL EBCDIC.
1700   05 002A6   3238C0039                    LW,X1     CNAMETBL,X2     NOW CHECK DISPLAY TABLE FOR
1701   05 002A7   224C00007 A                  LI,X2     7               PRESENCE OF THIS COMMAND.
```

```
1702  05 002A8  31380317 02          CW,X1    BDISPTBL,X2
1703  05 002A9  683002AC            BE       BRK40
1704  05 002AA  644002A8            BDR,X2   $-2                IF NOT FOUND,
1705  05 002AB  680002D2            B        BRK80              ASK TO CONTINUE.
1706                          *
1707  05 002AC  21400004 A    BRK40 CI,X2    4
1708  05 002AD  68200288            BLE      BRK50
1709  05 002AE  325002C3 02         LW,P1    INTFLAG1           THESE COMMANDS TAKE SINGLE SEQUENCE
1710  05 002AF  691002D2            BLZ      BRK80              NUMBER - DE,FD,FT
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           THESE COMMANDS TAKE A DOUBLE SEQ. #
1721  05 002B9  691002D2            BLZ      BRK80              DISPLAY
1722  05 002BA  22600BF0 02         LI,P2    BA(UTSM5)+24
1723  05 002BB  6A700B97            BAL,LNK  MOVESEQ            SET UP DDD.DD (
1724  05 002BC  404D0000 A          GEN4     BL,LP,0,0
1725  05 002BD  30600008 A          AW,P2    R1                 INCR MSG BYTE ADDR
1726  05 002BE  20B00017 A          AI,R1    23                 AND MSG LENGTH
1727  05 002BF  325002C4 02         LW,P1    INTFLAG2           IF SECOND SEQ. # NOT SET UP,
1728  05 002C0  681002CD            BGEZ     BRK60              WE MUST BE DELETING.
1729  05 002C1  723003O7 02         LB,X1    UTSM7
1730  05 002C2  30B00003 A          AW,R1    X1                 THEREFORE, INSERT DELETING
1731  05 002C3  22400001 A          LI,X2    1                  MESSAGE.
1732  05 002C4  7258O307 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              ADJUST BYTE COUNT OF TOTAL
1738  05 002CA  6A700DC4            BAL,LNK  TYPEMSG            MESSAGE.
```

```
1739  05 002CB  00000002F6 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 TYPEMSG         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 TYPEMSG         RETURN CARRIAGE.
1752  05 002D6  0000027A 02            DATA    MSGO
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  08100198 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  228FFFFF 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   EQU     $
```

```
1776   05 002ED                              D01       MODE=2
1777   05 002ED     32B00E27                 LW,R1     L(X'00200000')     IF OPEN FOR OUTPUT,
1778   05 002EE     31B00000 X               CW,R1     F:EO
1779   05 002EF     684002F1                 BAZ       $+2
1780   05 002F0     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     31B00000 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      *
                              *                                     *
                              ***************************************
                              *
                              *
        05 00301              MASTEREXECUTIVE   EQU $
05 00301    22800049 02               LI,T1     CDT+1              SET CDTADR=FIRST COMMAND IN CDT
05 00302    358000AC 02               STW,T1    CDTADR
05 00303    3280016A 02               LW,T1     SVBPFLAG           RESTORE LAST DFLT VALUE OF BPFLAG
05 00304    35800023 02               STW,T1    BPFLAG
        00000001                      DO        MODE=2
05 00305    22800001 A                LI,T1     1
05 00306    358002C9 02               STW,T1    XEQFLAG
05 00307    330000E9 02               MTW,0     SETFLAG            ARE WE IN SET MODE
05 00308    6930030C                  BNEZ      RESTART$EXECUTIVE YES, DON'T RESET TAB FLAGS
05 00309    22800000 A               LI,T1     0
05 0030A    358002C6 02              STW,T1    TABCFLAG
05 0030B    358002C7 02              STW,T1    TABXFLAG
                                      FIN
                              *
                              *
                              *
        05 0030C              RESTART$EXECUTIVE EQU $              (INTRALINE CMND LOOP ENTERS HERE)
05 0030C    22300000 A                LI,X1     0                  INDICATE 'ALL' MODE IS
05 0030D    35300021 02               STW,X1    ALLOK              POTENTIALLY LEGAL
05 0030E    22300001 A                LI,X1     1                  GET NUMBER OF COMMAND
05 0030F    F24600AC 02               LB,X2     *CDTADR,X1
05 00310    6830034A                  BEZ       EXC50              IS CMND=0 (END OF CDT)
05 00311    21400004 A                CI,X2     FIRST$FICMND       IS IT A FILE COMMAND
05 00312    69100322                  BL        EXC6               NO - ITS A CONTROL CMND
```

```
1836  05 00313  2140000B A          CI,X2   FIRST$R:CMND     NO - IS IT A FILE COMMAND
1837  05 00314  69100325            BL      EXC5             YES - SKIP TEST
1838  05 00315  330000B6 02         MTW,0   FILETYPE         NO - IS INP FILE PRESENT AND KEYED
1839  05 00316  69100347            BLZ     EXC40            NO - ERROR
1840  05 00317  2140001E A          CI,X2   FIRST$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  330000E9 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  33000167 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  33000167 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                DO      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  33000020 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$FLAG         TURN OFF 'CT' FLAG
1877  05 00331  F28000AC 02       LB,T1   *CDTADR         INCR CDTADR TO NEXT COMMAND
1878  05 00332  668000AC 02       AWM,T1  CDTADR
1879  05 00333  68000300          B       RESTART$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  68000328          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  33000167 02     EXC20   MTW,0   STEPFLAG        IS SYSTEM IN STEP MODE
1893  05 00338  69300328          BNEZ    EXC10
1894  05 00339  33000E9 02       MTW,0   SETFLAG         NO - MUST SET LOOP BE INITIALIZED
1895  05 0033A  69100328          BLZ     EXC10           NO - GO EXECUTE I:CMND
1896  05 0033B  228FFFFF 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 SETEOD          SET EOD MARKER
1904  05 00343  68000328          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    ERRC4
```

```
05 00346  68000004             B         MASTERPARSER      EXIT TO PARSER
                       *
                       *  ERROR: NO SOURCE FILE NAMED
                       *
05 00347  6A700DC4     EXC40   BAL,LNK   TYPEMSG           TYPE: '-NO FILE NAMED'
05 00348  000001F2 02          DATA      ERRM13
05 00349  68000004             B         MASTERPARSER      EXIT TO PARSER
                       *
                       *  END OF CDT: IF IN SET OR STEP MODES, GO TO APPROPRIATE LOOP
                       *
05 0034A  33000167 02  EXC50   MTW,0     STEPFLAG          IS SYSTEM IN STEP MODE
05 0034B  693006AD             BNEZ      STEP$LOOP         YES - GO TO STEP LOOP
05 0034C  3300019A 02          MTW,0     TTYIMGSZ          WAS INPUT LINE NULL
05 0034D  68300351             BEZ       EXC55             YES - ERROR
05 0034E  330000E9 02          MTW,0     SETFLAG           IS SYSTEM IN SET MODE
05 0034F  69300748             BNEZ      SET$LOOP          YES - GO TO SET LOOP
05 00350  68000004             B         MASTERPARSER      EXIT TO PARSER
                       *
                       *  ERROR: NULL COMMAND
                       *
      05 00351         EXC55    EQU      $
05 00351  68000004             B         MASTERPARSER      EXIT TO PARSER
                       *
                       *  FILE COMMANDS CAN ONLY APPEAR ONE PER LINE
                       *
      05 00351         CMNDTBL  EQU      $-1
05 00352  6AD0037E             BAL,F:LNK F:BLANK$PRESERV     1: BP
05 00353  6AD00534             BAL,F:LNK S(MASTERPARSER,F:TA) TA
05 00354  6AD00481             BAL,F:LNK F:CR                3: CR
05 00355  6AD00390             BAL,F:LNK F:BUILD             4: BUILD
05 00356  6AD003D9             BAL,F:LNK F:COPY              5: COPY
05 00357  6AD00497             BAL,F:LNK F:DELETE            6: DELETE
05 00358  6AD004A3             BAL,F:LNK F:EDIT              7: EDIT
05 00359  6AD004BA             BAL,F:LNK F:END               8: END
05 0035A  6AD004BE             BAL,F:LNK F:MERGE             9: MERGE
05 0035B  6AD0048C             BAL,F:LNK F:RP               10: RP
                       *
```

```
1947                                *  RECORD COMMANDS CAN ONLY APPEAR ONE PER LINE
1948                                *
1949   05 0035C   6AD00058C             BAL,F:LNK R:DELETE          11: DE
1950   05 0035D   6AD00596              BAL,R:LNK R:FIND$DELETE     12: FD
1951   05 0035E   6AD00598              BAL,R:LNK R:FIND$TYPE       13: FT
1952   05 0035F   6AD005DB              BAL,R:LNK R:INSERT          14: IN
1953   05 00360   6AD005D8              BAL,R:LNK R:INSERT$SUP$SEQ  15: IS
1954   05 00361   6AD00614              BAL,R:LNK R:MOVE$DELETE     16: MD
1955   05 00362   6AD00616              BAL,R:LNK R:MOVE$KEEP       17: MK
1956   05 00363   6AD00686              BAL,R:LNK R:RENUMBER        18: RN
1957   05 00364   6AD00699              BAL,R:LNK R:SET$STEP        19: SS
1958   05 00365   6AD0069B              BAL,R:LNK R:SET$STEP$TYPE   20: ST
1959   05 00366   6AD006C5              BAL,R:LNK R:TYPE$SUP$SEQ    21: TS
1960   05 00367   6AD006C3              BAL,R:LNK R:TYPE            22: TY
1961   05 00368   6AD006C3              BAL,R:LNK R:TYPE$COMPRESSED 23: TC
1962   05 00369   6AD00594              BAL,R:LNK R:FIND$SEQUENCE   24: FS
1963   05 0036A   6AD00545              BAL,R:LNK R:COMMENTARY      25: CM=CT
1964   05 0036B                         RES     4
1965                                *
1966                                *  INTRALINE COMMANDS MAY BE COMPOUNDED ON ONE LINE
1967                                *
1968   05 0036F   6AD0072D              BAL,I:LNK I:SET             30: SE (MUST BE FIRST I:CMND)
1969   05 00370   6AD00773              BAL,I:LNK I:DELETE          31: D
1970   05 00371   6AD00787              BAL,I:LNK I:OVERWR$EXTEND   32: E
1971   05 00372   6AD00798              BAL,I:LNK I:FOLLOW$BY       33: F
1972   05 00373   6AD007A7              BAL,I:LNK I:SHIFT$LEFT      34: L
1973   05 00374   6AD007B1              BAL,I:LNK I:OVERWRITE       35: O
1974   05 00375   6AD007BC              BAL,I:LNK I:PRECEDE$BY      36: P
1975   05 00376   6AD007CA              BAL,I:LNK I:SHIFT$RIGHT     37: R
1976   05 00377   6AD007D4              BAL,I:LNK I:SUBSTITUTE      38: S
1977   05 00378   6AD007ED              BAL,I:LNK I:JUMP            39: JU
1978   05 00379   6AD00800              BAL,I:LNK I:NO$CHANGE       40: NO
1979   05 0037A   6AD00808              BAL,I:LNK I:REVERSE$BPFLAG  41: RF
1980   05 0037B   6AD00810              BAL,I:LNK I:TYPE$SUP$SEQ    42: TS
1981   05 0037C   6AD0080C              BAL,I:LNK I:TYPE            43: TY
1982   05 0037D   6AD00814              BAL,I:LNK I:TYPEX           44: TX
```

```
                                  PAGE
                                  ****************************************
                                  *  FILE COMMAND: SET BLANK PRESERVATION  *
                                  ****************************************
                                  *
                                  *
        05 0037E                  F:BLANK$PRESERV   EQU $
05 0037E   22300005 A                      LI,X1     5               SET TO GET PARAMETER FROM CDT
05 0037F   F24600AC 02                     LB,X2     *CDTADR,X1
05 00380   B28800AC 02                     LW,T1     *CDTADR,X2      GET 'ON' OR 'OFF' AS A TEXTC-STRING
05 00381   3180038E                        CW,T1     BPVON
05 00382   69300386                        BNE       BPV5            IS STRING='ON'
05 00383   22800001 A                      LI,T1     1               YES - SET BPFLAG=1
05 00384   3580016A 02                     STW,T1    SVBPFLAG
05 00385   E800000D A                      B         *F:LNK          EXIT
                                  *
                                  *  TEST FOR 'OFF'
                                  *
05 00386   3180038F               BPV5     CW,T1     BPVOFF
05 00387   6930038B                        BNE       BPV10           IS STRING='OFF'
05 00388   22800000 A                      LI,T1     0               YES - SET BPFLAG=0
05 00389   3580016A 02                     STW,T1    SVBPFLAG
05 0038A   E800000D A                      B         *F:LNK          EXIT
                                  *
                                  *  ERROR: NOT ON OR OFF
                                  *
05 0038B   6A700DC4               BPV10    BAL,LNK   TYPEMSG         TYPE: '-NOT ON/OFF'
05 0038C   000001E3 02                     DATA      ERRM5
05 0038D   E800000D A                      B         *F:LNK          EXIT
                                  *
                                  *
05 0038E   02D6D540 A             BPVON    TEXTC     'ON'
05 0038F   03D6C6C6 A             BPVOFF   TEXTC     'OFF'
```

```
2016                                    PAGE
2017                            ************************
2018                            *  FILE COMMAND: BUILD  *
2019                            ************************
2020                            *
2021                            *
2022          05 00390          F:BUILD  EQU      *
2023    05 00390    6A700D78              BAL,LNK  TESTEDITACTIVE    CHECK IF EDIT FILE ACTIVE
2024    05 00391    22300005 A            LI,X1    5
2025    05 00392    F2560AC  02           LB,P1    *CDTADR,X1        SET P1=ADR OF FID IN CDT
2026    05 00393    30500AC  02           AW,P1    CDTADR
2027    05 00394                          DO1      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                          DO1      MODE=2
2032    05 00397    041002CC 02           CAL1,1   NOPROMPT+FPT
2033    05 00398    225003E8 A            LI,P1    DFLTSEQ           NO - 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    F2B600AC 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    F2460AC  02           LB,X2    *CDTADR,X1        YES - SET P1=SEQ # FROM CDT
2040    05 0039F    B2580AC  02           LW,P1    *CDTADR,X2
2041                            *
2042                            *  PROCESS INCREMENT PARAMETER
2043                            *
2044    05 003A0    22300008 A  BLD5      LI,X1    8
2045    05 003A1    F2B600AC 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    F2460AC  02           LB,X2    *CDTADR,X1        YES - SET T1=INCR FROM CDT
2049    05 003A5    B2880AC  02           LW,T1    *CDTADR,X2
2050                            *
2051          00000C01                    DO       MODE=2
2052    05 003A6    33000 2C1 02  BLD08   MTW,0    BUILDFLAG
```

```
2053  05 003A7  683003B2              BEZ     BLD12
2054  05 003A8  21C00000 A  BLD10     CI,R2   0                 HAVE WE BUILT FIRST LINE
2055  05 003A9  683003B2              BEZ     BLD12               YES--BYPASS RE-OPEN
2056  05 003AA  6A700B33              BAL,LNK CLOSE               CLOSE AND SAVE BUILD FILE
2057  05 003AB  46500000C A           XW,P1   R2                POSITION FID FOR OPENING
2058  05 003AC  6A700BC8              BAL,LNK OPEN              REOPEN IN INPUT SO ESC LEAVES INTACT
2059  05 003AD  32500000C A           LW,P1   R2                RESET P1 TO SEQ. #
2060  05 003AE  22C00001 A            LI,R2   1                 NOW MARK FILE AS IF WE ARE IN
2061  05 003AF  35C000B6 02           STW,R2  FILETYPE            EDIT MODE - RECORD COMM O.K.
2062  05 003B0  22C00000 A            LI,R2   0                 AND R2 SO WILL NOT RE-OPEN EO
2063  05 003B1  35C002C5 02           STW,R2  TABERRFLAG        EDIT WOULD DO THIS, SO SHALL BUILD
2064                                  ELSE
2065                          *S* BLD08 EQU   $
2066                          *S* BLD10 EQU   $
2067                                  FIN
2068                              *   TYPE NEXT SEQ # AND READ INPUT LINE
2069                              *
2070  05 003B2  6A700DDB      BLD12   BAL,LNK TYPESEQ  TYPE 'DDDD,DDD'
2071  05 003B3  40080000 A            GEN4    BL,EOM,0,0
2072        00000001                  DO      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  32400000B A            LW,X2    R1           GET BYTE CNT., INTO INDEX REG.
2079  05 003B9  204FFFFF 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  223000004 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.
```

```
2090  05 003C3  35B0000E7 02        STW,R1   RECSIZE
2091  05 003C4  21B00008C A        CI,R1    MAXCLMN
2092  05 003C5  682003C8           BLE      BLD25
2093  05 003C6  6A700DC4           BAL,LNK  TYPEMSG          NO - TYPE: !=>OVERFLOW!
2094  05 003C7  000001DC 02        DATA     ERRM3
2095                               *
2096                               *  WRITE INPUT LINE AND INCREMENT SEQ. #
2097                               *
2098  05 003C8  6A700A56           BLD25    BAL,LNK  SETEOD           FINDS COL. OF LAST NON-BLANK
2099  05 003C9  6A700E1F           BAL,LNK  WRITEANDOM       WRITE CARD IMAGE; P1 CONTAINS SEQ. #
2100  05 003CA  30500008 A         AW,P1    T1               INCREMENT SEQ. #
2101  05 003CB  315000BF 02        CW,P1    MAXSEQ           IS SEQ. NO. TOO BIG
2102  05 003CC  682003A8           BLE      BLD10            NO.  GO READ MORE INPUT
2103  05 003CD  6A700DC4           BAL,LNK  TYPEMSG          YES.
2104  05 003CE  0000021D 02        DATA     ERRM20
2105  05 003CF  21C00000 A         CI,R2    0                IF THIS IS BANG BUILD,
2106  05 003D0  683003D2           BEZ      BLD30            R2 IS NON-ZERO
2107  05 003D1  6A700B33           BAL,LNK  CLOSE
2108                               *
2109                               *  NULL INPUT LINE: EXIT
2110                               *
2111        05 003D2               BLD30    EQU      $
2112        00000001                        DO       MODE=2
2113  05 003D2  330002C1 02        MTW,0    BUILDFLAG        IF ENTERED BY BUILD COMMAND, EXIT
2114  05 003D3  683004BA           BEZ      F:END            TO TEL.
2115                               FIN
2116  05 003D4  E800000D A         B        *F:LNK           EXIT
2117                               *
2118                               *  ERROR: NAMED FILE ALREADY EXISTS
2119                               *
2120  05 003D5  6A700DC4           BLD40    BAL,LNK  TYPEMSG          TYPE: !=FILE EXISTS; CAN'T BUILD'
2121  05 003D6  000001FA 02        DATA     ERRM15
2122  05 003D7  6A700B33           BAL,LNK  CLOSE            CLOSE F:EI
2123  05 003D8  6800003D2          B        BLD30
```

```
2124                                        PAGE
2125                                *************************
2126                                *  FILE COMMAND: COPY   *
2127                                *************************
2128                                *
2129                                *
2130          05 003D9              F:COPY    EQU       *
2131    05 003D9    6A700D78                  BAL,LNK   TESTEDITACTIVE    CHECK IF EDIT FILE ACTIVE
2132    05 003DA    22100000 A                LI,X3     0                 INITIALIZE FLAG FOR
2133    05 003DB    351000AE 02               STW,X3    COPYFL              FID1=FID2
2134    05 003DC    22300005 A                LI,X1     5                 OBTAIN FID 1 AND FID 2
2135    05 003DD    F25600AC 02               LB,P1     *CDTADR,X1          AS
2136    05 003DE    305000AC 02               AW,P1     CDTADR            BYTE
2137    05 003DF    22300009 A                LI,X1     9                   ADDRESSES
2138    05 003E0    F26600AC 02               LB,P2     *CDTADR,X1          IN
2139    05 003E1    306000AC 02               AW,P2     CDTADR              REGISTERS
2140    05 003E2    25500002 A                SLS,P1    2                   P1 AND
2141    05 003E3    25600002 A                SLS,P2    2                   P2
2142                                *
2143                                *  SEARCH LOOP TO DETERMINE IF FID1 = FID2
2144                                *
2145    05 003E4    20500001 A    CPY1        AI,P1     1
2146                                *                     (OK TO BYPASS TEXTC BYTE IN COMPR)
2147    05 003E5    20600001 A                AI,P2     1
2148    05 003E6    723A0000 A                LB,X1     0,P1              GET FID 1 BYTE
2149    05 003E7    683003E8                  BEZ       CPY1A               QUIT WHEN END OF FID
2150    05 003E8    713C0000 A                CB,X1     0,P2
2151    05 003E9    693003ED                  BNE       CPY1B               OR WHEN NOT EQUAL
2152    05 003EA    683003E4                  B         CPY1              LOOP
2153                                *
2154                                *  FINISH FID COMPARISON - FID STRING HAS ENDED
2155                                *
2156    05 003EB    713C0000 A    CPY1A       CB,X1     0,P2              CHECK LAST BYTE
2157    05 003EC    683004 3E                 BE        CPY32
2158    05 003ED    22300007 A    CPY1B       LI,X1     7                 FIND OUT WHETHER ON
2159    05 003EE    F24600AC 02               LB,X2     *CDTADR,X1        OR OVER SPECIFIED
2160    05 003EF    B28800AC 02               LW,T1     *CDTADR,X2        T1='ON' OR 'OVER'
```

```
2161  05 003F0  318000013 02          CW,T1    X:BVER
2162  05 003F1  693000430             BNE      CPY30          NOT EQUAL ==> BN
2163                                  *
2164                                  *  BPEN FOR COPY A OVER B
2165                                  *
2166  05 003F2  223000005 A  CPY2     LI,X1    5
2167  05 003F3  F256000AC 02          LB,P1    *CDTADR,X1     SET P1=ADR OF FID1 IN CDT
2168  05 003F4  305000AC 02          AW,P1    CDTADR
2169  05 003F5  222000000 A          LI,X4    0              SET X4=0 TO SHOW FILE UNKEYED
2170  05 003F6  6A700BCC             BAL,LNK  BPEN1          BPEN INPUT FILE WITH THIS FID
2171  05 003F7  698000046E           BCS,8    CPY40          DOES FILE EXIST
2172  05 003F8  694003FA             BCS,4    *+2            YES - IS FILE KEYED
2173  05 003F9  222000001 A          LI,X4    1              YES - SET X4=1 TO SHOW FILE KEYED
2174  05 003FA  223000009 A          LI,X1    9
2175  05 003FB  F256000AC 02          LB,P1    *CDTADR,X1     SET P1=ADR OF FID2 IN CDT
2176  05 003FC  305000AC 02          AW,P1    CDTADR
2177  05 003FD  6A700BEF             BAL,LNK  BPEN2          BPEN COPY FILE WITH THIS FID
2178  05 003FE  698000401            BCS,8    CPY3           FILE 2 NO EXIST YET
2179  05 003FF  6A700B37             BAL,LNK  CLOSE3         .. EXISTS-RELEASE GRANS
2180  05 00400  6A700BEB             BAL,LNK  BPEN3          BPEN FOR OUTPUT
2181                                  *
2182                                  *  FINISH INITIALIZATION AND PROCESS PARAMETER 4
2183                                  *
2184  05 00401  6A700DC4    CPY3     BAL,LNK  TYPEMSG        TYPE: '..COPYING'
2185  05 00402  000000278 02          DATA     MSG1
2186  05 00403  223000004 A          LI,X1    10
2187  05 00404  F2C600AC 02          LB,R2    *CDTADR,X1     IS 'STARTING SEQ #' PARAM PRESENT
2188  05 00405  693000410             BNEZ     CPY10          YES - GO COPY AND RESEQ
2189                                  *
2190                                  *  COPY SOURCE FILE THROUGH EOF
2191                                  *
2192  05 00406  6A700C7F    CPY5     BAL,LNK  READSEQUEN     READ SOURCE RECORD
2193  05 00407  318000E28             CW,R1    L(EOF)         IS IT AN EOF
2194  05 00408  683000428             BE       CPY20          YES - GO FINISH UP
2195  05 00409  212000001 A          CI,X4    1
2196  05 0040A  693000471             BNE      CPY50
2197  05 0040B  325000008 A  CPY5A    LW,P1    R1             GET SEQ # IN P1
```

```
2198  05 0040C  6A700DFD          BAL,LNK  WRITE2          WRITE RECORD IN COPY FILE
2199        00000001              DO       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                        *  PROCESS STARTING SEQ. # AND INCREMENT PARAMETERS
2206                        *
2207  05 00410  22300008 A  CPY10  LI,X1    11
2208  05 00411  F2460AAC 02      LB,X2    *CDTADR,X1
2209  05 00412  B2580AAC 02      LW,P1    *CDTADR,X2       SET P1=STARTING SEQ #
2210  05 00413  22803E8 A        LI,T1    1000                 T1=1 (DEFAULT INCR)
2211  05 00414  2230000C A       LI,X1    12
2212  05 00415  F2C60AAC 02      LB,R2    *CDTADR,X1       GET PARAM4 TYPE
2213  05 00416  68300419          BEZ      CPY15            TEST IF PARAM4 PRESENT
2214  05 00417  20300001 A       AI,X1    1
2215  05 00418  F2460AAC 02      LB,X2    *CDTADR,X1       YES - SET T1=INCR FROM CDT
2216  05 00419  B2880AAC 02      LW,T1    *CDTADR,X2
2217                        *
2218                        *  COPY AND RESEQUENCE SOURCE FILE THROUGH EOF
2219                        *
2220  05 0041A  6A700C7F    CPY15  BAL,LNK  READSEQUEN       READ SOURCE RECORD
2221  05 0041B  31B00E28          CW,R1    L(EOF)           IS IT AN EOF
2222  05 0041C  68300428          BE       CPY20            YES - GO FINISH UP
2223  05 0041D  6A700DFD          BAL,LNK  WRITE2           WRITE RECORD IN COPY FILE
2224  05 0041E  69800471          BCS,8    CPY50            DOES RECORD ALREADY EXIST
2225        00000001              DO       MODE=2
2226  05 0041F  35B002C3 02      STW,R1   INTFLAG1
2227  05 00420  35500AC4 02      STW,P1   INTFLAG2
2228                              FIN
2229  05 00421  30500008 A       AW,P1    T1               NO - INCR SEQ #
2230  05 00422  31500ABF 02      CW,P1    MAXSEQ           IS SEQ. NO. TOO BIG
2231  05 00423  68200419          BLE      CPY15            NO.
2232  05 00424  6A700DC4          BAL,LNK  TYPEMSG          YES.
2233  05 00425  00000210 02      DATA     ERRM20
2234  05 00426  33000AAE 02      MTW,0    COPYFL           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        *F:LNK         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        *F:LNK         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  3050000AC 02               AW,P1    CDTADR
2253  05 00433  6A700BEF                   BAL,LNK  OPEN2          OPEN IN=OUT=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  3050000AC 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  69800468                   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  3310000AE 02               MTW,1    COPYFL         SET TO SHOW FID1=FID2
2269  05 00440  F25600AC 02                LB,P1    *CDTADR,X1
2270  05 00441  3050000AC 02               AW,P1    CDTADR         P1 = FILE NAME
2271  05 00442  F2300005 A                 LB,X1    *P1            BYTE CNT OF FILE NAME
```

```
05 00443   2530007E A        SLS,X1   -2            BYTE TO WORD COUNT
05 00444   20300001 A        AI,X1    1             GET NEXT WORD
05 00445   30500003 A        AW,P1    X1
05 00446   F2300005 A        LB,X1    *P1           BYTE CNT OF ACCOUNT
05 00447   2530007E A        SLS,X1   -2
05 00448   20300001 A        AI,X1    1             POINT TO PASSWORD
05 00449   30500003 A        AW,P1    X1
05 0044A   B2300005 A        LW,X1    *P1           FETCH PASSWORD
05 0044B   6930047E          BNEZ     CPY60         PASSWORD GIVEN - ERROR
05 0044C   22300009 A        LI,X1    9             NOW GO DO SAME FOR FID2
05 0044D   F25600AC 02       LB,P1    *CDTADR,X1
05 0044E   305000AC 02       AW,P1    CDTADR        P1 = FILE NAME
05 0044F   F2300005 A        LB,X1    *P1           BYTE CNT OF FILE NAME
05 00450   2530007E A        SLS,X1   -2            BYTE TO WORD COUNT
05 00451   20300001 A        AI,X1    1             GET NEXT WORD
05 00452   30500003 A        AW,P1    X1
05 00453   F2300005 A        LB,X1    *P1           BYTE CNT OF ACCOUNT
05 00454   2530007E A        SLS,X1   -2
05 00455   20300001 A        AI,X1    1             POINT TO PASSWORD
05 00456   30500003 A        AW,P1    X1
05 00457   B2300005 A        LW,X1    *P1           FETCH PASSWORD
05 00458   6930047E          BNEZ     CPY60         PASSWORD GIVEN - ERROR
                       *
05 00459   22300005 A        LI,X1    5             OBTAIN FID 1 AND FID 2
05 0045A   F25600AC 02       LB,P1    *CDTADR,X1      AS
05 0045B   305000AC 02       AW,P1    CDTADR        BYTE
05 0045C   6A700BEB          BAL,LNK  OPEN3         OPEN FOR OUTPUT
05 0045D   22300005 A        LI,X1    5             OBTAIN FID1 AND FID2
05 0045E   F25600AC 02       LB,P1    *CDTADR,X1      AS
05 0045F   305000AC 02       AW,P1    CDTADR        BYTE
05 00460   22200000 A        LI,X4    0             X4=4 MEANS NOT KEYED
05 00461   6A700BCC          BAL,LNK  OPEN1         OPEN1 OPEN1 IN, CONTINUE
05 00462   6980046A          BCS,8    CPY36         IF FILE DOES NOT EXIST
05 00463   69400401          BCS,4    CPY3          IS FILE KEYED
05 00464   22200001 A        LI,X4    1               MARK AS KEYED
05 00465   68000401          B        CPY3          GO TO BODY OF COPY
                       *
```

```
2309                                *  ERROR: COPY FILE EXISTS AND PARAMETER 2 IS 'ON'
2310                                *
2311  05 00466  6A700DC4     CPY35  BAL,LNK  TYPEMSG          TYPE: '=P2:FILE EXISTS'
2312  05 00467  00000250 02         DATA     ERRP13
2313  05 00468  6A700B35             BAL,LNK  CLOSE2
2314  05 00469  E8000000D A          B        *F:LNK           EXIT
2315                                *
2316                                *  ERROR: SOURCE FILE NAMED DOESN'T EXIST
2317                                *
2318  05 0046A  6A700B37     CPY36  BAL,LNK  CLOSE3
2319  05 0046B  68000470             B        CPY40
2320  05 0046C  6A700B35     CPY37  BAL,LNK  CLOSE2           CLOSE E0 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  E8000000D A          B        *F:LNK           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  3300000AE 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  F256000AC 02        LB,P1    *CDTADR,X1      SET P1=ADR OF FID2 IN CDT
2336  05 00479  3050000AC 02        AW,P1    CDTADR
2337  05 0047A  6A700B6A             BAL,LNK  DELETEFILE      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  E8000000D A  CPY58  B        *F:LNK           EXIT
2341  05 0047E  6A700DC4     CPY60  BAL,LNK  TYPEMSG          TYPE: 'PASSWORD ERROR'
2342  05 0047F  00000214 02         DATA     ERRM19
2343  05 00480  E8000000D A          B        *F:LNK           EXIT
```

```
2344                                    PAGE
2345                                ******************************************************
2346                                *  FILE COMMAND: SET TERMINATOR (X'15') MODE          *
2347                                ******************************************************
2348                                *
2349                                *
2350          05 00481              F:CR      EQU     $
2351    05 00481    22300005 A                LI,X1   5
2352    05 00482    F24600AC 02               LB,X2   *CDTADR,X1
2353    05 00483    B28800AC 02               LW,T1   *CDTADR,X2      GET PARAMETER AS A TEXTC STRING.
2354    05 00484    31800388                  CW,T1   BPV0N           CHECK FOR 'ON'
2355    05 00485    69300489                  BNE     CR5
2356                                *
2357    05 00486    22800000 A                LI,T1   0               TURN 'ON'
2358    05 00487    358000AF 02   CR3         STW,T1  CRFLAG          SET FLAG TO INCLUDE TERMINATOR
2359    05 00488    E800000D A                B       *F:LNK          IN OUTPUT RECORDS.
2360                                *
2361    05 00489    31800388F     CR5         CW,T1   BPV0FF          CHECK FOR 'OFF'
2362    05 0048A    68300487                  BE      CR3             TURN 'OFF'
2363    05 0048B    68000388                  B       BPV10           ERROR: -NOT ON/OFF
2364                                ******************************************************
2365                                *  FILE COMMAND: RECORD SIZE PRESERVATION             *
2366                                ******************************************************
2367                                *
2368          05 00486              F:RP      EQU     $
2369    05 0048C    22300005 A                LI,X1   5
2370    05 0048D    F24600AC 02               LB,X2   *CDTADR,X1
2371    05 0048E    B28800AC 02               LW,T1   *CDTADR,X2      GET PARAMETER AS A TEXTC STRING
2372    05 0048F    31800388E                 CW,T1   BPV0N           CHECK FOR 'ON'
2373    05 00490    69300494                  BNE     RP5
2374    05 00491    22800000 A                LI,T1   0               TURN 'ON'
2375    05 00492    358001A0 02   RP3         STW,T1  RP$FLAG         FLAG TO RETAIN RECORD SIZE
2376    05 00493    E800000D A                B       *F:LNK          EVEN WITH TRAILING BLANKS
2377                                *
2378    05 00494    31800388F     RP5         CW,T1   BPV0FF          CHECK FOR 'OFF'
2379    05 00495    68300492                  BE      RP3             TURN 'OFF'
2380    05 00496    68000388                  B       BPV10           ERROR: -NOT ON/OFF
```

