Central Florida Regional Planning Model
Version 6.1
Final Draft SubArea Application

Prepared for the
Florida Department of Transportation

Updated By
HTNB
1276 Metropolitan Boulevard
Suite 304
Tallahassee, Fl 32312
September 19, 2016

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Acronyms

CFRPM	Central Florida Regional Planning Model
FDOT	Florida Department of Transportation
GUI	Graphical User Interface
MPO	Metropolitan Planning Organization
SERPM	South East Florida Regional Planning Model
TAZ	Transportation analysis zone
TPO	Transportation Planning Organization

1. SUBAREA APPLICATION

The FDOT District 5 requested a Sub Area application be incorporated into the CFRPM v6.1 to allow users to reduce model run times. A review of models in the state found that the SERPM v6.5.4 already had a good sub area application in it therefore it was used the basis for this new application.

The following sections describe how the SubArea application was created in both the Daily and Time of Day (TOD) versions of CFRPM and how to use it.

1. Catalog Keys

There are 3 new keys required for the SubArea application in both the Daily and TOD modes which include 2 note keys and 1 "check box" key. 2 other keys are used in the application that already exist and were documented in the *Graphical User Interface & MPO Reporting Tech Memo*. The existing keys are shown as the first 2 keys below. The 5 keys used in the SubArea application are described and shown in Table 1-1 and are found in the model as shown in Figure 1-1.

Table 1-1: SubArea Application Catalog Key Definitions

Key	Definition
MPO_DIST	Key used to designate how the user wishes the districting to be run. Setup as a drop down list to avoid errant entries in the GUI.
MPO NOTES	Note key used to define the districting options for selection in the MPO_DIST key in the GUI.
Subnote0	New Note key used as a heading for the User-Defined Subarea/Windowed Network section of the GUI.
Subnote1	New Note key used to define the steps the user must take to execute their own Subarea for selection.
UserDefinedSubarea	New key used to select the users Defined Windowed Subarea step in the GUI.

Figure 1-1: SubArea Application Keys

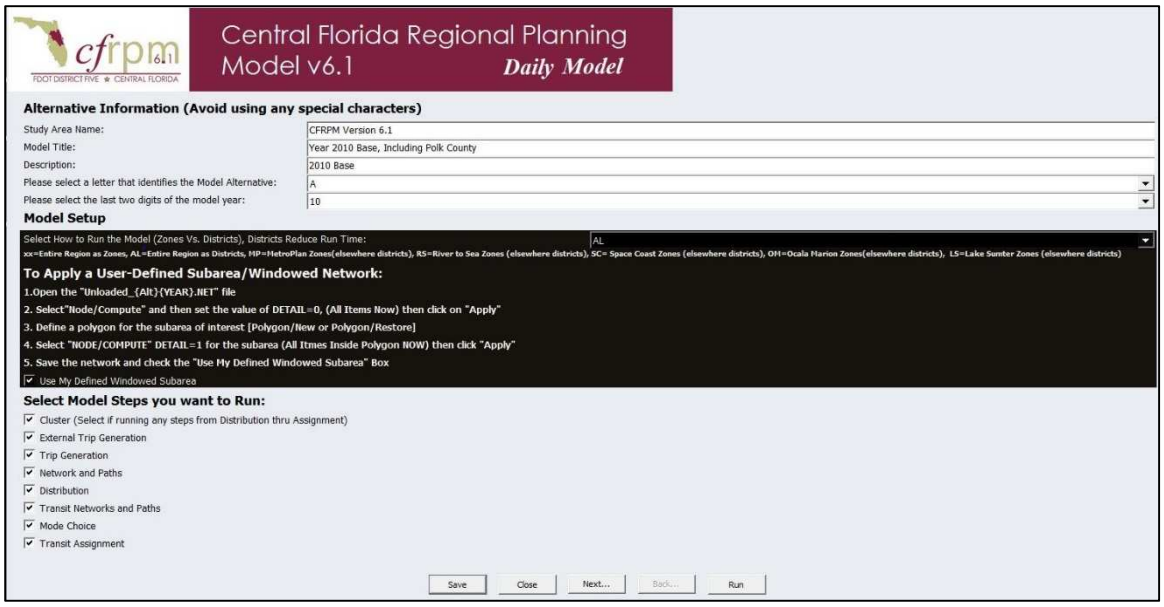
2. Design and Use of the SubArea Application in the Updated GUI

In order to implement the new SubArea application, the design of the GUI for both the Daily and TOD models needed to be updated. This section discusses those updates and explains how to use the new application.

1. Design of the Updated GUI

The implementation of the 5 keys of the SubArea application immediately follow the "MODEL SETUP" note on Page 1 of the GUI and are shown in Figure 1-2.

Figure 1-2: SubArea Application GUI Page 1



2. Use of the SubArea Application

The SubArea section of the GUI in both the Daily and TOD models allow the user to select how the districting is implemented during the model run.

The model can be run 4 different ways and is accomplished through a drop down list for the first 3 options and through a polygon procedure for the last.

1. The user can select a specific MPO/TPO area. Once selected, the MPO/TPO area is run as TAZs with the rest of the model run as districts. See Table 1-2.
2. The user can select the entire region as TAZ. Once selected, the whole model area is run as TAZs.
3. The user can select the entire region as districts. Once selected, the whole model area is run as districts.

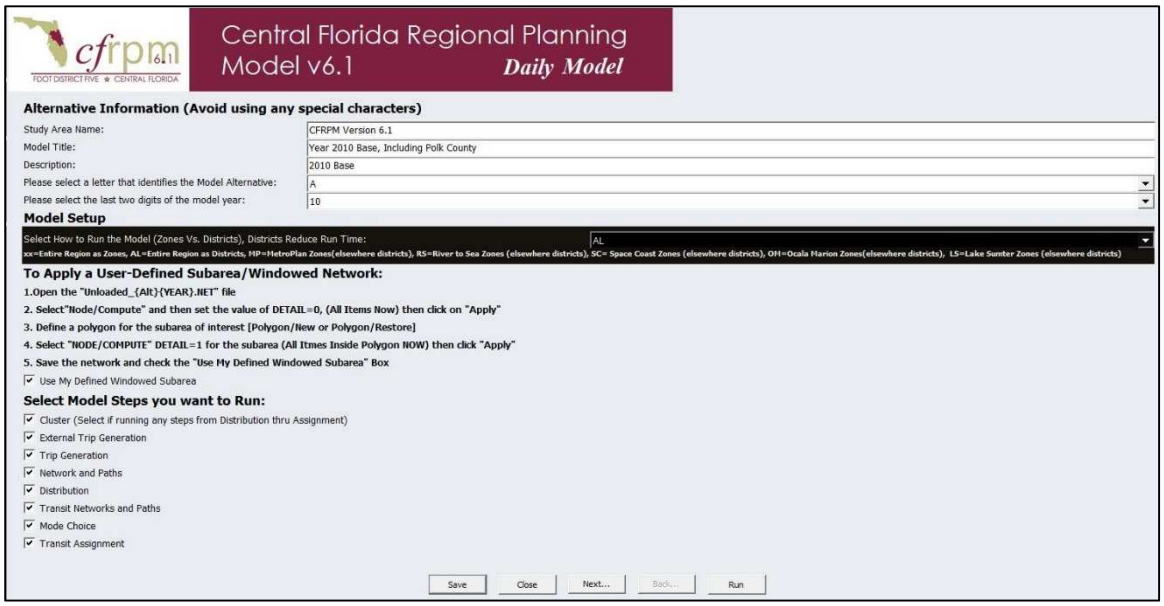
4. The user can define their own SubArea. Once selected, the model is run with the selected SubArea as TAZs with the rest of the model run as districts.

Table 1-2: MPO_DIST Catalog Key Definitions

Key Value	Definition
xx	This selection runs the Entire Region as Zones.
AL	This selection runs the Entire Region as Districts.
MP	This selection runs MetroPlan as Zones and the rest of the model area as Districts.
RS	This selection runs River to the Sea TPO as Zones and the rest of the model area as Districts.
SC	This selection runs Space Coast TPO as Zones and the rest of the model area as Districts.
OM	This selection runs Ocala/Marion TPO as Zones and the rest of the model area as Districts.
LS	This selection runs Lake/Sumter MPO as Zones and the rest of the model area as Districts.

To run the model by MPO, the entire region as DISTRICTS or TAZs the user simply selects that option from the drop down menu. As an example, Figure 1-3 below shows the selection of "AL" to run the entire model as Districts.

Figure 1-3: Select District/Zone Option



To run the model with a user Defined Windowed SubArea the user follows the instructions located under the heading “To Apply a User-Defined Subarea/Windowed Network”. Figure 1-4 shows these steps.

Figure 1-4: User Defined Subarea Steps

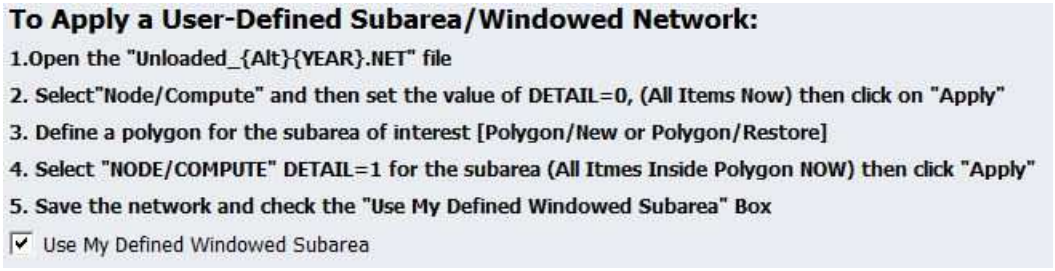


Table 1-3 shows the values for the UserDefinedSubarea Key.

Table 1-3: UserDefinedSubarea Catalog Key Definitions

Key Value	Definition
0	Does not run the User Defined Subarea scripts
1	Runs the User Defined Subarea scripts

Figures 1-5 through 1-9 show steps 1-4 which are the same in both the Daily and TOD models except for the first step shown in Figures 1-5 & 1-6 as the application names are different.

