

## General Disclaimer

### One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.
- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.
- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.
- This document is paginated as submitted by the original source.
- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.



78-29791  
CR151810

LOCKHEED ELECTRONICS COMPANY  
JUL 19 1978

SYSTEMS AND SERVICES DIVISION

16811 EL CAMINO REAL □ HOUSTON, TEXAS 77058 □ TELEPHONE (AREA CODE 713) 488-0080

Ref: 642-6920  
Contract NAS 9-15200  
Job Order 73-783-18

TECHNICAL MEMORANDUM

MODIFICATIONS TO THE ACCURACY ASSESSMENT ANALYSIS  
ROUTINE SPATL TO PRODUCE AN OUTPUT FILE

By

J. G. Carnes

Approved By:

*Elmer M. Hsu*  
E. M. Hsu, Supervisor  
Accuracy Assessment  
Section



(NASA-CR-151810) MODIFICATIONS TO THE  
ACCURACY ASSESSMENT ANALYSIS ROUTINE SPATL  
TO PRODUCE AN OUTPUT FILE (Lockheed  
Electronics Co.) 36 p HC A03/ME A01

N78-29791

CSCI 09B G3/E1

Unclass  
28481

June 1978

LEC-12175

## 1. BACKGROUND

SPATL is an analysis program in the Accuracy Assessment Software System which makes comparisons between ground truth information and dot labeling for an individual segment. (For information on this program see "As-Built Design Specification for PDP 11/45 Accuracy Assessment System Using Disk Data File," Job Order 71-695, TIRF 77-0048, LEC-11881.) Some of the information produced by this program is of interest for areas greater than a single segment. In order to facilitate the aggregation of this information, SPATL was modified to produce a disk output file containing the necessary information about each segment.

## 2. GENERAL DESCRIPTION

The modifications to SPATL that produce the disk output file do not change the operation of the program. The data needed for the output file are stored in arrays and then written to a disk file when processing is complete. The line printer output from the program was modified to include the contents of the output file.

The output file is named SPATL.OUT. It is created in the UIC where the task is located. The output file is an unformatted, sequential access file. The contents of the file are described in appendix A. Records 2, 3, 4, and 5 are two-dimensional arrays. The second subscript in records 2 and 3 is a running index for the transformations. In record 4, the second subscript is a running index for crop codes. The second subscript in record 5 is a running index for mixed pixel AI dots.

To access the output data file, the following statement should be used:

```
OPEN(UNIT=LUN,NAME=FLNM,TYPE='OLD',ACCESS='SEQUENTIAL',  
* FORM='UNFORMATTED',CARRIAGE CONTROL='NONE',ERR=N)
```

LUN is the logical unit number associated with the file, FLNM is an array containing the file name for the output file with an explicit version number, and N is a statement number to go to if there is an error in opening the file. The arrays to accept the data should be dimensioned as follows:

```
DIMENSION IREC1(16),IREC2(3,100),IREC3(2,10),IREC4(24,100),  
* IREC5(3,100)
```

To load the data into the arrays the following statements can be used:

```
READ(LUN)(IREC1(I),I=1,16)  
READ(LUN)((IREC2(I,J),I=1,3),J=1,IREC1(13))  
READ(LUN)((IREC3(I,J),I=1,2),J=1,IREC1(14))  
READ(LUN)((IREC4(I,J),I=1,IREC1(15)+14),J=1,IREC1(12))  
READ(LUN)((IREC5(I,J),I=1,3),J=1,IREC1(11))
```

In record 1, all of the words except word 2 are integers. Word 2 is an alphanumeric variable (A2 format). Record 2 contains only integers. In record 3, the first word in each pair is alphanumeric (A1 format). In records 4 and 5, the words are all integers. To calculate the number of subpixels with a crop code from the data in record 4, the following statement should be used:

$$\text{NSBPX}(J)=\text{IREC4}(2,J)*32768+\text{IREC4}(3,J)$$

where NSBPX(J) must be an INTEGER\*4 variable.

### 3. DETAILS OF THE MODIFICATIONS

Appendix B is a compiled listing of the routines used in the modified version of SPATL. The modifications made to the original program are marked off in blocks on the listings. SPATL uses the following routines: SPATL (main), CROPL, FILNAM, SUBSTR, TAB, and ZAP. The routines which were modified to produce the output file are SPATL, TAB, and ZAP. The modified version of SPATL requires 32,370 words of storage.

#### 3.1 SPATL (MAIN) MODIFICATIONS

The arrays used to store the data are called IREC1, IREC2, IREC3, IREC4, and IREC5. All except IREC4 are dimensioned only in SPATL. IREC4 is in common block BK1. It is in common because IREC4 is loaded in the subroutine TAB. The logical unit number (LUN) for the output file is 7. The data in IREC1 concerning the AI dots is obtained from the character data in the first AI dot record.

#### 3.2 TAB MODIFICATIONS

The calling argument FLG was given a third value (3) to correspond to the AI dots. An additional argument, LCD, is used to store the number of ground truth crop codes in the segment.

#### 3.3 ZAP MODIFICATIONS

The only modification to ZAP was the addition of IREC4 to the common statement.

APPENDIX A  
DISK OUTPUT FILE FORMAT FOR SPATL

APPENDIX A  
DISK OUTPUT FILE FORMAT FOR SPATL

Record 1 -- General information

WD1 -- Segment number  
WD2 -- State  
WD3 -- Processing date for analyst labeled dots  
WD4 -- Base acquisition date (1)  
WD5 -- Previous acquisition date (2)  
WD6 -- Previous acquisition date (3)  
WD7 -- Previous acquisition date (4)  
WD8 -- Date for ground truth tape (GTT)  
WD9 -- Dot type  
WD10 -- Number of labeled dots  
WD11 -- Number of mixed pixel labeled dots  
WD12 -- Number of crop codes on ground truth tape  
WD13 -- Number of crop code transformations  
WD14 -- Number of dot label transformations  
WD15 -- Largest AI dot output code  
WD16 -- Not used

Record 2 -- Crop code transformations

WD(1,N) -- Beginning crop code  
WD(2,N) -- Ending crop code  
WD(3,N) -- Transformation

Record 3 -- Dot label transformations

WD(1,N) -- Alphanumeric label designation  
WD(2,N) -- Numeric code

Record 4 -- Totals for each crop code

WD(1,N) -- Crop code (CC)  
WD(2,N) -- High order bits } Number of subpixels  
WD(3,N) -- Low order bits } with crop code