```
2381                                    PAGE
2382                                ****************************
2383                                *  FILE COMMAND: DELETE    *
2384                                ****************************
2385                                *
2386                                *
2387          05 00497              F:DELETE EQU     $
2388    05 00497    6A700D78                BAL,LNK  TESTEDITACTIVE
2389    05 00498    22300005 A              LI,X1    5
2390    05 00499    F256000AC 02            LB,P1    *CDTADR,X1          SET P1=ADR OF FID IN CDT
2391    05 0049A    3U5000AC 02             AW,P1    CDTADR
2392    05 0049B    6A700B6A                BAL,LNK  DELETEFILE          DELETE FILE
2393    05 0049C    698004A0                BCS,8    DLT10               DID FILE EXIST
2394                                *
2395                                *  TYPE MESSAGE AND EXIT
2396                                *
2397    05 0049D    6A700DC4    DLT5        BAL,LNK  TYPEMSG             TYPE: '..DELETED'
2398    05 0049E    0U000281 02             DATA     MSG3
2399    05 0049F    E800000D A              B        *F:LNK              YES = EXIT
2400                                *
2401                                *  ERROR: FILE TO DELETE DOESN'T EXIST
2402                                *
2403    05 004A0    6A700DC4    DLT10       BAL,LNK  TYPEMSG             TYPE: '=NO SUCH FILE'
2404    05 004A1    0U0001F6 02             DATA     ERRM14
2405    05 004A2    E800000D A              B        *F:LNK
```

```
2406                                        PAGE
2407                                *************************
2408                                *  FILE C0MMAND: EDIT   *
2409                                *************************
2410                                *
2411                                *
2412          05 004A3              F:EDIT    EQU     $
2413    05 004A3    330000B6 02               MTW,0   FILETYPE          FILETYPE=-1 NEVER 0PENED
2414    05 004A4    691004A6                  BLZ     EDT5                      +1 0PENED AS IN0UT, KEYED
2415    05 004A5    6A700B33                  BAL,LNK CL0SE             CL0SE FILE IF EVER 0PENED
2416                                *
2417                                *  0PEN FILE AND SET FILE TYPE
2418                                *
2419    05 004A6    22300005 A    EDT5        LI,X1   5
2420    05 004A7    F25600AC 02               LB,P1   *CDTADR,X1        SET P1=ADR 0F FID IN CDT
2421    05 004A8    305000AC 02               AW,P1   CDTADR
2422    05 004A9    6A700BC8                  BAL,LNK 0PEN              0PEN FILE
2423    05 004AA    698004B1                  BCS,8   EDT10             D0ES FILE EXIST
2424    05 004AB    694004B6                  BCS,4   EDT20             YES - IS IT KEYED
2425    05 004AC    22800001 A                LI,T1   1                 YES - SET FILETYPE=1
2426    05 004AD    358000B6 02               STW,T1  FILETYPE
2427          00000001                        D0      M0DE=2
2428    05 004AE    22800000 A                LI,T1   0
2429    05 004AF    358002C5 02               STW,T1  TABERRFLAG
2430                                          FIN
2431    05 004B0    E800000D A                B       *F:LNK            EXIT
2432                                *
2433                                *  ERR0R: S0URCE FILE D0ESN'T EXIST
2434                                *
2435    05 004B1    6A700DC4      EDT10       BAL,LNK TYPEMSG           TYPE: '-N0 SUCH FILE'
2436    05 004B2    000001F6 02               DATA    ERRM14
2437    05 004B3    228FFFFF A    EDT15       LI,T1   -1                SH0W UNSUCCESSFUL 0PEN
2438    05 004B4    358000B6 02               STW,T1  FILETYPE
2439    05 004B5    E800000D A                B       *F:LNK
2440                                *
2441                                *  FILE EXISTS BUT IS N0T KEYED
2442                                *
```

```
2443   05 00486   6A700B33     EDT20    BAL,LNK  CLOSE      CLOSE FILE
2444   05 00487   6A700DC4              BAL,LNK  TYPEMSG    TYPE: !.FILE NOT KEYED, MUST COPY!
2445   05 00488   000001EB 02           DATA     ERRM12
2446   05 00489   6B0004B3              B        EDT15      EXIT.
```

```
                                                PAGE
                                       ***********************
                                       *  FILE COMMAND: END  *
                                       ***********************
                                       *
                                       *
                05 004BA               F:END    EQU      $
      05 004BA      33000086 02                 MTW,0    FILETYPE           WAS INPUT FILE EVER NAMED
      05 004BB      691004BD                    BLZ      $+2                NO - SKIP CLOSE
      05 004BC      6A700B33                    BAL,LNK  CLOSE              CLOSE INPUT FILE
                00000000                        DO       MODE=1
                                  *S*           CAL3,6   0                  EXIT TO BTM
                                                ELSE
      05 004BD      04900001 A                  M:EXIT                      EXIT TO UTS.
                                                FIN
```

```
                                                PAGE
                                                ************************
                                                *  FILE COMMAND: MERGE  *
                                                ************************
                                                *
                                                *
05 004BE   6A700D78          F:MERGE   BAL,LNK   TESTEDITACTIVE
05 004BF   22300000 A                  LI,X1      0                 RESET THE RECORD CNT.
05 004C0   3530019E 02                 STW,X1     MVD:REC:CNT
05 004C1   22300005 A                  LI,X1     5                  SET P1 TO ADDRESS OF FID1 IN CDT.
05 004C2   F25600AC 02                 LB,P1     *CDTADR,X1
05 004C3   305000AC 02                 AW,P1     CDTADR
05 004C4   355000B3 02                 STW,P1    FID1ADR
05 004C5   6A700BCC                    BAL,LNK   OPEN1              OPEN MERGE SOURCE IN INPUT MODE.
05 004C6   6980046E                    BCS,8     CPY40              ERROR IF NON-EXISTENT
05 004C7   6940052C                    BCS,4     MRG80              OR NOT HEYED.
                                       *
05 004C8   22500000 A                  LI,P1     0
05 004C9   355000B8 02                 STW,P1    FIRSTFROM          SET UP INPUT RANGE AS DEFALT
05 004CA   32500E28                    LW,P1     L(EOF)             ENTIRE FILE.
05 004CB   355000BC 02                 STW,P1    LASTFROM
                                       *
05 004CC   20300001 A                  AI,X1     1
05 004CD   F25600AC 02                 LB,P1     *CDTADR,X1         BUT READJUST IF SPECIFIL RANGE
05 004CE   20300001 A                  AI,X1     1                  GIVEN
05 004CF   21500003 A                  CI,P1     SEQ2
05 004D0   693004D9                    BNE       MRG10
                                       *
05 004D1   F25600AC 02                 LB,P1     *CDTADR,X1         COMPUTE ADDRESS OF SEQUENCE PAIR
05 004D2   305000AC 02                 AW,P1     CDTADR
05 004D3   B2600005 A                  LW,P2     *P1                AND STORE THEM AWAY.
05 004D4   356000B8 02                 STW,P2    FIRSTFROM
05 004D5   20500001 A                  AI,P1     1
05 004D6   B2600005 A                  LW,P2     *P1
05 004D7   356000BC 02                 STW,P2    LASTFROM
05 004D8   20300002 A                  AI,X1     2                  STEP AROUND 'INTO'
                                       *
```

```
2499  05 004D9  325000B8 02  MRG10   LW,P1    FIRSTFROM        VERIFY EXISTENCE OF RECORDS TO
2500  05 004DA  6A700C61             BAL,LNK  READNXTRANDOM    MOVE.
2501  05 004DB  31B00E28             CW,R1    L(EOF)          IF RECORD READ WAS 'EOF',
2502  05 004DC  68100052A            BGE      MRG70           OR GREATER THAN LAST FROM, THEN
2503  05 004DD  31B000BC 02          CW,R1    LASTFROM
2504  05 004DE  692000052A           BG       MRG70           'NOTHING TO MOVE'
2505                           *
2506  05 004DF  6A700B33             BAL,LNK  CLOSE           YES, CLOSE FILE SO WE CAN
2507                           *                               USE F:E ROUTINES TO DELETE
2508  05 004E0  203000002 A          AI,X1    2               'TO' RANGE.
2509  05 004E1  F256000AC 02         LB,P1    *CDTADR,X1      STEP TO FID2 AND OPEN
2510  05 004E2  305000AC 02          AW,P1    CDTADR
2511  05 004E3  355000B4 02          STW,P1   FID2ADR
2512  05 004E4  6A700BC8             BAL,LNK  OPEN
2513  05 004E5  698000051C           BCS,8    MRG30           IF NON-EXISTENT,CREATE NEW FILE.
2514  05 004E6  694000530            BCS,4    MRG82           ERROR IF NOT KEYED
2515                           *
2516  05 004E7  203000002 A          AI,X1    2               NOW GET SEQUENCE NUMERS OF 'TO'
2517  05 004E8  F258000AC 02         LB,T1    *CDTADR,X1      RANGE
2518  05 004E9  308000AC 02          AW,T1    CDTADR
2519  05 004EA  B2500008 A           LW,P1    *T1             IN P1,PL = TEMPORARILY.
2520  05 004EB  208000001 A          AI,T1    1
2521  05 004EC  B2600008 A           LW,P2    *T1
2522                           *
2523  05 004ED  6A700B39             BAL,LNK  DELETE          DELETE 'TO' RANGE
2524  05 004EE  698004F0             BCS,8    $+2             GET 'STOP' SEQ # IF LAST 'TO'
2525  05 004EF  6A700C7F             BAL,LNK  READSEQUEN      NOT HIT EXACTLY.
2526  05 004F0  321000B A            LW,X3    R1              'STOP' SEQ # TO X3
2527  05 004F1  355000008 A          STW,P1   T1
2528  05 004F2  356000009 A          STW,P2   T2
2529                           *
2530  05 004F3  6A700B33     MRG13   BAL,LNK  CLOSE           CLOSE FID2 AS F:E
2531  05 004F4  22A003E8 A   MRG14    LI,P3    1000            DEFAULT INCREMENT
2532  05 004F5  203000001 A          AI,X1    1
2533  05 004F6  F256000AC 02         LB,P1    *CDTADR,X1
2534  05 004F7  683004FD             BEZ      MRG15
2535  05 004F8  203000001 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  READNXTRANDOM
2550  05 00505  32500008 A           LW,P1    T1               FIRST 'TO' 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 TRAN 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  33100019E 02         MTW,1     MVD;REC;CNT     COUNT REC.S MOVED.
2560        00000001                 DO       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  30500000A 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
```

```
05 00519    68100526              BGE       MRG65           'STOP' SEQ # WE'RE CUT OFF.
05 0051A    6A700C7F              BAL,LNK   READSEQUEN      GET NEXT 'FROM' RECORD.
05 0051B    68000506              B         MRG20
                           *
                           *
       05 0051C            MRG30  EQU       $               OUTPUT FILE DOESN'T EXIST.
05 0051C    20300002 A            AI,X1     2               GET STARTING OUTPUT SEQUENCE.
05 0051D    F28600AC 02           LB,T1     *CDTADR,X1
05 0051E    308000AC 02           AW,T1     CDTADR
05 0051F    B2800008 A            LW,T1     *T1
05 00520    32100E28       MRG35  LW,X3     L(EOF)          SET 'STOP' SEQUENCE TO EOF.
05 00521    680004F4              B         MRG14
                           *
                           *
05 00522    3850000A A     MRG55  SW,P1     P3              SUCCESSFUL MERGE, MOVE DEST SEQ #
                           *                                BACK TO LAST USED.  THEN USE
05 00523    6A700B33       MRG56  BAL,LNK   CLOSE           'MK' CODE AFTER CLOSING.
05 00524    6A700B35              BAL,LNK   CLOSE2
05 00525    68000661              B         MVE40
                           *
05 00526    358000BC 02    MRG65  STW,R1    LASTFROM        SET LAST SEQ # READ.
05 00527    6A700B33              BAL,LNK   CLOSE
05 00528    6A700B35              BAL,LNK   CLOSE2
05 00529    68000674              B         MVE56           THEN USE 'MK' CODE.
                           *
05 0052A    6A700B33       MRG70  BAL,LNK   CLOSE           CLOSE INPUT FILE
05 0052B    68000683              B         MVE58           THEN USE 'MK' ROUTINE
05 0052C    6A700B33       MRG80  BAL,LNK   CLOSE
05 0052D    6A700DC4              BAL,LNK   TYPEMSG
05 0052E    00000206 02           DATA      ERRM17          'SOURCE NOT KEYED'
05 0052F    E800000D A            B         *F:LNK
                           *
05 00530    6A700B33       MRG82  BAL,LNK   CLOSE
05 00531    6A700DC4              BAL,LNK   TYPEMSG
05 00532    0000020C 02           DATA      ERRM18          DEST. NOT KEYED
05 00533    E800000D A            B         *F:LNK
                           *
```

```
2610                                        PAGE
2611                                  ************************
2612                                  *  FILE COMMAND: TA     *
2613                                  ************************
2614                                  *
2615                                  *
2616          05 00534                F:TA      EQU     $
2617          00000001                          DO      MODE=2
2618    05 00534    223C0005 A                  LI,X1   5                       COMPUTE ADDRESS OF TAB SPECIFIER
2619    05 00535    F2560CAC 02                 LB,P1   *CDTADR,X1              IN CDT.
2620    05 00536    305000AC 02                 AW,P1   CDTADR
2621                                  *
2622    05 00537    B2500005 A                  LW,P1   *P1                     GET SPECIFIER
2623    05 00538    223C0004 A                  LI,X1   4                       AND CHECK VALIDITY
2624    05 00539    31560015 02                 CW,P1   X:F-1,X1
2625    05 0053A    683C053F                    BE      TA5
2626    05 0053B    643C0539                    BDR,X1  $-2
2627                                  *
2628    05 0053C    6A7C0DC4                    BAL,LNK TYPEMSG                 ERROR: NOT F,M,S.
2629    05 0053D    00C002E5 02                 DATA    UTSM3
2630    05 0053E    68C00004                    B       MASTERPARSER
2631                                  *
2632    05 0053F    67060540            TA5     EXU     TABSET-1,X1             CHANGE MUC TABS FOR F,M OR S
2633                                  *
2634    05 00540    E8C0000D A                  B       *F:LNK                  RETURN
2635    05 00541    04100001 03         TABSET  M:DEVICE M:UC,(TAB,7,0,0,0)         FTABS
        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                 M:DEVICE M:UC,(TAB,10,19,37,0)  MTABS
        03 00005    28000000 X
        03 00006    80000000 A
        03 00007    04       A
```

```
          03 00007 1  0A          A
          03 00007 2  13          A
          03 00007 3  25          A
          03 00008    00          A
    2637  05 00543    041000009 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    04100000D 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
```

```
2640                                    PAGE
2641                                ************************************
2642                                *  RECORD COMMAND: ADD COMMENTARY  *
2643                                ************************************
2644                                *
2645                                *
2646          05 00054B             R:COMMENTARY      EQU *
2647    05 00545    223000005 A               LI,X1     5
2648    05 00546    F246000AC 02              LB,X2     *CDTADR,X1          SET P1=STARTING SEQ #
2649    05 00547    B258000AC 02              LW,P1     *CDTADR,X2
2650    05 00548    223000007 A               LI,X1     7
2651    05 00549    F246000AC 02              LB,X2     *CDTADR,X1          SET T1=STARTING COLUMN #
2652    05 0054A    B288000AC 02              LW,T1     *CDTADR,X2
2653    05 0054B    208FFFFF A                AI,T1     -1                  ADJ TO INTERNAL COL. #
2654    05 0054C    691000580                 BLZ       CMT40
2655    05 0054D    218000080 A               CI,T1     MAXCLMN             IS COL. # >= MAX COL. #
2656    05 0054E    681000580                 BGE       CMT40               YES - ERROR
2657    05 0054F    6A7000C6D                 BAL,LNK   READRANDOM          READ FIRST RECORD
2658    05 00550    698000583                 BCS,8     CMT50               DOES IT EXIST (IF NO, ERROR)
2659          00000001                        DO        MODE=2
2660    05 00551    330001A1 02               MTW,O     CT$FLAG             IS TYPE FLAG ON
2661    05 00552    683000554                 BEZ       $+2                 NO
2662    05 00553    041002CC 02               CAL1,1    NOPROMPT$FPT        YES, TURN OFF PROMPT
2663                                          FIN
2664                                *
2665                                *  TYPE SEQ. # AND READ IN COMMENTARY
2666                                *
2667    05 00554    6A7000DDB       CMT10     BAL,LNK   TYPESEQ             TYPE: 'DDDD.DDD'
2668    05 00555    400800000 A               GEN4      BL,EOM,C,O
2669    05 00556    330001A1 02               MTW,O     CT$FLAG             IS TYPE FLAG ON
2670    05 00557    683000562                 BEZ       CM10A               NO, CONTINUE
2671                                *
2672                                *         CT SPECIFIED, TYPE CARD TO COLUMN REQUESTED
2673                                *
2674    05 00558    32E000E7 02               LW,D0     RECSIZE             SAVE RECSIZE TEMPORARILY
2675    05 00559    358000E7 02               STW,T1    RECSIZE             SET NEW RECSIZE (COLUMN NO)
2676    05 0055A    331000E7 02               MTW,1     RECSIZE
```

```
05 0055B   32300008 A               LW,X1      T1                  COLUMN FOR PROMPT
05 0055C   72160024 02              LB,X3      CARDIMG,X1
05 0055D   2240005C A               LI,X2      '*'
05 0055E   75460024 02              STB,X2     CARDIMG,X1          ADD TO TYPE IMAGE
              00000001              DO         MODE=2
05 0055F   041002D2 02              CAL1,1     TPC$FPT             TYPE CARD TO COMMENTARY
                                    ELSE
                         *S*        LI,X2      0
                         *S*        LW,T2      T1                  GET NO. CHARS TO TYPE
                         *S*        AI,T2      1
                         *S* CM10B  LB,R0      CARDIMG,X2          GET NEXT CHARACTER
                         *S*        CAL3,1     0                   TYPE IT
                         *S*        AI,X2      1
                         *S*        BDR,T2     CM10B               CONTINUE UNTIL ALL CHARS. TYPED
                                    FIN
05 00560   75160024 02              STB,X3     CARDIMG,X1          RESTORE CHARACTER TO CARD IMAGE
05 00561   35E000E7 02              STW,D0     RECSIZE
                         *
              05 00562       CM10A  EQU        $
05 00562   6A700CAE                 BAL,LNK    READTELETYPE2       READ COMMENTARY
05 00563   20BFFFFF A               AI,R1      -1                  SET R1=# OF CHARS READ, LESS C/R
05 00564   E830000D A               BEZ        *R;LNK              IF ONLY  C/R READ - EXIT
05 00565   22300000 A               LI,X1      0
05 00566   32400008 A               LW,X2      T1
05 00567   22900040 A               LI,T2      ' '
                         *
                         *  MOVE COMMENTARY INTO SPECIFIED COLUMN OF CARD
                         *
05 00568   72E60176 02       CMT15  LB,D0      TTYIMG,X1           MOVE COMMENTARY INTO SPECIFIED
05 00569   75E80024 02              STB,D0     CARDIMG,X2           COLUMN
05 0056A   20300001 A               AI,X1      1
05 0056B   20400001 A               AI,X2      1
05 0056C   64B0056E                 BDR,R1     $+2                 TEST IF ANY MORE CHARS LEFT TO MOVE
05 0056D   68000574                 B          CMT20               NO - GO FINISH UP
05 0056E   2140008C A               CI,X2      MAXCLMN             YES - TEST IF ANY ROOM LEFT ON CARD
05 0056F   69100568                 BL         CMT15               YES - LOOP
05 00570   71960176 02              CB,T2      TTYIMG,X1           NO - TEST IF REMAINING CHARS ARE ALL
```

```
2714  05 00571  693 00589          BNE     CMT70           BLANKS (IF NOT, ERROR)
2715  05 00572  203 00001 A        AI,X1   1
2716  05 00573  64B 00570          BDR,R1  *-3             LOOP
2717                               *
2718                               *  BLANK OUT REST OF CARD AFTER NEW COMMENTARY
2719                               *
2720  05 00574  214 0008C A  CMT20  CI,X2    MAXCLMN         BLANK OUT REST OF CARD
2721  05 00575  681 00579          BGE     CMT30
2722  05 00576  759 80024 02       STB,T2  CARDIMG,X2
2723  05 00577  204 00001 A        AI,X2   1
2724  05 00578  680 00574          B       CMT20
2725                               *
2726                               *  WRITE NEW RECORD AND THEN GET NEXT RECORD TO PROCESS
2727                               *
2728  05 00579  6A7 00A56  CMT30    BAL,LNK SETEOD
2729  05 0057A  6A7 00E1F          BAL,LNK WRITERANDOM
2730  05 0057B  6A7 00C7F          BAL,LNK READSEQUEN      READ NEXT RECORD
2731  05 0057C  31B 00E28          CW,R1   L(EOF)          WAS IT AN EOF
2732  05 0057D  683 00586          BE      CMT60           YES - ERROR
2733  05 0057E  325 0000B A        LW,P1   R1              SET P1=SEQ # OF RECORD
2734  05 0057F  680 00554          B       CMT10           GO GET MORE COMMENTARY
2735                               *
2736                               *  ERROR: SPECIFIED COLUMN NUMBER > MAX COLUMN NUMBER
2737                               *
2738  05 00580  6A7 00DC4  CMT40    BAL,LNK TYPEMSG         TYPE: '=P2:COL>72'
2739  05 00581  000 00261 02       DATA    ERRP14
2740  05 00582  680 00004          B       MASTERPARSER    EXIT TO PARSER
2741                               *
2742                               *  ERROR: INITIAL SEQ. # DOESN'T EXIST
2743                               *
2744  05 00583  6A7 00DC4  CMT50    BAL,LNK TYPEMSG         TYPE: '=P1:NO SUCH REC'
2745  05 00584  000 0022E 02       DATA    ERRP1
2746  05 00585  680 00004          B       MASTERPARSER    EXIT TO PARSER
2747                               *
2748                               *  ERROR: EOF HIT
2749                               *
2750  05 00586  6A7 00DC4  CMT60    BAL,LNK TYPEMSG         TYPE: '=-EOF HIT'
```

```
2751   05 00587   000001D5 02          DATA     ERRM1
2752   05 00588   68000004             B        MASTERPARSER      EXIT TO PARSER
2753                            *
2754                            *  ERROR: COMMENTARY OVERFLOWS CARD
2755                            *
2756   05 00589   6A700DC4     CMT70    BAL,LNK  TYPEMSG           TYPE: '**OVERFLOW'
2757   05 0058A   000001DC 02          DATA     ERRM3
2758   05 0058B   68000579             B        CMT30             GO CONTINUE WITH NEXT RECORD
```

```
2759                                    PAGE
2760                                ******************************
2761                                *  RECORD COMMAND: DELETE    *
2762                                ******************************
2763                                *
2764                                *
2765          05 00058C             R;DELETE EQU     *
2766    05 00058C    22300005 A              LI,X1    5
2767    05 00058D    F2460AAC 02             LB,X2    *CDTADR,X1      GET ADDR OF FIRST SEQ # IN CDT
2768    05 00058E    B2580AAC 02             LW,P1    *CDTADR,X2      SET P1=FIRST SEQ #
2769    05 00058F    20400001 A              AI,X2    1                   P2=LAST SEQ #
2770    05 000590    B2680AAC 02             LW,P2    *CDTADR,X2
2771    05 000591                            D01      MODE=2
2772    05 000591    33100207 02             MTW,1    TABXFLAG
2773    05 000592    6A700B39                BAL,LNK  DELETE          DELETE ALL BETWEEN THESE SEQ #'S
2774    05 000593    E800000D A              B        *R;LNK          EXIT
```

```
2775                                        PAGE
2776                                ***********************************************
2777                                *  RECORD COMMANDS: FIND AND DELETE(TYPE)  *
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    F246000AC 02              LB,X2     *CDTADR,X1
2796    05 0059C    B258000AC 02              LW,P1     *CDTADR,X2          SET P1=FIRST SEQ # IN CDT
2797    05 0059D    3550000B9 02              STW,P1    FIRSTSET                FIRSTSET=1ST SEQ # IN CDT
2798    05 0059E    3580000B9 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    B288000AC 02              LW,T1     *CDTADR,X2
2801    05 005A1    3580000BE 02              STW,T1    LASTSET
2802    05 005A2    22300007 A                LI,X1     7
2803    05 005A3    F266000AC 02              LB,P2     *CDTADR,X1          SET P2=ABSOLUTE ADDR OF STRING TO
2804    05 005A4    3060000AC 02              AW,P2     CDTADR               MATCH
2805    05 005A5    22300008 A                LI,X1     8
2806    05 005A6    6A7000A34                 BAL,LNK   PROCESSCOL#PAIR     PROCESS COL # PARAMS
2807    05 005A7    6A7000C61                 BAL,LNK   READNXTRANDOM       READ FIRST SEQ # OR NEXT HIGHEST
2808                                *
2809                                *  READ EACH RECORD AND SEE IF IT CONTAINS THE SPECIFIED STRING
2810                                *
2811    05 005A8    31B000E28     FND20       CW,R1     L(EOF)              WAS IT AN EOF
```

```
05 005A9  683005CF               BE        FND70            YES - ERROR
05 005AA  318000BE 02            CW,R1     LASTSET          WAS INPUT SEQ # > LAST SEQ #
05 005AB  692005C1               BG        FND50            YES - FINISH UP
05 005AC  358000B9 02            STW,R1    FIRSTSET         NO - SAVE NEW SEQ #
05 005AD  325000B7 02            LW,P1     FRSTCLMN         CHECK IF REC CONTAINS STRING
05 005AE  6A7009EF    FNDTYP     BAL,LNK    FINDMATCH       STARTING AT SPECIFIED COL. #
05 005AF  698005BB               BCS,8     FND40
05 005B0  20A00001 A             AI,P3     1                YES - INCR MATCH COUNT
05 005B1  670405D2               EXU       FNDTBL1,X4       GO PERFORM APPRO ACTION
                      *
                      *  'FD' USED: DELETE RECORD
                      *
05 005B2  6A700B95    FND30      BAL,LNK   DELETERECORD     DELETE RECORD
05 005B3  680005BB               B         FND40            GO ON TO NEXT RECORD
                      *
                      *  'FT' USED: TYPE SEQ #, AND RECORD
                      *
05 005B4  325000B9 02 FND32      LW,P1     FIRSTSET
05 005B5  6A700A56               BAL,LNK   SETEOD
05 005B6  6A700D82               BAL,LNK   TYPECARD
05 005B7  680005BB               B         FND40
                      *
                      *  'FS' USED: TYPE SEQ #
                      *
05 005B8  325000B9 02 FND35      LW,P1     FIRSTSET         GET SEQ #
05 005B9  6A700DDB               BAL,LNK   TYPESEQ          TYPE: 'DDDD.DDD'
05 005BA  00000000 A             GEN4      0,0,0,0
                      *
                      *  TEST IF LAST RECORD HIT: IF YES, GO FINISH UP
                      *
05 005BB  328000B9 02 FND40      LW,R1     FIRSTSET         TEST IF LAST SEQ # = SEQ # TO STOP
05 005BC                         DO1       MODE=2
05 005BC  358002C3 02            STW,R1    INTFLAG1
05 005BD  318000BE 02            CW,R1     LASTSET           AT
05 005BE  683005C1               BE        FND50
05 005BF  6A700C7F               BAL,LNK   READSEQUEN       NO - READ NEXT RECORD
05 005C0  680005A8               B         FND20            LOOP
```

```
2849                                *
2850                                *  SEQ. # TO STOP AT HIT OR PASSED: FINISH UP
2851                                *
2852   05 005C1  670405D5           FND50    EXU      FNDTBL2,X4        GO FINISH UP
2853                                *
2854                                *  'FD' USED: TYPE '=NNN RECS DLTED'
2855                                *
2856   05 005C2  32F0000A A        FND60    LW,D1    P3
2857   05 005C3  683005CC                   BEZ      FND65A            WERE ANY MATCHES FOUND
2858   05 005C4  32500000A A                LW,P1    P3                GET RECORD COUNT IN P1
2859   05 005C5  226000A31 02               LI,P2    BA(MSG6)+1        GET BYTE ADR. OF PLACE TO PUT CNT.
2860   05 005C6  6A7000B11                  BAL,LNK  BINTODEC          GO PUT THE NUMBER THER
2861   05 005C7  6A7000DC4                  BAL,LNK  TYPEMSG
2862   05 005C8  000002BC 02                DATA     MSG6
2863   05 005C9  E800000D A                 B        *R:LNK            EXIT
2864                                *
2865                                *  'FT' USED: TYPE '=NONE' IF NO MATCHES FOUND
2866                                *
2867   05 005CA  21A000000 A        FND65    CI,P3    0                 WERE ANY MATCHES FOUND
2868   05 005CB  E930000D A                 BNE      *R:LNK            YES - EXIT
2869   05 005CC  6A7000DC4          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                                *  ERROR: EOF HIT
2874                                *
2875   05 005CF  6A7000DC4          FND70    BAL,LNK  TYPEMSG           TYPE: '=EOF HIT'
2876   05 005D0  000001D5 02                DATA     ERRM1
2877   05 005D1  68000C1                    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                                *
```

```
2886          05 005D5          FNDTBL2  EQU     *
2887    05 005D5    6800005C2             B       FND60
2888    05 005D6    6800005CA             B       FND65
2889    05 005D7    6800005CA             B       FND65
```

```
                                       PAGE
                              ***************************************************
                              *  RECORD COMMANDS: INSERT(SUPPRESSING SEQ. NUMBERS)  *
                              ***************************************************
                              *
                              *
        05 005D8              R:INSERT$SUP$SEQ  EQU $
05 005D8     22200002 A                LI,X4     2                  USE X4=2 FOR 'IS'
        00000001                       DO        MODE=2
05 005D9     041002CB 02               CAL1,1    PROMPT2$FPT
05 005DA     680005DD                  B         R:INSERT+2
                                       ELSE
                         *S*           B         R:INSERT+1
                                       FIN
                              *
                              *
        05 005DB              R:INSERT EQU       $
05 005DB                               DO1       MODE=2
05 005DB     041002CC 02               CAL1,1    NOPROMPT$FPT
05 005DC     22200000 A                LI,X4     0                  USE X4=0 FOR 'IN'
05 005DD     22300005 A                LI,X1     5
05 005DE     F24600AC 02               LB,X2     *CDTADR,X1         SET P1=STARTING SEQ #
05 005DF     B25800AC 02               LW,P1     *CDTADR,X2
05 005E0     328000B0 02               LW,T1     DFLTINCR           SET T1=LAST INCR USED
05 005E1                               DO1       MODE=2
05 005E1     331002C7 02               MTW,1     TABXFLAG
05 005E2     22300006 A                LI,X1     6
05 005E3     F2B600AC 02               LB,R1     *CDTADR,X1         GET PARAM2 TYPE
05 005E4     683005E9                  BEZ       INS10              TEST IF PARAM2 PRESENT
05 005E5     20300001 A                AI,X1     1
05 005E6     F24600AC 02               LB,X2     *CDTADR,X1         YES - SET T1=INCR FROM CDT
05 005E7     B28800AC 02               LW,T1     *CDTADR,X2
05 005E8     358000B0 02               STW,T1    DFLTINCR           SET NEW DEFAULT INCR
                              *
                              *  GET SEQ. # AT WHICH TO STOP INSERTING
                              *
05 005E9     6A700C61         INS10    BAL,LNK   READNXTRANDOM      READ 1ST SEQ # OR NEXT HIGHEST
```

```
2927  05 005EA  698005EC          BCS,8   $+2             WAS NEXT HIGHEST READ
2928  05 005EB  6A700C7F          BAL,LNK READSEQUEN      NO - SO READ NEXT HIGHEST
2929  05 005EC  329 0000B A       LW,T2   R1              SET T2=SEQ # AT WHICH TO STOP INSERT
2930                        *
2931                        *  TYPE NEXT SEQ. # AND READ INPUT LINE
2932                        *
2933  05 005ED  680405EE    INS20 B       $+1,X4          TYPE 'DDDD.DDD' AS REQD
2934  05 005EE  6A700DDB          BAL,LNK TYPESEQ
2935  05 005EF  40080000 A        GEN4    BL,EOM,C,0
2936        00000001              DO      MODE=2
2937  05 005F0  67040610          EXU     INS50,X4
2938                              FIN
2939  05 005F1  6A700CB2          BAL,LNK READTELETYPE    READ INSERT
2940  05 005F2  218 00001 A       CI,R1   1
2941  05 005F3  683 0060F         BE      INS40
2942  05 005F4  324 0000B A        LW,X2   R1          GET BYTE CNT., INTO INDEX REG.
2943  05 005F5  204FFFFF A        AI,X2   -1              MAKE X2 A BINARY COUNT
2944  05 005F6  72F80024 02       LB,D1   CARDIMG,X2      GET LAST BYTE INPUT
2945  05 005F7  223 00004 A       LI,X1   GNTBL1SZ        GET THE COUNT OF DIFFERENT TYPE
2946                        *                             OF LINE TERMINATORS.
2947  05 005F8  71F608CD          CB,D1   GNTBL1,X1       FIND A CARRAGE RETURN
2948  05 005F9  683 005FC         BE      $+3
2949  05 005FA  643 005F8         BDR,X1  $-2
2950  05 005FB  680 005FF         B       $+4
2951  05 005FC  22F 00040 A       LI,D1   ' '             BLANK OUT C/R
2952  05 005FD  75F80024 02       STB,D1  CARDIMG,X2
2953  05 005FE  33F 0000B A       MTW,-1  R1              IF CR DECREMENT CHAR. COUNT.
2954  05 005FF  35B 000E7 02      STW,R1  RECSIZE
2955  05 00600  218 0008C A       CI,R1   MAXCLMN
2956  05 00601  682 00604         BLE     INS35
2957  05 00602  6A700DC4          BAL,LNK TYPEMSG         NO - TYPE! !=-OVERFLOW!
2958  05 00603  000001DC 02       DATA    ERRM3
2959                        *
2960                        *  WRITE INPUT IMAGE, INCREMENT SEQ. #, AND CHECK AGAINST # TO STOP AT
2961                        *
2962  05 00604  6A700A56    INS35 BAL,LNK SETEOD
2963  05 00605  6A700E1F          BAL,LNK WRITERANDOM     WRITE CARD IMAGE
```