Figure 1-5: User Defined Subarea Step 1 Daily Model

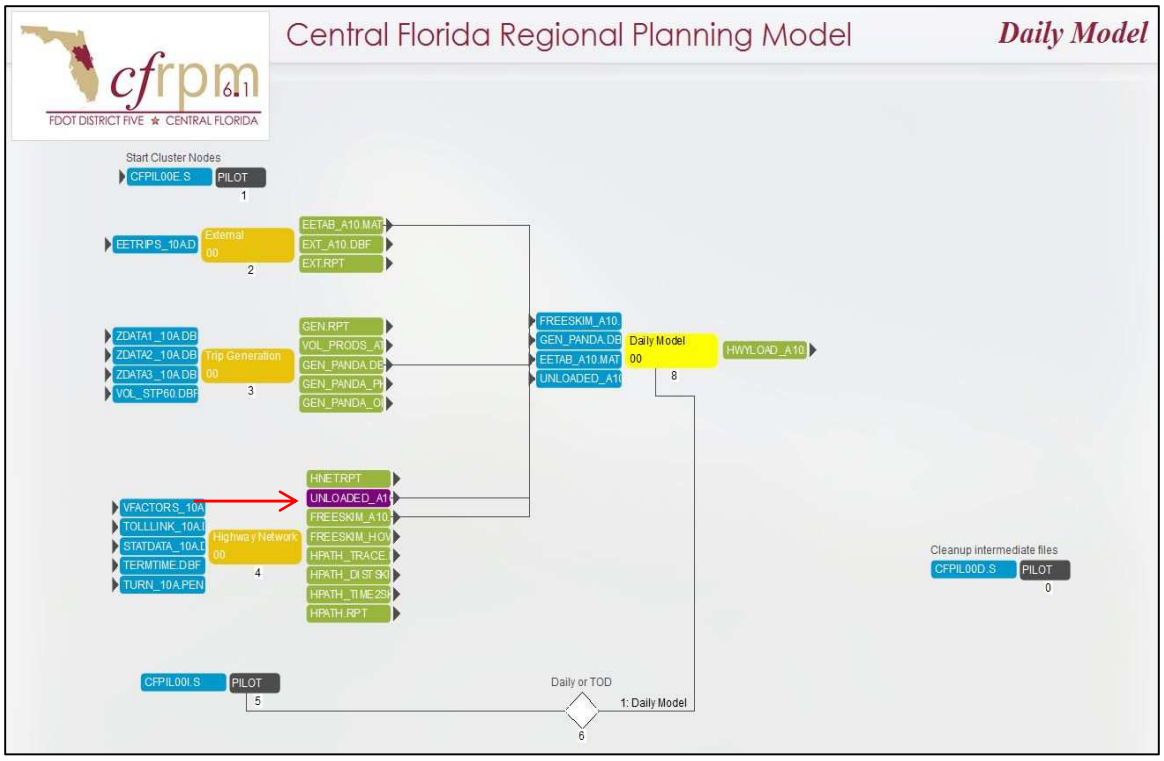


Figure 1-6: User Defined Subarea Step 1 TOD Model

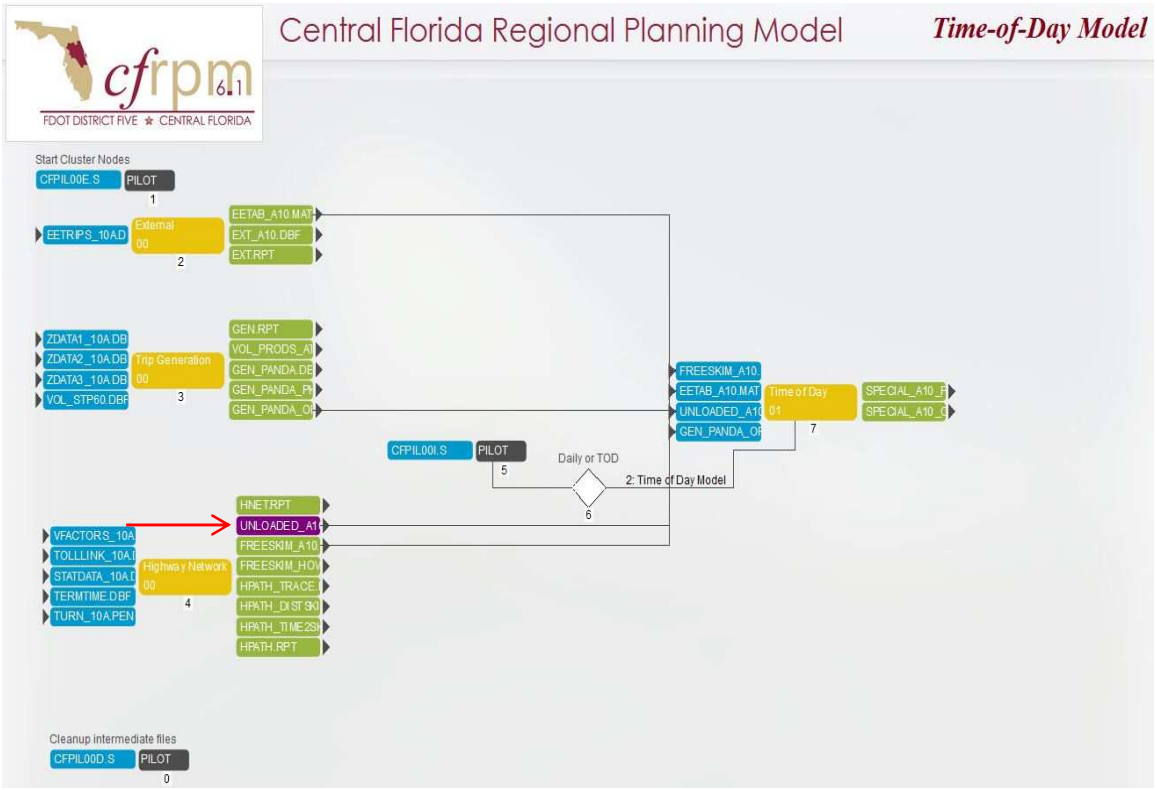


Figure 1-7: User Defined Subarea Step 2

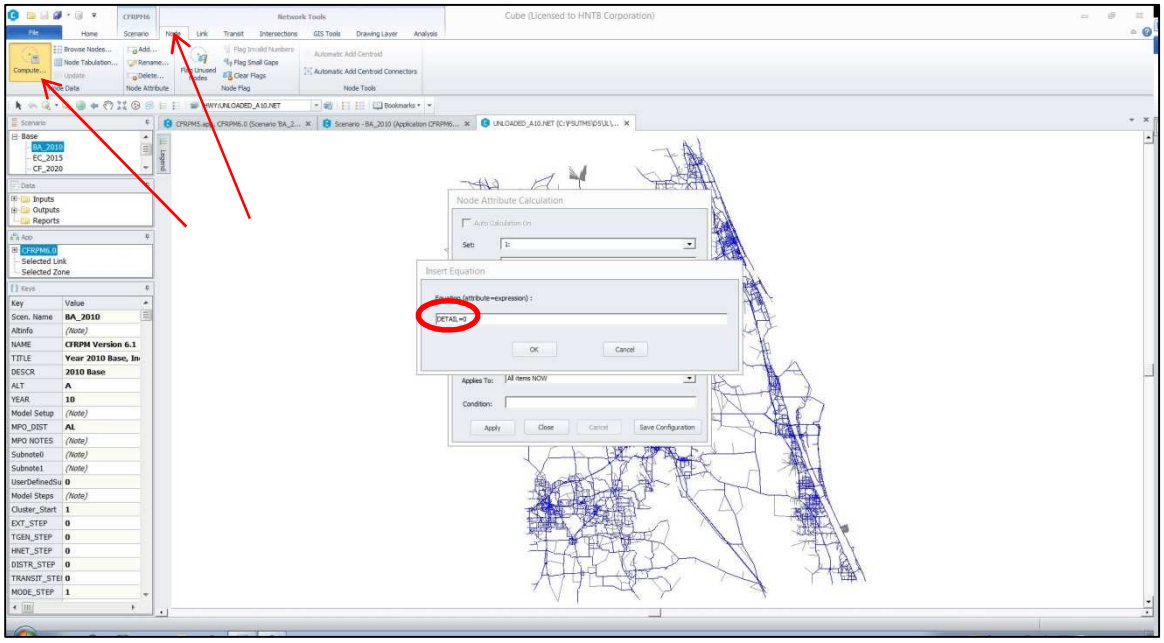


Figure 1-8: User Defined Subarea Step 3

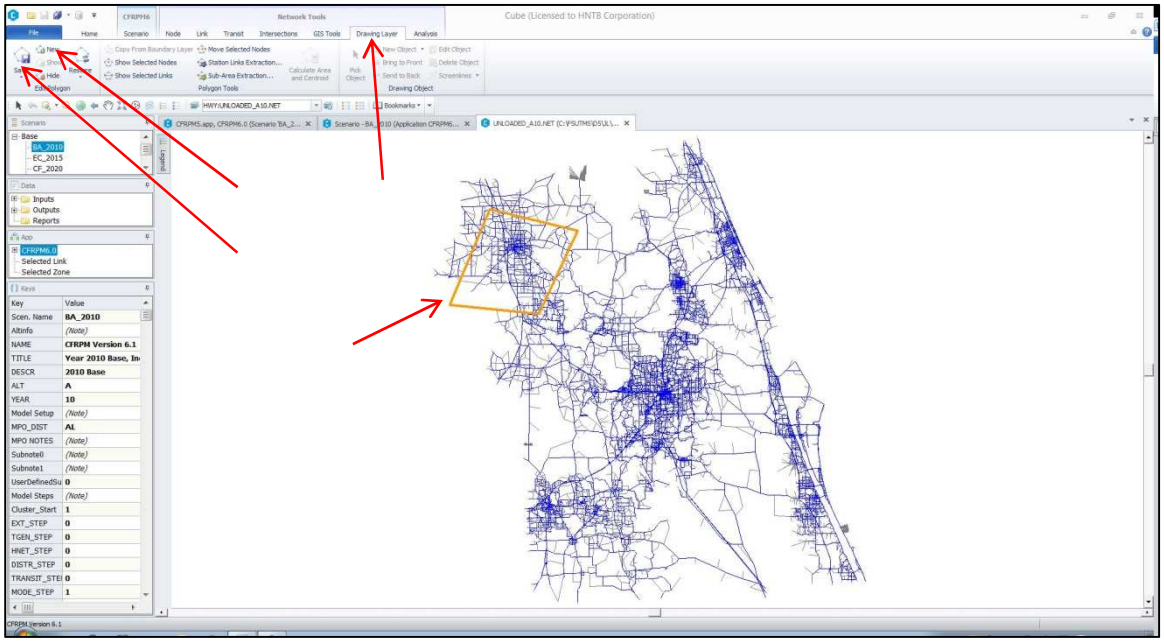
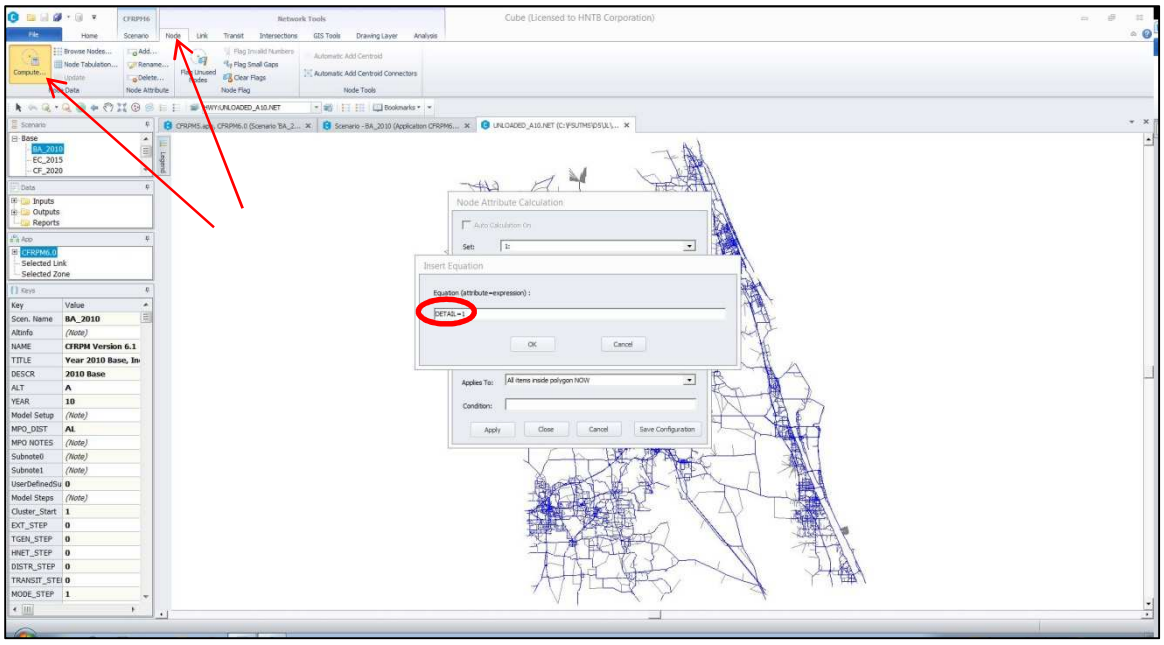


Figure 1-9: User Defined Subarea Step 4



The final step is the select the “check box” for “Use My Defined Windowed Subarea” on the GUI.

3. Application & Script Modifications

In order to accommodate the use of the SubArea application in the model, modifications were made to the DISTRIBUTION, MODE CHOICE and HIGHWAY ASSIGNMENT applications.

1. Daily Application

The SubArea application was inserted as step 19 in DISTRIBUTION. This application is where the networks are configured based on the user’s selection for SubArea windowing. See Figure 1-10 on the following page. All scripts are contained in Appendix A.