WD(4,N) – Number of pixels with one subpixel of CC on GTT  
 WD(5,N) – Number of pixels with two subpixels of CC on GTT  
 WD(6,N) – Number of pixels with three subpixels of CC on GTT  
 WD(7,N) – Number of pixels with four subpixels of CC on GTT  
 WD(8,N) – Number of pixels with five subpixels of CC on GTT  
 WD(9,N) – Number of pixels with six subpixels of CC on GTT  
 WD(10,N) – Number of 209 dots with CC on GTT  
 WD(11,N) – Number of 209 dots with three subpixels of CC on GTT  
 WD(12,N) – Number of 209 dots with four subpixels of CC on GTT  
 WD(13,N) – Number of 209 dots with five subpixels of CC on GTT  
 WD(14,N) – Number of 209 dots with six subpixels of CC on GTT  
 WD(15,N) – Number of dots labeled "1" by analyst with CC  
 WD(16,N) – Number of dots labeled "2" by analyst with CC  
 ⋮  
 WD(4+n,N) – Number of dots labeled "n" by analyst with CC

Record 5 – Labeling accuracy for each mixed pixel labeled dot

WD(1,N) – Crop code for mixed pixel labeled dot  
 WD(2,N) – Number of subpixels in labeled dot with crop code  
 WD(3,N) – Analyst label for dot

APPENDIX B  
PROGRAM LISTINGS

0001 PROGRAM SPATL

C MODIFIED TO PRODUCE DISK FILE OUTPUT

0002 IMPLICIT INTEGER (A=0), (S=2)

0003 BYTE (M=3000), (I=8), (J=8), (K=8)

0004 PARAMETER SIZE=64, MAXDSK=3

0005 BYTE CHAR(50)

C COMMON /BX1/RA(256,6), MC2RE(256)

0006 COMMON /BX1/RA(256,6), MC0DE(256), REC(4(24,100))

0007 DIMENSION GT(11,19), KT(26), AI(11,1), JT(256),

• REA(256,10), MI(6)

0008 DIMENSION CP(11,19)

0009 DIMENSION LC0DE(10)

0010 DIMENSION GTFIL(7), FILGT(3), AIFIL(7), AIFIL(13)

0011 DIMENSION SIZE, SIZE, MAXDSK

0012 DIMENSION IREC(116), IREC(3,100), IREC(2,10), IREC(5(3,100),

• MONTH(12)

C IREC1 = DISK OUTPUT (GENERAL INFORMATION)

C IREC2 = DISK OUTPUT (CRP CODE TRANSFORMATIONS)

C IREC3 = DISK OUTPUT (DOT LABEL TRANSFORMATIONS)

C IREC4 = DISK OUTPUT (TOTALS FOR EACH CRP CODE 0= GTT)

C IREC5 = DISK OUTPUT (LABELING ACCURACY FOR MIXED PIXEL

C Labeled DPTS)

C MONTH = NUMBER OF DAYS IN YEAR BEFORE FIRST DAY OF MONTH

0013 EQUIVALENCE (S,RUEIAZ1)

0014 DATA FILGT/08,'21','11','31','XX','3X','XX','XX','XX','XX',

• 'XX','XX',C

0015 DATA MONTH/0,31,59,90,120,151,181,212,243,273,304,334/

0016 CALL TIME(2)

0017 CALL DATE(0)

0018 NR004

NR003

NR002A

0021 NR007

C \*OUT = LOG FOR DISK OUTPUT

0022 WRITE(NPRT,703) N,T

0023 \* REVEUNIT=NRDD, NAME=SPATL, DAT, TYPE=OLD,

• ACCESS=SEQUENTIAL, FJRM=FORMATTED, CARRIAGE CONTROL=NONE)

0024 703 FORMATTING: JOB INITIATED ON 'SAL' AT '081'//11

1 PROGRAM ADSPATL,FT)

C \* \* \* \* \* READ CONTROL CARD WITH FILE NAMES \* \* \* \* \*

C \* \* \* \* \* AT 703S AND 704S \* \* \* \* \*

0025 READ(NR00,704) AIFIL,GTFIL

0026 F004AY (2(6A2,4(1X))

0027 WRITE (NPRT,705) (AIFIL(I),1(7)), (GTFIL(I),1(1,7))

0028 F004AY(140,10X,'AI DPTS FILE NAME = '7(A2)

• //,10X,'GROUND TRUTH FILE NAME = '7(A2))

ORIGINAL PAGE 15  
OF POOR QUALITY

```

0028 CALL FILNAM(GFIL,FILGT)
0029 CALL SUBSTP (FILGT,1,26,FILAI,1,26)
0030 CALL FILNAM(AFIL,FILAI)
0031 PRINT (NPR,70A) (FILAI(1),I=1,12)
0032 FORMAT (1H0,10X,A1,12D)
0033 WRITE(NPRT,710)(FILGT(I),I=1,12)
0034 FORMAT (//,10X,1GT FILE NAME = ,1,12(A2))

```

```

C *** OPEN FILES ***
0035 OPEN(UNIT=1,NAME=FILEGT,TYPE='OLD',ACCESS='SEQUENTIAL',
      * CARRIAGE CONTROL='NONE',FORM='UNFORMATTED')
0036 OPEN(UNIT=NRDR,NAME=FILEAI,TYPE='OLD',FORM='FORMATTED')
0037 Z09 WRITE (NPR,709)
      * FORMAT(//,10X,'FILES OPEN'

```

```

0038 READ (1) BUF
0039 DO 20 M=1,256
0040 JT(I)=0
0041 CONTINUE
0042 WRITE (NPR,304) S,(BUF(I),I=1,163)
0043 304 FORMAT(' SITE= ',115,5X,' DATE=',15,5X,' TIME=',15,5X,' YEAR=',15)

```

```

C IREC(1) IS ENTER SITE NUMBER
0044 IREC(1)=S
0045 CONVERT DATE ON GYT FROM DD/MM/YY TO JULIAN DATE
      * YY=BUF(13)=70
0046 IF(YY.GP.10)YY=YY+10
0047 IREC(1)=1000*YY+MONTH(BUF(62))+BLF(61)
0048 A=BUF(63)
0049 IF(MOD(A,4).EQ.0.A.D.BUF(62).GT.2)IREC(1)=IREC(1)+1

```

```

0050 CALL ZAP
0051 WRITE (NPR,905)
0052 905 FORMAT(1H0,10X,'CODE TO CODE TRANSFORMATION',//,9X,'BEGIN',7X,
      * 1'END',7X,'CODE')