```
2964  05 00606  305000008 A          AW,P1    T1          INCR SEQ #
2965  05 00607  315000009 A          CW,P1    T2
2966  05 00608  691005ED             BL       INS20       IS NEW SEQ # > SEQ # TO STOP AT
2967  05 00609  315000BF 02          CW,P1    MAXSEQ      IS SEQ. NO. TOO BIG
2968  05 0060A  682006OD             BLE      INS38       NO.
2969  05 0060B  6A700DC4             BAL,LNK  TYPEMSG     YES.
2970  05 0060C  000002ID 02          DATA     ERRM20
2971  05 0060D  6A700DC4     INS38   BAL,LNK  TYPEMSG     RING BELL TWICE
2972  05 0060E  00000613             DATA     INSMSG
2973        05 0060F        INS40   EQU      *
2974  05 0060F  E800000D A           B        *R:LNK      RETURN
2975        00000001                 DO       MODE=2
2976  05 00610  22F00009 A   INS50   LI,D1     9
2977  05 00611  02000000 A           NOP      0           X4 IS NEVER ONE
2978  05 00612  22F00001 A           LI,D1    1           OFFSET FOR PROMPT ONLY
2979                          *
2980                                 FIN
2981                          *
2982  05 00613  03070708 A   INSMSG  TEXTC    'GGH'       X'07'+X'07'+EOM
```

```
2983                                    PAGE
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    353001 9E 02              STW,X1     MVD:REC:CNT         ZERO OUT 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    F2B600AC 02               LB,R1     *CDTADR,X1          GET PARAM3 TYPE
3015    05 00626    68300628                  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                                *  CHECK FOR OVERLAPPING SEQ #'S AND SET UP MOVE
3022                                *
3023  05 0062B  32E00008 A  MVE20   LW,D0    T1              PUT 'FROM' SEQ #'S IN DW
3024  05 0062C  32F00009 A          LW,D1    T2
3025  05 0062D  195000CE A          CLM,P1   D0              MAKE SURE 'TO' AND 'FROM' RANGES
3026  05 0062E  68900066E           BIL      MVE50            ARE MUTUALLY EXCLUSIVE
3027  05 0062F  196000CE A          CLM,P2   D0
3028  05 00630  68900066E           BIL      MVE50
3029  05 00631  32E00005 A          LW,D0     P1
3030  05 00632  32F00006 A          LW,D1     P2
3031  05 00633  198000CE A        CLM,T1     D0
3032  05 00634  68900066E          BIL        MVE50
3033  05 00635  199000CE A          CLM,T2   D0
3034  05 00636  68900066E           BIL      MVE50
3035  05 00637  46800005 A          XW,T1    P1              EXCHANGE FIRST 'FROM' AND 'TO'
3036  05 00638  6A700C61           BAL,LNK  READNXTRAND0M     CHECK 'FROM' RANGE
3037  05 00639  46800005 A          XW,T1    P1             RESTORE
3038  05 0063A  31B00E28           CW,R1    L(E0F)      M    IF RECORD READ WAS AN E0F,
3039  05 0063B  68300683           BE       MVE58
3040  05 0063C  31B00009 A          CW,R1    T2              OR SEQUENCE GREATER THAN SECOND
3041  05 0063D  69200683           BG       MVE58           'FROM', NOTHING TO MOVE
3042  05 0063E  6A700B39           BAL,LNK  DELETE          DELETE 'TO' RECORDS
3043  05 0063F  69800641           BCS,8    $+2             WAS LAST 'TO' SEQ # HIT BY DELETE
3044  05 00640  6A700C7F           BAL,LNK  READSEQUEN      YES - READ NEXT RECORD
3045  05 00641  32100008 A          LW,X3    R1              SET X3=SEQ # AT WHICH TO STOP MOVE
3046  05 00642  46500008 A          XW,P1    T1
3047  05 00643  6A700C61           BAL,LNK  READNXTRAND0M     READ 1ST 'FROM' REC OR NEXT HIGHEST
3048  05 00644  32500008 A          LW,P1    T1              SET P1=NEW 'TO' SEQ # + INCR
3049  05 00645  38500004 A          SW,P1    P3
3050                              *
3051                              *  READ EACH 'FROM' RECORD AND WRITE UNDER 'TO' SEQ #
3052                              *
3053  05 00646  31B00E28     MVE30  CW,R1    L(E0F)          WAS AN E0F READ
3054  05 00647  68300671           BE       MVE53           YES - GO TYPE ERROR MESSAGE
3055  05 00648  31900008 A          CW,T2    R1              WAS 'FROM' SEQ # >= LAST 'FROM' SEQ
3056  05 00649  68200659           BLE      MVE35           YES - GO FINISH UP
```

```
05 0064A   3050000A A          AW,P1     P3                   INCR 'TO' SEQ #
05 0064B   31500001 A          CW,P1     X3                   IS NEW 'TO' SEQ # > SEQ # TO STOP AT
05 0064C   68100674            BGE       MVE56                YES - GO TYPE ERROR MESSAGE
05 0064D   35B000BC 02         STW,R1    LASTFROM
05 0064E   6804064F            B         $+1,X4               DELETE 'FROM' RECORD AS REQD
05 0064F   6A700B95            BAL,LNK   DELETERECORD
05 00650   3310019E 02         MTW,1      MVD:REC:CNT         INCERMENT REC. COUT
05 00651   6A700E1F            BAL,LNK   WRITERANDOM          WRITE RECORD WITH NEW 'TO' SEQ #
              00000001         DO        MODE=2
05 00652   35B002C3 02         STW,R1    INTFLAG1
05 00653   355002C4 02         STW,P1    INTFLAG2
                               FIN
05 00654   465000BC 02         XW,P1     LASTFROM             MUST REREAD LAST 'FROM' RECORD TO
05 00655   6A700C6D            BAL,LNK   READRANDOM            GET DCB BACK IN SEQ
05 00656   465000BC 02         XW,P1     LASTFROM             RESTORE P1 AND LASTFROM
05 00657   6A700C7F            BAL,LNK   READSEQUEN           READ NEXT 'FROM' RECORD
05 00658   68000646            B         MVE30                LOOP
                      *
                      *  LAST 'FROM' SEQ # HIT OR PASSED: FINISH UP
                      *
05 00659   69100661   MVE35    BL        MVE40                WAS LAST 'FROM' SEQ # PASSED
05 0065A   3050000A A          AW,P1     P3                   NO, WAS HIT - INCR 'TO' SEQ #
05 0065B   31500001 A          CW,P1     X3                   IS NEW 'TO' SEQ # > SEQ # TO STOP AT
05 0065C   68100674            BGE       MVE56                YES - GO TYPE ERROR MESSAGE
05 0065D   6804065E            B         $+1,X4               DELETE 'FROM' REC AS REQD
05 0065E   6A700B95            BAL,LNK   DELETERECORD
05 0065F   3310019E 02         MTW,1      MVD:REC:CNT         INCERMENT RECORD COUNT.
05 00660   6A700E1F            BAL,LNK   WRITERANDOM          WRITE REC WITH NEW 'TO' SEQ #
                      *
                      *  TYPE OUT LAST 'TO' SEQ # AND EXIT
                      *
05 00661   22600AA7 02 MVE40   LI,P2     BA(MVEMSG1)+11       BUILD MSG: '--DONE AT DD.D' + NL
05 00662   6A700B97            BAL,LNK   MOVESEQ               FROM LAST 'TO' SEQ #
05 00663   00000000 A          GEN4      0,0,0,0
05 00664   20B0000A A          AI,R1     10                   ADJ CNT OF TEXTC-STRING
05 00665   75B002A7 02         STB,R1    MVEMSG1
05 00666   6A700DC4            BAL,LNK   TYPEMSG              TYPE MSG
```

```
3094  05 00667  000002A7 02          DATA     MVEMSG1
3095  05 00668  22600A4D 02          LI,P2    BA(MSG7)+1
3096  05 00669  32500019E 02         LW,P1    MVD:REC:CNT        GET THE NUMBER OF RECS MOVED
3097  05 0066A  6A700B11             BAL,LNK  BINTODEC           CONVERT IT, STUFF IT AWAY
3098  05 0066B  6A700DC4             BAL,LNK  TYPEMSG            AND PRINT IT OUT.
3099  05 0066C  00000293 02          DATA     MSG7
3100  05 0066D  E8000000D A          B        *R:LNK             EXIT
3101                                 *
3102                                 *  ERROR: SEQ #'S OVERLAP
3103                                 *
3104  05 0066E  6A700DC4             MVE50    BAL,LNK  TYPEMSG   TYPE; '=RNG OVERLAP'
3105  05 0066F  000001DF 02          DATA     ERRM4
3106  05 00670  E8000000D A          B        *R:LNK             EXIT
3107                                 *
3108                                 *  ERROR: EOF HIT
3109                                 *
3110  05 00671  6A700DC4             MVE53    BAL,LNK  TYPEMSG   TYPE: '=EOF HIT'
3111  05 00672  000001D5 02          DATA     ERRM1
3112  05 00673  68000661             B        MVE40              GO EXIT
3113                                 *
3114                                 *  ERROR: 'TO' SEQ # HIT NEXT UNDELETED RECORD
3115                                 *
3116  05 00674  38500000A A          MVE56    SW,P1    P3         ADJ P1 TO LAST 'TO' 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 'TO' SEQ #
3119  05 00677  404D0000 A           GEN4     BL,LP,0,0
3120  05 00678  30600000B A          AW,P2    R1                 INCR MSG BYTE ADDR
3121  05 00679  20B0000C A           AI,R1    12                 CALC AND SAVE MSG LENGTH
3122  05 0067A  32800008 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  30800008 A           AW,T1    R1                 ADJ CNT OF TEXTC-STRING
3127  05 0067F  758002AD 02          STB,T1   MVEMSG2
3128  05 00680  6A700DC4             BAL,LNK  TYPEMSG            TYPE: '=CUTOFF AT DDD.D (DD.DD)' +
3129  05 00681  000002AD 02          DATA     MVEMSG2             NL
3130  05 00682  E8000000D A          B        *R:LNK             EXIT
```

```
3131                                    *
3132    05 00683    6A700DC4            MVE58     BAL,LNK   TYPEMSG
3133    05 00684    00000201 02                   DATA      ERRM16
3134    05 00685    E8000000D A                   B         *R;LNK
```

```
3135                                    PAGE
3136                                ********************************
3137                                *  RECORD COMMAND: RENUMBER    *
3138                                ********************************
3139                                *
3140                                *
3141          05 00686              R:RENUMBER        EQU *
3142    05 00686    223000005 A                 LI,X1     5
3143    05 00687    F246000AC 02                LB,X2     *CDTADR,X1          SET P1=OLD SEQ #
3144    05 00688    B258000AC 02                LW,P1     *CDTADR,X2
3145    05 00689    223000007 A                 LI,X1     7
3146    05 0068A    F246000AC 02                LB,X2     *CDTADR,X1          SET T1=NEW SEQ #
3147    05 0068B    B288000AC 02                LW,T1     *CDTADR,X2
3148    05 0068C    6A7000C6D                   BAL,LNK   READRANDOM          READ OLD RECORD
3149    05 0068D    698000693                   BCS,8     RNM10               DID IT EXIST
3150    05 0068E    325000008 A                 LW,P1     T1                  YES - SET P1=NEW SEQ #
3151    05 0068F    6A7000E0E                   BAL,LNK   WRITENEWRANDOM      WRITE RECORD UNDER NEW SEQ #
3152    05 00690    698000696                   BCS,8     RNM13               DID THIS SEQ # ALREADY EXIST
3153    05 00691    6A7000B95                   BAL,LNK   DELETERECORD        NO - DELETE OLD RECORD
3154    05 00692    E8000000D A                 B         *R:LNK              EXIT
3155                                *
3156                                *  ERROR: OLD RECORD DOESN'T EXIST
3157                                *
3158    05 00693    6A7000DC4       RNM10       BAL,LNK   TYPEMSG             TYPE: '=P1:NO SUCH REC'
3159    05 00694    00000022E 02                DATA      ERRP1
3160    05 00695    E8000000D A                 B         *R:LNK              EXIT
3161                                *
3162                                *  ERROR: NEW RECORD ALREADY EXISTS
3163                                *
3164    05 00696    6A7000DC4       RNM13       BAL,LNK   TYPEMSG             TYPE: '=P2:REC EXISTS'
3165    05 00697    000000232 02                DATA      ERRP2
3166    05 00698    E8000000D A                 B         *R:LNK              EXIT
```

```
3167                                    PAGE
3168                                ***********************************************
3169                                *  RECORD COMMANDS: SET AND STEP (AND TYPE)   *
3170                                ***********************************************
3171                                *
3172                                *
3173          05 00699              R:SET$STEP        EQU $
3174    05 00699    22300001 A                LI,X1   1                   USE STEPFLAG=1 FOR 'SS'
3175    05 0069A    68000069C                 B       R:SET$STEP$TYPE+1
3176                                *
3177                                *
3178          05 0069B              R:SET$STEP$TYPE   EQU $
3179    05 0069B    223FFFFF A                LI,X1   -1                  USE STEPFLAG=-1 FOR 'ST'
3180    05 0069C    35300167 02               STW,X1  STEPFLAG            TURN ON 'SET AND STEP MODE' FLAGS
3181    05 0069D    353000E9 02               STW,X1  SETFLAG
3182    05 0069E    22300005 A                LI,X1   5                   GET STARTING SEQ # FROM CDT
3183    05 0069F    F24600AC 02               LB,X2   *CDTADR,X1
3184    05 006A0    B25800AC 02               LW,P1   *CDTADR,X2
3185    05 006A1    22300006 A                LI,X1   6
3186    05 006A2    6A700A34                  BAL,LNK PROCESSCOL#PAIR     PROCESS COL # PARAMS
3187    05 006A3    6A700C61                  BAL,LNK  READNXTRANDOM
3188    05 006A4    35B00005 A                STW,R1   P1                 PUT FIRST REC. NO. IN P1.
3189    05 006A5    31B00E28                  CW,R1   L(EOF)
3190    05 006A6    683006C0                  BE      SPL20
3191    05 006A7    35B000B9 02               STW,R1   FIRSTSET           NO , SO USE THE FIRST RECORD
3192    05 006A8    6800006B8                 B        FINISH$STEP$LOOP   NUMBER FOUND THAT IS HIGHER
3193                                *                                       THAN THE INPUT RECORD NO.
3194                                *
3195                                *  NULL COMMAND OR ERROR: TURN OFF 'SET MODE' AND 'STEP MODE' FLAGS
3196                                *
3197    05 006A9    22800000 A    STP10       LI,T1   0                   TURN OFF MODE FLAGS
3198    05 006AA    358000E9 02               STW,T1  SETFLAG
3199    05 006AB    35800167 02               STW,T1  STEPFLAG
3200    05 006AC    68000004                  B       MASTERPARSER        EXIT TO PARSER
3201                                *
3202                                *
3203                                *
```

```
3204          05 006AC              STEP$LOOP       EQU $               (EXC ENTERS HERE AT 'END OF CDT')
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    6A7000E1F                 BAL,LNK   WRITERANDOM
3209    05 006B1    33000194 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    6A7000C7F     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  EQU $             ('JU' ENTERS HERE TO FINISH)
3223    05 006B8    6A7000A56                 BAL,LNK   SETEOD          SET EOD MARKER
3224    05 006B9    33000167 02               MTW,0     STEPFLAG
3225    05 006BA    692006BD                  BGZ       SPL15           WAS 'ST' CMND USED
3226    05 006BB    6A7000D82                 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    6A7000DDB     SPL15       BAL,LNK   TYPESEQ         TYPE: 'DDDD.DDD:'
3232    05 006BE    08000000 A                GEN4      EOM,0,0,0
3233    05 006BF    68000004                  B         MASTERPARSER    EXIT TO PARSER
3234                                *
3235                                *  ERROR: EOF HIT
3236                                *
3237    05 006C0    6A7000DC4     SPL20       BAL,LNK   TYPEMSG         TYPE: '**EOF HIT'
3238    05 006C1    000001D5 02               DATA      ERRM1
3239    05 006C2    680006A9                  B         STP10           GO EXIT
```

```
                                        PAGE
                              ***************************************************
                              *  RECORD COMMANDS: TYPE(SUPPRESSING SEQ. NUMBERS) *
                              ***************************************************
                              *
         05 006C3             R:TYPE$COMPRESSED EQU $
                              *
         05 006C3             R:TYPE    EQU       $
05 006C3      22200001 A                LI,X4     1                  USE X4=1 FOR 'TY'
05 006C4      680006C6                  B         R:TYPE$SUP$SEQ+1
                              *
                              *
         05 006C5             R:TYPE$SUP$SEQ      EQU $
05 006C5      22200000 A                LI,X4     0                  USE X4=0 FOR 'TS'
05 006C6      22300001 A                LI,X1      1
05 006C7      353000E9 02               STW,X1     SETFLAG           SET THE SETFLAG TO ONE
                              *                                      THE RANGE FROM TY IS USED FOR
                              *                                      AN SE COMMAND.
05 006C8      22C00000 A                LI,R2     0                  START COUNT OF RECORDS OUTPUT.
05 006C9      22300005 A                LI,X1     5
05 006CA      F24600AC 02               LB,X2     *CDTADR,X1         GET ADDR OF FIRST SEQ $ IN CDT
05 006CB      B25800AC 02               LW,P1     *CDTADR,X2         SET P1=FIRST SEQ #
05 006CC      20400001 A                AI,X2     1                      P2=LAST SEQ #
05 006CD      B26800AC 02               LW,P2     *CDTADR,X2
05 006CE      356000BE 02               STW,P2     LASTSET           SAVE ENDING SEQ #
05 006CF      20300001 A                AI,X1     1                  SET UP COL. NUMBERS
05 006D0      6A700A34                  BAL,LNK   PROCESSCOL#PAIR
         00000001                       DO        MODE=2
05 006D1      323000B7 02               LW,X1     FRSTCLMN           MUST EXPAND TABS, IF
05 006D2      693006D9                  BNEZ      TYP5
05 006D3      323000BB 02               LW,X1     LASTCLMN           COL. NO. SPECIFIED, OR
05 006D4      2130008C A                CI,X1     MAXCLMN
05 006D5      691006D9                  BL        TYP5
05 006D6      21200001 A                CI,X4     1                  'TY' AND 'TC'
05 006D7      683006D9                  BE        TYP5
05 006D8      331002C7 02               MTW,1     TABXFLAG
05 006D9                      TYP5      RES       0
```

```
3277                                        FIN
3278   05 006D9  6A700C61                   BAL,LNK  READNXTRANDOM     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)           WAS IT AN EOF
3285   05 006DD  683006F4                   BE        TYP20            YES - GO TYPE ERROR MESSAGE
3286   05 006DE  31600000B A                CW,P2     R1               WAS INPUT SEQ # >= LAST SEQ #
3287   05 006DF  682006E9                   BLE       TYP15            YES - FINISH UP
3288   05 006E0  32500000B A                LW,P1     R1
3289   05 006E1  680406E2                   B         $+1,X4           SET TO TYPE SEQ # AS REQD
3290   05 006E2  225FFFFF A                 LI,P1     -1
3291   05 006E3  6A7006FE                   BAL,LNK   TYP40
3292   05 006E4  6A700A56                   BAL,LNK   SETEOD           SET EOD MARKER
3293   05 006E5  6A700D82                   BAL,LNK   TYPECARD         TYPE CARD IMAGE WITH INPUT SEQ #
3294   05 006E6  20C00001 A                 AI,R2     1
3295   05 006E7  6A700C7F                   BAL,LNK   READSEQUEN       READ NEXT RECORD
3296   05 006E8  680006DC                   B         TYP10            LOOP
3297                              *
3298                              *  LAST SEQ # HIT OR PASSED: FINISH UP
3299                              *
3300   05 006E9  691006F1        TYP15      BL        TYP17            WAS LAST SEQ # PASSED
3301   05 006EA  32500000B A                LW,P1     R1               NO, WAS HIT - PREPARE TO TYPE CARD
3302   05 006EB  680406EC                   B         $+1,X4           SET TO TYPE SEQ # AS REQD
3303   05 006EC  225FFFFF A                 LI,P1     -1
3304   05 006ED  6A7006FE                   BAL,LNK   TYP40
3305   05 006EE  6A700A56                   BAL,LNK   SETEOD           SET EOD MARKER
3306   05 006EF  6A700D82                   BAL,LNK   TYPECARD         TYPE CARD IMAGE
3307   05 006F0  20C00001 A                 AI,R2     1
3308   05 006F1  20C00000 A        TYP17    AI,R2     0                CHECK OUTPUT COUNT
3309   05 006F2  682006F9                   BLEZ      TYP25
3310   05 006F3  6800072C                   B         TYP90
3311                              *
3312                              *  ERROR: EOF HIT
3313                              *
```

```
3314  05 006F4  6A700DC4     TYP20   BAL,LNK  TYPEMSG          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  TYPEMSG          TYPE! '=NONE'
3321  05 006FA  000001E6 02         DATA     ERRM6
3322  05 006FB  22800000 A   TYP25A  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    FRSTCLMN         ADJUST THE IMAGE FOR COLUMN BOUNDS
3328  05 006FF  683000A              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    CARDIMG,X1       ZERO.
3331  05 00702  75080024 02         STB,R0   CARDIMG,X2
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    X1
3338  05 00709  6800070B            B        $+2              SET FINISH COULUMN FOR NEXT ROUTINE
3339  05 0070A  323000BB 02  TYP50   LW,X1    LASTCLMN         IF LESS THAN FULL IMAGE DESIRED,
3340  05 0070B  353000CE A          STW,X1   D0               SAVE TERMINAL POSITION FOR COMPRESS
3341  05 0070C  213000BC A          CI,X1    MAXCLMN
3342  05 0070D  68100713            BGE      TYP60
3343  05 0070E  22400040 A          LI,X2    ' '              BLANK OUT REGION
3344  05 0070F  75460024 02  TYP55   STB,X2   CARDIMG,X1       (LASTCLMN,MAXCLMN-1)
3345  05 00710  20300001 A          AI,X1    1
3346  05 00711  213000BB A          CI,X1    MAXCLMN-1
3347  05 00712  68200710            BLE      TYP55
3348                       *
3349  05 00713  22300001 A   TYP60   LI,X1    1                FINALLY CHECK FOR COMPRESSION .
3350  05 00714  F24600AC 02         LB,X2    *CDTADR,X1
```

```
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,LASTCLMN) COMPRESS
3358  05 00719  22000040 A          LI,R0    ' '            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 UPPRR LIMIT.
3368  05 00723  69100718            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  68000718            B        TYP75          MOVE NEXT STRING DOWN.
3379        05 0072C          TYP90  EQU      $
3380  05 0072C  E8000000 A          B        *R:LNK
```

```
3381                                PAGE
3382                          ****************************
3383                          *  INTRALINE COMMAND: SET  *
3384                          ****************************
3385                          *
3386                          *
3387          05 0072D        I:SET   EQU     *
3388    05 0072D  228 00001 A         LI,T1   1               TURN 'SET MODE' FLAG ON
3389    05 0072E  358 000E9 02        STW,T1  SETFLAG
3390    05 0072F  223 00005 A         LI,X1   5
3391    05 00730  F24 600AC 02        LB,X2   *CDTADR,X1      GET ADDR OF FIRST SEQ # IN CDT
3392    05 00731  B25 800AC 02        LW,P1   *CDTADR,X2      SET P1=FIRST SEQ #
3393    05 00732  204 00001 A         AI,X2   1                   P2=LAST SEQ #
3394    05 00733  B26 800AC 02        LW,P2   *CDTADR,X2
3395    05 00734  356 000BE 02        STW,P2  LASTSET
3396    05 00735  223 00006 A         LI,X1   6
3397    05 00736  6A7 00A34           BAL,LNK PROCESSCOL#PAIR PROCESS COL # PARAMS
3398    05 00737  323 000AC 02        LW,X1   CDTADR          CALC X1=ADDR IN CDT OF NEXT COMMAND
3399    05 00738  F2B 000AC 02        LB,R1   *CDTADR          AFTER 'SE'
3400    05 00739  303 00000B A        AW,X1   R1
3401    05 0073A  353 000E8 02        STW,X1  SETADR          PUT THIS IN SETADR FOR I:CMND LOOP
3402    05 0073B  6A7 00C61           BAL,LNK  READNXTRANDOM READ FIRST RECORD IN RANGE.
3403    05 0073C  35B 00169 02        STW,R1   SV1STSET       SET FIRST SEQ NO.
3404    05 0073D  35B 000B9 02        STW,R1   FIRSTSET       SET LOOP CONTROL
3405    05 0073E  35B 00005 A         STW,R1   P1
3406    05 0073F  31B 000BE 02        CW,R1    LASTSET        MAKE SURE THAT THE FIRST RECORD IS
3407    05 00740  682 00746           BLE      SET10          IN TH P1=P2 RANGE.
3408    05 00741  6A7 00DC4           BAL,LNK TYPEMSG         NO - TYPE: '=P1:NO SUCH REC'
3409    05 00742  000 001E6 02        DATA     ERRM6          'NONE'
3410    05 00743  228 00000 A         LI,T1   0               TURN OFF 'SET MODE' FLAG
3411    05 00744  358 000E9 02        STW,T1  SETFLAG
3412    05 00745  680 00004           B       MASTERPARSER    EXIT TO PARSER
3413                                *
3414                                *  SET EOD MARKER AND EXIT
3415                                *
3416    05 00746  6A7 00A56     SET10 BAL,LNK SETEOD          SET EOD MARKER
3417    05 00747  E80 0000D A         B       *I:LNK          EXIT
```

```
3418                                *
3419                                *
3420                                *
3421          05 00748              SET$L00P 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                            DO1      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    68300758                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    69300758                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 'TX' 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                                *
```

```
3455  05 00760  6A700E1F      STL10   BAL,LNK  WRITERANDBM        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  692007 4F             BG       STL5               YES - GO EXIT
3463  05 00767  35B000B9 02           STW,R1   FIRSTSET           NO - SAVE NEW SEQ #
3464  05 00768  228FFFFE 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  SETEOD             SET EOD MARKER
3469  05 0076D  328001 6A 02          LW,T1    SVBPFLAG            REATORE LAST DFLT VALUE
3470  05 0076E  358000 23 02          STW,T1   BPFLAG             OF BPFLAG FOR NEXT ITRATION
3471  05 0076F  68000 30C             B        RESTART$EXECUTIVE  GO RESTART I:CMND LOOP
3472                                  *
3473                                  *  ERROR: EOF HIT
3474                                  *
3475  05 00770  6A700DC4      STL20   BAL,LNK  TYPEMSG            TYPE: '=EOF HIT'
3476  05 00771  000001D5 02           DATA     ERRM1
3477  05 00772  68000 74F             B        STL5               GO EXIT
```

```
3478                                    PAGE
3479                                ************************************
3480                                *  INTRALINE COMMAND: 'DELETE' X   *
3481                                ************************************
3482                                *
3483                                *
3484          05 00773              I:DELETE EQU     *
3485                                * IT IS AN ERROR TO HAVE STRING SUB 1 ALL BLANKS.
3486                                *
3487                                *
3488    05 00773    33000020 02             MTW,0    ALLFLAG          SEE IF ALL FLAG IS SET.
3489    05 00774    69100779                BLZ      I:DELETE01
3490    05 00775    F2300175 02             LB,X1    *TEXTCADR        GET THE CHARACTER COUNT.
3491    05 00776    F2660175 02  I:DELETE02 LB,P2    *TEXTCADR,X1     SERACH THE STRING FOR ANY
3492    05 00777    21600040 A              CI,P2    X'40'            NON BLANK CHARACTER.
3493    05 00778    69300779                BNE      I:DELETE01       CONTINUE
3494    05 00779    64300776                BDR,X1   I:DELETE02
3495    05 00779    6A700DC4                BAL,LNK  TYPEMSG          ALL BLANKS MESSAGE
3496    05 00778    00000223 02             DATA     ERRM21
3497    05 0077C    68000004                B        MASTERPARSER
3498          05 0077D              I:DELETE01  EQU  $
3499    05 0077D    6A7009A4                BAL,LNK  FINDCOLUMN       FIND COLUMN CORRES TO FIRST PARAM
3500    05 0077E    E9800000D A             BCS,8    *I:LNK           NONE FOUND - EXIT
3501    05 0077F    30500006 A              AW,P1    P2               SET P1=CHAR AFTER PARAM STRING
3502    05 00780    32A00006 A              LW,P3    P2                   P2=0 (FIELD WIDTH)
3503    05 00781    22600000 A              LI,P2    0                    P3=# TO SHIFT (=LENGTH OF STRG)
3504    05 00782    6A700A70                BAL,LNK  SHIFTLEFT        SHIFT LEFT TO DELETE STRING
3505    05 00783    38500004 A              SW,P1    P3               IF ALLFLAG IS ON, SET TO RESUME
3506    05 00784    6A700978                BAL,LNK  ADJUSTALLFLAG     MATCHING AFTER X AS DELETED
3507    05 00785    6A700A56                BAL,LNK  SETEOD           RESET EOD MARKER
3508    05 00786    E8000000D A             B        *I:LNK           EXIT
```

```
3509                                    PAGE
3510                                ********************************************************
3511                                *  INTRALINE COMMAND: 'OVERWRITE AND EXTEND' X BY Y    *
3512                                ********************************************************
3513                                *
3514                                *
3515          05 00787              I:OVERWR$EXTEND   EQU $
3516    05 00787    35D00021 02               STW,I:LNK ALLOK
3517    05 00788    6A700A4                   BAL,LNK   FINDCOLUMN        FIND COLUMN CORRES TO FIRST PARAM
3518    05 00789    E980000D A                BCS,8     *I:LNK
3519    05 0078A    20300001 A                AI,X1     1
3520    05 0078B    F26600AC 02               LB,P2     *CDTADR,X1        GET ADDR OF 2ND STRING IN CDT
3521    05 0078C    306000AC 02               AW,P2     CDTADR            SET P2=ABSOLUTE ADDR OF STRING
3522    05 0078D    6A700A19                  BAL,LNK   MOVESTRING        OVERWRITE WITH NEW STRING
3523    05 0078E    F2800006 A                LB,T1     *P2               SET P1=COL. AFTER LAST NEW CHAR
3524    05 0078F    30500008 A                AW,P1     T1
3525    05 00790    22800040 A                LI,T1     ' '
3526                                *
3527                                *  BLANK OUT REST OF CARD IMAGE
3528                                *
3529    05 00791    315000BB 02   OEX10       CW,P1     LASTCLMN          BLANK OUT BUFFER FROM CHAR AFTER
3530    05 00792    68100796                  BGE       OEX20              LAST NEW CHAR TO COL. TO STOP AT
3531    05 00793    758A0024 02               STB,T1    CARDIMG,P1
3532    05 00794    20500001 A                AI,P1     1
3533    05 00795    6800079I                  B         OEX10
3534                                *
3535                                *  SET EOD AND EXIT
3536                                *
3537    05 00796    6A700A56      OEX20       BAL,LNK   SETEOD            RESET EOD MARKER
3538    05 00797    E800000D A                B         *I:LNK            EXIT
```

```
                                       PAGE
                                       ****************************************
                                       *  INTRALINE COMMAND: 'FOLLOW' X BY Y  *
                                       ****************************************
                                       *
                                       *
            05 00798                   I:FOLLOW$BY       EQU $
  05 00798      6A7009A4                        BAL,LNK   FINDCOLUMN          FIND COLUMN CORRES TO FIRST PARAM
  05 00799      E980000D A                      BCS,8     *I:LNK              NONE FOUND - EXIT
  05 0079A      30500006 A                      AW,P1     P2                  SET P1=CHAR AFTER PARAM STRING
  05 0079B      22600000 A                      LI,P2     0                       P2=0 (FIELD WIDTH)
  05 0079C      20300001 A                      AI,X1     1
  05 0079D      F24600AC 02                     LB,X2     *CDTADR,X1          GET ADDR OF 2ND STRING IN CDT
  05 0079E      304000AC 02                     AW,X2     CDTADR              SET X2=ABSOLUTE ADDR OF STRING
  05 0079F      F2A00004 A                      LB,P3     *X2                     P3=LENGTH OF STRING
  05 007A0      6A700A99                        BAL,LNK   SHIFTRIGHT          SHIFT RIGHT TO MAKE ROOM FOR 2ND
  05 007A1      32600004 A                      LW,P2     X2                   STRING
  05 007A2      6A700A19                        BAL,LNK   MOVESTRING          MOVE STRING INTO HOLE
  05 007A3      3050000A A                      AW,P1     P3                  IF ALLFLAG IS ON, SET TO RESUME
  05 007A4      6A700978                        BAL,LNK   ADJUSTALLFLAG        MATCHING AFTER Y AS ADDED
  05 007A5      6A700A56                        BAL,LNK   SETEOD              RESET EOD MARKER
  05 007A6      E800000D A                      B         *I:LNK              EXIT
```

```
                                        PAGE
                              ***********************************************
                              *  INTRALINE COMMAND: SHIFT X 'LEFT' BY N  *
                              ***********************************************
                              *
                              *
               05 007A7       I:SHIFT$LEFT       EQU $
05 007A7     35D00021 02              STW,I:LNK ALLOK
05 007A8     6A7009A4                 BAL,LNK  FINDCOLUMN          FIND COLUMN CORRES TO FIRST PARAM
05 007A9     E980000D A               BCS,8    *I:LNK              NONE FOUND - EXIT
05 007AA     20300001 A               AI,X1    1
05 007AB     F24600AC 02              LB,X2    *CDTADR,X1          GET ADDR OF N IN CDT
05 007AC     B2A800AC 02              LW,P3    *CDTADR,X2          SET P3=NUMBER TO SHIFT (N)
05 007AD     E830000D A               BEZ      *I:LNK              IF N=0 - EXIT
05 007AE     6A700A70                 BAL,LNK  SHIFTLEFT           SHIFT LEFT N SPACES
05 007AF     6A700A56                 BAL,LNK  SETEOD              RESET EOD MARKER
05 007B0     E800000D A               B        *I:LNK              EXIT
```