Step 28 of the DISTRIBUTION application was also added to create the SubArea trip tables for assignment. See Figure 1-10.

The final modification in is the Mode Choice Application. Step 2 was modified to use the output trip table from the SubArea application named “SAHTTAB_TEM.MAT”. See Figure 1-12 for the Mode Choice application. The modified Script is located in Appendix A.

Figure 1-10: Daily DISTRIBUTION Application

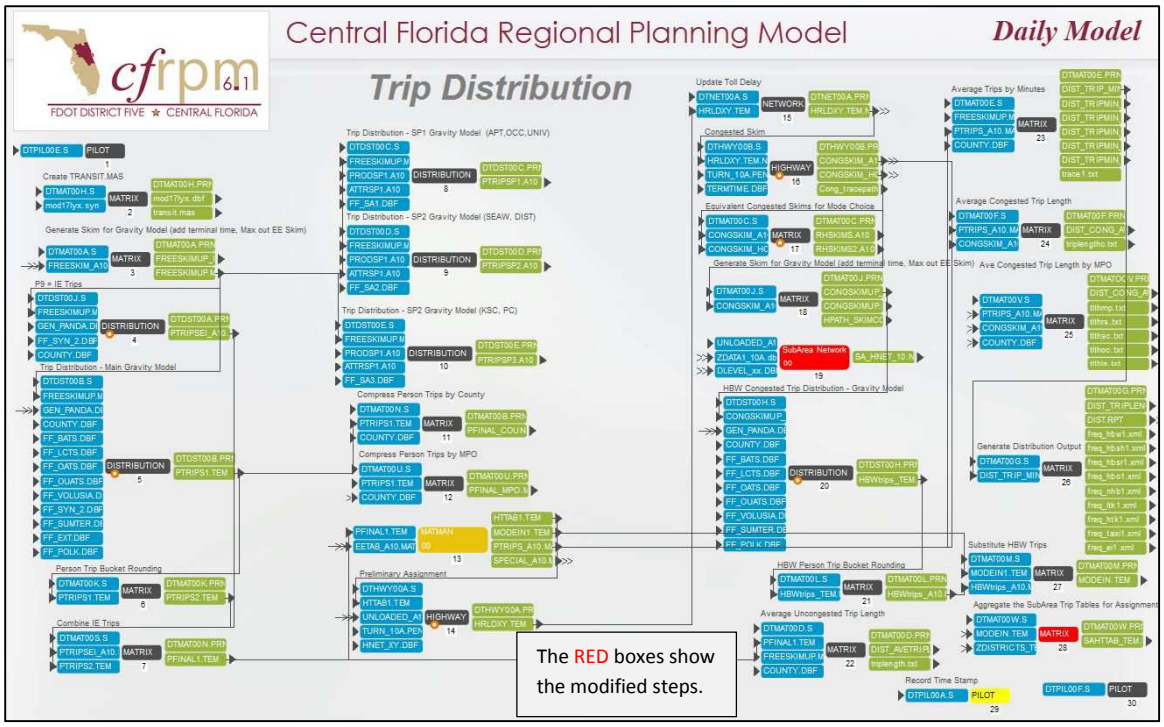


Figure 1-11: Daily SubArea Application

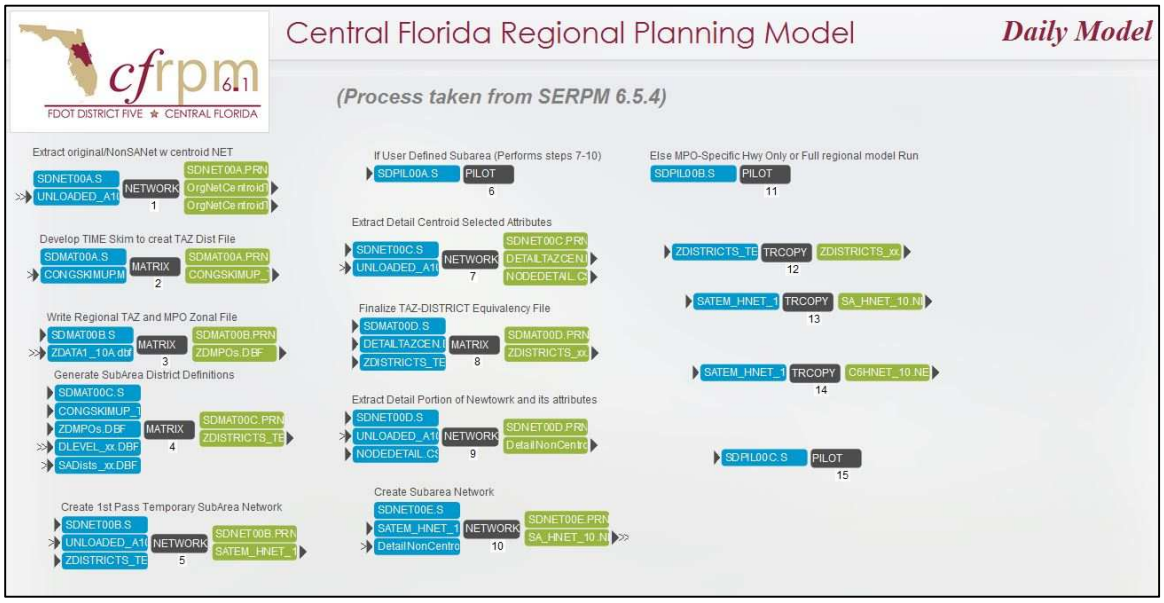
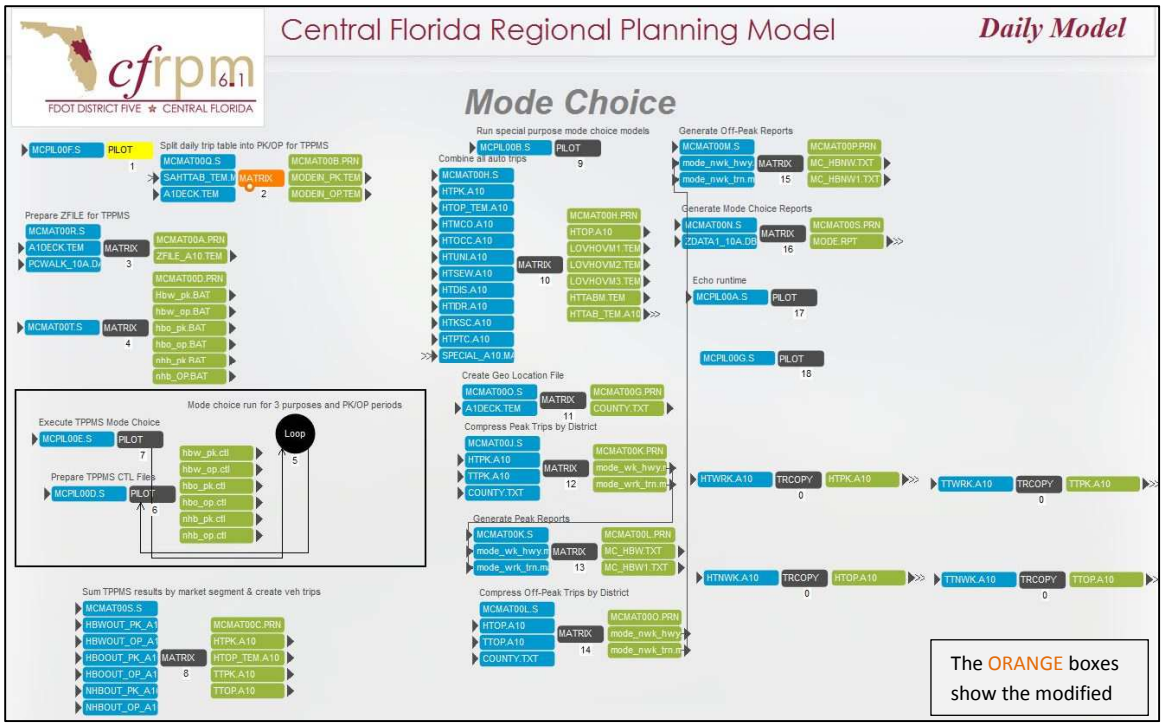
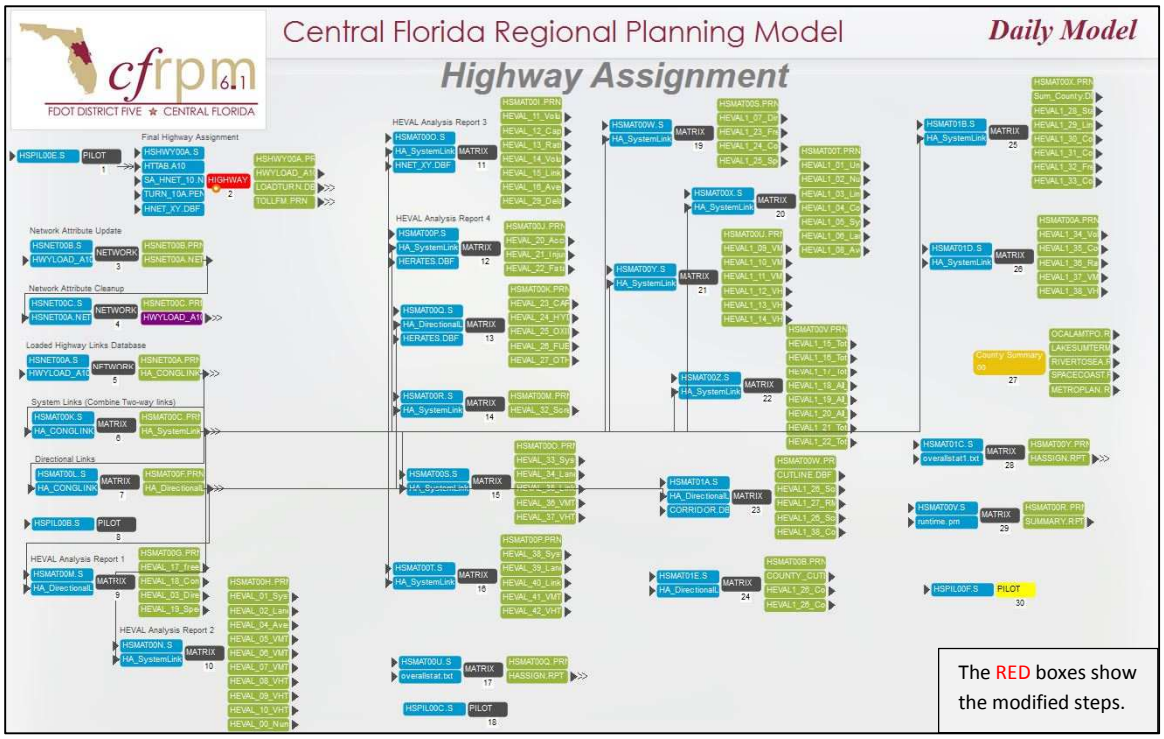


Figure 1-12: Daily Mode Choice Application



The final modification was done in the Highway Assignment application where the UNLOADED_{ALT}{YEAR}.NET network was replaced with the SA_HNET_{YEAR}.NET network file. See Figure 1-13 on the next page for the Daily Model.

Figure 1-13: Daily Highway Assignment Application



2. TOD Application

The DISTRIBUTION application in the TOD model already had 30 steps in it which is the maximum and application can have. In order to insert the SubArea application, the Peak Distribution steps were made into their own sub application as the new step 8. This allowed the SubArea application to be inserted as new step 15 in DISTRIBUTION. The SubArea application is where the networks are configured based on the user's selection for SubArea windowing. See Figure 1-14 on the following page. All scripts are contained in Appendix B.

Step 19 of the DISTRIBUTION application was also added to create the SubArea trip tables for assignment. See Figure 1-15 on the following page.