```

```

C NCCT=0 NCCT = NUMBER OF CRP CODE TRANSFORMATIONS
0053 NCCT=0
0054 121 CONTINUE
0055 READ(NRD,110) NR,NE,N0
0056 118 FORMAT(15)
0057 WRITE(NPRT,117) NR,NE,N0
0058 117 FORMAT(1H ,310)

```

```

C LEAD CRP CPDE TRANSFERPATIONS INTO IREC2
0059 NCCT=NCCT+1
0060 IREC2(1,NCCT)=NB
0061 IREC2(2,NCCT)=NE
0062 IREC2(3,NCCT)=N0
0063 IF((NR.EQ.0).AND.(NE.EQ.0).AND.(N0.EQ.0)) GO TO 122
0064 IF((NR.FO.0).AND.(NE.FO.0).AND.(N0.FO.0)) GO TO 224
0065 DO 119 N=NR,NE
0066 JT(N)=N0
0067 119 CONTINUE
0068 GO TO 121

```



ADSPATL.FIN /TRISBLOCKS/WR

```

0122      SA(L,N) = A(L,N)
0123      A CONTINUE
0124      N = N + 1
0125      IF ((L10.NE.0).OR.(S10.NE.0)) GO TO 8
0126      GP(L/10,S/10) = CRP * 10 + A
0127      IF (N.NE.6) CRP = CRP * 1000
0128      G1(L/10,S/10) = CRP
0129      * CONTINUE
0130      3 CONTINUE
0131      1 CONTINUE
0132      WRITE(NPRT,850)
0133      850 FORMAT(/,10X,'GROUND TRUTH INFERRATION FOR THE WHOLE SEGMENT')
0134      CHECK = 17.0 + 196.0 * A * D
0135      *****CHECK
0136      FLG = 1
0137      CALL TAB(NCODE,RPURE,RTT,FLG)
0138      CALL TAB(NCODE,RPURE,RTT,FLG,LCU)
0139      LCU = NCODE
0140      LCU = NUMBER OF CROP CEROES IN GTT
0141      IF (RTT.NE.CHECK) WRITE(NPRT,223)
0142      223 FORMAT(/,10X,'THE WHOLE SEGMENT WAS NOT GROUND TRUTHED')
0143      WRITE(NPRT,222) RT
0144      222 FORMAT(/,10X,'COMPUTATIONS BASED ON ',F10.2,' SUBPIXELS',/)
0145      10P FORMAT(' PERCENTAGE OF SCENE IN FIRE PIXELS',F10.2)
0146      WRITE(NPRT,903)
0147      903 FORMAT(/,10X,'A MATRIX OF GROUND TRUTH DOT LABELS')
0148      *****WRITE(NPRT,103) (GT(L,S),S=1,19),L=1,11)
0149      103 FORMAT(1H ,191A)
0150      CALL ZAP
0151      DO 200 I=1,11
0152      DO 200 J=1,19
0153      LOGP(I,J) = 10
0154      SGGP(I,J) = 10
0155      PA(L,S) = RA(L,S) * 1
0156      200 CONTINUE
0157      WRITE(NPRT,851)
0158      851 FORMAT(/,10X,'GROUND TRUTH INFERRATION FOR THE 209 DOTS')
0159      RTT = 209
0160      FLG = 2
0161      C CALL TAB(NCODE,RPURE,RTT,FLG)
0162      CALL TAB(NCODE,RPURE,RTT,FLG,LCU)
0163      WRITE(NPRT,255) RPURE
0164      255 FORMAT(/,10X,'PERCENTAGE OF THE 209 DOTS WHICH ARE PURE PIXELS',
0165      F10.2)
0166      DO 16 KL=1,11
0167      DO 17 KP=1,19
0168      AL(KL,KP) = 0
0169      17 CONTINUE
0170      1A CONTINUE
0171      DO 14 I=1,26

```

ORIGINAL PAGE IS  
OF POOR QUALITY

```

0170 K1(1)=0
0171 14 CONTINUE
0172 NCAI=0
0173 D0 666 I=1,10
0174 666 LCODE(I)=0
0175 WRITE(NPRT,702)
0176 702 FORMAT(//10X,'TYPE TO CODE TRANSFORMATIONS')
0177 WRITE(NPRT,300)
0178 300 FORMAT(//3X,'TYPE',6Y,'CODE')
0179 NAI=0

```

NCCT=0  
 NCCT = NUMBER OF DOT LABEL TRANSFORMATIONS

```

0180
0181 1X CONTINUE
0182 READ(NRDB,508) KS,NT
0183 508 FORMAT(1A1,4Y,115)
0184 WRITE(NPRT,105) KS,NT
0185 IF(NT.GT.10) WRITE(NPRT,158)
0186 158 FORMAT (1H0,10X,'THERE ARE TOO MANY A I DOT TYPES')
0187 IF(NT.GT.10) STOP
0188 10X FORMAT(1H , 5X,A1,110)

```

LOAD DOT LABEL TRANSFORMATIONS INTO IREC3

```

0189 NCCT=NCCT+1
0190 IREC3(1,NCCT)=KS
0191 IREC3(2,NCCT)=NT
0192 IF(NT.EQ.0) GO TO 400
0193 IF(KS.EQ.88) GO TO 12
0194 K1(KS)=NAI+NT
0195 LCODE(NT)=NT
0196 GO TO 13
0197 12 CONTINUE

```

IREC1(14)=NCCT  
 IREC1(15) = LARGEST AI DOT OUTPUT CODE

```

0198
0199 A=IREC3(2,1)
0200 D0 31 I=1,NCCT
0201 31 IF(IREC3(2,I).GT.A) A=IREC3(2,I)
0202 IREC1(15)=A
0203 CALL CLUSE(NRDB)
0204 D0 777 I=1,10
0205 777 IF(LCODE(I).GT.0) NCAI=NCAI+1
0206 WRITE(NPRT,778)
0207 778 FORMAT (1H0,10X,'THE A I DOTS',//)

```

FLAG TO PICK UP ONLY CHARACTER DATA ON FIRST CARD

```

0208
0209 CONTINUE
0210 READ(NRDB,106) KL,KP,KS,(CHAR(I),I=1,50)
0211 104 FORMAT(10X,11Z,1X,11Z,1X,1A1,13X,50A1)
0212 WRITE(NPRT,162) KL,KP,KS,(CHAR(I),I=1,50)
0213 162 FORMAT(1H ,10X,1Z,1X,1Z,1X,A1,13X,50A1)