```
3578                                PAGE
3579                        ********************************************
3580                        *  INTRALINE COMMAND: 'OVERWRITE' X BY Y   *
3581                        ********************************************
3582                        *
3583                        *
3584          05 007B1      I:OVERWRITE       EQU $
3585    05 007B1  6A7009A4                    BAL,LNK  FINDCOLUMN       FIND COLUMN CORRES TO FIRST PARAM
3586    05 007B2  E980000D A                  BCS,8    *I:LNK           NONE FOUND - EXIT
3587    05 007B3  20300001 A                  AI,X1    1
3588    05 007B4  F26600AC 02                 LB,P2    *CDTADR,X1       GET ADR OF 2ND STRING IN CDT
3589    05 007B5  30600AC 02                  AW,P2    CDTADR           CALC P2=ABSOLUTE ADDR OF STRING
3590    05 007B6  6A700A19                    BAL,LNK  MOVESTRING       OVERWRITE WITH NEW STRING
3591    05 007B7  F2300006 A                  LB,X1    *P2              IF ALLFLAG IS ON, SET TO RESUME
3592    05 007B8  30500003 A                  AW,P1    X1                MATCHING AFTER Y AS OVERWRITTEN
3593    05 007B9  6A700978                    BAL,LNK  ADJUSTALLFLAG
3594    05 007BA  6A700A56                    BAL,LNK  SETEOD           RESET EOD MARKER
3595    05 007BB  E800000D A                  B        *I:LNK           EXIT
```

```
3596                                    PAGE
3597                                ****************************************
3598                                *  INTRALINE COMMAND: 'PRECEDE' X BY Y  *
3599                                ****************************************
3600                                *
3601                                *
3602          05 007BC              I:PRECEDE,BY      EQU $
3603    05 007BC    6A700A44              BAL,LNK   FINDCOLUMN        FIND COLUMN CORRES TO FIRST PARAM
3604    05 007BD    E980000D A            BCS,8     *I:LNK            NONE FOUND - EXIT
3605    05 007BE    20300001 A            AI,X1     1
3606    05 007BF    F24600AC 02           LB,X2     *CDTADR,X1        GET ADDR OF 2ND STRING IN CDT
3607    05 007C0    304000AC 02           AW,X2     CDTADR            SET X2=ABSOLUTE ADDR OF STRING
3608    05 007C1    F2A00004 A            LB,P3     *X2                   P3=LENGTH OF STRING
3609    05 007C2    6A700A99              BAL,LNK   SHIFTRIGHT        SHIFT RIGHT TO MAKE ROOM FOR 2ND
3610    05 007C3    46600004 A            XW,P2     X2                 STRING
3611    05 007C4    6A700A19              BAL,LNK   MOVESTRING        MOVE STRING INTO HOLE
3612    05 007C5    30500004 A            AW,P1     X2                IF ALLFLAG IS ON, SET TO RESUME
3613    05 007C6    30500000A A           AW,P1     P3                 MATCHING AFTER X AS PRECEDED BY Y
3614    05 007C7    6A700978              BAL,LNK   ADJUSTALLFLAG
3615    05 007C8    6A700A56              BAL,LNK   SETEOD            RESET EOD MARKER
3616    05 007C9    E800000D A            B         *I:LNK            EXIT
```

```
                                    PAGE
                             ******************************************
                             *  INTRALINE COMMAND: SHIFT X 'RIGHT' BY N  *
                             ******************************************
                             *
                             *
          05 007CA           I:SHIFT$RIGHT      EQU $
05 007CA     35D00021 02            STW,I:LNK ALLBK
05 007CB     6A7009A4               BAL,LNK  FINDCOLUMN         FIND COLUMN CORRES TO FIRST PARAM
05 007CC     E980000D A             BCS,8    *I:LNK             NONE - FOUND ERROR
05 007CD     20300001 A             AI,X1    1
05 007CE     F24600AC 02            LB,X2    *CDTADR,X1         GET ADDR OF N IN CDT
05 007CF     B2A800AC 02            LW,P3    *CDTADR,X2         SET P3=NUMBER TO SHIFT (N)
05 007D0     E830000D A             BEZ      *I:LNK             IF N=0 - EXIT
05 007D1     6A700A99               BAL,LNK  SHIFTRIGHT         SHIFT RIGHT N SPACES
05 007D2     6A700A56               BAL,LNK  SETEOD             RESET EOD MARKER
05 007D3     E800000D A             B        *I:LNK             EXIT
```

```
3634                                PAGE
3635                        ***********************************************
3636                        *  INTRALINE COMMAND: FOR X 'SUBSTITUTE' Y    *
3637                        ***********************************************
3638                        *
3639          05 007D4      I:SUBSTITUTE      EQU $
3640    05 007D4  6A7009A4          BAL,LNK   FINDCOLUMN        FIND COLUMN CORRES TO FIRST PARAM
3641    05 007D5  E980000D A        BCS,8     *I:LNK            NONE FOUND - EXIT
3642    05 007D6  20300001 A        AI,X1     1
3643    05 007D7  F24600AC 02       LB,X2     *CDTADR,X1        GET ADDR OF 2ND STRING IN CDT
3644    05 007D8  304000AC 02       AW,X2     CDTADR            SET X2=ABSOLUTE ADDR OF STRING
3645    05 007D9  F2A00004 A        LB,P3     *X2                   P3=LENGTH OF STRING
3646    05 007DA  32800005 A        LW,T1     P1                SAVE P1
3647    05 007DB  30500006 A        AW,P1     P2                SET P1=CHAR AFTER PARAM1 STRING
3648    05 007DC  38A00006 A        SW,P3     P2                CALC NUMBER TO SHIFT IN P3
3649    05 007DD  6820007E1         BLEZ      SBS10             IS NEW STRING LONGER THAN OLD STRING
3650    05 007DE  22600000 A        LI,P2     0                 YES - SET P2=0 (FIELD WIDTH)
3651    05 007DF  6A700A99          BAL,LNK   SHIFTRIGHT        SHIFT RIGHT AMOUNT OF DIFFERENCE
3652    05 007E0  6800007E5         B         SBS15             GO TO MOVE IN NEW STRING
3653                                *
3654                                *  NEW STRING SHORTER OR EQUAL THAN OLD ONE
3655                                *
3656    05 007E1  6830007E5         SBS10     BE        SBS15           ARE NEW AND OLD STRINGS OF = LENGTH
3657    05 007E2  3AA0000A A                  LCW,P3    P3              NO - NEW SHORTER
3658    05 007E3  22600000 A                  LI,P2     0               SET P2=0 (FIELD WIDTH)
3659    05 007E4  6A700A70                    BAL,LNK   SHIFTLEFT       SHIFT LEFT AMOUNT OF DIFFERENCE
3660                                *
3661                                *  MOVE NEW STRING INTO POSITION
3662                                *
3663    05 007E5  32500008 A        SBS15     LW,P1     T1              SET P1=COL. OF PARAM1 STRING
3664    05 007E6  32600004 A                  LW,P2     X2                  P2=ADDR OF NEW STRING
3665    05 007E7  6A700A19                    BAL,LNK   MOVESTRING      MOVE NEW STRING IN PLACE
3666    05 007E8  F2800006 A                  LB,T1     *P2             IF ALLFLAG IS ON, SET TO RESUME
3667    05 007E9  30500008 A                  AW,P1     T1               MATCHING AFTER Y AS SUBSTITUTED
3668    05 007EA  6A700978                    BAL,LNK   ADJUSTALLFLAG
3669    05 007EB  6A700A56                    BAL,LNK   SETEOD          RESET EOD MARKER
3670    05 007EC  E800000D A                  B         *I:LNK          EXIT
```

```
3671                                    PAGE
3672                                ******************************
3673                                *  INTRALINE C0MMAND: JUMP   *
3674                                ******************************
3675                                *
3676                                *
3677          05 007ED                I:JUMP    EQU     $
3678    05 007ED    33000167 02               MTW,0   STEPFLAG          IS SYSTEM IN 'STEP M0DE'
3679    05 007EE    683007F8                  BEZ     JMP10             N0 - ERR0R
3680    05 007EF    325000B9 02               LW,P1   FIRSTSET
3681    05 007F0    6A700E1F                  BAL,LNK WRITERAND0M       WRITE CURRENT REC0RD
3682    05 007F1    22300005 A                LI,X1   5                 GET SEQ # F0R JUMP FR0M CDT
3683    05 007F2    F24600AC 02               LB,X2   *CDTADR,X1
3684    05 007F3    B25800AC 02               LW,P1   *CDTADR,X2
3685    05 007F4    6A700C6D                  BAL,LNK READRAND0M        READ THIS REC0RD
3686    05 007F5    698007FB                  BCS,8   JMP15             DID IT EXIST
3687    05 007F6    355000B9 02               STW,P1  FIRSTSET          SAVE NEW SEQ #
3688    05 007F7    680006B8                  B       FINISH$STEP$L00P  YES - G0 FINISH JUMP
3689                                *
3690                                *  ERR0R: 'JU' ILLEGAL AT THIS P0INT
3691                                *
3692    05 007F8    6A700D8F        JMP10     BAL,LNK TYPECERR          TYPE: '=CN:CMND ILGL HERE'
3693    05 007F9    000001B1 02               DATA    ERRC4
3694    05 007FA    E8000C0D A                B       *I:LNK            EXIT
3695                                *
3696                                *  ERR0R: REC0RD T0 JUMP T0 D0ESN'T EXIST
3697                                *
3698    05 007FB    6A700D8F        JMP15     BAL,LNK TYPECERR          TYPE: '=CN:N0 SUCH REC'
3699    05 007FC    000001AD 02               DATA    ERRC3
3700    05 007FD    325000B9 02               LW,P1   FIRSTSET
3701    05 007FE    6A700C6D                  BAL,LNK READRAND0M        RESTORE 0LD REC0RD
3702    05 007FF    E8000C0D A                B       *I:LNK            EXIT
```

```
3703                                    PAGE
3704                                ************************************
3705                                *  INTRALINE COMMAND: NO CHANGE  *
3706                                ************************************
3707                                *
3708                                *
3709          05 00800              I:NO$CHANGE       EQU $
3710    05 00800    33000167 02               MTW,0     STEPFLAG          IS SYSTEM IN STEP MODE
3711    05 00801    68300805                  BEZ       NCG10             NO = ERROR
3712    05 00802    22800001 A                LI,T1     1
3713    05 00803    35800CC0 02               STW,T1    NOCHGFLG          TURN ON 'NO CHANGE' FLAG
3714    05 00804    E800000D A                B         *I:LNK            EXIT
3715                                *
3716                                *  ERROR: 'NO' ILLEGAL AT THIS POINT
3717                                *
3718    05 00805    6A700D8F        NCG10     BAL,LNK   TYPECERR          TYPE: '-CN:CMND ILGL HERE'
3719    05 00806    000001B1 02               DATA      ERRC4
3720    05 00807    E800000D A                B         *I:LNK            EXIT
```

```
                                         PAGE
                                ***********************************************************
                                *  INTRALINE COMMAND: REVERSE BLANK PRESERVATION FLAG  *
                                ***********************************************************
                                *
                                *
            05 00808            I:REVERSE$BPFLAG  EQU $
   05 00808    32800023 02               LW,T1     BPFLAG              REVERSE BPFLAG
   05 00809    48800000 02               EOR,T1    K1
   05 0080A    35800023 02               STW,T1    BPFLAG
   05 0080B    E800000D A                B         *I:LNK              EXIT
```

```
3732                                    PAGE
3733                                ****************************************************
3734                                *  INTRALINE COMMANDS: TYPE(SUPPRESSING SEQ. NUMBERS)  *
3735                                ****************************************************
3736                                *
3737                                *
3738          05 00810              I:TYPE    EQU       $
3739    05 00810    331001A4 02               MTW,1     ZERO:STG:FLG      DON'T OUTPUT ON TY
3740    05 00811    325000B9 02               LW,P1     FIRSTSET          GET SEQ #
3741    05 00812    6A700D82                  BAL,LNK   TYPECARD          TYPE CARD IMAGE WITH SEQ #
3742    05 00813    E800000D A                B         *I:LNK            EXIT
3743                                *
3744                                *
3745          05 00814              I:TYPE$SUP$SEQ    EQU $
3746    05 00814    331001A4 02               MTW,1     ZERO:STG:FLG      DON'T OUTPUT ON TS
3747    05 00815    225FFFFF A                LI,P1     -1
3748    05 00816    6A700D82                  BAL,LNK   TYPECARD          TYPE CARD IMAGE WITHOUT SEQ #
3749    05 00817    E800000D A                B         *I:LNK            EXIT
3750                                *
3751          05 00818              I:TYPEX   EQU       $
3752    05 00818    330001A2 02               MTW,0     TXFLAG            ANYTHING CHANGED ON THIS LINE
3753    05 00819    E830000D A                BEZ       *I:LNK            NO
3754    05 0081A    325000B9 02               LW,P1     FIRSTSET          GET SEQUENCE NO.
3755    05 0081B    6A700D82                  BAL,LNK   TYPECARD          TYPE CARD IMAGE WITH SEQ NO.
3756    05 0081C    22500000 A                LI,P1     0                 ZERO 'TX' FLAG
3757    05 0081D    355001A2 02               STW,P1    TXFLAG
3758    05 0081E    E800000D A                B         *I:LNK            RETURN
3759                                *
```

```
3760                                    PAGE
3761                                ********************************
3762                                *  ADD NEW PARAMETER TO CDT    *
3763                                *     P1 = TYPE OF PARAMETER   *
3764                                ********************************
3765                                *
3766                                *
3767          05 00081B             ADDCDTPARAM       EQU $
3768    05 00081B   022000030 A               PUSH      (X1,P1)          SAVE REGS
       05 00081C   0B30019C 02
3769    05 00081D   323000E5 02               LW,X1     PARAMPSN         BUILD CONTROL HW FOR PARAM IN CDT:
3770    05 00081E   F55600AC 02               STB,P1    *CDTADR,X1         BYTE 0: PARAM TYPE
3771    05 00081F   20300001 A                AI,X1     1                  BYTE 1: LOC OF PARAM VALUE RELA-
3772    05 00820    F25000AC 02               LB,P1     *CDTADR                  TIVE TO CURRENT CDTADR
3773    05 00821    F55600AC 02               STB,P1    *CDTADR,X1
3774    05 00822    332000E5 02               MTW,2     PARAMPSN         INCR TO NEXT HW
3775    05 00823    305000E6 02               AW,P1     PRMBUFSZ         ADJUST COUNT OF # OF WORDS IN ENTRY
3776    05 00824    F55000AC 02               STB,P1    *CDTADR           BY SIZE OF PARAM
3777    05 00825    385000E6 02               SW,P1     PRMBUFSZ
3778    05 00826    305000AC 02               AW,P1     CDTADR           SET P1=ABSOLUTE ADR TO PUT VALUE AT
3779    05 00827    323000E6 02               LW,X1     PRMBUFSZ
3780    05 00828    22400000 A                LI,X2     0
3781    05 00829    32E800C1 02               LW,D0     PARAMBUF,X2      MOVE PARAM VALUE TO CDT ENTRY
3782    05 0082A    B5E80005 A                STW,D0    *P1,X2
3783    05 0082B    20400001 A                AI,X2     1
3784    05 0082C    64300829                  BDR,X1    $-3
3785    05 0082D    B23000AC 02               LW,X1     *CDTADR          BUILD 'END OF CDT' MARKER USING
3786    05 0082E    4B300005 02               AND,X1    XFF00             NUMBER OF NEXT CMND IN CDT
3787    05 0082F    20300100 A                AI,X1     X'0100'
3788    05 00830    B5380005 A                STW,X1    *P1,X2           SET 'END OF CDT' MARKER
3789    05 00831    022000030 A               PULL      (X1,P1)          RESTORE REGS
       05 00832    0A30019C 02
3790    05 00833    680E0000 A                B         0,LNK            EXIT
```

```
3791                                    PAGE
3792                                ************************************
3793                                *  CHECK IF ONLY ONE ENTRY IN CDT  *
3794                                ************************************
3795                                *
3796                                *
3797          05 00834              CHECK1CDTENTRY    EQU $
3798    05 00834    32E00048 02               LW,D0     CDT               CHECK IF ONLY ONE ENTRY IN CDT
3799    05 00835    21E00001 A                CI,D0     1
3800    05 00836    683E0000 A                BE        0,LNK             YES - EXIT
3801    05 00837    6A700D8F                  BAL,LNK   TYPECERR          NO - TYPE: !=CN: CMND !LGL HERE!
3802    05 00838    000001B1 02               DATA      ERRC4
3803    05 00839    68000004                  B         MASTERPARSER      EXIT TO PARSER
```

```
3804                                    PAGE
3805                                ****************************************
3806                                *  GET FILE IDENTIFICATION             *
3807                                ****************************************
3808                                *
3809                                *
3810          05 00083A             GETFILEID       EQU $
3811    05 0083A    022000060 A             PUSH    (X1,T1)             SAVE REGS
        05 0083B    053000198 02
3812    05 0083C    223000000 A             LI,X1   0                   USE X1 AS COUNT OF # OF WDS PUSHED
3813                                        NXTNAM  ERRP3,,
3814    05 0083D    6A7000889               (NAME,*)
        05 0083E    010000236 02
        05 0083F    010000840
3815    05 00840    728000C1 02             LB,T1   PARAMBUF            ALLOW ONLY <= 31 BYTES IN FILE
3816    05 00841    218000IF A              CI,T1   31                  NAME.
3817    05 00842    682000845               BLE     $+3
3818    05 00843    227000E29       GF5     LI,LNK  L(ERRP3)
3819    05 00844    680008A1                B       GETNEXT$ERROR
3820    05 00845    6A7000B7F               BAL,LNK GF$PUSH$SUBR        PUSH 'FILE NAME' PARAM
3821    05 00846    328000AD 02             LW,T1   CHARPSN             SAVE NEXT SCAN PSN
3822                                        NXTNAM  ERRP4,,
3823                                        (NAME,*),,
3824                                        (S(LPAR,PERIOD),GF10),,
3825                                        (SCOL,ILGL$SEMICOLON),,
3826                                        (COM,*),,
3827    05 00847    6A7000889               (END,*)
        05 00848    053000239 02
        05 00849    010000B4E
        05 0084A    053000854
        05 0084B    083000035
        05 0084C    070000B4E
        05 0084D    000000B4E
3828    05 0084E    358000AD 02             STW,T1  CHARPSN             RESTORE TO SCAN , OR C/R AGAIN
3829    05 0084F    226000000 A             LI,P2   0
3830    05 00850    096000198 02            PUSH    P2                  SET 'ACCT #' & 'PASSWORD' PARAMS =0
3831    05 00851    096000198 02            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  01000850
      05 00857  05000858
3840  05 00858  22600000 A          LI,P2   0
3841  05 00859  09600190 02        PUSH    P2               SET 'ACCT #' 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         8 CHARACTERS MAX.
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  05000867
      05 00864  01000874
      05 00865  07000874
      05 00866  00000874
3858                               *
3859                               *  PASSWORD PRESENT: GET AND PROCESS IT
3860                               *
```

```
3861                                GF18    NXTNAM  ERRP3,1
3862  05 00867  6A700889                            (NAME,*)
      05 00868  01000236 02
      05 00869  0100086A
3863  05 0086A  728000C1 02                LB,T1   PARAMBUF          R 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                             DO1     MODE=2
3868  05 0086E  328000AD 02                LW,T1   CHARPSN
3869                                       NXTNAM  ERRP3,1
3870                                               (S(RPAR,NAME),GF30),1
3871                                               (S(RPAR,COM),GF30),1
3872  05 0086F  6A700889                           (S(RPAR,END),GF30)
      05 00870  03000236 02
      05 00871  01000877
      05 00872  07000877
      05 00873  00000877
3873                                *
3874                                *  NO PASSWORD PRESENT
3875                                *
3876  05 00874  22600000 A          GF20    LI,P2   0
3877  05 00875  09600198 02                PUSH    P2                SET 'PASSWORD' PARAM = 0
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          SET # OF PARAMS = X1
3883  05 00878                             DO1     MODE=2
3884  05 00878  358000AD 02                STW,T1  CHARPSN           SET TO RE-SCAN LAST POSITION
3885  05 00879  08600198 02                PULL    P2                RECONSTRUCT FID IN PARAMBUF
3886  05 0087A  356600C0 02                STW,P2  PARAMBUF-1,X1
3887  05 0087B  64300879                   BDR,X1  *-2
3888  05 0087C  02200060 A                 PULL    (X1,T1)           RESTORE REGS
      05 0087D  0A300198 02
3889  05 0087E  680E0000 A                 B       0,LNK             EXIT
3890                                *
```

```
3891                                    *  SUBR TO PUSH A NAME ONTO THE STACK
3892                                    *
3893          05 0087F                  GF$PUSH$SUBR    EQU $
3894    05 0087F    725000C1 02                 LB,P1     PARAMBUF          SET P1=LENGTH OF NAME IN BYTES
3895    05 00880    20500004 A                  AI,P1     4                 ADD 1 AND ROUND SO P1=LENGTH OF
3896    05 00881    2550007E A                  SLS,P1    -2                 TEXTC-STRING IN WDS
3897    05 00882    30300005 A                  AW,X1     P1                ADJ PUSH COUNT
3898    05 00883    22400000 A                  LI,X2     0
3899    05 00884    326800C1 02                 LW,P2     PARAMBUF,X2       PUSH TEXTC-STRING ONTO STACK
3900    05 00885    09600019C 02                PUSH      P2                 BACKWARDS
3901    05 00886    20400001 A                  AI,X2     1
3902    05 00887    64500884                    BDR,P1    $-3               LOOP
3903    05 00888    680E0000 A                  B         0,LNK             EXIT
```

```
3904                                    PAGE
3905                                ***********************************************
3906                                *  GET NEXT NAME FROM TELETYPE INPUT BUFFER     *
3907                                *    GEN,8,24  # OF BRANCHES,ADDR OF ERROR MSG  *
3908                                *    GEN,8,24  TYPE 1,BRANCH ADDR 1             *
3909                                *      ...       ...  ...  ...                  *
3910                                *    GEN,8,24  TYPE N,BRANCH ADDR N             *
3911                                ***********************************************
3912                                *
3913                                *
3914          05 00889              GETNEXTNAME       EQU $
3915    05 00889    02200040 A                PUSH    (X1,P2)           SAVE REGS
       05 0088A    05300198 02
3916    05 0088B    32600AD 02                LW,P2   CHARPSN           SET P2=PSN OF NEXT INPUT CHAR
3917    05 0088C    725C0176 02               LB,P1   TTYIMG,P2         GET INPUT CHAR
3918    05 0088D    20600001 A                AI,P2   1                 INCR CHAR PSN
3919    05 0088E    21500040 A                CI,P1   ' '               SKIP LEADING BLANKS
3920    05 0088F    68300888                  BE      $+3
3921    05 00890    22300004 A                LI,X1   GNTBL1SZ          CHECK IF CHAR CORRESPONDS TO ONE
3922    05 00891    715608CD                  CB,P1   GNTBL1,X1          OF THE 'GETNEXTNAME' TYPES
3923    05 00892    683008CB                  BE      GN50
3924    05 00893    64300891                  BDR,X1  $-2               NO - LOOP
3925    05 00894    22400001 A                LI,X2   1                 USE X2 AS INDEX INTO PARAMBUF
3926                                *
3927                                *  TEST IF CHAR CAN BELONG TO A FILE ID 'NAME'; IF SO, BUILD NAME
3928                                *  IN PARAMBUF
3929                                *
3930    05 00895    1950001C 02   GN10        CLM,P1  LETTERS           IS CHAR A LETTER OR DIGIT
3931    05 00896    689008AC                  BIL     GN30
3932    05 00897    1950001A 02               CLM,P1  DIGITS
3933    05 00898    689008AC                  BIL     GN30
3934          00000001                        DO      MODE=2
3935    05 00899    1950001E 02               CLM,P1  LCLETTERS
3936    05 0089A    689008AC                  BIL     GN30
3937                                          FIN
3938    05 0089B    22300008 A                LI,X1   GNTBL2SZ          NO - IS CHAR ONE OF THE OTHER LEGAL
3939    05 0089C    715608D1                  CB,P1   GNTBL2,X1          'NAME' CHARS
```

```
3940  05 0089D  683008AC          BE      GN30            YES - GO PUT CHAR IN PARAMBUF
3941  05 0089E  643008AC          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$ERROR   EQU $    (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  33E000E5 02             MTW,-2  PARAMPSN    YES - DECR PARAM PSN BY 1
3952  05 008A5  485000007 02            AND,P1  X1FFFF
3953  05 008A6  21500022E 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                              *  ERROR TYPE 'C': GO TO PRINT MESSAGE
3959                              *
3960  05 008AA  355002B7 02      GN25   STW,P1  DMY$TYPECERR+1  PUT ERROR MSG ADDR IN DUMMY CALL
3961  05 008AB  680002B6 02             B       DMY$TYPECERR    GO PRINT ERROR MSG
3962                              *
3963                              *  A LEGAL 'NAME' CHAR FOUND: PROCESS THIS
3964                              *
3965  05 008AC  7558C0C1 02      GN30   STB,P1  PARAMBUF,X2     PUT CHAR IN PARAMBUF
3966  05 008AD  20400001 A              AI,X2   1               INCR PARAMBUF INDEX
3967  05 008AE  725C0176 02             LB,P1   TTYIMG,P2       GET NEXT INPUT CHAR
3968  05 008AF  20600001 A              AI,P2   1               INCR CHAR PSN
3969  05 008B0  21500040 A              CI,P1   ' '             IS CHAR=BLANK
3970  05 008B1  693008B5                BNE     GN10            NO - GO GET NEXT CHAR
3971                              *
3972                              *  END OF 'NAME' FOUND: ADD TRAILING BLANKS AND FINISH BUILDING PARAMBUF
3973                              *
3974  05 008B2  22300003 A       GN35   LI,X1    3
3975  05 008B3  22500040 A              LI,P1   ' '
3976  05 008B4  7558C0C1 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  $-2
3979  05 008B7  204FFFFC A          AI,X2   -4                PUT COUNT IN PARAMBUF TO FORM
3980  05 008B8  754000C1 02         STB,X2  PARAMBUF           TEXTC-STRING
3981  05 008B9  20400004 A          AI,X2   4                 SET PARAMBUF SIZE = # WDS OF TEXT
3982  05 008BA  2540007E A          SLS,X2  -2
3983  05 008BB  22500001 A          LI,P1   NAME              SET TYPE='NAME'
3984  05 008BC  206FFFFF A          AI,P2   -1                SET CHAR PSN TO RESCAN LAST CHAR
3985                          *
3986                          *
3987                          *
3988        05 008BD          GETNEXT$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  B2F80007 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  0A300198 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   GNTYTBL1,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                          *
```

```
        05 008CD          GNTBL1   EQU      $
05 008CD      00     A             DATA,1   0                    (FILLER)
05 008CD 1    00     A             DATA,1   CR                    0: C/R HIT
05 008CD 2    6B     A             DATA,1   CM                   7: COMMA
05 008CD 3    4B     A             DATA,1   S(LP,PR)              9,11: LEFT PAREN,PERIOD
05 008CE      1B     A             DATA,1   S(RP,LF)             10,0 : RIGHT PAREN, LINE FEED.
        00000004          GNTBL1SZ EQU      BA($)-BA(GNTBL1)-1
                                   BOUND    4
                          *
                          *  TABLE OF TYPES CORRESPONDING TO LEGAL CHARS
                          *
        05 008CF          GNTYTBL1 EQU      $
05 008CF      00     A             DATA,1   0                    (FILLER)
05 008CF 1    00     A             DATA,1   0                     0: C/R HIT
05 008CF 2    07     A             DATA,1   7                    7: COMMA
05 008CF 3    0B     A             DATA,1   S(9,11)               9,11: LEFT PAREN,PERIOD
05 008D0      00     A             DATA,1   S(10,0)              10,0 : RIGHT PAREN, LINE FEED.
                                   BOUND    4
                          *
                          *  TABLE OF LEGAL SPECIAL CHARS IN A 'NAME'
                          *
        05 008D1          GNTBL2   EQU      $
05 008D1      00     A             DATA,1   0
05 008D1 1    6D     A             DATA,1   ' '
05 008D1 2    5B     A             DATA,1   '$'
05 008D1 3    5C     A             DATA,1   '*'
05 008D2      60     A             DATA,1   '-'
05 008D2 1    6C     A             DATA,1   '%'
05 008D2 2    7A     A             DATA,1   ':'
05 008D2 3    7B     A             DATA,1   '#'
05 008D3      7C     A             DATA,1   '@'
05 008D3 1                         DO1      MODE=1
                    *S*            DATA,1   ' '
        00000008          GNTBL2SZ EQU      BA($)-BA(GNTBL2)-1
                                   BOUND    4
```

```
4048                                    PAGE
4049                                ******************************************************
4050                                *  GET NEXT PARAMETER FROM TELETYPE INPUT BUFFER     *
4051                                *    GEN,8,24  # OF BRANCHES,ADDR OF ERROR MSG       *
4052                                *    GEN,8,24  TYPE 1,BRANCH ADDR 1                  *
4053                                *      ...       ...  ...  ...                       *
4054                                *    GEN,8,24  TYPE N,BRANCH ADDR N                  *
4055                                ******************************************************
4056                                *
4057                                *
4058          05 008D4              GETNEXTPARAM      EQU $
4059    05 008D4    02200040 A                PUSH    (X1,P2)           SAVE REGS
       05 008D5    05300198 02
4060    05 008D6    32600AD 02                LW,P2   CHARPSN           SET P2=PSN OF NEXT INPUT CHAR
4061    05 008D7    725C0176 02               LB,P1   TTYIMG,P2         GET INPUT CHAR
4062    05 008D8    20600001 A                AI,P2   1                 INCR CHAR PSN
4063    05 008D9    21500040 A                CI,P1   ' '               SKIP LEADING BLANKS
4064    05 008DA    683008D7                  BE      $-3
4065    05 008DB    22300005 A                LI,X1   GPTBLSZ           CHECK IF CHAR CORRESPONDS TO ONE
4066    05 008DC    7156095F                  CB,P1   GPTBL,X1          OF THE 'GETNEXTPARAM' TYPES
4067    05 008DD    683008F4                  BE      GP20
4068    05 008DE    643008DC                  BDR,X1  $-2               NO - LOOP
4069    05 008DF    1950001A 02               CLM,P1  DIGITS            CHECK IF CHAR IS A DIGIT
4070    05 008E0    68900917                  BIL     GP50
4071    05 008E1    2150004B A                CI,P1   '.'               CHECK IF CHAR IS A '.'
4072    05 008E2    68300917                  BE      GP50
4073    05 008E3    22400001 A                LI,X2   1                 NO - USE X2 AS INDEX INTO PARAMBUF
4074    05 008E4    21500061 A                CI,P1   '/'               CHECK IF A STRING FOUND
4075    05 008E5    683008F7                  BE      GP30
4076          00000001                        DO      MODE=2
4077    05 008E6    1950001E 02               CLM,P1  LCLETTERS
4078    05 008E7    689008EA                  BIL     GP10
4079                                          FIN
4080    05 008E8    1950001C 02               CLM,P1  LETTERS           NO - CHECK IF ALPHA TEXT FOUND
4081    05 008E9    699008A1                  BOL     GETNEXT$ERROR     NO - ERROR
4082                                *
4083                                *  ALPHABETIC TEXT FOUND; BUILD TEXTC-STRING IN PARAMBUF
```