Figure 1-14: TOD DISTRIBUTION Application

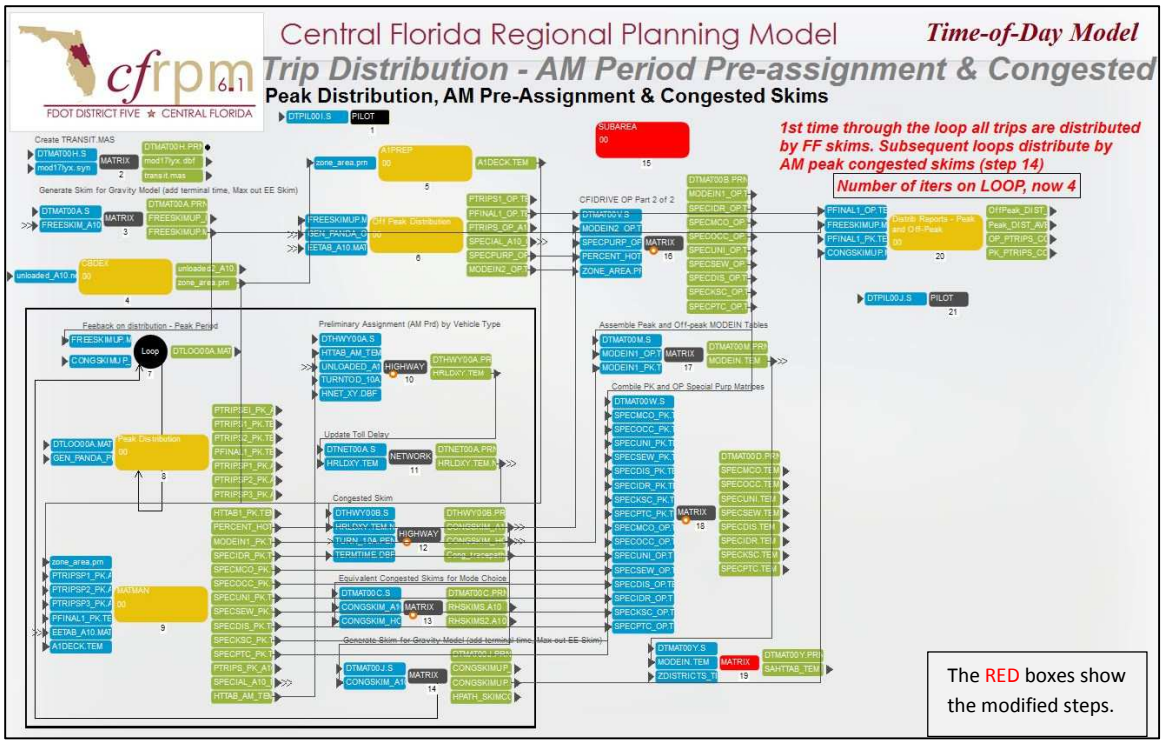
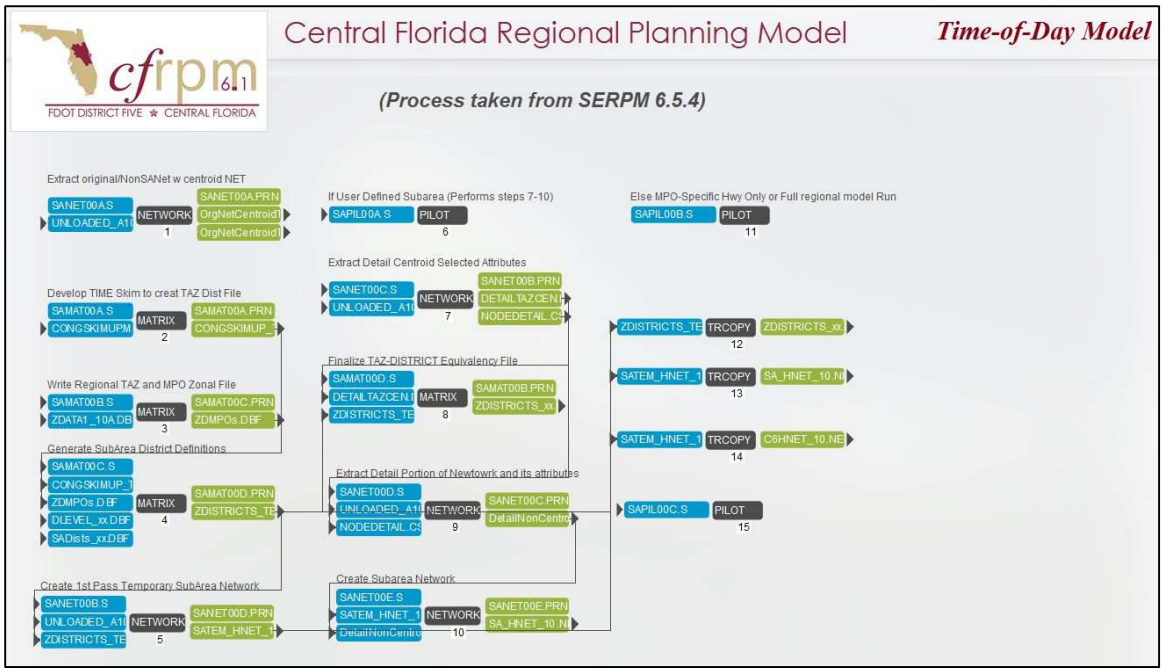
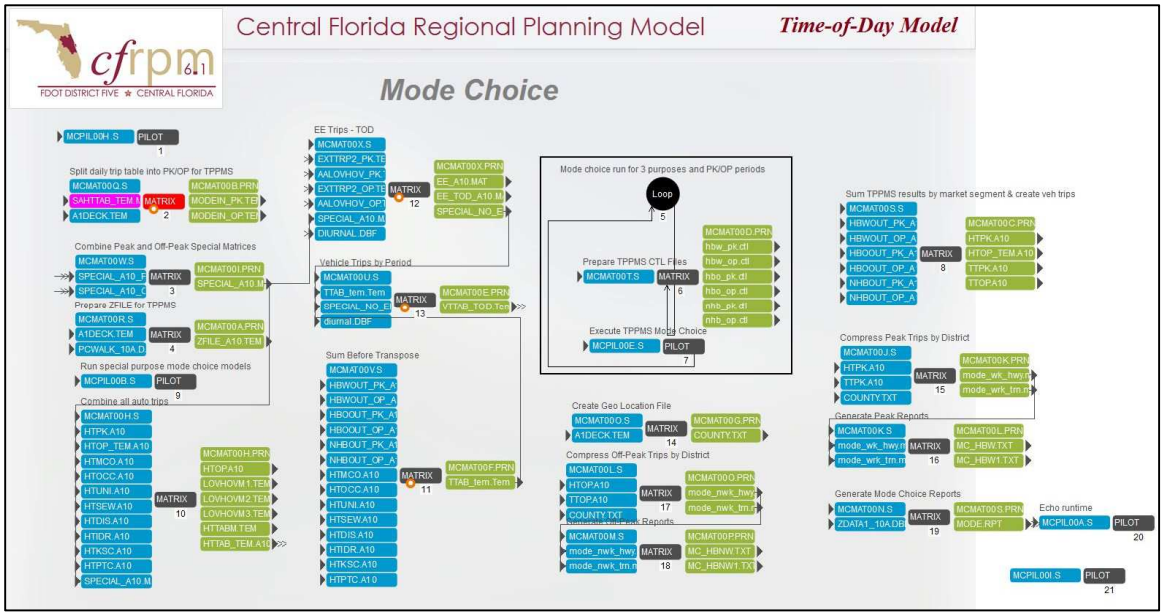


Figure 1-15: TOD SubArea Application



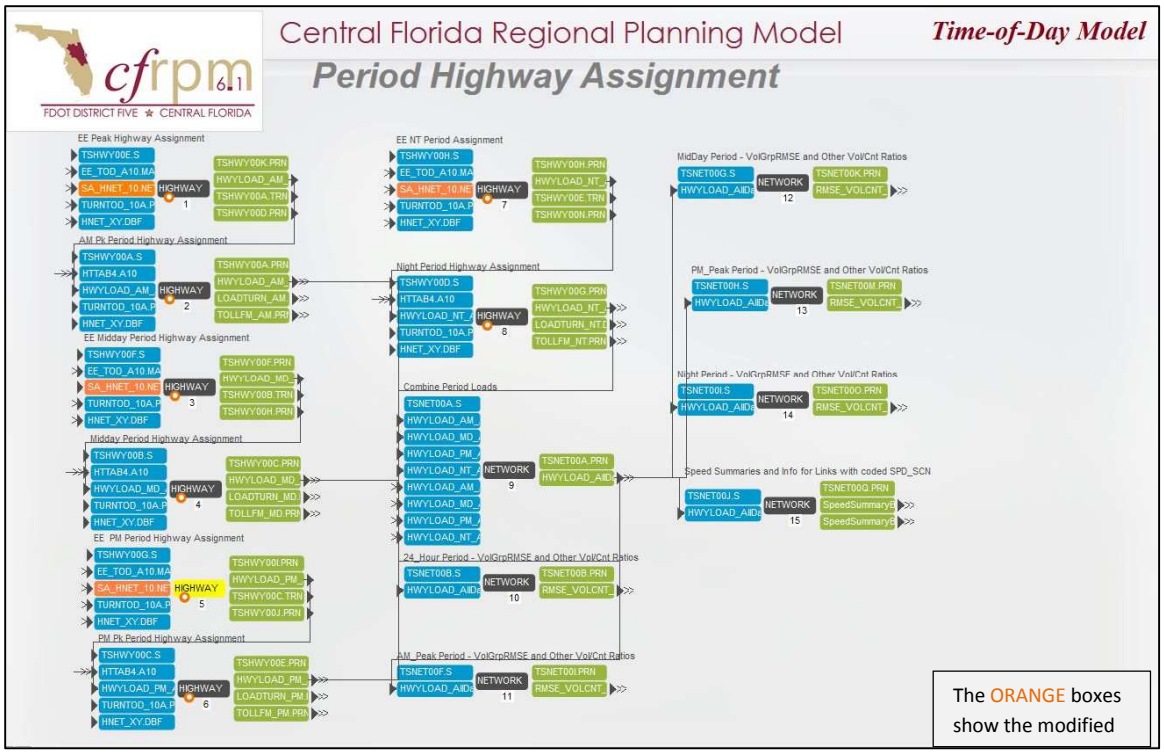
Another modification was made in the Mode Choice Application. Step 2 was modified to use the output trip table from the SubArea application named “SAHTTAB_TEM.MAT”. See Figure 1-16 below for the Mode Choice application. The modified Script is located in Appendix B.

Figure 1-16: TOD MODE CHOICE Application



The final modification was done in the Period Highway Assignment sub application of HIGHWAY ASSIGNMENT application, where the UNLOADED_{ALT}{YEAR}.NET network was replaced with the SA_HNET_{YEAR}.NET network file. See Figure 1-17 on the next page for the TOD Model.