```

C LOAD SITE DATA FROM FIRST DOT CARD INTO IREC1

```

0214 IF(Z,G,T,0) G= T0 607
0215 DECIDE(4,601,CHAR(5))IREC1(3)
0216 601 FORMAT(14)
0217 DECIDE(4,601,CHAR(10))IREC1(4)
0218 DECIDE(4,601,CHAR(15))IREC1(5)
0219 DECIDE(4,601,CHAR(20))IREC1(6)
0220 DECIDE(4,601,CHAR(25))IREC1(7)
0221 DECIDE(1,606,CHAR(44))IREC1(9)
0222 606 FORMAT(11)
0223 A=CHAR(2)
0224 B=CHAR(3)
0225 2=1
0226 2=1
0227 607 CONTINUE
    
```

```

0228 IF(KL,EO,0) G0 T0 15
0229 NAID=NAID+1
0230 NT=NY(KS=64)
0231 IF(AI(KL, KP),NE,0) WRITE(NPRT,313) KL,KP
0232 313 FORMAT(//,10X,'DUPLICATE DOT LABEL FOR DOT = ',2I5,///)
0233 AI(KL,KP)ENT
0234 GP T0 212
0235 15 CONTINUE
0236 WRITE(NPRT,502) NAID
0237 502 FORMAT(//,10X,'NO. OF AI DOTS=',15)
0238 WRITE(NPRT,902)
    
```

```

0239 IREC1(10)=NAID
0240 902 FORMAT(//,10X,'A MATRIX OF AI DOTS LABELS')
0241 WRITE(NPRT,103) ((AI(L,S),S=1,15),L=1,11)
0242 CALL ZAP
0243 D0 201 I=1,11
0244 D0 201 J=1,19
0245 IF(AI(L,J),EQ,0) G= T0 201
0246 L=GP(I,J)/10
0247 S=GP(I,J)-L*10
0248 RA(L,S)=RA(L,S)+1
0249 201 CONTINUE
0250 WRITE(NPRT,852)
0251 852 FORMAT(//,10X,'GROUND TRUTH INFORMATION FOR THE AI DOTS')
0252 RTT=NAID
    
```

```

0253 FLG=3
0254 CALL TAB(NDUMY,RPURE,RTT,FLG,LCD)
0255 CALL TAB(NDUMY,RPURE,RTT,FLG)
0256 WRITE(NPRT,256) RPURE
0257 256 FORMAT(//,10X,'PERCENTAGE OF THE AI DOTS WHICH ARE PURE PIXELS',
1F10.2)
0258 IF(FLG,NE,=1) WRITE(NPRT,901)
0259 901 FORMAT(//,10X,'MISAPPEL DOTS')
0260 107 FORMAT(1 LINE SAMPLE TRUE ANALYST1)
    
```



```
0261 ERRPR=0  
0262 NLABEL=0  
0263 DO 10 KL=1,11  
0264 DO 10 KP=1,19  
0265 A=AI(KL,KP)  
0266 IF (A=EQ,0) GO TO 19  
0267 NLABEL=NLABEL+1  
0268 GG=GT(KL,KP)  
0269 IF (GG=SE,1000) GG=GG-1000  
0270 PEA(GG,AA)=PEA(GG,AA)+1,0  
0271 IF (GG=EQ,AA) GO TO 22  
0272 ERRPR=ERRPR+1  
0273 WRITE(NPRT,110) KL,KP,GG,AA  
0274 10 FORMAT(1H ,4110)  
0275 20 CONTINUE  
0276 19 CONTINUE  
0277 18 CONTINUE
```

C LOAD DOT LABELING INFORMATION INTO IRECA

```
0278 DO 30 I=1,LCD  
0279 DO 30 J=1,IREC1(15)  
0280 30 IREC4((J+14),I)=PEA(IREC4(1,I),J)  
ERRPR=ERRPR  
PNLABEL=NLABEL  
RPMC=100.0*ERRPR/PNLABEL  
IF (A=NE,-1) WRITE (NPRT,109) RPMC  
100 FORMAT(1, PROPORTION OF DOTS MISLABELED=,1F6.2)  
WRITE(NPRT,951)  
951 FORMAT(/,10X,'COMPARISON OF THE GROUND TRUTH AND THE AI DOTS')  
DO 800 L=1,11  
WRITE(NPRT,103) (GP(L,S),S=1,19)  
WRITE(NPRT,801) (AI(L,S),S=1,15)  
801 FORMAT(1H ,19(15.1V),/)  
800 CONTINUE  
WRITE(NPRT,301)  
301 FORMAT(/,10X,'MIXED PIXELS')  
WRITE(NPRT,107)  
800 CONTINUE
```

C NMIX=0 NMIX = NUMBER OF MIXED PIXEL LABELED DOTS

```
0286 NMIX=0  
0287 DO 302 I=1,11  
0288 DO 302 J=1,19  
0289 IF (AI(I,J)=EQ,0) GO TO 302  
0290 N=GP(I,J)-GP(I,J)/10*10  
0291 IF (N=EQ,6) GO TO 302  
0292 WRITE(NPRT,110) I,J,GP(I,J),AI(I,J)  
0293 NMIX=NMIX+1  
0294 302 CONTINUE
```

C LOAD INFORMATION ABOUT MIXED PIXEL DOTS INTO IRECS

```
0295 NMIX=NMIX+1  
0296 IRECS(1,NMIX)=GP(I,J)/10  
0297 IRECS(2,NMIX)=N  
0298 IRECS(3,NMIX)=AI(I,J)  
0299 302 CONTINUE
```