```
                                    *
05 008EA   758000C1 02  GP10      STB,P1    PARAMBUF,X2       PUT CHAR IN PARAMBUF
05 008EB   20400001 A             AI,X2     1                 INCR PARAMBUF INDEX
05 008EC   725C0176 02            LB,P1     TTYIMG,P2         GET NEXT CHAR
05 008ED   20600001 A             AI,P2     1                 INCR CHAR PSN
              00000001            D0        MODE=2
05 008EE   1950001E 02            CLM,P1    LCLETTERS
05 008EF   689008EA               BIL       GP10
                                  FIN
05 008F0   1950001C 02            CLM,P1    LETTERS           IS CHAR A LETTER
05 008F1   689008EA               BIL       GP10              YES - LOOP
05 008F2   22500006 A             LI,P1     ALPH              NO - SET TYPE='ALPH'
05 008F3   68000905               B         GP40              GO FINISH UP
                                  *
                                  *  A LEGAL 'GETNEXTPARAM' TYPE  FOUND
                                  *
05 008F4   72560961     GP20      LB,P1     GPTYTBL,X1        SET P1=TYPE OF MATCH FOUND
05 008F5   22400001 A             LI,X2     1                 SET INDEX IN CASE OF DEFAULT
05 008F6   680008BD               B         GETNEXT$FINISH    GO FINISH UP
                                  *
                                  *  STRING FOUND: BUILD TEXTC-STRING IN PARAMBUF
                                  *
05 008F7   725C0176 02  GP30      LB,P1     TTYIMG,P2         GET NEXT INPUT CHAR
05 008F8   20600001 A             AI,P2     1                 INCR CHAR PSN
05 008F9   3160019A 02            CW,P2     TTYIMGSZ          CHECK IF END OF CMND HIT
05 008FA   69200914               BG        GP45              YES - ERROR
05 008FB   21500061 A             CI,P1     '/'               IS CHAR='/'
05 008FC   68300900               BE        GP35
05 008FD   758000C1 02  GP30A     STB,P1    PARAMBUF,X2       NO - PUT CHAR IN PARAMBUF
05 008FE   20400001 A             AI,X2     1                 INCR PARAMBUF INDEX
05 008FF   680008F7               B         GP30              LOOP
                                  *
                                  *  '/' FOUND: DETERMINE IF IT IS END OF STRING OR '//'
                                  *
05 00900   725C0176 02  GP35      LB,P1     TTYIMG,P2         GET NEXT INPUT CHAR
05 00901   20600001 A             AI,P2     1                 INCR CHAR PSN
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    ' '
4129  05 00907  75F800C1 02       STB,D1   PARAMBUF,X2     PUT 3 TRAILING BLANKS ON TEXT OR
4130  05 00908  20400001 A        AI,X2    1                STRING
4131  05 00909  643009O7          BDR,X1   $-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: '-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: '-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 SEG #
4155  05 00918  224FFFFF A        LI,X2    -1              USE X2 TO SHOW INTG(=1) OR SEG(>=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  195C0001A 02  GP52      CLM,P1    DIGITS            IS CHAR A DIGIT
4162  05 0091B  689C0920              BIL       GP52A             YES - GO ACCUMULATE IT
4163  05 0091C  215C004B A            CI,P1     '.'               IS CHAR A '.'
4164  05 0091D  693C0938              BNE       GP60
4165  05 0091E  224C0003 A            LI,X2     3                 YES - USE X3 TO CNT DIGITS AFTER '.'
4166  05 0091F  68C00928              B         GP53              GO PROCESS '.'
4167  05 00920  23F0000A A    GP52A     MI,D1     10                ACCUMULATE DIGIT
4168  05 00921  205FFF10 A            AI,P1     -'0'
4169  05 00922  30F00005 A            AW,D1     P1
4170  05 00923  31F00E2A              CW,D1     L(10000)
4171  05 00924  681C0931              BGE       GP53A
4172  05 00925  725C0176 02           LB,P1     TTYIMG,P2         GET NEXT INPUT CHAR
4173  05 00926  206C0001 A            AI,P2     1                 INCR CHAR PSN
4174  05 00927  68C0091A              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  206C0001 A            AI,P2     1                 INCR CHAR PSN
4180  05 0092A  195C0001A 02          CLM,P1    DIGITS            IS CHAR A DIGIT
4181  05 0092B  699C0934              BOL       GP55
4182  05 0092C  23F0000A A            MI,D1     10                YES - ACCUMULATE IT
4183  05 0092D  205FFF10 A            AI,P1     -'0'
4184  05 0092E  30F00005 A            AW,D1     P1
4185  05 0092F  204FFFFF A            AI,X2     -1                CHECK IF >3 DIGITS FOUND
4186  05 00930  681C0928              BGEZ      GP53              NO - LOOP
4187  05 00931  6A7C0D99      GP53A     BAL,LNK   TYPEPERR          YES - TYPE; '-PN:ILGL SEQ #'
4188  05 00932  0C000250 02           DATA      ERRP10
4189  05 00933  68C00004              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  683C0938              BE        GP60
```

```
4195  05 00936  23F0000A A          MI,D1   10              NO - ADJ SEQ # FOR MISSING DIGITS
4196  05 00937  644n0936            BDR,X2  $-1
4197                          *
4198                          *  END OF INTEGER OR SEQ #: SEE IF SEQ # PAIR PRESENT
4199                          *
4200  05 00938  213n0001 A    GP60    CI,X1   1               WAS THIS 2ND SEQ # OF PAIR
4201  05 00939  683n094B            BE      GP63
4202  05 0093A  215n0060 A          CI,P1   '-'             NO - DOES A '-' FOLLOW FIRST
4203  05 0093B  693n0957            BNE     GP66
4204  05 0093C  214FFFFF A          CI,X2   -1              YES - WAS FIRST AN INTEGER
4205  05 0093D  693n093F            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  223n0001 A          LI,X1   1               SET X1=2ND SEQ #
4209  05 00941  224FFFFF 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  206n0001 A          AI,P2   1               INCR CHAR PSN
4213  05 00945  1950001A 02         CLM,P1  DIGITS          IS CHAR A DIGIT
4214  05 00946  689n0920            BIL     GP52A           YES - GO ACCUMULATE IT
4215  05 00947  215n004B A          CI,P1   '.'             IS CHAR A '.'
4216  05 00948  693n0931            BNE     GP53A           NO - ERROR
4217  05 00949  224n0003 A          LI,X2   3               YES - USER X3 TO CNT DIGITS AFTER '.
4218  05 0094A  685n0928            B       GP53            GO PROCESS '.'
4219                          *
4220                          *  DONE WITH SECOND SEQ # OF PAIR: FINISH UP
4221                          *
4222  05 0094B  214FFFFF A    GP63    CI,X2   -1              WAS SECOND AN INTEGER
4223  05 0094C  693n094E            BNE     $+2
4224  05 0094D  23F003E8 A          MI,D1   1000            YES - CONVERT TO A SEQ #
4225  05 0094E  35F000C2 02         STW,D1  PARAMBUF+1      PUT VALUE IN PARAMBUF
4226  05 0094F  225n0003 A          LI,P1   SEQ2            SET TYPE='SEQ2'
4227  05 00950  224n0002 A          LI,X2   2               SET PARAMBUF SIZE = 2
4228  05 00951  206FFFFF A          AI,P2   -1              SET CHAR PSN TO RESCAN LAST CHAR
4229  05 00952  31F000C1 02         CW,D1   PARAMBUF        IS SEQ # 2 >= SEQ # 1
4230  05 00953  681n08BD            BGE     GETNEXT$FINISH  YES - GO FINISH UP
4231  05 00954  6A7n0D99            BAL,LNK TYPEPERR        NO - TYPE: '-PN:SEQ2<SEQ1'
```

```
4232  05 00955  00000254 02          DATA    ERRP11
4233  05 00956  68000004             B       MASTERPARSER        GO TO PARSER
4234                                 *
4235                                 *  NO '=' FOLLOWS FIRST; FINISH UP
4236                                 *
4237  05 00957  22500002 A  GP66     LI,P1   SEQ                 SET TYPE='INTG' OR 'SEQ' AS APPRO
4238  05 00958  214FFFFF A          CI,X2   -1
4239  05 00959  69300095B          BNE     $+2
4240  05 00959A 22500004 A          LI,P1   INTG
4241  05 00959B 35F000C1 02         STW,D1  PARAMBUF            PUT VALUE IN PARAMBUF
4242  05 00959C 22400001 A          LI,X2   1                   SET PARAMBUF SIZE = 1
4243  05 00959D 206FFFFF A          AI,P2   -1                  SET CHAR PSN TO RESCAN LAST CHAR
4244  05 00959E 680008BD            B       GETNEXT$FINISH      GO FINISH UP
4245                                 *
4246                                 *  TABLE OF LEGAL 'GETNEXTPARAM' MATCH CHARS
4247                                 *
4248        05 00959F               GPTBL   EQU     $
4249  05 00959F    00          A             DATA,1  0                   (FILLER)
4250  05 00959F 1  00          A             DATA,1  CR                   0: C/R HIT
4251  05 00959F 2  15          A             DATA,1  LF                   0: LINE FEED HIT.
4252  05 00959F 3                            DB1     MODE=2
4253  05 00959F 3  0C          A             DATA,1  FF                  0: FORM FEED HIT
4254  05 00960     6B          A             DATA,1  CM                   7: COMMA
4255  05 00960 1   5E          A             DATA,1  SC                   8: SEMI=COLON
4256        00000005            GPTBLSZ EQU     BA($)-BA(GPTBL)-1
4257                                 BOUND   4
4258                                 *
4259                                 *  TABLE OF TYPES CORRESPONDING TO LEGAL CHARS
4260                                 *
4261        05 00961                GPTYTBL EQU     $
4262  05 00961     00          A             DATA,1  0                   (FILLER)
4263  05 00961 1   00          A             DATA,1  0                    0: C/R HIT
4264  05 00961 2   00          A             DATA,1  0                    0: LINE FEED HIT.
4265  05 00961 3                            DB1     MODE=2
4266  05 00961 3   00          A             DATA,1  0                   0: FORM FEED HIT
4267  05 00962     07          A             DATA,1  7                    7: COMMA
4268  05 00962 1   08          A             DATA,1  8                    8: SEMI=COLON
```

4269 BOUND 4

```
4270                                PAGE
4271                        ***********************************************
4272                        *  CREATE NEW ENTRY IN CDT                    *
4273                        *     P1 = NUMBER OF COMMAND TO ADD           *
4274                        *     WORD AFTER BAL = NUMBER OF PARAMETERS   *
4275                        ***********************************************
4276                        *
4277                        *
4278          05 00963      NEWCDTENTRY       EQU $
4279    05 00963  022000020 A                 PUSH      (P1,P2)           SAVE REGS
        05 00964  0B50019C 02
4280    05 00965  255000008 A                 SLS,P1    8                 BUILD CONTROL WORD OF ENTRY:
4281    05 00966  495000048 02                BR,P1     CDT                 BYTE 0: LENGTH OF ENTRY (=0)
4282    05 00967  255000008 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  206000003 A                AI,P2     3                   (# OF PARAMETERS)/2+1
4287    05 0096C  2560007F 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 'END OF CDT' MARKER USING
4290    05 0096F  205000100 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  225000000 A                LI,P1     0
4293    05 00972  680000974                  B         $+2
4294    05 00973  B55C00AC 02                STW,P1    *CDTADR,P2        SET ALL PARAM CONTROL HW'S TO ZERO
4295    05 00974  646000973                  BDR,P2    $-1
4296    05 00975  022000020 A                PULL      (P1,P2)           RESTORE REGS
        05 00976  0A50019C 02
4297    05 00977  680E0001 A                 B         1,LNK             EXIT
```

```
                                              PAGE
                                     ************************************************
                                     *  ADJUST ALL FLAG                             *
                                     *    P1 = COLUMN NUMBER TO RESUME MATCHING AT  *
                                     ************************************************
                                     *
                                     *
             05 00978                ADJUSTALLFLAG    EQU $
     05 00978     33000020 02                 MTW,0   ALLFLAG           IS ALLFLAG ON
     05 00979     691E0000 A                  BLZ     0,LNK             NO - EXIT
     05 0097A     35500020 02                 STW,P1  ALLFLAG           YES - SET IT TO COL. TO RESUME MATCH
     05 0097B     680E0000 A                  B       0,LNK             EXIT
```

```
                                    PAGE
                                ******************************************************************
                                *  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 0097C              ANLZRIGHT          EQU $
  05 0097C    02200020 A                  PUSH     (P1,P2)           SAVE REGS
  05 0097D    0850019C 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    1300019C 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    EODCLMN           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 N0N-BLANK
4347  05 00992  6820098D            BLE      AR10A            N0 - C0NTINUE SCAN
4348                               *
4349                               *  N0W PAST LAST N0N-BLANK, CALC R2=NUMBER 0F BLANKS T0 END
4350                               *
4351  05 00993  22C0008C A  AR12    LI,R2    MAXCLMN          CALC R2=DISTANCE FR0M P1 T0 END 0F
4352  05 00994  38C00005 A          SW,R2    P1                BUFFER
4353  05 00995  6800099A            B        AR18             G0 EXIT
4354                               *
4355                               *  AT END 0F N0N-BLANKS, C0UNT BLANK FIELD
4356                               *
4357  05 00996  20C00001 A  AR15    AI,R2    1                INCR R2 & P1
4358  05 00997  20500001 A          AI,P1    1
4359  05 00998  716A0024 02         CB,P2    CARDIMG,P1       IS CHAR AT P1=BLANK
4360  05 00999  68300996            BE       AR15             YES - KEEP C0UNTING BLANKS
4361                               *
4362                               *  EXIT WITH CC1=0
4363                               *
4364  05 0099A  02200020 A  AR18    PULL     (P1,P2)          RESTORE REGS
      05 0099B  0A50019C 02
4365  05 0099C  02200000 A          LCI      0                SET CC1=0
4366  05 0099D  680E0000 A          B        0,LNK            EXIT
4367                               *
4368                               *  BP FLAG 0N, CALC R1 & R2
4369                               *
4370  05 0099E  32B000B1 02  AR20    LW,R1    E0DCLMN          CALC R1=DISTANCE FR0M P1 T0 LAST
4371  05 0099F  38B00005 A          SW,R1    P1                N0N-BLANK
4372  05 009A0  20B00001 A          AI,R1    1
4373  05 009A1  22C0008B A          LI,R2    MAXCLMN-1        CALC R2=NUMBER 0F TRAILING BLANKS
4374  05 009A2  38C000B1 02         SW,R2    E0DCLMN
4375  05 009A3  6800099A            B        AR18             G0 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             *
                                     ************************************************************
                                     *
            05 009A4                 FINDCOLUMN        EQU $
      05 009A4     0940019C 02                PUSH     X2,(LNK,T1)         SAVE REGS
      05 009A5     02200020 A
      05 009A6     0870019C 02
      05 009A7     22800000 A                 LI,T1    0                   SET T1=ALL OCCURRENCES
      05 009A8     32500020 02                LW,P1    ALLFLAG                 P1=COL. TO START MATCHING AT
      05 009A9     681009CC                   BGEZ     FC15                IS SYSTEM IN 'ALL' MODE
      05 009AA     325000B7 02                LW,P1    FRSTCLMN            NO - SET P1=COL. TO START AT
      05 009AB     22300003 A                 LI,X1    3                   GET NUMBER OF PARAMS IN CDT
      05 009AC     F28600AC 02                LB,T1    *CDTADR,X1
      05 009AD     21800002 A                 CI,T1    2                   ARE THERE > 2 PARAMS
      05 009AE     692009C1                   BG       FC10
      05 009AF     22300004 A                 LI,X1    4
      05 009B0     22800001 A                 LI,T1    1
      05 009B1     F24600AC 02                LB,X2    *CDTADR,X1          NO - GET PARAM1 TYPE
      05 009B2     22300005 A                 LI,X1    5
      05 009B3     21400005 A                 CI,X2    STRG                IS IT A STRING
      05 009B4     683009CD                   BE       FC15A               YES - FORM IS: /ST/ X -
      05 009B5     F24600AC 02                LB,X2    *CDTADR,X1          NO - FORM IS: C X -
      05 009B6     B25800AC 02                LW,P1    *CDTADR,X2          GET COL. # FROM CDT
      05 009B7     205FFFFF A                 AI,P1    -1                  ADJUST TO INTERNAL COL. #
      05 009B8     315000B7 02                CW,P1    FRSTCLMN            IS IT BELOW COL. TO START AT
      05 009B9     691009EC                   BL       FC45                YES - ERROR
      05 009BA     315000BB 02                CW,P1    LASTCLMN            IS TO BEYOND COL. TO STOP AT
      05 009BB     681009E9                   BGE      FC40                YES - ERROR
      05 009BC     22600001 A                 LI,P2    1                   NO - SET FIELD WIDTH = 1
      05 009BD     22300006 A                 LI,X1    6                   SET NEXT CDT CTRL BYTE = 6
```

```
4411  05 009BE  331001A4 02          MTW,1   ZER0:STG:FLG      DON'T OUTPUT '0 STRINGS'
4412  05 009BF  331001A2 02          MTW,1   TXFLAG            BUT TYPE IF 'TX' COMMAND
4413  05 009C0  680009DD             B       FC20              GO EXIT
4414                                 *
4415                                 *  THERE ARE 3 PARAMETERS: GET 'OCCURRENCE' COUNT
4416                                 *
4417  05 009C1  22300005 A  FC10     LI,X1   5
4418  05 009C2  F246000AC 02         LB,X2   *CDTADR,X1        SET T1=OCCURRENCE CNT IN CDT
4419  05 009C3  B2880AC 02           LW,T1   *CDTADR,X2
4420  05 009C4  21800000 A           CI,T1   0                 CHECK IF ALL IS LEGAL FOR THIS
4421  05 009C5  692009CC             BG      FC15              COMMAND
4422  05 009C6  33000021 02          MTW,0   ALLOK
4423  05 009C7  683009CC             BEZ     FC15
4424  05 009C8  6A7000D8F            BAL,LNK TYPECERR
4425  05 009C9  000001BF 02          DATA    ERRC7
4426  05 009CA  331000B2 02          MTW,1   ERRORCNT          ALLOW ONE MORE CERR
4427  05 009CB  22800001 A           LI,T1   1                 SUBSTITUTE 1
4428                                 *
4429                                 *  FIND CORRECT OCCURRENCE OF STRING IF IT EXISTS
4430                                 *
4431  05 009CC  22300007 A  FC15     LI,X1   7
4432  05 009CD  F266000AC 02 FC15A   LB,P2   *CDTADR,X1        SET P2=ABSOLUTE ADDR OF PARAM2
4433  05 009CE  306000AC 02          AW,P2   CDTADR             STRING
4434  05 009CF  6A7009EF             BAL,LNK FINDMATCH         FIND MATCH FOR STRING
4435  05 009D0  698009E2             BCS,8   FC30              IF NONE = ERROR
4436  05 009D1  33F0019F 02          MTW,-1   CHG:STG:CNT      SLOPY WAY OF FIXING THIS PROBLEM
4437  05 009D2  32500008 A           LW,P1   R1                SET P1=COL. TO RESUME MATCHING
4438  05 009D3  20500001 A           AI,P1   1
4439  05 009D4  648009CF             BDR,T1   $-5              LOOP IF NOT ON CORRECT OCCURRENCE
4440  05 009D5  3310019F 02          MTW,1    CHG:STG:CNT      INCREMENT COUNT HERE
4441  05 009D6  331001A2 02          MTW,1   TXFLAG            SET 'TX' FLAG
4442  05 009D7  21800000 A           CI,T1   0                 IF T1<0, 'ALL' MODE IS ACTIVE; IN
4443  05 009D8  681009DA             BGE     $+2                THIS MODE ALLFLAG>=0
4444  05 009D9  35500020 02          STW,P1  ALLFLAG
4445  05 009DA  205FFFFF A           AI,P1   -1                SET P1=COLUMN OF MATCH
4446  05 009DB  F2600006 A           LB,P2   *P2                   P2=LENGTH OF STRING
4447  05 009DC  20300001 A           AI,X1   1                     X1=NEXT CDT CONTROL BYTE
```

```
4448                                *
4449                                *  EXIT WITH CC1=0
4450                                *
4451   05 009DD  022000020 A   FC20     PULL     X2,(LNK,T1)        RESTORE REGS
       05 009DE  0A700019C 02
       05 009DF  084000019C 02
4452   05 009E0  022000000 A            LCI      0
4453   05 009E1  680E00000 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  358000020 02           STW,T1   ALLFLAG
4459                                *
4460                                *  EXIT WITH CC1=1
4461                                *
4462   05 009E4  022000020 A   FC35     PULL     X2,(LNK,T1)        RESTORE REGS
       05 009E5  0A700019C 02
       05 009E6  084000019C 02
4463   05 009E7  022000080 A            LCI      8
4464   05 009E8  680E00000 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
```

```
4477                                    PAGE
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    680000A04                     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 COLMN>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  022000080 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  223000001 A  FM20   LI,X1    1               SET X1=POSITION IN TEXTC-STRING
4518  05 00A07  324000005 A        LW,X2    P1                  X2=COL. # ON CARD
4519  05 00A08  F2600175  02       LB,P2    *TEXTCADR           P2=# OF CHARS TO COMPARE
4520  05 00A09  206FFFFF  A        AI,P2    -1
4521  05 00A0A  683000A11         BEZ      FM30            IF STRING IS 1 CHAR LONG - EXIT
4522  05 00A0B  203000001 A  FM20A  AI,X1    1               INCR X1 & X2
4523  05 00A0C  204000001 A        AI,X2    1
4524  05 00A0D  F2960175  02       LB,T2    *TEXTCADR,X1    DO 2 CHARS MATCH
4525  05 00A0E  719800024 02       CB,T2    CARDIMG,X2
4526  05 00A0F  693009FF          BNE      FM10B           NO - GO START 1ST CHAR SEARCH AGAIN
4527  05 00A10  646000A0B         BDR,P2   FM20A           YES - LOOP UNTIL CORRECT # MATCH
4528                                *
4529                                *  EXIT WITH MATCH FOUND (CC1=0)
4530                                *
4531  05 00A11  328000005 A  FM30   LW,R1    P1              MATCH FOUND - SET R1=COL. # OF MATCH
4532  05 00A12  21700A5AF          CI,LNK    FNDTYP+1       IF I CAME FROM A FT,FD OR FS TYPE OF
4533  05 00A13  683000A15          BE        *+2            COMMAND DON;T INCERMENT CHG;STG;CNT.
4534  05 00A14  331001 9F 02       MTW,1     CHG;STG;CNT    COUNT THE NO. OF HITS.
4535  05 00A15  022000070 A        PULL      (X1,T2)        RESTORE REGS
      05 00A16  0A300019C 02
4536  05 00A17  022000000 A        LCI       0
4537  05 00A18  680E0000  A        B         0,LNK          EXIT WITH CC1=0
```

```
4538                                    PAGE
4539                                ******************************************
4540                                *  MOVE STRING TO CARD                   *
4541                                *    P1 = COLUMN AT WHICH TO PUT STRING  *
4542                                *    P2 = ADDR OF TEXTC-STRING TO MOVE   *
4543                                ******************************************
4544                                *
4545                                *
4546        05 00A19                MOVESTRING      EQU $
4547   05 00A19   022000050 A               PUSH    (X1,LNK)            SAVE REGS
       05 00A1A   0B300019C 02
4548   05 00A1B   256000002 A               SLS,P2  2                   CONVERT P2 TO A BYTE ADDR
4549   05 00A1C   723C0000 A               LB,X1   0,P2                SET X1=# OF CHARS TO MOVE
4550   05 00A1D   2150008C A               CI,P1   MAXCLMN             IS STARTING COL. BEYOND END OF CARD
4551   05 00A1E   681000A2B               BGE     MS20A-1             YES - GO CHECK
4552                                *
4553                                *  MOVE CHAR FROM TEXTC-STRING TO CARD
4554                                *
4555   05 00A1F   206000001 A       MS5     AI,P2   1                   INCR TO NEXT TEXTC-STRING CHAR
4556   05 00A20   724C0000 A               LB,X2   0,P2                MOVE CHAR TO CARD
4557   05 00A21   754A0024 02              STB,X2  CARDIMG,P1
4558   05 00A22   205000001 A              AI,P1   1                   INCR COLUMN
4559   05 00A23   2150008C A               CI,P1   MAXCLMN
4560   05 00A24   681000A29               BGE     MS20                HAS END OF BUFFER BEEN PASSED
4561   05 00A25   643000A1F               BDR,X1  MS5                 NO - LOOP UNTIL ALL CHARS MOVED
4562                                *
4563                                *  EXIT
4564                                *
4565   05 00A26   022000050 A       MS10    PULL    (X1,LNK)            RESTORE REGS
       05 00A27   0A300019C 02
4566   05 00A28   680E0000 A               B       0,LNK               EXIT
4567                                *
4568                                *  AT END OF BUFFER: IF MORE NON-BLANKS TO MOVE, TYPE ERROR MESSAGE
4569                                *
4570   05 00A29   203FFFFF A       MS20    AI,X1   -1                  END OF BUFFER: ARE THERE MORE CHARS
4571   05 00A2A   683000A26               BEZ     MS10                 TO MOVE
4572   05 00A2B   224000040 A               LI,X2   ' '
```

```
4573  05 00A2C  20600001 A  MS20A   AI,P2     1          YES - IS NEXT CHAR OF TEXTC-STRING
4574  05 00A2D  714C0000 A          CB,X2     0,P2        A BLANK
4575  05 00A2E  693 00A31           BNE       MS20B      NO - TYPE ERROR MSG
4576  05 00A2F  643 00A2C           BDR,X1    MS20A      YES - LOOP UNTIL ALL CHARS CHECKED
4577  05 00A30  68 000A26           B         MS10       ALL BLANKS - GO EXIT
4578  05 00A31  6A7 00D8F   MS20B   BAL,LNK   TYPECERR   TYPE: !--CN:OVERFLOW!
4579  05 00A32  000001A5 02         DATA      ERRC1
4580  05 00A33  68 000A26           B         MS10       GO EXIT
```

```
4581                                    PAGE
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    022000040 A               PUSH      (X1,P2)             SAVE REGS
        05 00A35    053000198 02
4590    05 00A36    226000000 A               LI,P1     0                   SET P1=DFLT STARTING COL. #
4591    05 00A37    226000088 A               LI,P2     MAXCLMN                 P2=DFLT STOPPING COL. #
4592    05 00A38    F24600AC 02               LB,X2     *CDTADR,X1          GET NEXT PARAM TYPE
4593    05 00A39    683000A3F                 BEZ       PP10                IS PARAM PRESENT
4594    05 00A3A    203000001 A               AI,X1     1
4595    05 00A3B    F24600AC 02               LB,X2     *CDTADR,X1          YES - SET P1=STARTING COL #
4596    05 00A3C    825800AC 02               LW,P1     *CDTADR,X2
4597    05 00A3D    205FFFFF A                AI,P1     -1                  ADJUST TO INTERNAL COL #
4598    05 00A3E    203FFFFF A                AI,X1     -1
4599                                *
4600                                *  PROCESS SECOND COLUMN NUMBER PARAMETER
4601                                *
4602    05 00A3F    203000002 A   PP10          AI,X1     2
4603    05 00A40    F24600AC 02               LB,X2     *CDTADR,X1          GET NEX PARAM TYPE
4604    05 00A41    683000A45                 BEZ       PP20                IS PARAM PRESENT
4605    05 00A42    203000001 A               AI,X1     1
4606    05 00A43    F24600AC 02               LB,X2     *CDTADR,X1          YES - SET P2=STOPPING COL # + 1
4607    05 00A44    826800AC 02               LW,P2     *CDTADR,X2
4608                                *
4609                                *  FINISH INITIALIZATION AND EXIT
4610                                *
4611    05 00A45    355000B7 02   PP20          STW,P1    FRSTCLMN            SET STARTING AND STOPPING COL #'S
4612    05 00A46    356000BB 02               STW,P2    LASTCLMN
4613    05 00A47    315000006 A               CW,P1     P2
4614    05 00A48    681000A50                 BGE       PP25
4615    05 00A49    215000000 A               CI,P1     0
4616    05 00A4A    691000A50                 BL        PP25
```

```
4617  05 00A4B  2160008C A        CI,P2    MAXCLMN
4618  05 00A4C  69200A50          BG       PP25
4619  05 00A4D  02200040 A        PULL     (X1,P2)          RESTORE REGS
      05 00A4E  0A30019C 02
4620  05 00A4F  680E0000 A        B        0,LNK            EXIT
4621                        *
4622  05 00A50  6A700DC4     PP25  BAL,LNK  TYPEMSG          TYPE! '-BAD COL. NO. PAIR'
4623  05 00A51  000001D0 02       DATA     ERRC11
4624  05 00A52  22700000 A        LI,LNK   0
4625  05 00A53  357000E9 02       STW,LNK  SETFLAG
4626  05 00A54  35700167 02       STW,LNK  STEPFLAG
4627  05 00A55  68000004          B        MASTERPARSER
```

```
4628                                    PAGE
4629                                ************************************
4630                                *  FIND COLUMN OF LAST NON-BLANK   *
4631                                ************************************
4632                                *
4633                                *
4634          05 00A56              SETEOD  EQU     $
4635    05 00A56    33000140 02             MTW,O   RP$FLAG         CHECK RECORD PRESERVATION
4636    05 00A57    683E0000 A              BEZ     0,LNK           'ON', DONT CHANGE RECSIZE
4637    05 00A58    02200020 A              PUSH    (X1,X2)         SAVE REGS
        05 00A59    03300198 02
4638    05 00A5A    22300022 A              LI,X1   MAXCLMN/4-1
4639    05 00A5B    32400008 02             LW,X2   4BLNKS
4640    05 00A5C    31460024 02             CW,X2   CARDIMG,X1      MAKE GROSS COMPARISON FOR ALL
4641    05 00A5D    69300A67                BNE     SRS10           BLANK WORDS.
4642    05 00A5E    64300A5C                BDR,X1  $-2
4643                                *
4644                                *
4645    05 00A5F    22300003 A              LI,X1   3               CHECK FIRST WORD BY BYTE.
4646    05 00A60    71460024 02 SRS5        CB,X2   CARDIMG,X1      ITERATE THROUGH BYTES OF
4647    05 00A61    69300A6A                BNE     SRS15           TARGET WORD.
4648    05 00A62    64300A60                BDR,X1  $-2
4649                                *
4650    05 00A63    71400024 02             CB,X2   CARDIMG         CHECK FIRST BYTE OF FIRST WORD,
4651    05 00A64    69300A6A                BNE     SRS15           FOR BLANK.
4652    05 00A65    223FFFFF A              LI,X1   -1              IF BLANK, RECORD SIZE =0.
4653    05 00A66    68000A6A                B       SRS15
4654    05 00A67    25300002 A  SRS10       SLS,X1  2               REVERT TO BYTE INDEXING, TO GET
4655    05 00A68    20300003 A              AI,X1   3               BYTE WITHIN WORD.
4656    05 00A69    68000A60                B       SRS5
4657                                *
4658    05 00A6A    353000B1 02 SRS15       STW,X1  EODCLMN         SAVE ENDING COLUMN (BYTE INDEX)
4659    05 00A6B    20300001 A              AI,X1   1
4660    05 00A6C    353000E7 02             STW,X1  RECSIZE         AND RECORD SIZE (TRUE BYTE COUNT)
4661    05 00A6D    02200020 A              PULL    (X1,X2)
        05 00A6E    0A300198 02
4662    05 00A6F    680E0000 A              B       0,LNK           EXIT
```

```
4663                                PAGE
4664                        ****************************************************
4665                        *  SHIFT STRING LEFT                               *
4666                        *    P1 = COLUMN AT WHICH TO START SHIFT           *
4667                        *    P2 = WIDTH OF FIELD STARTING AT THIS COLUMN    *
4668                        *    P3 = NUMBER TO SHIFT LEFT                     *
4669                        ****************************************************
4670                        *
4671                        *
4672          05 00A70      SHIFTLEFT       EQU *
4673   05 00A70  022000080 A                PUSH    (P1,R2)         SAVE REGS
       05 00A71  0B50019C 02
4674   05 00A72  305000006 A                AW,P1   P2              START ANLZ AFTER ORIG FIELD
4675   05 00A73  6A7000976C                 BAL,LNK ANLZRIGHT       ANLZ FIELD AT P1
4676   05 00A74  698000A93                  BCS,8   SL30            OOPS - FIELD IS BEYOND END OF CARD
4677   05 00A75  385000006 A                SW,P1   P2              RESTORE P1
4678                        *
4679                        *  COMPUTE WHERE TO SHIFT TO, COMPENSATING IF SHIFT PUSHES DATA OFF
4680                        *  LEFT END OF CARD
4681                        *
4682   05 00A76  30B000006 A  SL3            AW,R1   P2              SET R1=WIDTH OF FIELD AT P1 TO SHIFT
4683   05 00A77  326000005 A                LW,P2   P1              CALC: P1=BEGINNING OF 'FROM' FIELD
4684   05 00A78  3860000A A                 SW,P2   P3                    P2=BEGINNING OF 'TO' FIELD
4685   05 00A79  681000A80                  BGEZ    SL5             DOES THIS SHIFT OFF LEFT END OF CARD
4686   05 00A7A  6A7000D8F                  BAL,LNK TYPECERR        YES - TYPE: '-=CN:UNDERFLOW'
4687   05 00A7B  000001A9 02                DATA    ERRC2
4688   05 00A7C  385000006 A                SW,P1   P2              FIX UP 'FROM' COL. AND WIDTH SO AS
4689   05 00A7D  30B000006 A                AW,R1   P2               TO SHIFT ONLY TO COL. 0
4690   05 00A7E  68200A8E                   BLEZ    SL20            DOES SHIFT PUSH ENTIRE FIELD OFF CRD
4691   05 00A7F  226000000 A                LI,P2   0               NO - SET 'TO'=COL. 0
4692                        *
4693                        *  SHIFT FIELD AT P1 LEFT
4694                        *
4695   05 00A80  21B000000 A  SL5            CI,R1   0               IS WIDTH OF FIELD TO SHIFT = 0
4696   05 00A81  683000A87                  BE      SL10            YES - SKIP SHIFT
4697   05 00A82  728A0024 02  SL5A           LB,T1   CARDIMG,P1      SHIFT LEFT
4698   05 00A83  758C0024 02                STB,T1  CARDIMG,P2
```