Figure 1-17: TOD Period Highway Assignment Application



A. SubArea Application Scripts - Daily Model

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDNET00A.PRN" MSG='Extract
    original/NonSANet w centroid NET'
3   FILEI LINKI[2] = "{SCENARIO_DIR}\Output\UNLOADED_{ALT}{Year}.NET"
4   FILEO NETO = "{SCENARIO_DIR}\Output\Temp\OrgNetCentroidTime.NET"
5   FILEO LINKO = "{SCENARIO_DIR}\Output\Temp\OrgNetCentroidTime.TEM",
6      include=a,b,TimeOrg
7
8   PROCESS  PHASE=LINKMERGE
9     if ((a >5406)& (b >5406)) delete
10    TimeOrg=Time
11  ENDPROCESS
12  ENDRUN
13
```

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2  RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDMAT00A.PRN"
    MSG='Develop TIME Skim to creat TAZ Dist File'
3  FILEI MATI[1] = "{SCENARIO_DIR}\Output\Temp\CONGSKIMUP.MAT"
4  FILEO MATO[1] = "{SCENARIO_DIR}\Output\Temp\CONGSKIMUP_TEM.MAT",
5  mo=1,NAME=TIME
6
7  PAR ZONEMSG=100
8  MW[1]=mi.1.1*1      ; TIME With Terminal Time (in Minutes) - Step Not really
   needed as TT is already in Minutes. Done for consistency with SERPM
9  MW[1][I]=1000000    ; QUICKER (WW)
10
11 ENDRUN
12
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an
editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDMAT00B.PRN"
 MSG='Write Regional TAZ and MPO Zonal File'
FILEI ZDATI[1] = "{SCENARIO_DIR}\Input\ZDATA1_{Year}{ALT}.dbf"
FILEO RECO[1] = "{SCENARIO_DIR}\Output\ZDMPOs.DBF",
form=10.0, FIELDS=N,MPO
PAR ZONES={ZONES}
PAR ZONEMSG=100


;Get Zonal TAZ Info as Node Record...
N=ZI.1.TAZ_REG
MPO=ZI.1.MPO

_ztemp=_ztemp+1

;Set Data for Indian River  and Polk Internal and Dummy Zones

if(z>=4601 & z<={ZONESI})
   N=z

    MPO=0

endif

;Set Data for MetroPlan Orladndo Internal and Dummy Zones
if(z>=1 & z<=1400)
  N=z

    MPO=1

endif

;Set Data for Volusia Internal and Dummy Zones

if(z>=1801 & z<=2900)
  N=z

    MPO=2

endif

;Set Data for Flagler Internal and Dummy Zones

if(z>=4401 & z<=4600)
  N=z

    MPO=2
ENDIF

;Set Data for Space Coast Internal and Dummy Zones

if(z>=2901 & z<=3700)
   N=z

    MPO=3

endif

;Set Data for Ocala Marion Internal and Dummy Zones

```

```
61   if(z>=3701 & z<=4200)
62     N=z
63
64       MPO=4
65
66   endif
67
68   ;Set Data for Lake Internal and Dummy Zones
69
70   if(z>=1401 & z<=1800)
71      N=z
72
73       MPO=5
74
75   endif
76
77
78   ;Set Data for Sumter Internal and Dummy Zones
79
80   if(z>=4201 & z<=4400)
81      N=z
82
83       MPO=5
84
85   ENDIF
86
87   WRITE RECO=1
88
89   ;External Zones
90   if(z={ZONESI})
91     loop jj={ZONESI}+1,{ZONES}
92       N=jj
93
94       if (N>=5351 & N<=5357)          ;Indian River Polk Externals
95        MPO=0
96       ENDIF
97
98        if (N>=5361 & N<=5377)          ;Polk Externals
99         MPO=0
100       ENDIF
101
102       if (N>=5358 & N<=5360)          ;MetroPlan Orlando Externals
103         MPO=1
104      endif
105
106      if (N>=5401 & N<=5406)          ;River to Sea Externals
107        MPO=2
108      endif
109
110      if (N>=5384 & N<=5400)          ;Ocala Marion Externals
111         MPO=4
112      endif
113
114      if (N>=5378 & N<=5383)          ;Lake Sumter Externals
115        MPO=5
116      endif
117
118
119      WRITE RECO=1
120
121    endloop
122  ENDIF
```

```
123    ENDRUN
124
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDMAT00C.PRN" MSG='Generate SubArea District Definitions'

FILEI ZDATI[1] = "{SCENARIO_DIR}\Output\ZDMPOs.DBF",
    z=N
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\Input\DLEVEL_{MPO_DIST}.DBF"
FILEI LOOKUPI[2] = "{SCENARIO_DIR}\Input\SADists_{MPO_DIST}.DBF"
FILEO RECO[1] = "{SCENARIO_DIR}\Output\Temp\ZDISTRICTS_TEM_{MPO_DIST}.DBF",
    Fields= TAZ,DTAZ,NEAR,SADIST,MPO
FILEI MATI[1] = "{SCENARIO_DIR}\Output\Temp\CONGSKIMUP_TEM.MAT"

LOOKUP LOOKUPI=2,  ; One record per District
       NAME=DCENTROID,
         LOOKUP[1]=CENTERTAZ, RESULT=SADIST,
         LOOKUP[2]=CENTERTAZ, RESULT=CENTERTAZ,
         LOOKUP[3]=CENTERTAZ, RESULT=MPOCODE,
         LOOKUP[4]=CENTERTAZ, RESULT=SADST_TW,
       FAIL=0,0,0

LOOKUP LOOKUPI=1,  ; One record per MPO
       NAME=LEVEL,
         LOOKUP[1]=MPO, RESULT=DLEVEL, ; 1=District Level, 0=TAZ Level
       FAIL=0,0,0

Parameters ZONES={ZONES}
PAR ZONEMSG=100


near=1000
TAZ=i
MPO=zi.1.MPO
mylevel=LEVEL(1,MPO)
IF(i>{ZONESI}) MPO=99
  jloop
   place=DCENTROID(2,j)
   thisdist=DCENTROID(1,j)
   if(place<>0)
      if(i=place)
        near=0
        DTAZ=j
        SADIST=thisdist
      else
        ctime=mi.1.time[j]
        if(ctime < near)
           near=ctime
           DTAZ=j
           SADIST=thisdist
        endif
      endif
   endif
  endjloop
if(mylevel=0) DTAZ=i ; At the TAZ level
if(near<1000 & MPO>=0) WRITE RECO=1 ;added = in MPO>0 for Polk and Indian River  zones 4601-5350

ENDRUN
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDNET00B.PRN" MSG='Create 1st Pass
    Temporary SubArea Network'
3   FILEI LINKI[2] = "{SCENARIO_DIR}\Output\UNLOADED_{ALT}{Year}.NET"
4   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\Output\Temp\ZDISTRICTS_TEM_{MPO_DIST}.DBF"
5   FILEO NETO = "{SCENARIO_DIR}\Output\SATEM_HNET_{YEAR}.NET"
6
7   array gone=99999
8
9   PROCESS  PHASE=NODEMERGE
10    LOOKUP LOOKUPI=1,
11         NAME=SADIST,
12           LOOKUP[1]=TAZ, RESULT=DTAZ,
13         FAIL[3]=0
14    if(N<={ZONES}) _NN=_NN+1
15    if(N<={ZONES}) SA_Centroid=1
16    IF((N<>SADIST(1,N)) & (N<={ZONESI}))
17
18    ;| (NODETYPE=3,4))   ; for SERPM Only
19       gone[N]=1
20      _dd=_dd+1
21    delete
22    endif
23  ENDPROCESS
24
25  PROCESS  PHASE=LINKMERGE
26    _AD=gone[a]
27    _BD=gone[b]
28    if(_AD>0 | _BD>0)
29      print list= A(5.0),B(5.0),' deleted'
30      delete
31    endif
32  ENDPROCESS
33
34  PROCESS  PHASE=SUMMARY
35    _LL=_NN-_dd
36    print list=' ******* Deleted ',_dd(4.0),' nodes out of a total of ',_NN(4.0),
    ' leaving ',_LL(4.0),' active centroids'
37  ENDPROCESS
38
39
40
41
42  ENDRUN
43
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   if ({UserDefinedSubarea}=1) ; User defined Subarea
3
4
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDNET00C.PRN" MSG='Extract Detail
    Centroid Selected Attributes'
3   FILEO PRINTO[1] = "{SCENARIO_DIR}\Output\NODEDETAIL.CSV"
4   FILEO NODEO = "{SCENARIO_DIR}\Output\Temp\DETAILTAZCEN.DBF",
5      INCLUDE=TAZ,DTAZ,NEAR,SADIST,MPO,DETAIL
6   FILEI LINKI[1] = "{SCENARIO_DIR}\Output\UNLOADED_{ALT}{Year}.NET"
7
8   PROCESS  PHASE=NODEMERGE
9   ; ---------------------------------------------------------------
10  ; NEXT STEP IS TO CAPTURE THE DETAIL ATTRIBUTE ON THE NODES TO
11  ; SAVE THE USER NEEDING TO COMPUTE IT FOR LINKS TOO
12    PRINT CSV=T, LIST=N(6.0),DETAIL(2.0), PRINTO=1
13  ; ---------------------------------------------------------------
14    IF (DETAIL=1 & N <={ZONES})
15      TAZ=NI.1.N
16      DTAZ=NI.1.N
17      NEAR=0
18      SADIST=5000+TAZ
19      MPO=NI.1.MPO
20    ELSE
21      DELETE
22    ENDIF
23  ENDPROCESS
24
25  ENDRUN
26
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an
editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDMAT00D.PRN"
 MSG='Finalize TAZ-DISTRICT Equivalency File'

FILEI ZDATI[2] = "{SCENARIO_DIR}\Output\Temp\ZDISTRICTS_TEM_{MPO_DIST}.DBF",
  Z=TAZ
FILEI ZDATI[1] = "{SCENARIO_DIR}\Output\Temp\DETAILTAZCEN.DBF",
  z=taz
FILEO RECO[1] = "{SCENARIO_DIR}\Output\ZDISTRICTS_{MPO_DIST}.DBF",
   FIELDS=TAZ,DTAZ,NEAR,SADIST,MPO,DETAIL

PAR ZONES={ZONES}
PAR ZONEMSG=100

;xxxxxx
TAZ=ZI.2.TAZ
DETAIL=ZI.1.DETAIL

IF (DETAIL=1)
  DTAZ=ZI.1.DTAZ
  NEAR=ZI.1.NEAR
  SADIST=ZI.1.SADIST
  COUNTY=ZI.1.MPO
  DETAIL=ZI.1.DETAIL
 Else
  DTAZ=ZI.2.DTAZ
  NEAR=ZI.2.NEAR
  SADIST=ZI.2.SADIST
  COUNTY=ZI.2.MPO
  DETAIL=0
ENDIF

WRITE RECO=1
ENDRUN

```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDNET00D.PRN" MSG='Extract Detail
    Portion of Newtowrk and its attributes'
3   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\Output\NODEDETAIL.CSV"
4   FILEO NETO = "{SCENARIO_DIR}\Output\Temp\DetailNonCentroid.NET"
5   FILEI LINKI[1] = "{SCENARIO_DIR}\Output\UNLOADED_{ALT}{YEAR}.NET"
6   PROCESS  PHASE=NODEMERGE
7     IF (DETAIL=1 & N <={ZONES})SA_Centroid=1
8   ENDPROCESS
9
10  PROCESS  PHASE=LINKMERGE
11    LOOKUP NAME=NODEDETAIL, LOOKUP[1]=1, RESULT=2,
12           FAIL[1]=0, FAIL[2]=0, FAIL[3]=0,
13           LOOKUPI=1
14    _ADETAIL=NODEDETAIL(1,A.N)
15    _BDETAIL=NODEDETAIL(1,B.N)
16    ; IF EITHER THE ANODE OR BNODE IS A KEEPER THEN KEEP
17    _KEEP=MAX(_ADETAIL,_BDETAIL)
18    IF (_KEEP=0) DELETE
19
20  ;  IF (DETAILNET=0) delete
21  ENDPROCESS
22
23  ENDRUN
24
```

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2  RUN PGM=NETWORK PRNFILE=
   "C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\SDNET00E.PRN" MSG='Create Subarea
   Network'
3  FILEI LINKI[3] = "{SCENARIO_DIR}\Output\Temp\DetailNonCentroid.NET"
4
5  FILEI LINKI[1] = "{SCENARIO_DIR}\Output\SATEM_HNET_{YEAR}.NET"
6  FILEO NETO = "{SCENARIO_DIR}\Output\SA_HNET_{YEAR}.NET"
7
8  merge MAX=SA_Centroid
9
10 PROCESS  PHASE=LINKMERGE
11
12 ENDPROCESS
13
14 ENDRUN
15
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2
3   else
4
```