C 302 CONTINUE

ORIGINAL PAGE IS  
OF POOR QUALITY



```

0308 TREC(11)=MIX
0309 D0 125 I=1,256
0310 D0 126 J=1,10
0311 REA(I,J)=100.0*REA(I,J)/RNLABL
0312 12A CONTINUE
0313 125 CONTINUE
0314 WRITE(NPRT,960)
0315 960 FORMAT(/,10X,'THE CONFUSION MATRIX')
0316 WRITE(NPRT,965)
0317 965 FORMAT(/,10X,'THE FIRST INDEX IS THE GROUND TRUTH CODE',
1/,10X,'THE SECOND INDEX IS THE AI CODE')
0318 D0 910 I=1,NCODE
0319 K=MCODE(I)
0320 WRITE(NPRT,911) (MCODE(I),LCODE(J),REA(K,J),J=1,NCAL)
0321 911 FORMAT(/,15( 5X,'P(',I3,',',I3,',')=',F6.2))
0322 910 CONTINUE
0323 WRITE(NPRT,999)
0324 999 FORMAT(//)
0325 CALL ZAP
0326 D0 998 I=1,NCODE
0327 D0 966 J=1,NCAL
0328 K=MCODE(I)
0329 RA(K,1)=RA(K,1)+REA(K,J)
0330 966 CONTINUE
0331 WRITE(NPRT,967) MCODE(I),RA(K,1)
0332 967 FORMAT(/,10X,'P(',I3,',',I3,',')= ',F6.2)
0333 968 CONTINUE
0334 WRITE(NPRT,999)
0335 D0 968 J=1,NCAL
0336 D0 969 I=1,NCODE
0337 K=MCODE(I)
0338 RA(J,2)=RA(J,2)+REA(K,J)
0339 969 CONTINUE
0340 WRITE(NPRT,970) LCMDE(J),RA(J,2)
0341 970 FORMAT(/,10X,'P(=,=, ',I3,',',I3,',')= ',F6.2)
0342 968 CONTINUE
0343 WRITE(NPRT,999)
0344 WRITE(NPRT,974)
0345 974 FORMAT(/,10X,'CONDITIONED ON AI CODES')
0346 D0 971 I=1,NCODE
0347 D0 972 J=1,NCAL
0348 K=MCODE(I)
0349 IF(RA(J,2).EQ.0.0) RA(J,3)=1.0
0350 IF(RA(J,2).EQ.0.0) G0 T0 972
0351 RA(J,3)=REA(K,J)/RA(J,2)+0.0
0352 972 CONTINUE
0353 WRITE(NPRT,973) (MCODE(I),LCODE(J),RA(J,3),J=1,NCAL)
0354 973 FORMAT(/,15( 5X,'P(',I3,',',I3,',')= ',F6.2))
0355 971 CONTINUE
0356 WRITE(NPRT,999)
0357 WRITE(NPRT,975)
0358 975 FORMAT(/,10X,'CONDITIONED ON GROUND TRUTH CODES')
0359 D0 976 J=1,NCAL
0360 D0 977 I=1,NCODE

```

```

0361      KEMCDELI)
0362      IF (RA(K,1).EQ.0.0) RA(K,4)=-1.0
0363      IF (RA(K,1).EQ.0.0) GO TO 977
0364      RA(K,4)=RA(K,J)/RA(K,1)*100.0
0365      977 CONTINUE
0366      WRITE(NPRT,973) (LCODE(J),NCODE(I),RA(CODE(I),4),I=1,NCODE)
0367      97A CONTINUE
0368      CONTINUE
0369      CALL CLOSE(1)
0370      CALL CLOSE(NRDP)
0371      40A CONTINUE

```

```

C      CREATE DISK OUTPUT FILE
0372      OPEN(UNIT=OUT,NAME='SPATL'OUT,OLT,TYPE='NEW',ACCESS='SEQUENTIAL',
      *FORM='UNFORMATTED',CARRIAGE='CONTINUED',NAME=I)
0373      WRITE(NOUT) (IREC1(I),I=1,16)
0374      JMAX=IREC1(13)
0375      WRITE(NOUT) ((IREC2(I,J),I=1,3),J=1,JMAX)
0376      JMAX=IREC1(14)
0377      WRITE(NOUT) ((IREC3(I,J),I=1,3),J=1,JMAX)
0378      IMAX=IREC1(15)*14
0379      JMAX=IREC1(12)
0380      WRITE(NOUT) ((IREC4(I,J),I=1,[MAX]),J=1,JMAX)
0381      JMAX=IREC1(11)
0382      WRITE(NOUT) ((IREC5(I,J),I=1,3),J=1,JMAX)
0383      CALL CLOSE(NOUT)
C      PRINT OUT CONTENTS OF IREC ARRAYS
0384      407 WRITE(NPRT,408) (IREC1(I),I=1,16)
0385      408 FORMAT(1H,' IREC1: 2X,1I4,2X,1A2,14(2X,1I4))
0386      WRITE(NPRT,409)
0387      409 FORMAT(1H,' IREC2:')
0388      DO 410 J=1,25
0389      410 WRITE(NPRT,411) J,(IREC2(I,J),I=1,3),J=25,(IREC2(I,J*25),I=1,3),
      *J=50,(IREC2(I,J*50),I=1,3),J=75,(IREC2(I,J*75),I=1,3)
0390      411 FORMAT(1H,' 4(1I4,2X)')
0391      WRITE(NPRT,412)
0392      412 FORMAT(1H,' IREC3:')
0393      WRITE(NPRT,413) ((IREC3(I,J),I=1,2),J=1,10)
0394      413 FORMAT(1H,' 1A1,1I4)')
0395      WRITE(NPRT,414)
0396      414 FORMAT(1H,' IREC4:')
0397      415 WRITE(NPRT,415) ((IREC4(I,J),I=1,24),J=1,IREC1(12))
0398      415 FORMAT(1H,' 1X,12,7I7,15I5)')
0399      WRITE(NPRT,416)
0400      416 FORMAT(1H,' IREC5:')
0401      DO 417 J=1,25
0402      417 WRITE(NPRT,411) J,(IREC5(I,J),I=1,3),J=25,(IREC5(I,J*25),I=1,3),
      *J=50,(IREC5(I,J*50),I=1,3),J=75,(IREC5(I,J*75),I=1,3)
0403      41A CONTINUE
0404      WRITE(NPRT,999)
0405      CALL DATE(0)
0406      CALL TIME(1)
0407      CALL CLOSE(NRDP)
0408      WRITE(NPRT,104) D,T
0409      WRITE(NPRT,104) T,T

```

PAGE 10

14-MAR-78

10159123

F0PTRAM IV-PLUS V02-51  
ADSPATL.FIN /TRIDLOCKS/AR

0410 104 FORMATS JMD COMPLETED ON 1,9A1,1 AT 1,8A11  
0411 STOP  
0412 END

ORIGINAL PAGE IS  
OF POOR QUALITY

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES	
1	SCODE1	014006	3075	RKJ,C0A,LCL
2	SPDATA	00004	18	RKJ,C0A,LCL
3	3IDATA	00350	944	RKJ,C0A,LCL
4	SVARS	041314	8550	RKJ,C0A,LCL
5	STEMPS	000012	5	RKJ,C0A,LCL
6	BK1	026300	5728	RKJ,EVR,GBL