```
4699  05 00A84  20500001 A            AI,P1    1
4700  05 00A85  20600001 A            AI,P2    1
4701  05 00A86  64B00A82              BDR,R1   SL5A
4702                          *
4703                          *  BLANK OUT CLEARED CHARS ON RIGHT
4704                          *
4705  05 00A87  22800040 A    SL10    LI,T1    ' '                 BLANK OUT
4706  05 00A88  758C0024 02           STB,T1   CARDIMG,P2
4707  05 00A89  20600001 A            AI,P2    1
4708  05 00A8A  64A00A88              BDR,P3   $-2
4709  05 00A8B  02200080 A            PULL     (P1,R2)             RESTORE REGS
      05 00A8C  0A50019C 02
4710  05 00A8D  680E0000 A            B        0,LNK               EXIT
4711                          *
4712                          *  SHIFT PUSHES EVERYTHING, INCLUDING FIELD AT P1, OFF CARD, SO BLANK
4713                          *  OUT AND EXIT
4714                          *
4715  05 00A8E  30A00006 A    SL20    AW,P3    P2                  CALC P3=# OF COLUMNS WIPED OUT
4716  05 00A8F  38B00006 A            SW,R1    P2
4717  05 00A90  30A0000B A            AW,P3    R1
4718  05 00A91  22600000 A            LI,P2    0                   SET 'TO' FOR BLANKING=0
4719  05 00A92  68000A87              B        SL10                GO BLANK OUT
4720                          *
4721                          *  FIELD TO SHIFT IS BEYOND END OF CARD: SET UP TO SHIFT IN BLANKS
4722                          *
4723  05 00A93  38500006 A    SL30    SW,P1    P2                  RESTORE R1
4724  05 00A94  2150008C A            CI,P1    MAXCLMN             IS FIELD BEYOND END OF CARD
4725  05 00A95  69100A76              BL       SL3                 NO - CONTINUE NORMALLY
4726  05 00A96  32600005 A            LW,P2    P1                  SET P2=COL. AT WHICH TO START
4727  05 00A97  3860000A A            SW,P2    P3                   BLANKING OUT
4728  05 00A98  68000A87              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                   *
                                   ****************************************************
                                   *
                                   *
            05 00A99               SHIFTRIGHT         EQU $
     05 00A99    2150008C A                CI,P1      MAXCLMN           IS FIELD BEYOND END OF CARD
     05 00A9A    681E0000 A                BGE        0,LNK             YES - EXIT
     05 00A9B    022000A0 A                PUSH       (X1,R2)           SAVE REGS
     05 00A9C    0B30019C 02
     05 00A9D    22800000 A                LI,T1      0                 SET CNTS=0
     05 00A9E    358000B5 02               STW,T1     FIELDCNT
     05 00A9F    35800022 02               STW,T1     BLANKCNT
     05 00AA0    30500006 A                AW,P1      P2                START ANLZ AFTER ORIG FIELD
     05 00AA1    2150008C A                CI,P1      MAXCLMN           DOES FIELD ABUTT END OF CARD
     05 00AA2    68300B03                  BE         SR70              YES - GO PROCESS
                                   *
                                   *  BUILD 2-WD DATA BLOCK FOR EACH FIELD TO BE COMPRESSED AND PUSH
                                   *  ON STACK
                                   *
     05 00AA3    6A70097C          SR5     BAL,LNK    ANLZRIGHT         ANLZ FIELD AT P1
     05 00AA4    69800ADD                  BCS,8      SR50              OOPS - END OF CARD
     05 00AA5    66C00022 02               AWM,R2     BLANKCNT          CNT BLNKS TO COMPRESS
     05 00AA6    55B0000C A                STH,R1     R2
     05 00AA7    30B00005 A                AW,R1      P1                BUILD: R1=COLUMN AT END OF NON-BLNKS
     05 00AA8    20BFFFFF A                AI,R1      -1                       R2=(# OF NON-BLNKS,# TO SHFT)
     05 00AA9    331000B5 02               MTW,1      FIELDCNT          CNT FIELDS COMPRESSED
     05 00AAA    31A00022 02               CW,P3      BLANKCNT          ARE ENOUGH BLNKS COMPRESSED YET
     05 00AAB    68200AB3                  BLE        SR8               YES
     05 00AAC    02200020 A                PUSH       (R1,R2)           NO - SAVE FIELD DATA BLOCK
     05 00AAD    0BB0019C 02
     05 00AAE    3250000B A                LW,P1      R1                INCR P1 TO NEXT FIELD
     05 00AAF    4BC00006 02               AND,R2     XFFFF
```

```
4764  05 00AB0  305000C A          AW,P1   R2
4765  05 00AB1  205000002 A        AI,P1   2
4766  05 00AB2  68000AA3           B       SR5              ANLZ NEXT FIELD
4767                               *
4768                               *  INITIALIZE TO DO 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  358000022 02       STW,T1  BLANKCNT         CLEAR BLNK CNT
4776  05 00AB9  33F000B5 02        MTW,-1  FIELDCNT         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  528000004 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  560000004 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  556000004 A        STH,P2  X2
4787                               *
4788                               *  SET UP PARAMETERS FOR CURRENT SHIFT
4789                               *
4790  05 00AC1  528000004 A  SR12    LH,T1   X2               SET T1=# OF CHARS IN FIELD TO SHIFT
4791  05 00AC2  208000001 A        AI,T1   1                 (INCLUDING PRECEDING BLANK)
4792  05 00AC3  4B4000006 02 SR12A   AND,X2  XFFFF            KEEP CUMULATIVE CNT OF BLNKS
4793  05 00AC4  664000022 02       AWM,X2  BLANKCNT          COMPRESSED OUT
4794  05 00AC5  324000003 A        LW,X2   X1               CALC: X1=END OF 'FROM' FIELD
4795  05 00AC6  304000022 02       AW,X2   BLANKCNT               X2=END OF 'TO' FIELD
4796  05 00AC7  218000000 A        CI,T1   0                IS # OF CHARS TO SHIFT = 0
4797  05 00AC8  68300ACE           BE      SR15A            YES - SKIP SHIFT
4798                               *
4799                               *  DO CURRENT SHIFT, THEN CHECK NUMBER LEFT TO DO
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  203FFFFF A          AI,X1   =1
4804  05 00ACC  204FFFFF A          AI,X2   =1
4805  05 00ACD  648 00AC9           BDR,T1  SR15
4806  05 00ACE  33F 000B5 02 SR15A  MTW,=1  FIELDCNT        DECR FIELD CNT
4807  05 00ACF  691 00AD5           BLZ     SR20            <0 = ALL SHIFTS DONE
4808  05 00AD0  02200020 A          PULL    (X1,X2)         >=0 = GET NEXT FIELD DATA BLOCK
      05 00AD1  0A300 19C 02
4809  05 00AD2  330 000B5 02       MTW,0   FIELDCNT        TEST FIELD CNT
4810  05 00AD3  692 00AC1          BGZ     SR12            >0 = 1 OR MORE FIELDS LEFT
4811  05 00AD4  680 00ABE          B       SR10            =0 = AT 1ST FIELD
4812                          *
4813                          *  ALL SHIFTS DONE, SO BLANK OUT CLEARED CHARS ON LEFT
4814                          *
4815  05 00AD5  32800022 02  SR20   LW,T1    BLANKCNT
4816  05 00AD6  22900040 A   SR20A  LI,T2    ' '             BLANK OUT
4817  05 00AD7  75980024 02         STB,T2  CARDIMG,X2
4818  05 00AD8  204FFFFF A          AI,X2   =1
4819  05 00AD9  648 00AD7           BDR,T1  $-2
4820  05 00ADA  02200 0A0 A         PULL    (X1,R2)         RESTORE REGS
      05 00ADB  0A300 19C 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  6A7 00D8F    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 00AE0  468 00022 02        XW,T1   BLANKCNT        SET T1=(# OF NON-BLNKS TO DESTROY)
4829  05 00AE1  38800000A A         SW,T1   P3
4830  05 00AE2  02200020 A          PULL    (R1,R2)         START ON LAST FIELD
      05 00AE3  0AB 019C 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                          *
```

```
4835  05 00AE4  5U80000C A  SR52    AH,T1    R2              IS CURRENT FIELD (+OTHERS ALREADY
4836  05 00AE5  6820OAEE           BLEZ     SR55             WIPED OUT) LONG ENOUGH FOR OVERFLOW
4837  05 00AE6  58B0000C A  SR52A   SH,R1    R2              YES -BUILD R1 & R2 AS BEFORE:
4838  05 00AE7  30B00008 A          AW,R1    T1               R1=COLUMN AT END OF NON-BLNKS
4839  05 00AE8  50C0000C A          AH,R2    R2                   NOT DESTROYED
4840  05 00AE9  4BC00006 02         AND,R2   XFFFF            R2=(# OF NON-BLNKS NOT DESTROYED,
4841  05 00AEA  38C00008 A          SW,R2    T1                   ,# TO SHIFT)
4842  05 00AEB  30C00022 02         AW,R2    BLANKCNT
4843  05 00AEC  5580000C A          STH,T1   R2
4844  05 00AED  6800OAB5           B        SR8A             GO SHIFT
4845                              *
4846                              *  NOT ENOUGH ROOM FOUND YET, GET NEXT FIELD DOWN AND DESTROY PART OF IT
4847                              *
4848  05 00AEE  5290000C A  SR55    LH,T2    R2               KEEP CUMULATIVE CNT OF CHARS
4849  05 00AEF  3090000C A          AW,T2    R2                DESTROYED
4850  05 00AF0  4B900006 02         AND,T2   XFFFF
4851  05 00AF1  6690OO22 02         AWM,T2   BLANKCNT
4852  05 00AF2  33F000B5 02         MTW,-1   FIELDCNT         DECR FIELD CNT
4853  05 00AF3  6830OAF9           BEZ      SR58             =0 - AT 1ST FIELD
4854  05 00AF4  02200020 A          PULL     (R1,R2)          >0 - GET NEXT FIELD DATA BLOCK
      05 00AF5  0AB00190 02
4855  05 00AF6  20B00001 A          AI,R1    1                INC. FOLLOWING BLANK IN FIELD
4856  05 00AF7  20C10000 A          AI,R2    X'10000'
4857  05 00AF8  6800OAE4           B        SR52
4858                              *
4859                              *  AT 1ST FIELD AND STILL NOT ENOUGH ROOM
4860                              *
4861  05 00AF9  30800006 A  SR58    AW,T1    P2               ADD IN ORIG FIELD AT P1 AND CHK ROOM
4862  05 00AFA  6820OAFF           BLEZ     SR60
4863  05 00AFB  58B0000C A          SH,R1    R2               ENOUGH FOUND - FIX R1 & R2 TO
4864  05 00AFC  22C00000 A          LI,R2    0                 DESTROY PART OF ORIG FIELD AT P1
4865  05 00AFD  5560000C A          STH,P2   R2
4866  05 00AFE  6800OAE6           B        SR52A
4867                              *
4868                              *  SHIFT PUSHS ALL FIELDS OFF CARD, SO BLANK OUT AND EXIT
4869                              *
4870  05 00AFF  6660OO22 02  SR60   AWM,P2   BLANKCNT         SET UP TO BLANK FROM ORIG P1
```

```
4871  05 00B00  32400005 A          LW,X2    P1
4872  05 00B01  204FFFFE A          AI,X2    =2
4873  05 00B02  68000AD5            B        SR20              GO BLANK OUT
4874                          *
4875                          *  FIELD TO SHIFT ABUTTS END OF CARD; SET UP TO PERFORM THIS SHIFT
4876                          *
4877  05 00B03  6A700D8F      SR70  BAL,LNK  TYPECERR          TYPE: '=-CN:OVERFLOW'
4878  05 00B04  00000A5 02          DATA     ERRC1
4879  05 00B05  38600000A A        SW,P2    P3                DOES SHIFT PUSH ORIG FIELD OFF CARD
4880  05 00B06  68200B0D            BLEZ     SR72
4881  05 00B07  35A00022 02        STW,P3   BLANKCNT          NO - SET BLANKCNT=# OF CHARS TO
4882  05 00B08  22300008B A        LI,X1    MAXCLMN-1          BLANK OUT
4883  05 00B09  38300000A A        SW,X1    P3                SET X1=END OF 'FROM' FIELD
4884  05 00B0A  22400008B A        LI,X2    MAXCLMN-1             X2=LAST COLUMN ON CARD
4885  05 00B0B  32800006 A         LW,T1    P2                    T1=# OF CHARS TO SHIFT
4886  05 00B0C  68000AC9            B        SR15              GO SHIFT THIS FIELD
4887                          *
4888                          *  ABUTTING FIELD IS SHIFTED OFF CARD, SO SET UP TO BLANK OUT
4889                          *
4890  05 00B0D  30600000A A  SR72   AW,P2    P3                SET T1=# OF CHARS TO BLANK OUT
4891  05 00B0E  32800006 A          LW,T1    P2                        (=ORIG FIELD WIDTH)
4892  05 00B0F  22400008B A         LI,X2    MAXCLMN-1             X2=LAST COLUMN ON CARD
4893  05 00B10  68000AD6            B        SR20A             GO BLANK OUT
```

```
                                      PAGE
                                 ****************************************
                                 *  CONVERT BINARY TO DECIMAL STRING    *
                                 *     P1 = BINARY NUMBER               *
                                 *     P2 = BYTE ADDR TO PUT STRING IN  *
                                 ****************************************
                                 *
                                 *
        05 00B11                 BINTODEC EQU     $
05 00B11    02200020 A                    PUSH     (P1,P2),X1      SAVE RGS
05 00B12    0B50019C 02
05 00B13    0930019C 02
05 00B14    20600007 A                    AI,P2    7               SET P2=LAST BYTE ADDR OF STRING
05 00B15    32F00005 A                    LW,D1    P1
05 00B16    22500007 A                    LI,P1    7               SET TO LOOP 7 TIMES
05 00B17    22E00000 A           BD10     LI,D0    0
05 00B18    36E00001 02                   DW,D0    K10             EXTRACT RIGHTMOST DIGIT
05 00B19    20E000F0 A                    AI,D0    '0'             CONVERT TO EBCDIC AND PUT IN STRING
05 00B1A    75EC0000 A                    STB,D0   0,P2
05 00B1B    206FFFFF A                    AI,P2    -1
05 00B1C    64500B17                      BDR,P1   BD10            LOOP
05 00B1D    22F00040 A                    LI,D1    ' '             SET 1ST BYTE = BLANK
05 00B1E    75FC0000 A                    STB,D1   0,P2
05 00B1F    20600001 A                    AI,P2     1               GET PAST THE BLANK
05 00B20    22300006 A                    LI,X1     6              NUMBER OF FIELDS TO CHECK.
05 00B21    22F00040 A                    LI,D1     X'40'          NULL CHARACTER
05 00B22    225000F0 A                    LI,P1     X'F0'
        05 00B23                 BD30     EQU      $
05 00B23    715C0000 A                    CB,P1    0,P2            CHECK FOR ZERO
05 00B24    69300B28                      BNE      BD20            NOT EQU IS NOT A LEADING ZERO
05 00B25    75FC0000 A                    STB,D1   0,P2            MAKE IT NULL
05 00B26    20600001 A                    AI,P2    1               INCREMENT BYTE ADR.
05 00B27    64300B23                      BDR,X1   BD30            GO FIND ANOTHER
        05 00B28                 BD20     EQU      $
05 00B28    0830019C 02                   PULL     (P1,P2),X1
05 00B29    02200020 A
05 00B2A    0A50019C 02
```

```
4927   05 0082B    680E0000 A          B          0,LNK              EXIT
```

```
4928                                    PAGE
4929                                ********************************
4930                                *       BLANK INPUT BUFFER     *
4931                                ********************************
4932                                *
4933                                *
4934   05 00B2C   097C019C 02   BLANKBUF PUSH     LNK
4935   05 00B2D   227C0023 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   647C0B2F              BDR,LNK   *-1
4940                                *
4941   05 00B31   087C019C 02           PULL      LNK
4942   05 00B32   680E0000 A            B         0,LNK
```

```
4943                                    PAGE
4944                                ************************
4945                                *  CLOSE UPDATE FILE   *
4946                                ************************
4947                                *
4948                                *
4949          05 00B33              CLOSE    EQU      $
4950    05 00B33    04100011 03              M:CLOSF  F:EI,(SAVE)
        03 00011    15000000 N
                    80000000
        03 00013    00000002 A
4951    05 00B34    680E0000 A               B        0,LNK
4952                                ***********************
4953                                *  CLOSE COPY FILE    *
4954                                ***********************
4955                                *
4956                                *
4957          05 00B35              CLOSE2   EQU      $
4958    05 00B35    04100014 03              M:CLOSF  F:EO,(SAVE)
        03 00014    15000000 N
                    80000000
        03 00016    00000002 A
4959    05 00B36    680E0000 A               B        0,LNK
4960                                *
4961                                *
4962    05 00B37    04100017 03  CLOSE3      M:CLOSF  F:EO,(REL)
        03 00017    15000000 N
                    80000000
        03 00019    00000001 A
4963    05 00B38    680E0000 A               B        0,LNK
```

```
                                   PAGE
                          *****************************************************
                          *  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    *
                          *****************************************************
                          *
                          *
        05 00B39          DELETE   EQU      $
05 00B39   02200040 A              PUSH     (P1,T1)          SAVE REGS
05 00B3A   0B50019C 02
05 00B3B   22800000 A              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  READNXTRANDOM    READ 1ST SEQ # OR NEXT HIGHEST #
                          *
                          *  READ AND DELETE UNTIL LAST SEQ # READ OR PASSED
                          *
05 00B3D   31B00E28       DL10     CW,R1    L(EOF)           WAS AN EOF READ
05 00B3E   68300B67                BE       DL30             YES - GO TYPE ERROR MESSAGE
05 00B3F   3160000B A              CW,P2    R1               NO - WAS INPUT SEQ # >= LAST SEQ #
05 00B40   68200B4D                BLE      DL15             YES - GO FINISH UP
05 00B41   35B0019B 02             STW,R1   DELNXT           N TO DELETE BUFFER
05 00B42   3250019B 02             LW,P1    DELNXT
05 00B43   20500001 A              AI,P1    1
05 00B44   6A700C61                BAL,LNK  READNXTRANDOM    READ N+1
05 00B45   22700003 A              LI,LNK   3                SET KEY LENGTH
05 00B46   7570019B 02             STB,LNK  DELNXT
```

```
5000  05 00B47  04100001A 03          M:DELRFC F:EI,(KEY,DELNXT) DELETE N
      03 0001A  00000000 N
                80000000
      03 0001C  000019B 02
5001  05 00B48  20800001 A           AI,T1    1                 BUMP DELETED RECORD COUNTER
5002        00000001                 DB       MODE=2
5003  05 00B49  32500198 02          LW,P1    DELNXT
5004  05 00B4A  4B500E28             AND,P1    =X'FFFFFF'       GET RID OF THE BYTE COUNT
5005  05 00B4B  355002C3 02          STW,P1   INTFLAG1
5006                                 FIN
5007  05 00B4C  68000B3D             B        DL10              N+1 IS OK, SET TO DELETE IT
5008                          *
5009                          *  LAST SEQ # HIT OR PASSED: IF HIT, FINISH UP AND EXIT WITH CC1=0
5010                          *
5011  05 00B4D  69100B5C      DL15   BL       DL20              WAS LAST SEQ # PASSED
5012  05 00B4E  6A700B95             BAL,LNK  DELETERECORD      NO, WAS HIT - DELETE IT
5013  05 00B4F                       DB1      MODE=2
5014  05 00B4F  35B002C3 02          STW,R1   INTFLAG1
5015  05 00B50  20800001 A           AI,T1    1                 INCR DELETE COUNT
5016  05 00B51  32500008 A          LW,P1    T1
5017  05 00B52  21500001 A           CI,P1     1                DON'T SAY ANYTHING IF ITS ONLY 1
5018  05 00B53  68300B58             BE        DL17
5019  05 00B54  22600A31 02          LI,P2    BA(MSG6)+1
5020  05 00B55  6A700B11             BAL,LNK  BINTODEC
5021  05 00B56  6A700DC4             BAL,LNK   TYPEMSG
5022  05 00B57  00000028C 02         DATA      MSG6
5023        05 00B58                 DL17     EQU       $
5024  05 00B58  02200040 A           PULL     (P1,T1)           RESTORE REGS
      05 00B59  0A50019C 02
5025  05 00B5A  02200000 A           LCI      0
5026  05 00B5B  680E0000 A           B        0,LNK             EXIT WITH CC1=0
5027                          *
5028                          *  LAST SEQ # WAS PASSED: EXIT WITH CC1=1
5029                          *
5030  05 00B5C  32500008 A   DL20    LW,P1    T1
5031  05 00B5D  21500001 A           CI,P1     1                DON'T SAY ANYTHING IF ITS ONLY 1
5032  05 00B5E  68200B63             BLE      DL25              OR LESS
```

```
5033  05 00B5F  226600A31 02          LI,P2    BA(MSG6)+1
5034  05 00B60  6A700B11              BAL,LNK  BINTODEC
5035  05 00B61  6A700DC4              BAL,LNK   TYPEMSG
5036  05 00B62  000002BC 02          DATA      MSG6
5037        05 00B63                  DL25      EQU       $
5038  05 00B63  022000040 A           PULL      (P1,T1)          RESTORE REGS
      05 00B64  0A500019C 02
5039  05 00B65  022000080 A           LCI       8
5040  05 00B66  680E0000 A            B         0,LNK            EXIT WITH CC1=1
5041                                  *
5042                                  *  ERROR: EOF HIT
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
```

```
5047                                        PAGE
5048                                ********************************************************
5049                                *  DELETE FILE                                         *
5050                                *    P1 = ADDR OF FILE ID IN CDT                       *
5051                                *    CC1=1 IF FILE DOES NOT EXIST; CC1=0 OTHERWISE     *
5052                                ********************************************************
5053                                *
5054                                *
5055                                        LOCAL   $20,$50
5056          05 00B6A              DELETEFILE      EQU $
5057   05 00B6A    02200080 A               PUSH    (X1,P3)
       05 00B6B    0B30019C 02
5058   05 00B6C    22600B7A                 LI,P2   DF$ABN
5059   05 00B6D    3560036A 02              STW,P2  B2$FPT+2
5060   05 00B6E    22600004 A               LI,P2   4                       INOUT
5061   05 00B6F    3560036D 02              STW,P2  B2$FPT+5
5062   05 00B70    22800371 02              LI,T1   B2$NAME
5063   05 00B71    2290037A 02              LI,T2   B2$ACCT
5064   05 00B72    22A0037D 02              LI,P3   B2$PASS
5065   05 00B73    6A700C08                 BAL,LNK OPENINIT
5066   05 00B74    04100368 02              CAL1,1  B2$FPT
5067   05 00B75    04100010 03              M:CLOSF F:E0,(REL)      FILE EXISTS, SO CLOSE AND RELEASE
       03 0001D    19000000 N
                   80000000
       03 0001F    00000001 A
5068   05 00B76    02200080 A               PULL    (X1,P3)
       05 00B77    0A30019C 02
5069   05 00B78    02200000 A               LCI     0
5070   05 00B79    680E0000 A               B       0,LNK
5071                                *
5072                                *
5073                                *
5074   05 00B7A                     DF$ABN  RES     0
5075   05 00B7A    72300000A A              LB,X1   P3
5076   05 00B7B    21300003 A               CI,X1   3
5077   05 00B7C    69300B82                 BNE     BADIO1
5078   05 00B7D    02200080 A               PULL    (X1,P3)
```

```
          05 00B7E   0A30019C 02
5079      05 00B7F   02200080 A              LCI       8
5080      05 00B80   680E0000 A              B         0,LNK
5081                                *
5082                                *  ERROR: BAD I/O
5083                                *
5084      05 00B81                 BADIO     RES       0
5085      05 00B81   323F000F A              LW,X1     D1                  MOVE CODE TO X1.
5086      05 00B82                 BADIO1    RES       0                   ENTER HERE IF CODE IN X1.
5087      05 00B82   253F027C A              SCS,X1    =4                  BUILD ERROR CODE
5088      05 00B83   7246000E 02             LB,X2     HEXCHAR,X1
5089      05 00B84   754002A5 02             STB,X2    IOERRCOD
5090      05 00B85   25300064 A              SLS,X1    =28
5091      05 00B86   7246000E 02             LB,X2     HEXCHAR,X1
5092      05 00B87   22100001 A              LI,X3     1
5093      05 00B88   754202A5 02             STB,X2    IOERRCOD,X3
5094      05 00B89   7242000A A              LB,X2     P3,X3               GET SUBCODE
5095      05 00B8A   2540007F A              SLS,X2    =1                  AND RIGHT JUSTIFY
5096      05 00B8B   2540027C A              SCS,X2    =4                  BUILD ERROR SUBCODE
5097      05 00B8C   7238000E 02             LB,X1     HEXCHAR,X2
5098      05 00B8D   20100002 A              AI,X3     2
5099      05 00B8E   753202A5 02             STB,X1    IOERRCOD,X3
5100      05 00B8F   25400064 A              SLS,X2    =28
5101      05 00B90   7238000E 02             LB,X1     HEXCHAR,X2
5102      05 00B91   753002A6 02             STB,X1    IOERRCOD+1
5103      05 00B92   6A700DC4               BAL,LNK   TYPEMSG
5104      05 00B93   000002A0 02             DATA      IOERRMSG
5105           00000000                     DO        MODE=1
5106                              *S*        CAL3,6    0
5107                                         ELSE
5108      05 00B94   04900002 A              M:ERR                         ERROR TO UTS.
5109                                         FIN
```

```
5110                                    PAGE
5111                                ********************************
5112                                *  DELETE LAST RECORD READ     *
5113                                ********************************
5114                                *
5115                                *
5116          05 00B95              DELETERECORD      EQU $
5117    05 00B95    04100020 03               M:DELREC F:E:,(KEY,LASTKEY)
        03 00020    00000000 N
                    80000000
        03 00022    000000BD 02
5118    05 00B96    680E0000 A                B        0,LNK
```

```
                                          PAGE
                                   **************************************************
                                   *  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   *
                                   **************************************************
                                   *
                                   *
               05 00B97            MOVESEQ  EQU      $
      05 00B97      02200060 A              PUSH     (X4,LNK)           SAVE REGS
      05 00B98      0B20019C 02
      05 00B99      32200007 A              LW,X4    LNK                SAVE LINK
      05 00B9A      3560016E 02             STW,P2   TEMPBLCK+3         SAVE P2
      05 00B9B      226005AC 02             LI,P2    BA(TEMPBLCK)
      05 00B9C      6A700B11                BAL,LNK  BINTODEC           CONVERT SEQ # TO EBCDIC: ' DDDDDDD'
      05 00B9D      3260016E 02             LW,P2    TEMPBLCK+3         RESTORE P2
      05 00B9E      22300001 A              LI,X1    1
      05 00B9F      22400003 A              LI,X2    3
      05 00BA0      22E000F0 A              LI,D0    '0'
      05 00BA1      71E6016B 02             CB,D0    TEMPBLCK,X1        CALC X1=POSITION OF 1ST NON-ZERO
      05 00BA2      69300BA5                BNE      MQ10                CHAR OR 4TH DIGIT
      05 00BA3      20300001 A              AI,X1    1
      05 00BA4      64400BA1                BDR,X2   $-3
                                   *
                                   *  SUPPRESS TRAILING ZEROES
                                   *
      05 00BA5      22500007 A     MQ10     LI,P1    7
      05 00BA6      22400003 A              LI,X2    3
      05 00BA7      71EA016B 02             CB,D0    TEMPBLCK,P1        CALC P1=POSITION OF 1ST NON-ZERO
      05 00BA8      69300BAB                BNEZ     MQ20                DIGIT FROM RIGHT OF 4TH DIGIT
      05 00BA9      205FFFFF A              AI,P1    -1
      05 00BAA      64400BA7                BDR,X2   $-3
                                   *
                                   *  BUILD STRING TO LEFT OF DECIMAL POINT
                                   *
```

```
5155  05 00BAB  72E60168 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   '.'              MOVE '.' TO ADDR IN P2
5162  05 00BB2  75EC0000 A           STB,D0  0,P2
5163  05 00BB3  20600001 A           AI,P2   1
5164                           *
5165                           *  BUILD STRING TO RIGHT OF DECIMAL POINT
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  72E60168 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                           *  APPEND 4 SPECIFIED CHARS
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
      05 00BC5  0A20019C 02
5187  05 00BC6  38B00006 A           SW,R1   P2               CALC R1=NUMBER OF CHARS IN STRING
5188  05 00BC7  680E0001 A           B       1,LNK            EXIT
```

```
                                         PAGE
                                  ****************************************************
                                  *  ØPEN UPDATE FILE                                *
                                  *  ØPEN UPDATE FILE (ØPEN1 ØPENS CØPY INPUT FILE)  *
                                  *    P1 = ADDR ØF FILE ID IN CDT                   *
                                  *    CC1=1 IF FILE DØES NØT EXIST; CC1=0 ØTHERWISE *
                                  *    CC2=1 IF FILE IS NØT KEYED; CC2=0 ØTHERWISE   *
                                  ****************************************************
                                  *
                                  *
                                          LØCAL     $20,$90
          05 00BC8                ØPEN    EQU       $
05 00BC8     02200080 A                   PUSH      (X1,P3)
05 00BC9     0B30019C 02
05 00BCA     22600004 A                   LI,P2     4                INØUT
05 00BCB     68000000 F                   B         $20
                                  *
                                  *
          05 00BCC                ØPEN1   EQU       $
05 00BCC     02200080 A                   PUSH      (X1,P3)
05 00BCD     0B30019C 02
05 00BCE     22600001 A                   LI,P2     1                INPUT
                                  *
                                  *
05 00BCF                          $20     RES       0
05 00BCF     35600356 02                  STW,P2    Ø$FPT+5
05 00BD0     22600BE4                     LI,P2     Ø$ABN
05 00BD1     35600353 02                  STW,P2    Ø$FPT+2
05 00BD2     2280035A 02                  LI,T1     Ø$NAME           SET ADDRESS REGISTERS FØR
05 00BD3     22900363 02                  LI,T2     Ø$ACCT               STØRING PARAMETERS INTØ
05 00BD4     22A00366 02                  LI,P3     Ø$PASS               FPT.
05 00BD5     6A700C08                     BAL,LNK   ØPENINIT
05 00BD6     04100351 02                  CAL1,1    Ø$FPT            ØPEN FILE
05 00BD7     32300005 N                   LW,X1     F:EI+5           FILE EXISTS.
05 00BD8     2530007C A                   SLS,X1    -4               ØRGANIZATIØN SHØULD BE KEYED.
05 00BD9     4B300003 02                  AND,X1    XF
05 00BDA     21300002 A                   CI,X1     2
```

```
5224   05 00BDB   693000000 F          BNE     $90
5225   05 00BDC   022000080 A          PULL    (X1,P3)          IT IS.
       05 00BDD   0A3001900 2
5226   05 00BDE   022000000 A          LCI     0
5227   05 00BDF   680E00000 A          B       0,LNK
5228                              *
5229                              *
5230   05 00BE0                   $90  RES     0
5231   05 00BE0   022000080 A          PULL    (X1,P3)
       05 00BE1   0A3001900 2
5232   05 00BE2   022000040 A          LCI     4
5233   05 00BE3   680E00000 A          B       0,LNK
5234                              *
5235                              *
5236                              *
5237   05 00BE4                   B$ABN RES     0
5238   05 00BE4   723000000A A         LB,X1   P3
5239   05 00BE5   213000003 A          CI,X1   3
5240   05 00BE6   693000882            BNE     BADIB1
5241   05 00BE7   022000080 A          PULL    (X1,P3)          NO FILE.
       05 00BE8   0A3001900 2
5242   05 00BE9   022000080 A          LCI     8
5243   05 00BEA   680E00000 A          B       0,LNK
```

```
5244                                    PAGE
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 00BEB  022000080 A         OPEN3   PUSH    (X1,P3)
      05 00BEC  0B300019C 02
5254  05 00BED  226000002 A                 LI,P2   2                       OUTPUT
5255  05 00BEE  680000000 F                 B       $20
5256                                *
5257                                *
5258        05 00BEF                OPEN2   EQU     $
5259  05 00BEF  022000080 A                 PUSH    (X1,P3)
      05 00BF0  0B300019C 02
5260  05 00BF1  226000004 A                 LI,P2   4                       INOUT
5261  05 00BF2  356000360 02       $20     STW,P2  O2$FPT+5
5262  05 00BF3  22600BFE                    LI,P2   O2$ABN
5263  05 00BF4  356000364 02                STW,P2  O2$FPT+2
5264                                *
5265  05 00BF5  228000371 02                LI,T1   O2$NAME                 SAME.
5266  05 00BF6  229000374 02                LI,T2   O2$ACCT
5267  05 00BF7  22A000377 02                LI,P3   O2$PASS
5268  05 00BF8  6A700C08                    BAL,LNK OPENINIT
5269  05 00BF9  041000368 02                CAL1,1  O2$FPT
5270  05 00BFA  022000080 A                 PULL    (X1,P3)
      05 00BFB  0A300019C 02
5271  05 00BFC  022000000 A                 LCI     0
5272  05 00BFD  680E00000 A                 B       0,LNK
5273                                *
5274                                *
5275                                *
5276  05 00BFE                      O2$ABN  RES     0
5277  05 00BFE  723000004 A                 LB,X1   P3
```

```
5278  05 00BFF  213000003 A   CI,X1    3
5279  05 00C00  693008B82     BNE      BADIO1
5280  05 00C01  223000002 A   LI,X1    2                  NO PREVIOUS FILE, OPEN FOR OUTPUT.
5281  05 00C02  353000360 02  STW,X1   02$FPT+5
5282  05 00C03  041000368 02  CAL1,1   02$FPT
5283  05 00C04  022000080 A   PULL     (X1,P3)
      05 00C05  0A3000190 02
5284  05 00C06  022000080 A   LCI      8
5285  05 00C07  680E00000 A   B        0,LNK
```

```
5286                                    PAGE
5287                                ***********************************************
5288                                *  INITIALIZE OPEN FPT                        *
5289                                *    P1 = ADDR OF FILE ID IN CDT              *
5290                                *    T1 = FPT ENTRY TO PUT FILE NAME IN       *
5291                                *    T2 = FPT ENTRY TO PUT ACCOUNT NUMBER IN  *
5292                                *    P3 = FPT ENTRY TO PUT PASSWORD IN        *
5293                                ***********************************************
5294                                *
5295                                *
5296                                        LOCAL   $50,$60,$65,$70,$80
5297          05 00C08              OPENINIT EQU    $
5298    05 00C08    323000A 02              LW,X1   4BLNKS
5299    05 00C09    B53000009 A             STW,X1  *T2
5300    05 00C0A    B530000A A              STW,X1  *P3
5301    05 00C0B    22400001 A              LI,X2   1
5302    05 00C0C    B5380009 A              STW,X1  *T2,X2
5303    05 00C0D    B538000A A              STW,X1  *P3,X2
5304    05 00C0E    224FFFFF A              LI,X2   -1
5305    05 00C0F    323000E2C               LW,X1   L(X'02000202')    INITIALIZE ACCOUNT AND PASS CONTROLS
5306    05 00C10    B5380009 A              STW,X1  *T2,X2
5307    05 00C11    323000E2D               LW,X1   L(X'03010202')
5308    05 00C12    B538000A A              STW,X1  *P3,X2
5309    05 00C13    F2400005 A              LB,X2   *P1               MOVE FILE NAME TO BUFFER.
5310    05 00C14    F5400008 A              STB,X2  *T1                   P1 POINTS TO IT.
5311    05 00C15    F2380005 A              LB,X1   *P1,X2
5312    05 00C16    F5380008 A              STB,X1  *T1,X2
5313    05 00C17    64400C15                BDR,X2  $-2
5314    05 00C18    F2400008 A              LB,X2   *T1               SKIP TO ACCOUNT. BYTE COUNT FROM FPT
5315    05 00C19    20400004 A              AI,X2   4
5316    05 00C1A    2540007E A              SLS,X2  -2
5317    05 00C1B    30500004 A              AW,P1   X2                P1 NOW AT ACCOUNT
5318    05 00C1C    F2400005 A              LB,X2   *P1
5319    05 00C1D    68300000 F              BEZ     $50               NO ACCOUNT
5320    05 00C1E    F2380005 A              LB,X1   *P1,X2            MOVE ACCOUNT TO BUFFER
5321    05 00C1F    204FFFFF A              AI,X2   -1                THIS LOOP PUTS NO BYTE COUNT INTO
5322    05 00C20    F5380009 A              STB,X1  *T2,X2            FPT.
```