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2
3  ;*if exist {SCENARIO_DIR}\Output\TEMP\ZDISTRICTS_TEM_XX.DBF copy
   {SCENARIO_DIR}\Output\TEMP\ZDISTRICTS_TEM_XX.DBF
   {SCENARIO_DIR}\Output\ZDISTRICTS_XX.DBF
4  ;*if exist {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET copy
   {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET
   {SCENARIO_DIR}\Output\SA_HNET_{Year}.NET
5  ;*if exist {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET copy
   {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET
   {SCENARIO_DIR}\Output\S6HNET_{Year}.NET
6  ENDIF
7
8
9
```

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2  RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\Cube\DTMAT00W.PRN"
    MSG='Aggregate the SubArea Trip Tables for Assignment'
3  FILEI MATI[2] = "{SCENARIO_DIR}\Output\MODEIN.TEM"
4  ;DISTRIBUTEINTRASTEP PROCESSID='CFRPMdist',
   PROCESSLIST=2-%NUMBER_OF_PROCESSORS%,MinGroupSize=20,SavePrn=F
5  ;DISTRIBUTEINTRASTEP ProcessID='CFRPMdist',ProcessList=1-4
6  FILEI ZDATI[1] = "{SCENARIO_DIR}\Output\Temp\ZDISTRICTS_TEM_{MPO_DIST}.DBF"
7  FILEO MATO[1] = "{SCENARIO_DIR}\Output\Temp\SAHTTAB_TEM.MAT",
8   MO=1-4, NAME=M1,M2,M3,M4 DEC=2*S
9
10
11 PAR ZONEMSG=100
12
13 FILLMW MW[1]=MI.2.1,2,3,4
14
15 RENUMBER ZONEO=ZI.1.DTAZ MISSINGZI=W MISSINGZO=W ZONES={ZONES}
16
17
18 ENDRUN
19
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\CUBE\MCMAT00B.PRN"
     MSG='Split daily trip table into PK/OP for TPPMS'
3   FILEI MATI[2] = "{SCENARIO_DIR}\Output\Temp\SAHTTAB_TEM.MAT"
4
5   DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
6   ;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4
7
8   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\output\A1DECK.TEM"
9   FILEO MATO[2] = "{SCENARIO_DIR}\OUTPUT\MODEIN_OP.TEM",
10  MO=4-10, name=HBW0,HBW1,HBW2,HBO0,HBO1,HBO2,NHB, Format=tranplan
11  FILEO MATO[1] = "{SCENARIO_DIR}\OUTPUT\MODEIN_PK.TEM",
12  MO=4-10, name=HBW0,HBW1,HBW2,HBO0,HBO1,HBO2,NHB,format=tranplan
13  zonemsg=100
14
15  LOOKUP NAME=A1DECK,LOOKUP[1]=1,RESULT=5, ;HBW0
16                     LOOKUP[2]=1,RESULT=6, ;HBW1
17                     LOOKUP[3]=1,RESULT=7, ;HBW2
18                     LOOKUP[4]=1,RESULT=8, ;NWK0
19                     LOOKUP[5]=1,RESULT=9, ;NWK1
20                     LOOKUP[6]=1,RESULT=10, ;NWK2
21                     FAIL=0,0,0,
22                     LOOKUPI=1
23
24  FILLMW MW[1]=MI.2.1,2,3
25
26  MW[4]=MW[1]*A1DECK(1,I)*0.5*0.01
27  MW[5]=MW[1]*A1DECK(2,I)*0.5*0.01
28  MW[6]=MW[1]*(100-A1DECK(1,I)-A1DECK(2,I))*0.5*0.01
29  MW[7]=MW[2]*A1DECK(4,I)*0.5*0.01
30  MW[8]=MW[2]*A1DECK(5,I)*0.5*0.01
31  MW[9]=MW[2]*(100-A1DECK(4,I)-A1DECK(5,I))*0.5*0.01
32  MW[10]=MW[3]*0.5
33
34  ENDRUN
35
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="C:\FSUTMS\D5\Latest\CFRPMv60_08152016\CUBE\HSHWY00A.PRN" MSG='Final Highway Assignment'
DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4

FILEI LOOKUPI[1] = "{SCENARIO_DIR}\Output\HNET_XY.DBF"
FILEI MATI[1] = "{SCENARIO_DIR}\output\HTTAB.{ALT}{YEAR}"

FILEI NETI = "{SCENARIO_DIR}\Output\SA_HNET_{Year}.NET"

FILEI TURNPENI = "{SCENARIO_DIR}\Input\TURN_{YEAR}{ALT}.PEN"

FILEO PRINTO[1] = "{CATALOG_DIR}\CUBE\TOLLFM.PRN"
FILEO NETO = "{SCENARIO_DIR}\Output\HWYLOAD_{ALT}{Year}_TEM.NET"
FILEO TURNVOLO[1] = "{SCENARIO_DIR}\Output\LOADTURN.DBF",
 FORMAT=DBF


ARRAY TOLLVOL={ITER}, TOLLREVENUE={ITER}, TOLLVMT={ITER}
PAR MAXITERS={ITER}, RELATIVEGAP=0,  GAP=0.001, AAD=0, RAAD=0, RMSE=0
;RELATIVEGAP=0.07    GAP=0.0005,
TURNS N=1-99999
ZONEMSG=100
CTOLL={CTOLL}

; look up deceleration rate based on approach speed
LOOKUP,
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = '30 4',
    '70 6.2'

LOOKUP LOOKUPI=1,
       NAME=DUMMYZONE, ;COUNTY LOOKUP
         LOOKUP[1]=N, RESULT=DUMMY,
         FAIL[1]=0,FAIL[2]=0,FAIL[3]=0,INTERPOLATE=N, LIST=N

FUNCTION  TC[1] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll links
FUNCTION  TC[2] = ({CTOLL} * LI.CARTOLL * 60); + LI.SVCMINUTES + (LI.SVCSECONDS / 60)             ;(MINS) time for toll links
FUNCTION  TC[3] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll acceleration links
FUNCTION  TC[4] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll deceleration links
FUNCTION  TC[5] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll TRUCK ONLY links

FUNCTION  COST[1] = TIME

;(MINS) congested COST for non-toll links
FUNCTION  COST[2] = TIME + ROUND((1 / (LW.SERVR - LW.ARRIVR)) * 60 * 1000) / 1000                  ;(MINS) congested COST for toll links
FUNCTION  COST[3] = TIME + ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000           ;(MINS) congested COST toll acceleration links MINS
FUNCTION  COST[4] = TIME + ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links MINS
FUNCTION  COST[5] = TIME
```

```
    ;(MINS) congested COST for non-toll TRUCK ONLY links
48
49  PROCESS PHASE=LINKREAD
50    ; basics
51    T0 = LI.TOTALCOST
52    C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR / LI.CONFAC
53    IF (LI.TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (LI.TIME / 60)   ; CONVERT TIME
    (MINS) TO HOURS
54
55    IF (LI.FACILITY=49)    ADDTOGROUP=1   ; TRUCK RESTRICTED LOCAL ROADS
56    IF (LI.FACILITY=80-89) ADDTOGROUP=2   ; HOV LANES
57    IF (LI.FACILITY=68-69)    ADDTOGROUP=3   ; Transit-only links
58
59    ; classify links based on presence/absence of tolls
60    LINKCLASS=1 ; no toll
61
62    IF (LI.TOLL > 0)
63      LINKCLASS = 2 ; with toll
64      T0 = LI.LINKCOST
65
66
67      IF (LI.TOLLTYPE = 1)
68        LW.DELAY  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)   ; CONVERT SECONDS TO
    (MINS)
69        LW.SERVR  = 1 / (LW.DELAY / 60)                    ; CONVERT (MINS) DELAY
    TO HOURS AND INVERT (VPH PER LANE)
70        LW.ARRIVR = 0
71      ELSE
72        LW.DELAY  = 0
73        LW.SERVR  = 1000000                                ; PREVENT DIVIDE BY
    ZERO FOR RAMP BOOTHS
74        LW.ARRIVR = 0
75      ENDIF
76
77    ENDIF
78
79    IF (LI.TOLL_ACC > 0)
80      LINKCLASS=3 ; Toll Plaza Acceleration link
81      LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000
              ; (MINS) congested COST toll acceleration links
82    ENDIF
83
84    IF (LI.TOLL_DEC > 0)
85      LINKCLASS=4 ; Toll Plaza Deceleration link
86      LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000)
    / 1000 ; (MINS) congested COST toll deceleration links
87    ENDIF
88
89    IF (LI.TOLL > MAXPLZNO) MAXPLZNO = LI.TOLL
90
91    IF (LI.FACILITY = 76) LINKCLASS=5
92
93  ENDPHASE
94
95  ;=============================================================================
    =========
96  ;=== ILOOP (ASSIGNMENT) PHASE ===
97  ;=============================================================================
    =========
98  PHASE=ILOOP
99
100 ; NORMAL MODE
```

```
101       if (DUMMYZONE(1,I)=0)
102         ;  PATHLOAD PATH=COST, VOL[1]=MI.1.1+MI.1.3+MI.1.4, PENI=1-3,
      EXCLUDEGROUP=2-3  ;,  patho=1,name='lovassignment',allj=f,includecosts=t
103         ;  PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1-3, EXCLUDEGROUP=3    ;,
      patho=1,name='hovassignment',allj=f,includecosts=t
104           PATHLOAD PATH=COST, VOL[3]=MI.1.3, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
105           PATHLOAD PATH=COST, VOL[4]=MI.1.4, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
106           PATHLOAD PATH=COST, VOL[1]=MI.1.1, PENI=1-3, EXCLUDEGROUP=2-3
      ;+MI.1.3+MI.1.4
107           PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1-3, EXCLUDEGROUP=3
108       endif
109
110   ; DYNASIM GENERATOR MODE
111   ;  PATHLOAD PATH=COST, VOL[1]=MI.1.1,VOL[2]=MI.1.2,VOL[3]=MI.1.3, PENI=1,
      patho=1,name='Assignment',allj=f,includecosts=f
112
113   ENDPHASE
114
115   ;================================================================================
      =========
116   ;=== ADJUST PHASE (WITH REPORTING)===
117   ;================================================================================
      =========
118   ; User Specified Functions
119   ; Delay functions, one for congested travel time, the other for toll links
      convert to time equivalent
120   PHASE=ADJUST
121
122
123     IF (TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (TIME / 60)                ;
      (MPH) CONVERT TIME (MINS) TO HRS
124
125     IF (LI.CARTOLL > 0)
126       IF (LI.TOLL > _TOLLLINKS) _TOLLLINKS = LI.TOLL
127
128       IF (LI.TOLLTYPE = 1)
129         LW.ARRIVR = ROUND((V / LI.UROADFACTOR) * LI.CONFAC / LI.PLZALNSMin) ;
      (HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
130         LW.SERVT  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)                  ;
      (MINS) Plaza lane service time in minutes per vehicle
131         LW.SERVR  = 1 / (LW.SERVT / 60)                                   ;
      (HRS)  Plaza lane service rate in vehicle per hour
132
133         IF ( (LW.ARRIVR >= LW.SERVR) && (LW.SERVR > 1000) )
134           LW.ARRIVR = 0.99 * LW.SERVR                                     ;
      (HRS)  Prevent infinite or negative queue
135         ELSEIF (LW.ARRIVR >= LW.SERVR)
136           LW.ARRIVR = 0.95 * LW.SERVR                                     ;
      (HRS)  Prevent infinite or negative queue
137         ENDIF
138
139         IF (LW.SERVR = 0) LW.SERVR = 1000                                 ;
      PREVENT DIVIDE BY ZERO
140
141         LW.DELAY = ROUND(1 / ((LW.SERVR - LW.ARRIVR) / 60) * 1000) / 1000   ;
      CONVERT HOURLY RATES TO (MINS) OF DELAY
142       ENDIF
143
144       _REV  = (ROUND(V * LI.PCTTRUCKS) * 3 * LI.CARTOLL) + (ROUND(V * (1 -
      LI.PCTTRUCKS)) * LI.CARTOLL)
```

```
145       _UTIL = (V / (LW.SERVR / (LI.UROADFACTOR * LI.CONFAC)))
146
147       _SVCSECONDS = '00' + LTRIM(STR(LI.SVCSECONDS,2,0))
148       _SVCSECLEN  = STRLEN(_SVCSECONDS)-1
149       _SVCSECONDS = SUBSTR(_SVCSECONDS,_SVCSECLEN,2)
150
151       PRINT LIST = LI.TOLL(7.0), LI.TOLLTYPE(5.0), LI.A(6.0), LI.B(6.0), " ",
152                    LI.PLAZADESC(30C), LI.PLZALNSMIN(4.0), "   $", LI.CARTOLL(5.2),
153                    LI.SVCMINUTES(6.0), ":", _SVCSECONDS(2.0C), "  ",
    (LI.PCTTRUCKS * 100)(7.2),
154                    V(10.0), LW.DELAY(6.2), LI.LINKCOST(6.2), " $", _REV(8.0C),
    _UTIL(7.3), LW.ARRIVR(7.2), LW.SERVR(7.2) printo=1
155
156       TOLLVOL[ITERATION] = TOLLVOL[ITERATION] + V
157       TOLLREVENUE[ITERATION] = TOLLREVENUE[ITERATION] + _REV
158     ENDIF
159
160     IF (LI.TOLL_ACC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2
    * 1000) / 1000           ;(MINS) congested COST toll acceleration links
161     IF (LI.TOLL_DEC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED))
    / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links
162
163     IF (LI.FACILITY=90-99) TOLLVMT[ITERATION] = TOLLVMT[ITERATION] + (LI.DISTANCE
    * V)
164
165     IF (LINKNO=1)
166       PRINT LIST = '\n','\n',"                          ITERATION ",ITERATION(2.0),
    "  SUMMARY REPORT ON TOLL LINKS -- CTOLL VALUE =",ctoll(6.3), printo=1
167       PRINT LIST = '\n','\n',
    "                                                    No.
    ServTim  %Heavy            Delay Toll    Plan", printo=1
168       PRINT LIST =           "  Class Type  ...Nodes..    ......... Name
    .........    Lanes   Toll   (mm:ss)  Trucks    Volume  (Min) (Min)   Rev.
    Util.", printo=1
169       PRINT LIST = '\n','\n'
170     ENDIF
171
172     IF (LINKNO=NUMLINKS)
173       PRINT LIST = '\n','\n',"
    *---------------------*************************** TOTAL VOLUME FOR THIS
    ITERATION IS     ",TOLLVOL[ITERATION](10.0C), printo=1
174       PRINT LIST =           "     *-TOLL REVENUE SUMMARY-**********************
    TOTAL TOLL REVENUE FOR THIS ITERATION IS    $",TOLLREVENUE[ITERATION](10.0C),
    printo=1
175       PRINT LIST =           "
    *---------------------**********************  TOLL FACILITY VMT FOR THIS
    ITERATION IS     ",TOLLVMT[ITERATION](10.0C), printo=1
176       PRINT LIST =           "     *----------------------*****************
    AVERAGE REVENUE PER VMT FOR THIS ITERATION IS    $",(TOLLREVENUE[ITERATION] /
    TOLLVMT[ITERATION])(14.3C), printo=1
177     ENDIF
178
179   ENDPHASE
180
181
182   ENDRUN
183
```