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
A	1e2	4-041142	AA	1e2	4-041264	B	1e2	4-041172
FLG	1e2	4-041230	GC	1e2	4-041266	I	1e2	4-041132
J	1e2	4-041158	JMAX	1e2	4-041310	K	1e2	4-041244
KS	1e1	4-006005	L	1e2	4-041164	LCD	1e2	4-041166
N	1e2	4-041154	NAID	1e2	4-041252	NB	1e2	4-041212
NCT	1e2	4-041144	NCODE	1e2	4-041232	NC	1e2	4-041150
NPJX	1e2	4-041304	NP	1e2	4-041152	NEUT	1e2	4-041126
NRDP	1e2	4-041124	NT	1e2	4-041202	P	1e2	4-041214
RCHECK	R04	4-041220	RERR0R	R04	4-041270	RNLABL	R04	4-041300
RY	R04	4-041160	RYT	R04	4-041224	S	1e2	4-000102
S1	1e2	4-041174	S10	1e2	4-041176	S2	1e2	4-041200
Z	1e2	4-041256						

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
AI	1e2	4-007016	000642	209 (11,15)
AIFIL	1e2	4-035632	000016	7 (7)
RUF	1e1	4-000000	003764	1530 (30,60)
CHAR	1e1	4-006006	000062	25 (50)
D	1e1	4-005774	000011	4 (9)
FILAI	1e2	4-035650	000032	13 (13)
FILGT	1e2	4-035600	000032	13 (13)
GP	1e2	4-034674	000642	209 (11,15)
GT	1e2	4-006070	000642	209 (11,15)
GTYL	1e2	4-035562	000016	7 (7)
IREC1	1e2	4-036802	000040	16 (16)
IREC2	1e2	4-036542	001130	300 (3,100)
IREC3	1e2	4-037672	000050	20 (2,10)
IREC4	1e2	6-015000	011300	2400 (24,100)
IREC5	1e2	4-037742	001130	300 (3,100)
JT	1e2	4-007460	001000	256 (256)
KY	1e2	4-006732	000064	26 (26)
LCODE	1e2	4-035536	000024	10 (10)
MCODE	1e2	6-014000	001000	256 (256)
MONTH	1e2	4-041072	000030	12 (12)
MY	1e2	4-034660	000014	6 (6)
RA	R04	6-000000	014000	1072 (256,6)

REPTRAN IV=PLUS V02-51  
ADSPATL.FTN /TRIPLECKS/WR

REA R04 4-010660 024000 5120 (256,10)  
SITE I02 4-035702 000600 192 (64,3)  
T L01 4-005764 000010 4 (8)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	00	2	00	3	00	4	00	5	00		
6	1-002470	7	1-002410	8	1-002664	12	1-004314	13	1-004032		
14	00	15	1-005422	16	00	17	00	19	00		
19	1-006464	20	00	22	1-006454	30	00	31	00		
102'	3-000726	103'	3-001052	104'	3-003130	105'	3-001374	106'	3-001436		
107'	3-002054	108'	3-001316	109'	3-002142	110'	3-002134	117'	3-000372		
118'	3-000366	119'	00	121	1-000760	122	1-001320	123	00		
124	00	125	00	126	00	158'	3-001324	162'	3-001454		
200	00	201	1-005776	212	1-004514	222'	3-000450	223'	3-000570		
224	1-001254	225	00	255'	3-001140	256'	3-001734	300'	3-001274		
301'	3-002312	302	1-007514	306'	3-000214	313'	3-001510	399	00		
400	1-011600	407	00	408'	3-002764	409'	3-003614	410	00		
411'	3-003030	412'	3-003044	413'	3-003054	414'	3-003064	415'	3-003100		
416'	3-003116	417	00	418	00	502'	3-001564	601'	3-001500		
606'	3-001504	607	1-005262	666	00	702'	3-001232	703'	3-000000		
704'	3-000102	705'	00	706'	3-000114	709'	00	710'	3-000160		
777	00	776'	3-001410	800	00	801'	3-002274	850'	3-000502		
851'	3-001060	852'	3-001454	880'	3-000400	881'	3-000462	901'	3-002024		
902'	3-001614	903'	3-001000	905'	3-000272	910	00	911'	3-002512		
951'	3-002210	960'	3-002334	965'	3-002370	966	00	967'	3-002550		
968	00	969	00	970'	3-002574	971	00	972	1-010772		
973'	3-002662	974'	3-002624	975'	3-002714	976	00	977	1-011374		
988	00	999'	3-002944								

FUNCTIONS AND SUBROUTINES REFERENCED

CLOSE CROBL DATE FILNAM OPEN\$ SUBSTR TAB TIME ZAP \$ISHT

TOTAL SPACE ALLOCATED = 107440 10320

```
C SUBROUTINE TABLNCODE,RPURE,RI,FLG)
C
C SUBROUTINE TABLNCODE,RPURE,RI,FLG,LCD)
C
C IMPLICIT INTEGER (A-Z), (S-Z)
C CRANKN /BA1/PA(256,6),NCODE(256)
C
C CRANKN /RA1/RA(256,6),NCODE(256),IREC4(24,100)
```

```
0004 PRINT*6
0005 WRITE(NPRT,950) (I,I=1,6)
0006 950 FORMAT(/,9X,'X',6X,'P(X)',6X,'A(A)',6(6X,'PIX',I))
0007 NCODE=0
0008 RPURE=0.0
0009 DO 9 M=1,256
0010 RN=0.0
0011 DO 10 P=1,6
0012 IF(FLG.EQ.1) RN=RN+PAR(M,P)
IF(FLG.EQ.2) RN=RN+RA(M,P)
```

```
0013 IF(FLG.EQ.2,OR,FLG.EQ.3)RN=RN+RA(M,P)
C
C 10 CONTINUE
0014 IF(RN.EQ.0.0) GO TO 9
0015 NCODE=NCODE+1
0016 IF(FLG.EQ.1) NCODE(NCODE)=M
0017 RP=100.0*RN/RT
0018 IF(FLG.EQ.1) RPURE=RPURE+6.0*RA(M,6)
0019 IF(FLG.EQ.2) RPURE=RPURE+ RA(M,6)
```

```
0020 IF(FLG.EQ.2,OR,FLG.EQ.3)RPURE=RPURE+RA(M,6)
C
C LOAD DATA INTO IREC4
C FLG=1 = GROUND TRUTH
C FLG=2 = 209 DOTS
C FLG=3 = LABELED DOTS (NO OUTPUT TO DISK)
```

```
0021 IF(FLG.EQ.2) GO TO 30
0022 IF(FLG.EQ.3) GO TO 50
0023 IREC4(1,NCODE)=M
0024 IREC4(2,NCODE)=RN/32768.0
0025 IREC4(3,NCODE)=RN-1*REC4(2,NCODE)*32768
0026 DO 20 I=1,6
0027 IREC4(I,3,NCODE)=RA(M,I)
0028 GO TO 8
0029 DO 40 I=1,LCD
0030 IF(M.EQ.IREC4(1,I)) GO TO 41
0031 40 CONTINUE
0032 41 IREC4(10,I)=PN
0033 DO 42 J=1,4
0034 IREC4(J+10,I)=RA(M,J+2)
0035 GO TO 8
0036 50 CONTINUE
```

```
0037 A WRITE(NPRT,101) M,RP,RN,(RA(M,P),P=1,6)
0038 101 FORMAT(1H ,110,'R',F10.2)
0039 9 CONTINUE
0040 RPURE=100.0*RPURE/RT
```