```
5323  05 00C21  20400000 A          AI,X2   0
5324  05 00C22  692000C1E          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  204FFFFF A          AI,X2   =1
5346  05 00C2F  F538000A A          STB,X1  *P3,X2
5347  05 00C30  20400000 A          AI,X2   0
5348  05 00C31  692000C2D          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
```

```
5359                                    PAGE
5360                                *****************************************************
5361                                *  OPEN NEW (OUTPUT ONLY) FILE                      *
5362                                *    P1 = ADDR OF FILE ID IN CDT                    *
5363                                *    CC1=1 IF FILE DOES NOT EXIST; CC1=0 OTHERWISE  *
5364                                *****************************************************
5365                                *
5366                                *
5367          05 00C36              OPENNEW   EQU     $
5368    05 00C36    02200080 A              PUSH    (X1,P3)
        05 00C37    0B30019C 02
5369    05 00C38    22600C45                LI,P2   ON$ABN
5370    05 00C39    35600353 02             STW,P2  O$FPT+2
5371    05 00C3A    22600004 A              LI,P2   4                       INOUT
5372    05 00C3B    35600356 02             STW,P2  O$FPT+5
5373    05 00C3C    2280035A 02             LI,T1   O$NAME
5374    05 00C3D    22900363 02             LI,T2   O$ACCT
5375    05 00C3E    22A00366 02             LI,P3   O$PASS
5376    05 00C3F    6A700C08                BAL,LNK OPENINIT
5377    05 00C40    04100351 02             CAL1,1  O$FPT
5378    05 00C41    02200080 A              PULL    (X1,P3)         FILE EXISTS.
        05 00C42    0A30019C 02
5379    05 00C43    02200000 A              LCI     0               NOTE.
5380    05 00C44    680E0000 A              B       0,LNK
5381                                *
5382                                *
5383                                *
5384    05 00C45                    ON$ABN  RES     0
5385    05 00C45    72300000A A             LB,X1   P3
5386    05 00C46    21300003 A              CI,X1   3
5387    05 00C47    69300B82                BNE     BADIO1
5388    05 00C48    22300002 A              LI,X1   2                       OPEN FOR OUTPUT.
5389    05 00C49    35300356 02             STW,X1  O$FPT+5
5390    05 00C4A    04100351 02             CAL1,1  O$FPT
5391    05 00C4B    02200080 A              PULL    (X1,P3)
        05 00C4C    0A30019C 02
5392    05 00C4D    02200080 A              LCI     8
```

```
05 00C4E    680E0000 A            B         0,LNK
                           *
                           *
                           ****************************************************************
                           *        VERIFY CARRIAGE RETURN EXISTS ON OUTPUT RECORD.       *
                           ****************************************************************
                           *
05 00C4F    0970019C 02   PUTCR    PUSH      LNK
05 00C50                           DO1       MODE=2
05 00C50    6A700D3B               BAL,LNK   TABCOMPRESS
05 00C51    327000AF 02            LW,LNK    CRFLAG              DO NOT INSERT CR WHEN FLAG IS
05 00C52    69300C5F               BNEZ      PUTCR2              NON-ZERO
05 00C53    327000E7 02            LW,LNK    RECSIZE
05 00C54    207FFFFF A             AI,LNK    -1
05 00C55    22F00015 A             LI,D1     X'15'
05 00C56    71FE0024 02            CB,D1     CARDIMG,LNK
05 00C57    68300C5F               BE        PUTCR2
                           *
05 00C58    20700001 A             AI,LNK    1                   IF NO CR
05 00C59    2170008C A             CI,LNK    MAXCLMN
05 00C5A    69100C5D               BL        $+3
05 00C5B    2270008B A             LI,LNK    MAXCLMN-1           (DO NOT GO BEYOND COL. 140)
05 00C5C    357000E7 02            STW,LNK   RECSIZE
05 00C5D    75FE0024 02            STB,D1    CARDIMG,LNK         INSERT ONE
05 00C5E    331000E7 02            MTW,1     RECSIZE
                           *
05 00C5F    0870019C 02   PUTCR2   PULL      LNK
05 00C60    680E0000 A             B         0,LNK
```

```
5421                                    PAGE
5422                                ***********************************************
5423                                *  READ RANDOM RECORD OR NEXT HIGHEST ONE     *
5424                                *    P1 = SEQ. NUMBER TO READ                 *
5425                                *    R1 = SEQ. NUMBER ACTUALLY READ           *
5426                                *    CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE  *
5427                                ***********************************************
5428                                *
5429                                *
5430                                        LOCAL   $20
5431          05 00C61              READNXTRANDOM   EQU $
5432   05 00C61   097C0019C 02              PUSH    LNK
5433   05 00C62   6A7C00C6D                 BAL,LNK READRANDOM
5434   05 00C63   698000000 F               BCS,8   $20
5435   05 00C64   32B000BD 02               LW,R1   LASTKEY         GET IT, RETURN KEY.
5436   05 00C65   4BB00008 02               AND,R1  XFFFFFF
5437   05 00C66   087C019C 02               PULL    LNK
5438   05 00C67   02200000 A                LCI     0
5439   05 00C68   680E0000 A                B       0,LNK
5440                                *
5441                                *
5442   05 00C69                     $20     RES     0
5443   05 00C69   6A7C00C7F                 BAL,LNK READSEQUEN      NOW GET NEXT KEY, IN R1.
5444   05 00C6A   087C019C 02               PULL    LNK
5445   05 00C6B   02200080 A                LCI     8
5446   05 00C6C   680E0000 A                B       0,LNK
```

```
                                              PAGE
                                    ********************************************
                                    *  READ RANDOM RECORD                       *
                                    *    P1 = SEQ. NUMBER TO READ               *
                                    *    CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE *
                                    ********************************************
                                    *
                                    *
             05 00C6D               READRANDOM       EQU $
 05 00C6D    02200040 A                       PUSH      (LNK,P3)
 05 00C6E    0B70019C 02
 05 00C6F    6A700B2C                         BAL,LNK   BLANKBUF
 05 00C70    6A700CE6                         BAL,LNK   SETKEY           (P1) ARE KEY,I.E. SEQUENCE
 05 00C71                                     DO1       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
```

```
5474   05 00C78   72F0000A A        LB,D1     P3
5475   05 00C79   21F00043 A        CI,D1     X'43'
5476   05 00C7A   693008B81         BNE       BADIO
5477   05 00C7B   02200040 A        PULL      (LNK,P3)
       05 00C7C   0A70019C 02
5478   05 00C7D   02200080 A        LCI       8
5479   05 00C7E   680E0000 A        B         0,LNK
```

```
5480                                        PAGE
5481                                ********************************
5482                                *  READ SEQUENTIAL RECORD      *
5483                                *    R1 = SEQ. NUMBER READ IN  *
5484                                ********************************
5485                                *
5486                                        LOCAL     $10,$20
5487                                *
5488          05 00C7F              READSEQUEN          EQU *
5489    05 00C7F    022000040 A             PUSH      (LNK,P3)
       05 00C80    05700019C 02
5490    05 00C81    6A700B2C                BAL,LNK   BLANKBUF
5491    05 00C82                            DB1       MODE=2
5492    05 00C82    041000028 03            M:SETDCB  F:EI,(ABN,RS$ABN)
       03 00028    060000000 N
                   400000000
       03 0002A    000000C90 05
5493                                        M:READ    F:EI,,
5494                                                  (WAIT),,
5495                                                  (SIZE,MAXCLMN),,
5496    05 00C83    041000029 03                      (ABN,RS$ABN)
       03 00029    100000000 N
                   500000010
       03 0002D    000000C90 05
       03 0002E    000000080 A
5497    05 00C84    6A700CEA                BAL,LNK   SETLASTKEY
5498    05 00C85    32F00005 N              LW,D1     F:EI+5              CHECK ORGANIZATION
5499    05 00C86    25F0007C A              SLS,D1    =4
5500    05 00C87    45F00003 02             AND,D1    XF
5501    05 00C88    B2B0000A N              LW,R1     *F:EI+10            RETURN SEQUENCE
5502    05 00C89    45B00008 02             AND,R1    XFFFFFF
5503    05 00C8A    21F00002 A              CI,D1     2
5504    05 00C8B    6830C8D                 BE        *+2
5505    05 00C8C    22B00000 A              LI,R1     0                   ZERO IF NOT KEYED.
5506    05 00C8D    022000040 A             PULL      (LNK,P3)
       05 00C8E    0A700019C 02
5507    05 00C8F    680E0000 A              B         0,LNK
```

```
5508                                    *
5509                                    *
5510                                    *
5511   05 00C90                         RS$ABN    RES       0
5512   05 00C90   72F00000A A                     LB,D1     P3
5513   05 00C91   21F00006 A                     CI,D1     6
5514   05 00C92   69300B81                       BNE       BADI0
5515   05 00C93   32B00E28                       LW,R1     L(E0F)
5516                                    *                                   PUT LAST SEQ # IN E0F MESG
5517   05 00C94   02200070 A                     PUSH      (P1,R1)
       05 00C95   0B50019C 02
5518   05 00C96   22500000 A                     LI,P1     0               INITIALIZE LASTKEY IN CASE SEQ
5519   05 00C97   3550008D 02                    STW,P1    LASTKEY         BEL0W TAKES ABN EXIT
5520   05 00C98   04100032 03                    M:PREC0RD F:EI,(ABN,RS$ABNABN),(REV)  P0SN BEF0RE LAST REC
       03 00032   10000000 N
                  40000010
       03 00034   00000C9B 05
5521   05 00C99   04100035 03                    M:READ    F:EI,(ERR,RS$ABNABN),(SIZE,MAXCLMN)  AND GET KEY
       03 00035   10000000 N
                  90000000
       03 00037   00000C9B 05
       03 00038   0000008C A
5522   05 00C9A   6A700CEA                       BAL,LNK   SETLASTKEY      IN C0RE L0C LASTKEY
5523        05 00C9B                    RS$ABNABN EQU       $               0R BYPASS SETTING IF TR0UBLES
5524   05 00C9B   22B00015 A                     LI,R1     21              NUMBER 0F TEXTC BYTES
5525   05 00C9C   3250008D 02                    LW,P1     LASTKEY         LAST SEQ # READ
5526   05 00C9D   4B500008 02                    AND,P1    XFFFFFF         ZAP TEXTC BYTE IN KEY
5527   05 00C9E   22600765 02                    LI,P2     BA(ERRM1)+17
5528   05 00C9F   6A700B97                       BAL,LNK   M0VESEQ
5529   05 00CA0   08000000 A                     GEN4      E0M,0,0,0
5530   05 00CA1   22500001 A                     LI,P1     1               START PAST TEXTC C0UNT
5531        05 00CA2                    RS$ABNE0M EQU       $
5532   05 00CA2   726A01D5 02                    LB,P2     ERRM1,P1        GET BYTE L00KING F0R E0M
5533   05 00CA3   21600008 A                     CI,P2     E0M
5534   05 00CA4   68300CA7                       BE        RS$ABN0UT       EXIT WHEN F0UND
5535   05 00CA5   20500001 A                     AI,P1     1
5536   05 00CA6   6B000CA2                       B         RS$ABNE0M       KEEP G0ING TILL F0UND
```

```
5537          05 00CA7            RS$ABNOUT       EQU $
5538    05 00CA7    205FFFFF A          AI,P1       =1          POINT TO BYTE BEFORE EOM
5539    05 00CA8    755001D5 02         STB,P1      ERRM1       AND USE IT AS TEXTC COUNT
5540    05 00CA9    02200070 A          PULL        (P1,R1)
        05 00CAA    0A50019C 02
5541    05 00CAB    02200040 A          PULL        (LNK,P3)
        05 00CAC    0A70019C 02
5542    05 00CAD    680E0000 A          B           0,LNK
```

```
5543                                    PAGE
5544                                ************************************
5545                                *  READ TELETYPE                     *
5546                                *    R1 = NUMBER OF CHARS READ       *
5547                                ************************************
5548                                *
5549                                *
5550          05 00CAE              READTELETYPE2     EQU $
5551    05 00CAE    022000020 A               PUSH    (X1,X2)           SAVE REGS
        05 00CAF    0B300019C 02
5552    05 00CB0    224000001 A               LI,X2   1                 USE X2=1 FOR 'READTELETYPE2'
5553    05 00CB1    68000CB5                  B       RT5
5554                                *
5555                                *
5556          05 00CB2              READTELETYPE      EQU $
5557    05 00CB2    022000020 A               PUSH    (X1,X2)           SAVE REGS
        05 00CB3    0B300019C 02
5558    05 00CB4    224000000 A               LI,X2   0                 USE X2=0 FOR 'READTELETYPE'
5559    05 00CB5    3200000A 02   RT5         LW,R0   4BLNKS
5560    05 00CB6    223000023 A               LI,X1   MAXCLMN/4
5561    05 00CB7    67080CE0                  EXU     RTSTWTBL,X2
5562    05 00CB8    643000CB7                 BDR,X1  $-1
5563          00000000                        DO      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    693000CD7                 BNEZ    RT10              FROM J:CCBUF.
5573    05 00CBB    331002C2 02               MTW,1   CFLAG
5574                                *
5575    05 00CBC    72300000 X                LB,X1   JB: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
```

```
05 00CBF    6430OCBD                    BDR,X1     $-2
                                *
05 00CC0    72000000 X                  LB,R0      J:CCBUF
05 00CC1    6708OCDE                    EXU        RTSTBTBL,X2
05 00CC2    72B00000 X                  LB,R1      JB:CCARS
05 00CC3    210000C2 A                  CI,R0      'B'             IS COMMAND 'BUILD' FROM TEL
05 00CC4    6830OCDB                    BE         RT17            YES. IT STARTS WITH 'B'.
                                *
05 00CC5    357002C1 02                 STW,LNK    BUILDFLAG
05 00CC6    6A700DC4                    BAL,LNK    TYPEMSG         TYPE
05 00CC7    000002E1 02                 DATA       UTSM1           'EDIT HERE'
05 00CC8    20BFFFFF A                  AI,R1      -1
05 00CC9    327002C1 02                 LW,LNK     BUILDFLAG
05 00CCA    72000000 X                  LB,R0      J:CCBUF
05 00CCB    210000C5 A                  CI,R0      'E'             IS THIS A FORM OF EDIT
05 00CCC    6930OCB5                    BNE        RT5             IF NOT, MUST BE A START, RUN,
                                *                                  OR GJOB SO READ FIRST COMMAND.
05 00CCD    22300004 A                  LI,X1      4               START LOOKING IN BYTE 5.
05 00CCE    22000040 A                  LI,R0      ' '
05 00CCF    3130000B A          RT8     CW,X1      R1              CHECK FOR END OF RECORD
05 00CD0    6810OCB5                    BGE        RT5             IF SO, GET NEXT COMMAND. OTHERWISE,
05 00CD1    71060176 02                 CB,R0      TTYIMG,X1       IF NON-BLANK ENCOUNTERED, ACCEPT
05 00CD2    6930OCD5                    BNE        RT9
05 00CD3    20300001 A                  AI,X1      1               INCREMENT TO NEXT BYTE.
05 00CD4    6800OCCF                    B          RT8
05 00CD5    20B00001 A          RT9     AI,R1      1               UPDATE TOTAL BYTE COUNT TO INCLUDE
05 00CD6    6800OCDB                    B          RT17            CR, THEN EXECUTE COMMAND.
                                *
05 00CD7    3238OCE2            RT10    LW,X1      RTADDTBL,X2
05 00CD8    041002DC 02                 CAL1,1     RT$FPT
                                *
05 00CD9    32800004 N          RT15    LW,R1      M:UC+4
05 00CDA    25B0006F A                  SLS,R1     -17
                                        FIN
05 00CDB    02200020 A          RT17    PULL       (X1,X2)         RESTORE REGS.
05 00CDC    0A30019C 02
05 00CDD    680E0000 A                  B          0,LNK           EXIT
```

```
5614                                    *
5615                                    *
5616          05 00CDC                  RTSTBTBL EQU     *
5617    05 00CDE    75060024 02                   STB,R0   CARDIMG,X1
5618    05 00CDF    75060176 02                   STB,R0   TTYIMG,X1
5619                                    *
5620          05 00CE0                  RTSTWTBL EQU     *
5621    05 00CE0    35060023 02                   STW,R0   CARDIMG-1,X1
5622    05 00CE1    35060175 02                   STW,R0   TTYIMG-1,X1
5623                                    *
5624          00000001                  MODE     DC       MODE=2
5625          05 00CE2                  RTADDTBL EQU      *
5626    05 00CE2    00000024 02                   DATA     CARDIMG
5627    05 00CE3    00000176 02                   DATA     TTYIMG
5628                                    FIN
```

```
5629                                    PAGE
5630                                ********************************
5631                                *  RE-OPEN LAST UPDATE FILE    *
5632                                ********************************
5633                                *
5634                                *
5635          05 00CE4              REOPEN   EQU      *
5636   05 00CE4    041000351 02              CAL1,1    OPFPT
5637   05 00CE5    680E0000 A                B         0,LNK
```

```
5638                                    PAGE
5639                                ****************************************
5640                                *  SET KEY FOR READ OR WRITE           *
5641                                *    P1 = SEQ. NUMBER TO PUT IN KEY    *
5642                                ****************************************
5643                                *
5644          05 00CE6              SETKEY  EQU     *
5645    05 00CE6    355000BA 02             STW,P1  KBUF
5646    05 00CE7    22F00003 A              LI,D1   3
5647    05 00CE8    75F000BA 02             STB,D1  KBUF
5648    05 00CE9    680E0000 A              B       0,LNK
5649                                *
5650                                *  SAVE KEY FROM LAST READ
5651                                *
5652          05 00CEA              SETLASTKEY      EQU *
5653    05 00CEA    09700019C 02            PUSH    LNK
5654    05 00CEB    B2F0000A N              LW,D1   *FIELI+10
5655    05 00CEC    35F000BD 02             STW,D1  LASTKEY
5656    05 00CED    32700004 N              LW,LNK  FIELI+4         SET RECORD SIZE RECEIVED
5657    05 00CEE    25700006F A             SLS,LNK =17
5658    05 00CEF    357000E7 02             STW,LNK RECSIZE
5659    05 00CF0    207FFFFF A              AI,LNK  =1              DELETE CR FROMIMAGE.  CHECKBOTH
5660    05 00CF1    357000B1 02             STW,LNK ENDCLMN         SAVE ENDING COLUMN
5661    05 00CF2    22F00015 A              LI,D1   X'15'           BTM
5662    05 00CF3    71FE0024 02             CB,D1   CARDIMG,LNK
5663    05 00CF4    683000CFA               BE      SETK2
5664    05 00CF5    22F0000D A              LI,D1   X'0D'           AND UTS CR'S
5665    05 00CF6    71FE0024 02             CB,D1   CARDIMG,LNK
5666    05 00CF7    683000CFA               BE      SETK2
5667    05 00CF8    6A7000A56               BAL,LNK SETEND
5668    05 00CF9    680000CFE               B       SETK6
5669    05 00CFA    22F00040 A    SETK2     LI,D1   ' '             BLANK WILL NOT INTERFERE
5670    05 00CFB    75FE0024 02             STB,D1  CARDIMG,LNK     WITH STRING EDITING.
5671    05 00CFC    357000E7 02             STW,LNK RECSIZE
5672    05 00CFD    33F000B1 02             MTW,-1  ENDCLMN         DECREMENT ENDING COLUMN
5673    05 00CFE                SETK6     DB1     MODE=2
5674    05 00CFE    6A7000D01               BAL,LNK TABEXPAND
```

```
5675   05 00CFF    087C0019C 02          PULL    LNK
5676   05 00D00    680E0000 A            B       0,LNK
```

```
                                           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.               *
                                 **************************************************
                                 *
                00000001                   DO        MODE=2
                05 00D01         TABEXPAND           EQU $
      05 00D01  330000B6 02                MTW,0     FILETYPE            IF NOT EDITING,
      05 00D02  691E0000 A                 BLZ       0,LNK               EXIT.
      05 00D03  330002C7 02                MTW,0     TABXFLAG
      05 00D04  693E0000 A                 BNEZ      0,LNK
      05 00D05  02200070 A                 PUSH      (X3,LNK)
      05 00D06  0B10019C 02
      05 00D07  321000E7 02                LW,X3     RECSIZE             SAVE RECSIZE
      05 00D08  351001A3 02                STW,X3    TRECSIZE
      05 00D09  22100000 A                 LI,X3     0                   START AT FIRST TAB IN DCB.
      05 00D0A  22200000 A                 LI,X4     0                   START AT FIRST CHAR. IN CARDIMG
      05 00D0B  352002C6 02                STW,X4    TABCFLAG            INDICATE DONT COMPRESS
      05 00D0C  22500005 A       TABX4     LI,P1     X'05'
      05 00D0D  71540024 02      TABX5     CB,P1     CARDIMG,X4
      05 00D0E  68300D15                   BE        TABX10
      05 00D0F  20200001 A                 AI,X4     1
      05 00D10  312000E7 02                CW,X4     RECSIZE
      05 00D11  69100D0D                   BL        TABX5               WHEN OUT OF CHARACTERS,
                                 *
      05 00D12  02200070 A       TABX7     PULL      (X3,LNK)            EXIT
      05 00D13  0A10019C 02
      05 00D14  680E0000 A                 B         0,LNK
                                 *
      05 00D15  7232000F N       TABX10    LB,X1     M:UC+15,X3
      05 00D16  69300D1F                   BNEZ      TABX15
                                 *
      05 00D17  20100000 A                 AI,X3     0                   IF NO MORE TABS IN DCB, WE CAN
      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  693000D12           BNEZ     TABX7
5714  05 00D1B  331002C5 02        MTW,1    TABERRFLAG
5715  05 00D1C  6A7000DC4          BAL,LNK  TYPEMSG         IN THAT CASE, ERROR.
5716  05 00D1D  0000030B 02        DATA     UTSM8
5717  05 00D1E  680000D12          B        TABX7
5718                         *
5719  05 00D1F  203FFFFF 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  692000D26          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  691000D15          BL       TABX10
5725  05 00D25  680000D12          B        TABX7
5726                         *
5727  05 00D26  22500040 A   TABX17  LI,P1     ' '          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  204FFFFF 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  683000D0C          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  203FFFFF A         AI,X1    -1
5743  05 00D35  204FFFFF A         AI,X2    -1
5744  05 00D36  31400002 A         CW,X2    X4           GO DOWN ONLY TO BYTE JUST ABOVE
5745  05 00D37  681000D31          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  68000D0C          B       TABX4           MOVED, AND LOOK FOR MORE TAB CODES.
5750                              *
5751                              *
5752                              ************************************************
5753                              *  ACCORDINGLY, EACH RECORD WRITTEN MUST BE    *
5754                              *  RE-COMPRESSED IN ORDER TO MINIMIZE RAD      *
5755                              *  STORAGE PER RECORD.                         *
5756                              ************************************************
5757                              *
5758                              *
5759        05 00D3B               TABCOMPRESS     EQU $
5760  05 00D3B  330000B6 02             MTW,0     FILETYPE
5761  05 00D3C  691E0000 A              BLZ       0,LNK
5762  05 00D3D  330002C6 02             MTW,0     TABCFLAG        IF NO COMPRESSION NEEDED, EXIT.
5763  05 00D3E  683E0000 A              BEZ       0,LNK
5764  05 00D3F  02200060 A              PUSH      (X3,P2)
      05 00D40  0B10019C 02
5765  05 00D41  22100000 A              LI,X3     0
5766  05 00D42  7232000F N              LB,X1     M:UC+15,X3
5767  05 00D43  69300D49                BNEZ      TABC13
5768  05 00D44  02200060 A  TABC5      PULL      (X3,P2)         EXIT.
      05 00D45  0A10019C 02
5769  05 00D46  680E0000 A              B         0,LNK
5770  05 00D47  7232000F N  TABC10     LB,X1     M:UC+15,X3      SKIP TO LAST TAB POSITION+1
5771  05 00D48  683000D4C               BEZ       TABC15          IN DCB.
5772  05 00D49  20100001 A  TABC13     AI,X3     1
5773  05 00D4A  21100010 A              CI,X3     16
5774  05 00D4B  691000D47               BL        TABC10
5775  05 00D4C  201FFFFF A  TABC15     AI,X3     -1              MOVE DOWN TO NEXT LOWER TAB
5776  05 00D4D  691000D6F               BLZ       TABC30          POSITION. IF ALL GONE, EXIT
5777  05 00D4E  7232000F N              LB,X1     M:UC+15,X3
5778  05 00D4F  313000E7 02             CW,X1     RECSIZE         DONT PUT ANY TAB CHARACTERS
5779  05 00D50  692000D4C               BG        TABC15          PAST END OF RECORD
5780  05 00D51  203FFFFE A              AI,X1     -2              MAKE INDEX TO NEXT LOWER BYTE.
5781  05 00D52  22500040 A              LI,P1     ' '             IS NEXT LOWER BYTE A BLANK,
5782  05 00D53  71560024 02             CB,P1     CARDIMG,X1
5783  05 00D54  693000D4C               BNE       TABC15          IF NOT, WE CAN'T COMPRESS IMAGE.
```

```
05 00D55  32200001 A             LW,X4     X3              IF BLANK, WE CAN COMPRESS DOWN
05 00D56  202FFFFF A             AI,X4     -1
05 00D57  69100D5A               BLZ       TABC17          TO NEXT LOWER TAB POSITION.
05 00D58  7224000F N             LB,X4     M:UC+15,X4
05 00D59  202FFFFE A             AI,X4     -2              TAB POSITION,
05 00D5A  32400003 A   TABC17    LW,X2     X1              CREATE NEW INDEX,
05 00D5B  71580024 02  TABC18    CB,P1     CARDIMG,X2      MOVE IT DOWN TO
05 00D5C  69300D60               BNE       TABC20          A NON-BLANK,
05 00D5D  204FFFFF A             AI,X2     -1
05 00D5E  31400002 A             CW,X2     X4              OR TAB BOUNDARY.
05 00D5F  69200D5B               BG        TABC18
                       *
05 00D60  20400001 A   TABC20    AI,X2     1               MOVE BACK UP TO BLANK.
05 00D61  20300001 A             AI,X1     1               MOVE BACK UP TO TAB COLUMN.
05 00D62  22600005 A             LI,P2     X'05'           PUT TAB CHARACTER OVER BLANK,
05 00D63  75680024 02            STB,P2    CARDIMG,X2
05 00D64  20400001 A             AI,X2     1               INCREMENT, AND CHECK IF MORE SPACE
05 00D65  31400003 A             CW,X2     X1              EXISTS BETWEEN INDICES.
05 00D66  68300D4C               BE        TABC15          IF NOT, TRY NEXT LOWER TAB.
                       *
05 00D67  72660024 02  TABC25    LB,P2     CARDIMG,X1      MOVE BYTES DOWN, STARTING AT TAB
05 00D68  75680024 02            STB,P2    CARDIMG,X2      COLUMN, AND CONTINUING UP TO END
05 00D69  20400001 A             AI,X2     1               OF RECORD.
05 00D6A  20300001 A             AI,X1     1
05 00D6B  313000E7 02            CW,X1     RECSIZE
05 00D6C  69100D67               BL        TABC25
                       *
05 00D6D  354000E7 02            STW,X2    RECSIZE         SET NEW, SMALLER RECORD SIZE,
05 00D6E  68000D4C               B         TABC15          AND GET NEXT TAB.
                       *
05 00D6F  330001A0 02  TABC30    MTW,0     RP#FLAG         PRESERVE RECORD SIZE
05 00D70  69300D44               BNEZ      TABC5           NO
05 00D71  324001A3 02            LW,X2     TRECSIZE        YES, RESET RECORD SIZE
05 00D72  354000E7 02            STW,X2    RECSIZE
05 00D73  75560024 02  TABC35    STB,P1    CARDIMG,X1      BLANK REST OF RECORD
05 00D74  20300001 A             AI,X1     1
05 00D75  31300004 A             CW,X1     X2              END OF RECORD
```

```
5821    05 00D76    69100D73    BL      TABC35      NO
5822    05 00D77    68000D44    B       TABC5       YES, RETURN
5823                            FIN
```

```
5824                                    PAGE
5825                                ************************************
5826                                *  TEST IF EDIT FILE IS ACTIVE  *
5827                                ************************************
5828                                *
5829                                *
5830          05 00D78              TESTEDITACTIVE    EQU $
5831    05 00D78    33000086 02               MTW,0     FILETYPE        TEST IF EDIT FILE ACTIVE
5832    05 00D79    691E0000 A                BLZ       0,LNK           NO - EXIT
5833    05 00D7A    09700198 02               PUSH      LNK             SAVE REG
5834    05 00D7B    6A700B33                  BAL,LNK   CLOSE           CLOSE IT
5835    05 00D7C    6A700DC4                  BAL,LNK   TYPEMSG         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    35F00086 02               STW,D1    FILETYPE
5839    05 00D80    08700198 02               PULL      LNK             RESTORE REG
5840    05 00D81    680E0000 A                B         0,LNK           EXIT
```

```
5841                                    PAGE
5842                                ***********************************
5843                                *  TYPE CARD IMAGE                *
5844                                *    P1 = SEQ. NUMBER TO TYPE     *
5845                                ***********************************
5846                                *
5847                                *
5848        05 00D82                TYPECARD EQU     $
5849  05 00D82  022000050 A                  PUSH    (X1,LNK)          SAVE REGS
      05 00D83  0B300019C 02
5850        00000000                        D0      MODE=1
5851                          *S*           LW,X2   EODCLMN           SET X2=NUMBER OF SIGNIFICANT CHARS
5852                          *S*           AI,X2   1
5853                                        FIN
5854  05 00D84  215000000 A                 CI,P1   0                 IS SEQ # < 0 (MEANING DON'T TYPE IT)
5855  05 00D85  681000D8C                   BGE     TC25
5856                                  *
5857        00000000                        D0      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 TYPEMSG           INTERSPERSE WITH CR/LF.
5868                          *S*           DATA    MSG0
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 TYPEMSG
5876  05 00D88  00000027A 02                DATA    MSG0
```

```
5877   05 00D89   022200050 A          PULL     (X1,LNK)          RESTORE REGS
       05 00D8A   0A3C0019C 02
5878   05 00D8B   680E0000 A          B        0,LNK             EXIT
5879                              *
5880                              *
5881   05 00D8C   6A700DDB      TC25  BAL,LNK  TYPESEQ           TYPE SEQ #
5882   05 00D8D   40080000 A          GEN4     BL,E8M,0,0
5883   05 00D8E                       D81      MODE=1
5884                         *S*     LI,P1    62                62 CHARACTERS ALOWWS FOR SEQ #
5885   05 00D8E   680000D86           B        TC5
```

```
5886                                    PAGE
5887                                ******************************************************
5888                                *  TYPE COMMAND OR PARAMETER ERROR                   *
5889                                *    WORD AFTER BAL = WORD ADDR OF TEXTC-STRING       *
5890                                ******************************************************
5891                                *
5892                                *
5893          05 00D8F              TYPECERR EQU     $
5894    05 00D8F    33F000B2 02             MTW,-1   ERRORCNT          IS ERROR CNT EXHAUSTED
5895    05 00D90    691E0001 A              BLZ      1,LNK             YES - SKIP PRINTING ERROR MSG
5896    05 00D91    02200030 A              PUSH     (X1,P1),LNK
        05 00D92    0B30019C 02
        05 00D93    09700198 02
5897    05 00D94    22300002 A              LI,X1    2
5898    05 00D95    224000C3 A              LI,X2    'C'
5899    05 00D96    F2E600AC 02             LB,D0    *CDTADR,X1        GET # OF CURRENT CMND IN CDT
5900    05 00D97    325E0000 A              LW,P1    0,LNK             SET P1=ADDR OF STRING
5901    05 00D98    680000DA3               B        TP10
5902                                *
5903                                *
5904          05 00D99              TYPEPERR EQU     $
5905    05 00D99    02200030 A              PUSH     (X1,P1),LNK
        05 00D9A    0B30019C 02
        05 00D9B    09700198 02
5906    05 00D9C    325E0000 A              LW,P1    0,LNK             SET P1=ADDR OF STRING
5907    05 00D9D    33F00048 02             MTW,-1   CDT               TEST IF ONLY ONE CMND IN CDT
5908    05 00D9E    692000DB2               BGZ      TP20              NO - GO FIX UP ERROR MSG
5909                                *
5910                                *  ONLY ONE COMMAND IN CDT: PRINT 'P' ERROR AS IT STANDS
5911                                *
5912    05 00D9F    224000D7 A      TP5     LI,X2    'P'
5913    05 00DA0    32E000E5 02             LW,D0    PARAMPSN          CALC POSITION OF CURRENT PARAMETER
5914    05 00DA1    20EFFFFE A              AI,D0    -2
5915    05 00DA2    25E0007F A              SLS,D0   -1
5916                                *
5917                                *  SEARCH FOR FIRST 'C(P)'
5918                                *
```