B. SubArea Application Scripts - TOD Model

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2  RUN PGM=NETWORK PRNFILE=
   "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SANET00A.PRN" MSG=
   'Extract original/NonSANet w centroid NET'
3  FILEO LINKO = "{SCENARIO_DIR}\OUTPUT\TEMP\OrgNetCentroidTime.TEM",
4   include=a,b,TimeOrg
5  FILEO NETO = "{SCENARIO_DIR}\OUTPUT\TEMP\OrgNetCentroidTime.NET"
6  FILEI LINKI[1] = "{SCENARIO_DIR}\OUTPUT\UNLOADED_{ALT}{YEAR}.NET"
7  PROCESS  PHASE=LINKMERGE
8    if ((a >5406)& (b >5406)) delete
9    TimeOrg=Time
10 ENDPROCESS
11 ENDRUN
12
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=MATRIX PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SAMAT00A.PRN" MSG=
    'Develop TIME Skim to creat TAZ Dist File'
3   FILEO MATO[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\CONGSKIMUP_TEM.MAT",
4   mo=1,NAME=TIME
5   FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\CONGSKIMUP.MAT"
6
7   PAR ZONEMSG=100
8   MW[1]=mi.1.1*1      ; TIME With Terminal Time (in Minutes) - Step Not really
    needed as TT is already in Minutes. Done for consistency with SERPM
9   MW[1][I]=1000000    ; QUICKER (WW)
10
11  ENDRUN
12
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=MATRIX PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SAMAT00C.PRN" MSG=
    'Write Regional TAZ and MPO Zonal File'
3   FILEO RECO[1] = "{SCENARIO_DIR}\OUTPUT\ZDMPOs.DBF",
4   form=10.0, FIELDS=N,MPO
5   PAR ZONES={ZONES}
6   PAR ZONEMSG=100
7   FILEI ZDATI[1] = "{SCENARIO_DIR}\INPUT\ZDATA1_{YEAR}{ALT}.DBF"
8
9
10  ;Get Zonal TAZ Info as Node Record...
11  N=ZI.1.TAZ_REG
12  MPO=ZI.1.MPO
13
14  _ztemp=_ztemp+1
15
16  ;Set Data for Indian River  and Polk Internal and Dummy Zones
17
18  if(z>=4601 & z<={ZONESI})
19     N=z
20
21      MPO=0
22
23  endif
24
25  ;Set Data for MetroPlan Orladndo Internal and Dummy Zones
26  if(z>=1 & z<=1400)
27    N=z
28
29      MPO=1
30
31  endif
32
33  ;Set Data for Volusia Internal and Dummy Zones
34
35  if(z>=1801 & z<=2900)
36    N=z
37
38      MPO=2
39
40  endif
41
42  ;Set Data for Flagler Internal and Dummy Zones
43
44  if(z>=4401 & z<=4600)
45    N=z
46
47      MPO=2
48  ENDIF
49
50  ;Set Data for Space Coast Internal and Dummy Zones
51
52  if(z>=2901 & z<=3700)
53     N=z
54
55      MPO=3
56
57  endif
58
59  ;Set Data for Ocala Marion Internal and Dummy Zones
```

```
60
61   if(z>=3701 & z<=4200)
62     N=z
63
64       MPO=4
65
66   endif
67
68   ;Set Data for Lake Internal and Dummy Zones
69
70   if(z>=1401 & z<=1800)
71      N=z
72
73       MPO=5
74
75   endif
76
77
78   ;Set Data for Sumter Internal and Dummy Zones
79
80   if(z>=4201 & z<=4400)
81      N=z
82
83       MPO=5
84
85   ENDIF
86
87   WRITE RECO=1
88
89   ;External Zones
90   if(z={ZONESI})
91     loop jj={ZONESI}+1,{ZONES}
92       N=jj
93
94       if (N>=5351 & N<=5357)          ;Indian River Polk Externals
95        MPO=0
96       ENDIF
97
98        if (N>=5361 & N<=5377)          ;Polk Externals
99         MPO=0
100       ENDIF
101
102       if (N>=5358 & N<=5360)          ;MetroPlan Orlando Externals
103         MPO=1
104      endif
105
106      if (N>=5401 & N<=5406)          ;River to Sea Externals
107        MPO=2
108      endif
109
110      if (N>=5384 & N<=5400)          ;Ocala Marion Externals
111         MPO=4
112      endif
113
114      if (N>=5378 & N<=5383)          ;Lake Sumter Externals
115        MPO=5
116      endif
117
118
119      WRITE RECO=1
120
121    endloop
```

```
122    ENDIF
123    ENDRUN
124
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SAMAT00D.PRN" MSG='Generate SubArea District Definitions'
FILEO RECO[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\ZDISTRICTS_TEM_{MPO_DIST}.DBF",
    Fields= TAZ,DTAZ,NEAR,SADIST,MPO
FILEI LOOKUPI[2] = "{SCENARIO_DIR}\INPUT\SADists_{MPO_DIST}.DBF"
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\INPUT\DLEVEL_{MPO_DIST}.DBF"
FILEI ZDATI[1] = "{SCENARIO_DIR}\OUTPUT\ZDMPOs.DBF",
   Z=N
FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\CONGSKIMUP_TEM.MAT"

LOOKUP LOOKUPI=2,  ; One record per District
       NAME=DCENTROID,
         LOOKUP[1]=CENTERTAZ, RESULT=SADIST,
         LOOKUP[2]=CENTERTAZ, RESULT=CENTERTAZ,
         LOOKUP[3]=CENTERTAZ, RESULT=MPOCODE,
         LOOKUP[4]=CENTERTAZ, RESULT=SADST_TW,
       FAIL=0,0,0

LOOKUP LOOKUPI=1,  ; One record per MPO
       NAME=LEVEL,
         LOOKUP[1]=MPO, RESULT=DLEVEL, ; 1=District Level, 0=TAZ Level
       FAIL=0,0,0

Parameters ZONES={ZONES}
PAR ZONEMSG=100


near=1000
TAZ=i
MPO=zi.1.MPO
mylevel=LEVEL(1,MPO)
IF(i>{ZONESI}) MPO=99
  jloop
   place=DCENTROID(2,j)
   thisdist=DCENTROID(1,j)
   if(place<>0)
      if(i=place)
        near=0
        DTAZ=j
        SADIST=thisdist
      else
        ctime=mi.1.time[j]
        if(ctime < near)
           near=ctime
           DTAZ=j
           SADIST=thisdist
        endif
      endif
   endif
  endjloop
if(mylevel=0) DTAZ=i ; At the TAZ level
if(near<1000 & MPO>=0) WRITE RECO=1 ;added = in MPO>0 for Polk and Indian River  zones 4601-5350