ORIGINAL PAGE IS  
OF POOR QUALITY

FOTRAN IV-PLUS V02-51  
AUSPALLEIN /TRISLMO/SWR

1101100

14-AR-7A

PAGE 14

0041  
0042

RETRW  
END

ORIGINAL PAGE IS  
OF POOR QUALITY

ORIGINAL PAGE IS  
OF POOR QUALITY





FORTRAN IV-PLUS V02-51  
ADSPATL.FT /TRIDLECKS/A3

```

0001 SUBROUTINE ZAP
0002 IMPLICIT INTEGER (A-D), (S-Z)
0003 COMMON /BK1/RA(254,6),MCODE(256)
0004 DO 20 I=1,254
0005 DO 21 P=1,6
0006 PA(I,P)=0
0007 21 CONTINUE
0008 20 CONTINUE
0009 RETURN
0010 END

```

ORIGINAL PAGE IS  
OF POOR QUALITY

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	CODE1	000120	40
4	SVARS	000004	2
6	BA1	026300	5728

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ZAP		1-000000						

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
M	I02	4-000000	P	I02	4-000004			

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
IREC4	I02	6-015000	010300	2400 (24,100)
MCODE	I02	6-014000	001000	256 (256)
RA	R04	6-000000	010000	1072 (256,6)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
20	00	21	00				

TOTAL SPACE ALLOCATED = 026424 5770

ADSPATL.FTN, LPI=C170, IJADSPATL.FTN

BEARTRAN IV-PLUS V02-51  
CRPL,FTM /P:RI:PCXS/W

```

0001 SUBROUTINE CRPL (CRP,NT,IC)
0002 IMPLICIT INTEGER (A-D), (S-Z)
0003 DIMENSION MT(6)
0004 M=0
0005 DO 10 I=1,A
0006 CC=MT(I)
0007 N=0
0008 DO 20 J=1,A
0009 IF (CC.EQ.MT(J))N=N+1
0010 20 CONTINUE
0011 IF (N.LE.NC) GO TO 10
0012 N=0
0013 CRP=CC
0014 IF (N.GE.3) RETURN
0015 10 CONTINUE
0016 RETURN
0017 END

```

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	IC00E1	79	RK1,C0A,LCL
3	3IDATA	5	RK1,C0A,LCL
4	3IVARS	4	RK1,C0A,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CR0PL		1-000000									

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CC	1*2	4-000002	CR0P	1*2	F-000002*	I	1*2	4-000000	J	1*2	4-000006
NC	1*2	F-000004*							N	1*2	4-000004

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
MT	1*2	F-000004*	000014	6 (6)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000212	20	**				

TOTAL SPACE ALLOCATED = 000260 08

NO FPP INSTRUCTIONS GENERATED

CR0PL.0B/JALP1517u.1JCR0PL.FTN

ORIGINAL PAGE IS  
OF POOR QUALITY

```
0001 SUBROUTINE SUBSTP(A,I,J,R,J,M)  
0002 IMPLICIT INTEGER (A-Z)  
0003 LOGICAL A(1),B(1)  
0004 DATA BLANK/2H /  
0005 IS=1  
0006 JS=J  
0007 L=0  
0008 IF(N.EQ.0) GO TO 20  
0009 L=N  
0010 IF( L.GT. M ) L=M  
0011 DO 10 K=1,L  
0012 B(JS)=A(IS)  
0013 IS=IS + 1  
0014 JS=JS + 1  
0015 CONTINUE  
0016 IF( N.GE. M ) RETURN  
0017 L=L + 1  
0018 DO 30 K=L,M  
0019 B(JS)=BLANK  
0020 JS=JS+1  
0021 CONTINUE  
0022 RETURN  
0023 END
```

FORTRAN IV PLUS V02-51  
SUBSTR.FTN /TRBLCKS/W2

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	SCDDE1	000312	LD1
3	DATA	000024	10
4	SVARS	000012	5
5	STEPS	000002	1

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
------	------	---------	------	------	---------	------	------	---------

SUBSTR 1-000000

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BLANK	I*2	4-000000	I	I*2	F-000004*	IS	I*2	4-000002
K	I*2	4-000010	L	I*2	4-000006	M	I*2	F-000014*
			N	I*2	F-000006*	N	I*2	F-000006*

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
A	L*1	F-000002*	000001	0 (1)
B	L*1	F-000010*	000001	0 (1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	**	20	1-000022	30	**		

TOTAL SPACE ALLOCATED = 000352 117

NZ FPP INSTRUCTIONS GENERATED

SUBSTR.00J.LPI=[170,1]SUBSTR.FTN

02RTRAN IV=PLUS V02=51  
FILNAM,FTN /TRI9LCK4S/WP

```

0001 SUBROUTINE FILNAM (INNAME,OUTNAM)
0002 IMPLICIT INTEGER (A-Z)
0003 PARAMETER SIZE=64,MAXDSS=3
0004 DIMENSION INNAME(7),OUTNAM(13),SITE(SIZE,MAXDSS)
0005 DIMENSION HSEGM(5)
0006 BYTE D(9),I(2)
0007 NPPT=6

```

```

C
C *** SUBROUTINE VALIDATES SEGMENT NUMBER AND
C MOVES FILE NAME FROM INNAME TO OUTNAM ***
C ALL FILE NAMES ARE CHECKED FOR SAME SEGMENT NO. *