```
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          BNE       $-2
5924  05 00DA8   20300001 A        AI,X1     1               SET X1=ADDR OF CHAR AFTER 'C(P)'
5925  05 00DA9   49E00004 02       OR,D0     XF0             CONVERT COUNT TO EBCDIC (MOD 10)
5926  05 00DAA   75E60000 A        STB,D0    0,X1             AND PUT IN STRING
5927          00000000             DO        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 TYPEMSG.
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                         * THERE IS MORE THAN ONE COMMAND IN CDT: ADD 'CN' TO ERROR MSG
5946                         *
5947  05 00DB2   32F00DC3    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
```

```
5954  05 00DB9  75F8016B 02          STB,D1  TEMPBLCK,X2     !=C1!
5955  05 00DBA  204FFFFF A          AI,X2   =1
5956  05 00DBB  643000DB8           BDR,X1  $-3             LOOP
5957  05 00DBC  22300002 A          LI,X1   2
5958  05 00DBD  F2E6000AC 02        LB,D0   *CDTADR,X1      GET # OF 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 !C! TO YIELD FORM:
5962  05 00DC1  2250016B 02         LI,P1   TEMPBLCK         !=CNP1:ERROR MSG!
5963  05 00DC2  6800D9F             B       TP5             GO PROCESS !P!
5964                          *
5965                          *
5966  05 00DC3  0360C3F1 A    TPMSG TEXTC   !=C1!
```

```
5967                                    PAGE
5968                                ******************************************************
5969                                *  TYPE MESSAGE
5970                                *     WORD AFTER BAL = WORD ADDR OF TEXTC-STRING   *
5971                                ******************************************************
5972                                *
5973                                *
5974          05 00DC4              TYPEMSG  EQU      *
5975    05 00DC4    022000020 A             PUSH     (X1,X2),LNK        SAVE REGS
       05 00DC5    0B30019C 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                                *                                   NOW RUN DOWN TO FIRST NON-ZERO
5995                                *                                   CHARACTER
5996    05 00DCA    F1380007 A             CB,X1    *LNK,X2
5997    05 00DCB    69300DCD               BNE      $+2
5998    05 00DCC    644000DCA              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.
```

```
05 00DCF   6930 0DD2                  BNE       TM4
                                  *
05 00DDC   204FFFFF A                 AI,X2     -1                  EOM FOUND.
05 00DD1   223FFFFF A                 LI,X1     -1
                                  *
05 00DD2   041002D7 02        TM4     CAL1,1     TYPM$FPT
05 00DD3   20300000 A                 AI,X1     0
05 00DD4   6910 0DD7                  BLZ       TM5
                                  *
05 00DD5   6A700DC4                   BAL,LNK   TYPEMSG             YES.  CALL RECURSIVELY TO
05 00DD6   0000027A 02                DATA      MSGO                SEND IT OUT.
                                      FIN
05 00DD7   0870019C 02        TM5     PULL      (X1,X2),LNK         RESTORE REGS.
05 00DD8   02200020 A
05 00DD9   0A30019C 02
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 #  *
                          **************************************************
                          *
                          *
           05 000DB       TYPESEQ  EQU      $
05 000DB   02200040 A              PUSH     (X2,LNK)          SAVE REGS
05 000DC   0B40019C 02
05 000DD   32400007 A              LW,X2    LNK
05 000DE   22600BAC 02             LI,P2    BA(TEMPBLCK)
05 000DF   6A700B11                BAL,LNK  BINTODEC          CONVERT SEQ # TO EBCDIC! 'DDDDDDD'
05 000E0   32E0016C 02             LW,D0    TEMPBLCK+1        PUT A '.' BETWEEN 4TH AND 5TH
05 000E1   32F80000 A              LW,D1    0,X2               DIGITS AND APPEND 4 SPECIFIED
05 000E2   25E00178 A              SLD,D0   -8                 CHARS TO END
05 000E3   7250016C 02             LB,P1    TEMPBLCK+1
05 000E4   25500008 A              SLS,P1   8
05 000E5   49500002 02             OR,P1    KPE
05 000E6   35E0016C 02             STW,D0   TEMPBLCK+1        PUT THIS BACK IN TEMP BLOCK
05 000E7   5550016C 02             STH,P1   TEMPBLCK+1
05 000E8   35F0016D 02             STW,D1   TEMPBLCK+2
05 000E9   32580000 A              LW,P1    0,X2              GET 4TH SPECIFIED CHAR AND PUT
05 000EA   25500018 A              SLS,P1   24                 IN TEMP BLOCK
05 000EB   3550016E 02             STW,P1   TEMPBLCK+3
05 000EC   22E00040 A              LI,D0     ' '              ADD ZEROS BETWEEN DECIMAL
05 000ED   22F000F0 A              LI,D1     '0'              POINT AND ANY NUMBER
05 000EE   224005B2 02             LI,X2     BA(TEMPBLCK)+6
05 000EF   71E80000 A              CB,D0     0,X2             IF NECCESSARY.
05 000F0   69300DF6                BNE       TS10             ZEROS COULD ONLY BE NEEDED
05 000F1   75F80000 A              STB,D1    0,X2             IN TEMPBLCK +6 AND +7.
05 000F2   20400001 A              AI,X2     1
05 000F3   71E80000 A              CB,D0     0,X2             SEE IF SECOND ONE IS NECCESSARY
05 000F4   69300DF6                BNE       TS10
05 000F5   75F80000 A              STB,D1    0,X2
                          *
```

```
6052                                *  MAKE STRING INTO A TEXTC-STRING AND TYPE
6053                                *
6054   05 00DF6   22500000C A   TS10     LI,P1     12                ATTACH COUNT TO MAKE A TEXTC-STRING
6055   05 00DF7   75500016B 02           STB,P1    TEMPBLCK
6056   05 00DF8   6A700DC4               BAL,LNK   TYPEMSG           TYPE: ,DDD,DDDXXX, WITH LEADING
6057   05 00DF9   00000016B 02           DATA      TEMPBLCK           0'S SUPPRESSED
6058   05 00DFA   02200040 A             PULL      (X2,LNK)          RESTORE REGS
       05 00DFB   0A40019C 02
6059   05 00DFC   680E0001 A             B         1,LNK             EXIT
```

```
6060                                    PAGE
6061                                ***********************************************
6062                                *  WRITE RECORD IN COPY FILE                  *
6063                                *    P1 = SEQ. NUMBER TO WRITE                *
6064                                *    CC1=1 IF RECORD EXISTS; CC1=0 OTHERWISE  *
6065                                ***********************************************
6066                                *
6067                                *
6068          05 00DFD              WRITE2  EQU     $
6069    05 00DFD    02200040 A              PUSH    (LNK,P3)
        05 00DFE    0B70019C 02
6070    05 00DFF    6A700CE6                BAL,LNK SETKEY
6071    05 00E00    6A700C4F                BAL,LNK PUTCR
6072    05 00E01                            D01     MODE=2
6073    05 00E01    04100039 03             M:SETDCB F:E0,(ABN,W2$ABN)
        03 00039    06000000 N
                    40000000
        03 0003B    00000E07 05
6074                                        M:WRITE F:E0,;
6075                                                (WAIT),;
6076                                                (KEY,KBUF),;
6077                                                (NEWKEY),;
6078                                                (ABN,W2$ABN),;
6079    05 00E02    04100003C 03                    (SIZE,*RECSIZE)
        03 0003C    11000000 N
                    58000030
        03 0003E    00000E07 05
        03 0003F    800000E7 02
        03 00040    000000BA 02
6080    05 00E03    02200040 A              PULL    (LNK,P3)
        05 00E04    0A70019C 02
6081    05 00E05    02200000 A              LCI     0                   NON-EXISTENT
6082    05 00E06    680E0000 A              B       0,LNK
6083                                *
6084                                *
6085    05 00E07                    W2$ABN  RES     0
6086    05 00E07    72F0000A A              LB,D1   P3
```

```
6087   05 00E08    21F00016 A          CI,D1    X'16'
6088   05 00E09    693008B1            BNE      BADIB
6089                            *
6090   05 00E0A    022000040 A         PULL     (LNK,P3)
       05 00E0B    0A700019C 02
6091   05 00E0C    022000080 A         LCI      8                  RECORD EXISTED
6092   05 00E0D    680E0000 A          B        0,LNK
```

```
6093                                        PAGE
6094                                ***********************************************
6095                                *  WRITE NEW RANDOM RECORD                    *
6096                                *    P1 = SEQ. NUMBER TO WRITE                *
6097                                *    CC1=0 IF RECORD EXISTS; CC1=1 OTHERWISE  *
6098                                ***********************************************
6099                                *
6100                                *
6101          05 00E0E              WRITENEWRANDOM    EQU $
6102   05 00E0E    022000040 A              PUSH      (LNK,P3)
       05 00E0F    0B700019C 02
6103   05 00E10    6A700CE6                 BAL,LNK   SETKEY
6104   05 00E11    6A700C4F                 BAL,LNK   PUTCR
6105   05 00E12                             DB1       MODE=2
6106   05 00E12    041000041 03             M:SETDCB  F:EI,(ABN,WNR$ABN)
       03 00041    060000000 N
                   400000000
       03 00043    000000E18 05
6107                                        M:WRITE   F:EI,;
6108                                                  (WAIT),;
6109                                                  (KEY,KBUF),;
6110                                                  (NEWKEY),;
6111                                                  (SIZE,*RECSIZE),;
6112   05 00E13    041000044 03                       (ABN,WNR$ABN)
       03 00044    110000000 N
                   580000030
       03 00046    000000E18 05
       03 00047    800000E7 02
       03 00048    000000BA 02
6113   05 00E14    022000040 A              PULL      (LNK,P3)
       05 00E15    0A7000198 02
6114   05 00E16    022000000 A              LCI       0
6115   05 00E17    680E0000 A               B         0,LNK
6116                                *
6117                                *
6118                                *
6119   05 00E18                     WNR$ABN  RES       0
```

```
6120   05 00E18   72F0000A A        LB,D1    P3
6121   05 00E19   21F00016 A        CI,D1    X'16'
6122   05 00E1A   693000B81         BNE      BADIB
6123   05 00E1B   02200040 A        PULL     (LNK,P3)
       05 00E1C   0A70019C 02
6124   05 00E1D   02200080 A        LCI      8
6125   05 00E1E   680E0000 A        B        0,LNK
```

```
6126                                    PAGE
6127                                ************************************
6128                                *  WRITE RANDOM RECORD             *
6129                                *    P1 = SEQ. NUMBER TO WRITE     *
6130                                ************************************
6131                                *
6132                                *
6133          05 00E1'              WRITERANDOM       EQU $
6134    05 00E1F    09700019C 02              PUSH      LNK
6135    05 00E20    6A700CE6                  BAL,LNK   SETKEY
6136    05 00E21    6A700C4F                  BAL,LNK   PUTCR
6137                                          M:WRITF   F:EI,,
6138                                                    (WAIT),,
6139                                                    (KEY,KBUF),,
6140                                                    (NEWKEY),,
6141    05 00E22    04100049 03                         (SIZE,*RECSIZE)
        03 00049    11000000 N
                    18000050
        03 00048    80000E7 02
        03 0004C    00000BA 02
6142    05 00E23    08700019C 02              PULL      LNK
6143    05 00E24    680E0000 A                B         0,LNK
6144          05 00E2'              ENDEDITOR         EQU     $+10
6145          05 0000U                          END       BEGINEDITOR
        05 00E25    00404040 A
        05 00E26    06700002 A
        05 00E27    00200000 A
        05 00E28    00989680 A
        05 00E29    00000236 02
        05 00E2A    00002710 A
        05 00E2B    00FFFFFF A
        05 00E2C    02000202 A
        05 00E2D    03010202 A

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

* SYMBOL VALUES

```
ADDCDTPARAM/05 0081B        ADJINT/05 00287             ADJUSTALLFLAG/05 00978
ALLFLAG/02 00020            ALLBK/02 00021              ALPH/00000006
ANLZRIGHT/05 0097C          AR10/05 00986               AR10A/05 0098D
AR12/05 00993               AR15/05 00996               AR18/05 0099A
AR20/05 0099E               BADIB/05 00B81              BADIB1/05 00B82
BDISPTBL/02 00317           BD10/05 00B17               BD20/05 00B28
BD30/05 00B23               BGD10/05 00001              BINTBDEC/05 00B11
BL/TEXT                     BLANK/0000000C              BLANKBUF/05 00B2C
BLANKCNT/02 00022           BLD08/05 003A6              BLD10/05 003A8
BLD12/05 00382              BLD25/05 003C8              BLD30/05 003D2
BLD40/05 003D5              BLD5/05 003A0               BPFLAG/02 00023
BPVBFF/05 0038F             BPVBN/05 0038E              BPV10/05 0038B
BPV5/05 00386               BR$FPT/02 002CD             BRK$KEY/05 0028F
BRK30/05 002A3              BRK40/05 002AC              BRK50/05 002B8
BRK53/05 002C4              BRK55/05 002C9              BRK60/05 002CD
BRK80/05 002D2              BRK90/05 002DC              BRK91/05 002ED
BRK99/05 002FC              BUILDFLAG/02 002C1          CARDIMG/02 00024
CBRCHTBL/05 0005E           CDT/02 00048                CDTADR/02 000AC
CFLAG/02 002C2              CHARPSN/02 000AD            CHECK1CDTENTRY/05 00834
CHG$STG$CNT/02 0019F        CL8SE/05 00B33              CL8SE2/05 00B35
CL8SE3/05 00B37             CM/TEXT                     CMNDTBL/05 00351
CMT10/05 00554              CMT15/05 00568              CMT20/05 00574
CMT30/05 00579              CMT40/05 00580              CMT50/05 00583
CMT60/05 00586              CMT70/05 00589              CM10A/05 00562
CNAMETBL/05 00039           CNMRTBL/05 00084            C8M/00000007
C8PYFL/02 000AE             CPY1/05 003E4               CPY1A/05 003EB
CPY1B/05 003ED              CPY10/05 00410              CPY15/05 0041A
CPY2/05 003F2               CPY20/05 0042B              CPY3/05 00401
CPY30/05 00430              CPY32/05 0043E              CPY35/05 00466
CPY36/05 0046A              CPY37/05 0046C              CPY40/05 0046E
CPY5/05 00406               CPY5A/05 0040B              CPY50/05 00471
CPY56/05 0047C              CPY58/05 0047D              CPY60/05 0047E
CR/0000000D                 CRFLAG/02 000AF             CR3/05 00487
CR5/05 00489                CT$FLAG/02 001A1            CTBLSZ/00000025
DELETE/05 00B39             DELETEFILE/05 00B6A         DELETERECORD/05 00B95
DELNXT/02 0019B             DF$ABN/05 00B7A             DFLTINCR/02 000B0
DFLTSEQ/000003E8            DIGITS/02 0001A             DL10/05 00B3D
DL10/05 00B4D               DL17/05 00B58               DL20/05 00B5C
DL25/05 00B63               DL30/05 00B67               DMY$TPM/02 002BE
DMY$TYPECERR/02 002B6       DMY$TYPEPERR/02 002B9       DMYSTKDW/02 0000C
D0/0000000E                 D1/0000000F                 EDT10/05 004B1
EDT15/05 004B3              EDT20/05 004B6              EDT5/05 004A6
END/00000000                ENDEDITBR/05 00E2F          E8DCLMN/02 000B1
E8F/00989680                E8M/00000008                ERRC1/02 001A5
ERRC10/02 001CC             ERRC11/02 001D0             ERRC2/02 001A9
ERRC3/02 001AD              ERRC4/02 001B1
```

```
ERRC5/02 001B6          ERRC6/02 001BB          ERRC7/02 001BF          ERRC8/02 001C4
ERRC9/02 001C8          ERRM1/02 001D5          ERRM12/02 001EB         ERRM13/02 001F2
ERRM14/02 001F6         ERRM15/02 001FA         ERRM16/02 00201         ERRM17/02 00206
ERRM18/02 0020C         ERRM19/02 00214         ERRM20/02 0021D         ERRM21/02 00223
ERRM3/02 001DC          ERRM4/02 001DF          ERRM5/02 001E3          ERRM6/02 001E6
ERRM8/02 001E8          ERRORCNT/02 000B2       ERRP1/02 0022E          ERRP10/02 00250
ERRP11/02 00254         ERRP12/02 00258         ERRP13/02 0025D         ERRP14/02 00261
ERRP14A/02 00261        ERRP15/02 00265         ERRP16/02 00269         ERRP17/02 00271
ERRP18/02 00276         ERRP2/02 00232          ERRP3/02 00236          ERRP4/02 00239
ERRP5/02 0023D          ERRP6/02 00241          ERRP7/02 00245          ERRP8/02 00249
ERRP9/02 0024D          EXC10/05 0032B          EXC15/05 00334          EXC20/05 00337
EXC30/05 00344          EXC40/05 00347          EXC5/05 00325           EXC50/05 0034A
EXC55/05 00351          EXC6/05 00322           F:BLANK$PRESERV/05 0037E
F:BUILD/05 00390        F:COPY/05 003D9         F:CR/05 00481           F:DELETE/05 00497
F:EDIT/05 004A3         F:EI/EXT                F:END/05 004BA.         F:EO/EXT
F:LNK/0000000D          F:MERGE/05 004BE        F:RP/05 0048C           F:TA/05 00534
FC10/05 009C1           FC15/05 009CC           FC15A/05 009CD          FC20/05 009DD
FC30/05 009E2           FC35/05 009E4           FC40/05 009E9           FC45/05 009EC
FF/0000000C             FID1ADR/02 000B3        FID2ADR/02 000B4        FIELDCNT/02 000B5
FILETYPE/02 000B6       FINDCOLUMN/05 009A4     FINDMATCH/05 009EF
FINISH$STEP$LOOP/05 006B8                       FIRST$F:CMND/00000004
FIRST$I:CMND/0000001E                           FIRST$R:CMND/0000000B
FIRSTFROM/02 000B8      FIRSTSET/02 000B9       FM10/05 009FB           FM10A/05 009FD
FM10B/05 009FF          FM15/05 00A04           FM20/05 00A06           FM20A/05 00A0B
FM30/05 00A11           FNDTBL1/05 005D2        FNDTBL2/05 005D5        FNDTYP/05 005AE
FND20/05 005A8          FND30/05 005B2          FND32/05 005B4          FND35/05 005B8
FND40/05 005BB          FND50/05 005C1          FND60/05 005C2          FND65/05 005CA
FND65A/05 005CC         FND70/05 005CF          FRSTCLMN/02 000B7       GET$COL#$PAIR/05 001F2
GET$INCREMENT/05 00117                          GET$SEQ$INCR/05 0010E
GETFILEID/05 0083A      GETNEXT$ERROR/05 008A1                          GETNEXT$FINISH/05 008BD
GETNEXTNAME/05 00889                            GETNEXTPARAM/05 008D4
GF$PUSH$SUBR/05 0087F                           GF10/05 00854           GF15/05 0085C
GF18/05 00867           GF20/05 00874           GF30/05 00877           GF5/05 00843
GNTBL1/05 008CD         GNTBL1SZ/00000004       GNTBL2/05 008D1         GNTBL2SZ/00000008
GNTYTBL1/05 008CF       GN10/05 00895           GN25/05 008AA           GN30/05 008AC
GN35/05 008B2           GN45/05 008C5           GN50/05 008CB           GPTBL/05 0095F
GPTBLSZ/00000005        GPTYTBL/05 00961        GP10/05 008EA           GP20/05 008F4
```

```
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 00773       I:DELETE01/05 0077D     I:DELETE02/05 00776     I:FULLWSBY/05 00798
I:JUMP/05 007ED         I:LNK/0000000D          I:NWSCHANGE/05 00800
I:OVERWRSEXTEND/05 00787                        I:OVERWRITE/05 007B1
I:PRECEDESBY/05 007BC                           I:REVERSEWBPFLAG/05 00808
I:SET/05 00720          I:SHIFTSLEFT/05 007A7                           I:SHIFTSRIGHT/05 007CA
I:SUBSTITUTE/05 007D4                           I:TSSCMNDSNMR/0000002A
I:TYSCMNDSNMR/00000028                          I:TYPE/05 0080C         I:TYPESUPSSEQ/05 00810
I:TYPEX/05 00814        ICBRCHTBL/05 000EF      ICNAMETBL/05 000E3      ICNMRTBL/05 000EC
ICS10/05 000AA          ICS20/05 000B2          ICS50/05 000B6          ICS90/05 000DE
ICTBLSZ/00000008        ILGLSSEMICOLON/05 00035                         ILGLSSEQ2/05 00129
INSMSG/05 00613         INS10/05 005E9          INS20/05 005ED          INS35/05 00604
INS38/05 00600          INS40/05 0060F          INS50/05 00610          INTFLAG1/02 002C3
INTFLAG2/02 002C4       INTG/00000004           IOERRCOD/02 002A5       IOERRMSG/02 002A0
JMP10/05 007F8          JMP15/05 007FB          KBUF/02 000BA           KPE/02 00002
K1/02 00000             K10/02 00001            LASTCLMN/02 000BB       LASTFROM/02 000BC
LASTKEY/02 000BD        LASTSET/02 000BE        LCLETTERS/02 0001E      LETTERS/02 0001C
LF/00000015             LNK/00000007            LPTEXT                  LPAR/00000009
MITRTN/05 002DB         MASTEREXECUTIVE/05 00301                        MASTERPARSER/05 00004
MAXCLMN/00000086        MAXSEQ/02 000BF         MODE/00000002           MOVESEQ/05 00B97
MOVESTRING/05 00A19     MQ10/05 00BA5           MQ20/05 00BAB           MQ25/05 00BB4
MQ30/05 00BBB           MQ30A/05 00BBD          MRG10/05 004D9          MRG13/05 004F3
MRG14/05 004F4          MRG15/05 004FD          MRG17/05 00503          MRG20/05 00506
MRG25/05 00518          MRG30/05 0051C          MRG35/05 00520          MRG55/05 00522
MRG56/05 00523          MRG65/05 00526          MRG70/05 0052A          MRG80/05 0052C
MRG82/05 00530          MSG0/02 0027A           MSG1/02 0027B           MSG2/02 0027E
MSG3/02 00281           MSG4/02 00284           MSG5/02 00288           MSG6/02 0028C
MSG7/02 00293           MSG8/02 00299           MS10/05 00A26           MS20/05 00A29
MS20A/05 00A2C          MS20B/05 00A31          MS5/05 00A1F            MVD:REC:CNT/02 0019E
MVEMSG1/02 002A7        MVEMSG2/02 002AD        MVE10/05 00619          MVE20/05 0062B
MVE30/05 00646          MVE35/05 00659          MVE40/05 00661          MVE50/05 0066E
MVE53/05 00671          MVE56/05 00674          MVE58/05 00683          NAME/00000001
NCG10/05 00805          NEWCDTENTRY/05 00963                            NOCHGFLG/02 000C0
NOPROMPTSFPT/02 002CC                           OSABN/05 00BE4          OSACCT/02 00363
```

```
   @$FPT/02 00351          @$NAME/02 0035A         @$PASS/02 00366         @EX10/05 00791
   @EX20/05 00796          @N$ABN/05 00C45         @PEN/05 00BC8           @PENINIT/05 00C08
   @PENNEW/05 00C36        @PEN1/05 00BCC          @PEN2/05 00BEF          @PEN3/05 00BEB
   @2$ABN/05 00BFE         @2$ACCT/02 0037A        @2$FPT/02 00368         @2$NAME/02 00371
   @2$PASS/02 00370        PARAMBUF/02 000C1       PARAMPSN/02 000E5       PARSE:BP/05 000F7
   PARSE:BUILD/05 00103                            PARSE:CM/05 00193       PARSE:COPY/05 00131
   PARSE:CR/05 000F7       PARSE:CT/05 00192       PARSE:DE/05 001AF       PARSE:DELETE/05 00181
   PARSE:EDIT/05 00181     PARSE:END/05 0018B      PARSE:FD/05 001DC       PARSE:FS/05 001DC
   PARSE:FT/05 001DC       PARSE:I:CMND$INTG/05 00096
   PARSE:I:CMND$STRG/05 0008E                      PARSE:IN/05 0020C       PARSE:IS/05 0020C
   PARSE:JU/05 00248       PARSE:MD/05 00210       PARSE:MERGE/05 00150
   PARSE:MK/05 00210       PARSE:NO/05 0018B       PARSE:RF/05 0025C       PARSE:RN/05 00228
   PARSE:RP/05 000F7       PARSE:SE/05 001B2       PARSE:SS/05 00245       PARSE:ST/05 00245
   PARSE:TA/05 000F7       PARSE:TC/05 00262       PARSE:TS/05 00269       PARSE:TX/05 0025C
   PARSE:TY/05 00269       PBU05/05 0012C          PBU10/05 00115          PBU20/05 00121
   PBU30/05 0011C          PCM10/05 0019C          PCM20/05 001AC          PCO10/05 00149
   PCO3/05 0013F           PCO5/05 00146           PDE10/05 001BA          PDE15/05 001BB
   PDE20/05 001C5          PDE5/05 001B9           PERIOD/0000000B         PFD10/05 001E5
   PFD15/05 001E6          PFD20/05 00209          PMD10/05 00219          PMD15/05 0021A
   PMD20/05 00225          PMD25/05 00226          PME15/05 00162          PME20/05 00167
   PME30/05 0017A          PME35/05 0017B          PME40/05 0017E          PME5/05 00161
   PP10/05 00A3F           PP20/05 00A45           PP25/05 00A50           PR/TEXT
   PRMBUFS4/02 000E6       PRN10/05 00231          PRN20/05 0023D          PRN30/05 00242
   PROCESSCBL#PAIR/05 00A34                        PROMPT$FPT/02 002CA     PROMPT2$FPT/02 002CB
   PRS10/05 00033          PSS10/05 00250          PSS20/05 00256          PTY10/05 00282
   PTY15/05 00283          PTY5/05 00281           PUTCR/05 00C4F          PUTCR2/05 00C5F
   P1/00000005             P2/00000006             P3/0000000A             R:COMMENTARY/05 00545
   R:DELETE/05 0058C       R:FIND$DELETE/05 00596
   R:FIND$SEQUENCE/05 00594                        R:FIND$TYPE/05 00598
   R:INSERT/05 005DB       R:INSERT$SUP$SEQ/05 005D8                       R:LNK/0000000D
   R:MOVE$DELETE/05 00614                          R:MOVE$KEEP/05 00616
   R:RENUMBER/05 00686     R:SET$STEP/05 00699     R:SET$STEP$TYPE/05 0069B
   R:TS$CMND$NMR/00000015                          R:TY$CMND$NMR/00000016
   R:TYPE/05 006C3         R:TYPE$COMPRESSED/05 006C3                      R:TYPE$SUP$SEQ/05 006C5
   READNXTRANDOM/05 00C61                          READRANDOM/05 00C6D     READSEQUEN/05 00C7F
   READTELETYPE/05 00CB2                           READTELETYPE2/05 00CAE
   RECSIZE/02 000E7        RELATIVE/02 0037F       REOPEN/05 00CE4         REPSEQ/05 0028B
```

```
RESTART$EXECUTIVE/05 0030C          RESUME$PARSING/05 0001F
RNM10/05 00693      RNM13/05 00696  RP/TEXT             RP$FLAG/02 001A0
RPAR/0000000A       RP3/05 00492    RP5/05 00494        RR$ERR/05 00C78
RS$ABN/05 00C90     RS$ABNABN/05 00C9B  RS$ABNEOM/05 00CA2  RS$ABNOUT/05 00CA7
RT$FPT/02 002DC     RTADDTBL/05 00CE2   RTSTBTBL/05 00CDE   RTSTWTBL/05 00CE0
RT10/05 00CD7       RT15/05 00CD9   RT17/05 00CDB       RT5/05 00CB5
RT8/05 00CCF        RT9/05 00CD5    R0/00000000         R1/00000008
R2/0000000C         S/FUNC          SBS10/05 007E1      SBS15/05 007E5
SC/TEXT             SCOL/00000008   SEQ/00000002        SEQLIM/0098967F
SEQ2/00000003       SET$LOOP/05 00748   SETADR/02 000E8     SETEOD/05 00A56
SETFLAG/02 000E9    SETKEY/05 00CE6 SETK2/05 00CFA      SETK6/05 00CFE
SETLASTKEY/05 00CEA SET10/05 00746  SHIFTLEFT/05 00A70  SHIFTRIGHT/05 00A99
SL10/05 00A87       SL20/05 00A8E   SL3/05 00A76        SL30/05 00A93
SL5/05 00A80        SL5A/05 00A82   SPL10/05 006B3      SPL15/05 006BD
SPL20/05 006CC      SRS10/05 00A67  SRS15/05 00A6A      SRS5/05 00A60
SR10/05 00ABE       SR12/05 00AC1   SR12A/05 00AC3      SR15/05 00AC9
SR15A/05 00ACE      SR20/05 00AD5   SR20A/05 00AD6      SR5/05 00AA3
SR50/05 00ADD       SR52/05 00AE4   SR52A/05 00AE6      SR55/05 00AEE
SR58/05 00AF9       SR60/05 00AFF   SR70/05 00B03       SR72/05 00B0D
SR8/05 00AB3        SR8A/05 00AB5   STACK/02 000EA      STACKDW/02 0019C
STACKSZ/0000007D    STEP$LOOP/05 006AD  STEPFLAG/02 00167   STL10/05 00760
STL20/05 00770      STL30/05 0075A  STL5/05 0074F       STOPCLMN/02 00168
STOPLASTCMD/05 002F7                STP10/05 006A9      STRG/00000005
SVBPFLAG/02 0016A   SV1STSET/02 00169   TABCFLAG/02 002C6   TABCOMPRESS/05 00D3B
TABC10/05 00D47     TABC13/05 00D49 TABC15/05 00D4C     TABC17/05 00D5A
TABC18/05 00D5B     TABC20/05 00D60 TABC25/05 00D67     TABC30/05 00D6F
TABC35/05 00D73     TABC5/05 00D44  TABERRFLAG/02 002C5  TABEXPAND/05 00D01
TABSET/05 00541     TABXFLAG/02 002C7   TABX10/05 00D15   TABX12/05 00D19
TABX15/05 00D1F     TABX17/05 00D26 TABX19/05 00D31     TABX4/05 00D0C
TABX5/05 00D0D      TABX7/05 00D12  TA5/05 0053F        TC15/05 00D87
TC25/05 00D8C       TC5/05 00D86    TEMPBLCK/02 0016B   TESTEDITACTIVE/05 00D78
TEXTCADR/02 00175   TM4/05 00DD2    TM5/05 00DD7        TPC$FPT/02 002D2
TPMSG/05 00DC3      TP10/05 00DA3   TP20/05 00DB2       TP5/05 00D9F
TRECSIZE/02 001A3   TSADDR/02 002C8 TS10/05 00DF6       TTYIMG/02 00176
TTYIMGSZ/02 0019A   TXFLAG/02 001A2   TYPE$ALPHA/05 000BD   TYPE$BETA/05 000C6
TYPE$I:CMND$D/05 000CF              TYPE$I:CMND$S/05 000D1
TYPECARD/05 00D82   TYPECERR/05 00D8F  TYPEMSG/05 00DC4    TYPEPERR/05 00D99
```

```
      TYPESEQ/05 000DB        TYPM$FPT/02 002D7       TYP10/05 006DC          TYP15/05 006E9
      TYP17/05 006F1          TYP20/05 006F4          TYP25/05 006F9          TYP25A/05 006FB
      TYP40/05 006FE          TYP42/05 00701          TYP45/05 00708          TYP5/05 006D9
      TYP50/05 0070A          TYP55/05 0070F          TYP60/05 00713          TYP65/05 00717
      TYP70/05 00718          TYP72/05 0071B          TYP75/05 0071D          TYP80/05 00725
      TYP82/05 00726          TYP90/05 0072C          T1/00000008             T2/00000009
      UTSM1/02 002E1          UTSM2/02 002E4          UTSM3/02 002E8          UTSM4/02 002E9
      UTSM5/02 002F6          UTSM6/02 00303          UTSM7/02 00307          UTSM8/02 0030B
      WNR$ABN/05 00E18        WRITENEWRANDBM/05 00E0E                         WRITERANDBM/05 00E1F
      WRITE2/05 00DFD         W2$ABN/05 00E07         X:C/02 00019            X:F/02 00016
      X:INTB/02 00014         X:M/02 00017            X:BN/02 00012           X:BVER/02 00013
      X:S/02 00018            X:TB/02 00015           XEQFLAG/02 002C9        XF/02 00003
      XFFFF/02 00006          XFFFFFF/02 00008        XFF00/02 00005          XF0/02 00004
      X1/00000003             X1FFFF/02 00007         X2/00000004             X3/00000001
      X4/00000002             X800000/02 00009        ZERB:STG:FLG/02 001A4
      4BLNKS/02 0000A
*     EXTERNAL DEFINITIONS
      BEGINEDITBR/05 00000                            EDITBASE/02 00000       PATCH/02 0031F
      SECT1/02 0000C          SECT5/05 00000
*     PRIMARY REFERENCES
      J:CCBUF        JB:CCARS       M:EI          M:EB          M:UC
* NB SECONDARY REFERENCES
* NB UNDEFINED SYMBOLS
*     ERROR SEVERITY LEVEL: 0
* NB ERROR LINES
```

XEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROXEROX

```
20:52 SEP 08,'75 ID=0116
ELAPSED JOB TIME                         :12
PARTITION NUMBER                          14
TOTAL CPU TIME                        8.0098
  PROCESSOR EXECUTION TIME            7.2411
  PROCESSOR SERVICE TIME               .7687
CARDS:  CARDS READ                        19
PAGES:  PROCESSOR PAGES                  446
CORE:   PEAK CORE(PAGES)                  32
        PAGE.MINUTES                278.3007
I/O:    OPERATIONS                     13239
        CALS                           30990
FILE SPACE
  PEAK RAD TEMPORARY                     296
  AVLBL DISK PERMANENT                   575
RESOURCES ALLOCATED
  CO= 32(PAGES)
```