```

```

0008 IF(COUNT.NE.0)GO TO 50
C
C *** READ IN SITE ID TABLE ***
C
PPFN (UNITX,NAME='C110,6)JHELOC,SIT11',TYPE='D10',
* FORM='UNFORMATTED',ACCESS='SEQUENTIAL',READONLY)
0010 READ (3) ((SITE(J,K), J=1,SIZE),K=1,MAXDSS)
0011 CALL CLOSE (3)
0012 CONTINUE
0013 COUNT=COUNT+1

```

```

C *** PICK UP GT DATA BASE NUMBER FROM TABLE ***
0014 DECADE(4,705),INNAME(1)) SEGNUM
0015 FORMAT(14)
0016 D2=60 K=1,MAXDSS
0017 DO 65 J=1,SIZE
0018 IF(SITE(J,K).EQ.SEGNUM)GO TO 69
0019 CONTINUE
0020 CONTINUE

```

```

0021 WRITE (NPRT,707) SEGNUM
0022 FORMAT (//,10X,'INVALID GRZLINE TRUTH SEGMENT NO. = ',I4)
0023 GO TO 88
0024 CONTINUE
0025 PRPDMK
0026 IF(DBNUM.NE.1)GO TO 72
0027 IF(OUTNAM(5)='1'
0028 GO TO 76
0029 CONTINUE
0030 IF(DBNUM.NE.2)GO TO 74
0031 IF(OUTNAM(5)='2'
0032 GO TO 76
0033 CONTINUE
0034 IF(OUTNAM(5)='3'
0035 CONTINUE

```

```

0036 HSEGM(COUNT)=SEGNUM
0037 DO 80 I=1,COUNT
0038 IF(HSEGM(I).NE.SEGNUM)GO TO 85
0039 CONTINUE
0040 GO TO 89
0041 CONTINUE
C *** INVALID SEG NO. ERROR DOES NOT MATCH OTHERS *
0042 WRITE (NPRT,708)HSEGM(1),COUNT,SEGNUM
0043 FORMAT (1H0,'SEGMENT NUMBERS DO NOT MATCH',5X,
* ISEG GT = ',I4,' SEG ',I4,' = ',I4)

```

ORIGINAL PAGE IS  
OF POOR QUALITY



FORTRAN IV-PLUS V02-51  
FILNAM.FIN /IR:ELCKNS/83

0044 HB CONTINUE  
0045 CALL DATE (P)  
0046 CALL TIME (T)  
0047 WRITE (NPRT,710)T,  
0048 WRITE (NPRT,710)DAT  
0049 STOP  
0050 710 F2EMAI(HH,LDX,LDGE ERRORED OFF = ON ' ,9A1.' AT ' ,8A1)  
0051 89 CONTINUE  
0052 CALL SUBSTR (INNAME,1,13,QUINAM,12,13)  
0053 RETURN  
0054 END

FORTRAN IV PLUS V02-51  
FILNAM.FTN /TSIBLCKS/AF

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	ICODE1	001136	303 RW,LCOM,LCL
2	IPDATA	000044	16 RW,LCOM,LCL
3	SIDATA	000366	123 RW,LCOM,LCL
4	SVARR	000652	213 RW,LCOM,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
FILNAM		1-000000						
NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
COUNT	I02	4-000634	DRNTH	I02	4-000646	I	I02	4-000650
NPRT	I02	4-000634	SEGLUM	I02	4-000644	J	I02	4-000640
						K	I02	4-000642

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
D	L01	4-000612	000011	4 (9)
HSEGM	I02	4-000600	000012	5 (5)
INAME	I02	F-000002*	000016	7 (7)
MUTNAM	I02	F-000004*	000032	13 (13)
SITE	I02	4-000000	000600	192 (64,3)
T	L01	4-00062X	000010	4 (8)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
50	1-000236	60	**	65	**	69	1-000454
74	1-000545	76	1-000566	80	**	85	1-000660
89	1-001076	706*	3-000000	707*	3-000004	708*	3-000060
						710*	3-000162

FUNCTIONS AND SUBROUTINES REFERENCED

CLOSE DATE OPENS SUBSTR TIME

TOTAL SPACE ALLOCATED = 002442 657

NO FPP INSTRUCTIONS GENERATED

FILNAM,08JULPI=C170.11FILNAM.FTN

ORIGINAL PAGE IS  
OF POOR QUALITY

ADSPATL/LPI/SH=ADSPATL/CC,CR=PL,SUESTR,FILNAM

UNITS=8

ASC=SY13

ASC=SY14

ASC=LPI6

ASC=SY17

ACTFIL=8

MAXBUF=3060

PRI=50

//

FILE ADSPAT.TSK12 MEMORY ALLOCATION MAP  
 THIS ALLOCATION WAS DONE ON 14-MAR-78  
 AT 1103 BY TASK BUILDER VERSION R10

\*\*\* ROOT SEGMENT: ADSPAT

R/W MEM LIMITS 00000 176343 176344  
 STACK LIMITS 00000 00077 00100  
 DISK BLK LIMITS 00000 00001 000177  
 IDENTIFICATION I 14MAR  
 PRG XFR ADDRESS: 046552  
 TASK ATTRIBUTES: FP,NF

PROGRAM SECTION ALLOCATION SYNOPSIS

< BLK> 00100 020251 017252  
 < BK1 > 020252 046551 026300  
 < CODE1 > 046552 066131 017340  
 < DATA > 066132 072373 004242  
 < PDATA > 072374 072507 000114  
 < STEPS > 072510 072527 000020  
 < SVARS > 072530 134767 042240  
 < SLOTS > 134770 135767 001000  
 < SDEV1 > 135770 137527 001540  
 < SDEV0 > 137530 137530 000000  
 < SFI00 > 137530 142123 002374  
 < SFI01 > 142124 142255 000132  
 < SFI02 > 142256 142761 000504  
 < SFI03 > 142762 142762 000000  
 < SFI04 > 142762 145113 002132  
 < SFI05 > 145114 145151 000036  
 < SFSR1 > 145152 155351 010200  
 < SFSR2 > 155352 155455 000104  
 < SSI001 > 155456 163441 003764  
 < SSI002 > 163442 163442 000000  
 < SSI01 > 163442 163541 000100  
 < SSI02 > 163542 163542 000000  
 < SSI03 > 163542 164345 000604  
 < SSI04 > 164346 175615 011250  
 < SRESL > 175616 176343 000526  
 < ABS > 000000 000000 000000