ENDRUN

```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SANET00D.PRN" MSG=
    'Create 1st Pass Temporary SubArea Network'
3   FILEO NETO = "{SCENARIO_DIR}\OUTPUT\SATEM_HNET_{YEAR}.NET"
4   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\ZDISTRICTS_TEM_{MPO_DIST}.DBF"
5   FILEI LINKI[1] = "{SCENARIO_DIR}\OUTPUT\UNLOADED_{ALT}{YEAR}.NET"
6
7   array gone=99999
8
9   PROCESS  PHASE=NODEMERGE
10    LOOKUP LOOKUPI=1,
11         NAME=SADIST,
12           LOOKUP[1]=TAZ, RESULT=DTAZ,
13         FAIL[3]=0
14    if(N<={ZONES}) _NN=_NN+1
15    if(N<={ZONES}) SA_Centroid=1
16    IF((N<>SADIST(1,N)) & (N<={ZONESI}))
17
18    ;| (NODETYPE=3,4))   ; for SERPM Only
19       gone[N]=1
20      _dd=_dd+1
21    delete
22    endif
23  ENDPROCESS
24
25  PROCESS  PHASE=LINKMERGE
26    _AD=gone[a]
27    _BD=gone[b]
28    if(_AD>0 | _BD>0)
29      print list= A(5.0),B(5.0),' deleted'
30      delete
31    endif
32  ENDPROCESS
33
34  PROCESS  PHASE=SUMMARY
35    _LL=_NN-_dd
36    print list=' ******* Deleted ',_dd(4.0),' nodes out of a total of ',_NN(4.0),
    ' leaving ',_LL(4.0),' active centroids'
37  ENDPROCESS
38
39
40
41
42  ENDRUN
43
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   IF ({UserDefinedSubarea}=1)   ; User defined Subarea
3
4
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SANET00B.PRN" MSG=
    'Extract Detail Centroid Selected Attributes'
3   FILEO PRINTO[1] = "{SCENARIO_DIR}\OUTPUT\NODEDETAIL.CSV"
4   FILEO NODEO = "{SCENARIO_DIR}\OUTPUT\TEMP\DETAILTAZCEN.DBF",
5     INCLUDE=TAZ,DTAZ,NEAR,SADIST,MPO,DETAIL
6   FILEI LINKI[1] = "{SCENARIO_DIR}\OUTPUT\UNLOADED_{ALT}{YEAR}.NET"
7
8   PROCESS  PHASE=NODEMERGE
9   ; ---------------------------------------------------------------
10  ; NEXT STEP IS TO CAPTURE THE DETAIL ATTRIBUTE ON THE NODES TO
11  ; SAVE THE USER NEEDING TO COMPUTE IT FOR LINKS TOO
12    PRINT CSV=T, LIST=N(6.0),DETAIL(2.0), PRINTO=1
13  ; ---------------------------------------------------------------
14    IF (DETAIL=1 & N <={ZONES})
15      TAZ=NI.1.N
16      DTAZ=NI.1.N
17      NEAR=0
18      SADIST=5000+TAZ
19      MPO=NI.1.MPO
20    ELSE
21      DELETE
22    ENDIF
23  ENDPROCESS
24
25  ENDRUN
26
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=MATRIX PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SAMAT00B.PRN" MSG=
    'Finalize TAZ-DISTRICT Equivalency File'
3   FILEO RECO[1] = "{SCENARIO_DIR}\OUTPUT\ZDISTRICTS_{MPO_DIST}.DBF",
4     FIELDS=TAZ,DTAZ,NEAR,SADIST,MPO,DETAIL
5   FILEI ZDATI[2] = "{SCENARIO_DIR}\OUTPUT\TEMP\ZDISTRICTS_TEM_{MPO_DIST}.DBF",
6     Z=TAZ
7   FILEI ZDATI[1] = "{SCENARIO_DIR}\OUTPUT\TEMP\DETAILTAZCEN.DBF",
8     z=taz
9
10  PAR ZONES={ZONES}
11  PAR ZONEMSG=100
12
13  ;xxxxxx
14  TAZ=ZI.2.TAZ
15  DETAIL=ZI.1.DETAIL
16
17  IF (DETAIL=1)
18    DTAZ=ZI.1.DTAZ
19    NEAR=ZI.1.NEAR
20    SADIST=ZI.1.SADIST
21    COUNTY=ZI.1.MPO
22    DETAIL=ZI.1.DETAIL
23   Else
24    DTAZ=ZI.2.DTAZ
25    NEAR=ZI.2.NEAR
26    SADIST=ZI.2.SADIST
27    COUNTY=ZI.2.MPO
28    DETAIL=0
29  ENDIF
30
31  WRITE RECO=1
32  ENDRUN
33
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SANET00C.PRN" MSG=
    'Extract Detail Portion of Newtowrk and its attributes'
3   FILEO NETO = "{SCENARIO_DIR}\OUTPUT\TEMP\DetailNonCentroid.NET"
4   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\NODEDETAIL.CSV"
5   FILEI LINKI[1] = "{SCENARIO_DIR}\OUTPUT\UNLOADED_{ALT}{YEAR}.NET"
6
7   PROCESS  PHASE=NODEMERGE
8     IF (DETAIL=1 & N <={ZONES})SA_Centroid=1
9   ENDPROCESS
10
11  PROCESS  PHASE=LINKMERGE
12    LOOKUP NAME=NODEDETAIL, LOOKUP[1]=1, RESULT=2,
13           FAIL[1]=0, FAIL[2]=0, FAIL[3]=0,
14           LOOKUPI=1
15    _ADETAIL=NODEDETAIL(1,A.N)
16    _BDETAIL=NODEDETAIL(1,B.N)
17    ; IF EITHER THE ANODE OR BNODE IS A KEEPER THEN KEEP
18    _KEEP=MAX(_ADETAIL,_BDETAIL)
19    IF (_KEEP=0) DELETE
20
21  ;  IF (DETAILNET=0) delete
22  ENDPROCESS
23
24  ENDRUN
25
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=NETWORK PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\SANET00E.PRN" MSG=
    'Create Subarea Network'
3   FILEO NETO = "{SCENARIO_DIR}\OUTPUT\SA_HNET_{YEAR}.NET"
4   FILEI LINKI[2] = "{SCENARIO_DIR}\OUTPUT\TEMP\DetailNonCentroid.NET"
5   FILEI LINKI[1] = "{SCENARIO_DIR}\OUTPUT\SATEM_HNET_{YEAR}.NET"
6
7   merge MAX=SA_Centroid
8
9   PROCESS  PHASE=LINKMERGE
10
11  ENDPROCESS
12
13  ENDRUN
14
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2
3   else
4
```

```
1  ; Do not change filenames or add or remove FILEI/FILEO statements using an
   editor. Use Cube/Application Manager.
2
3  ;*if exist {SCENARIO_DIR}\Output\TEMP\ZDISTRICTS_TEM_XX.DBF copy
   {SCENARIO_DIR}\Output\TEMP\ZDISTRICTS_TEM_XX.DBF
   {SCENARIO_DIR}\Output\ZDISTRICTS_XX.DBF
4  ;*if exist {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET copy
   {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET
   {SCENARIO_DIR}\Output\SA_HNET_{Year}.NET
5  ;*if exist {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET copy
   {SCENARIO_DIR}\Output\SATEM_HNET_{Year}.NET
   {SCENARIO_DIR}\Output\S6HNET_{Year}.NET
6
7  endif
8
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=MATRIX PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\CUBE\TOD\DTMAT00Y.PRN"
3   FILEO MATO[1] = "{SCENARIO_DIR}\output\temp\SAHTTAB_TEM.MAT",
4    MO=1-8, NAME=HBW_PK,HBNW_PK,NHB_PK,TTOT_PK,HBW_OP,HBNW_OP,NHB_OP,TTOT_OP,
    FORMAT=TRANPLAN
5   FILEI ZDATI[1] = "{SCENARIO_DIR}\output\temp\ZDISTRICTS_TEM_{MPO_DIST}.DBF"
6   FILEI MATI[1] = "{SCENARIO_DIR}\output\MODEIN.TEM"
7
8
9   PAR ZONEMSG=100
10
11  FILLMW MW[1]=MI.1.1,2,3,4,5,6,7,8
12
13  RENUMBER ZONEO=ZI.1.DTAZ MISSINGZI=W MISSINGZO=W ZONES={ZONES}
14
15
16  ENDRUN
17
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an
editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE=
"C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\Cube\TOD\MCMAT00B.PRN" MSG=
'Split daily trip table into PK/OP for TPPMS'
FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\temp\SAHTTAB_TEM.MAT"
DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4

FILEI LOOKUPI[1] = "{SCENARIO_DIR}\output\A1DECK.TEM"
FILEO MATO[2] = "{SCENARIO_DIR}\OUTPUT\MODEIN_OP.TEM",
MO=21-27, name=HBW0,HBW1,HBW2,HBO0,HBO1,HBO2,NHB, Format=tranplan
FILEO MATO[1] = "{SCENARIO_DIR}\OUTPUT\MODEIN_PK.TEM",
MO=11-17, name=HBW0,HBW1,HBW2,HBO0,HBO1,HBO2,NHB,format=tranplan
zonemsg=100

LOOKUP NAME=A1DECK,LOOKUP[1]=1,RESULT=5, ;HBW0
                   LOOKUP[2]=1,RESULT=6, ;HBW1
                   LOOKUP[3]=1,RESULT=7, ;HBW2
                   LOOKUP[4]=1,RESULT=8, ;NWK0
                   LOOKUP[5]=1,RESULT=9, ;NWK1
                   LOOKUP[6]=1,RESULT=10,;NWK2
                   FAIL=0,0,0,
                   LOOKUPI=1

FILLMW MW[1]=MI.1.1,2,3  ; PEAK
FILLMW MW[4]=MI.1.5,6,7  ; OFF-PEAK
; PEAK
MW[11]=MW[1]*A1DECK(1,I)*0.01
MW[12]=MW[1]*A1DECK(2,I)*0.01
MW[13]=MW[1]*(100-A1DECK(1,I)-A1DECK(2,I))*0.01
MW[14]=MW[2]*A1DECK(4,I)*0.01
MW[15]=MW[2]*A1DECK(5,I)*0.01
MW[16]=MW[2]*(100-A1DECK(4,I)-A1DECK(5,I))*0.01
MW[17]=MW[3]
; OFF-PEAK
MW[21]=MW[4]*A1DECK(1,I)*0.01
MW[22]=MW[4]*A1DECK(2,I)*0.01
MW[23]=MW[4]*(100-A1DECK(1,I)-A1DECK(2,I))*0.01
MW[24]=MW[5]*A1DECK(4,I)*0.01
MW[25]=MW[5]*A1DECK(5,I)*0.01
MW[26]=MW[5]*(100-A1DECK(4,I)-A1DECK(5,I))*0.01
MW[27]=MW[6]

ENDRUN

```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\Cube\TOD\TSHWY00K.PRN" MSG='EE Peak Highway Assignment'
FILEO PRINTO[1] = "{SCENARIO_DIR}\Output\TSHWY00D.PRN"
FILEO TURNVOLO[1] = "{SCENARIO_DIR}\Output\TSHWY00A.TRN",
FORMAT=DBF

FILEO NETO = "{SCENARIO_DIR}\OUTPUT\HWYLOAD_AM_{ALT}{YEAR}_EE.NET"
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\HNET_XY.DBF"
FILEI TURNPENI = "{SCENARIO_DIR}\INPUT\TURNTOD_{YEAR}{ALT}.PEN"
FILEI NETI = "{SCENARIO_DIR}\OUTPUT\SA_HNET_{YEAR}.NET"
FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\EE_TOD_{ALT}{Year}.MAT"
DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4



ARRAY TOLLVOL=1, TOLLREVENUE=1, TOLLVMT=1
PAR MAXITERS=1, RELATIVEGAP=0,  GAP=0.001, AAD=0, RAAD=0, RMSE=0
;RELATIVEGAP=0.07    GAP=0.0005,
PAR COMBINE=EQUI, ENHANCE=2 ; KDK
TURNS N=1-99999
ZONEMSG=100
CTOLL=.06 ;{CTOLL}

; look up deceleration rate based on approach speed
LOOKUP,
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = '30 4',
    '70 6.2'

LOOKUP LOOKUPI=1,
       NAME=DUMMYZONE, ;COUNTY LOOKUP
         LOOKUP[1]=N, RESULT=DUMMY,
         FAIL[1]=0,FAIL[2]=0,FAIL[3]=0,INTERPOLATE=N, LIST=N

FUNCTION  TC[1] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll links
FUNCTION  TC[2] = ( .005 * LI.CARTOLL * 60); + LI.SVCMINUTES + (LI.SVCSECONDS / 60)             ;(MINS) time for toll links
FUNCTION  TC[3] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll acceleration links
FUNCTION  TC[4] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll deceleration links
FUNCTION  TC[5] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll TRUCK ONLY links

FUNCTION  COST[1] = TIME

;(MINS) congested COST for non-toll links
FUNCTION  COST[2] = TIME + ROUND((1 / (LW.SERVR - LW.ARRIVR)) * 60 * 1000) / 1000                  ;(MINS) congested COST for toll links
FUNCTION  COST[3] = TIME + ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000           ;(MINS) congested COST toll acceleration links MINS
FUNCTION  COST[4] = TIME + ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links MINS
FUNCTION  COST[5] = TIME

;(MINS) congested COST for non-toll TRUCK ONLY links
```

```
47
48  PROCESS PHASE=LINKREAD
49
50     TODCONFAC={AMCAPFAC} ; different for each period
51
52    ; basics
53    T0 = LI.TOTALCOST
54    ;C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR / LI.CONFAC
55    C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR * TODCONFAC
56    IF (LI.TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (LI.TIME / 60)   ; CONVERT TIME
    (MINS) TO HOURS
57
58    IF (LI.FACILITY=49)    ADDTOGROUP=1   ; TRUCK RESTRICTED LOCAL ROADS
59    IF (LI.FACILITY=80-89) ADDTOGROUP=2   ; HOV LANES
60    IF (LI.FACILITY=68-69)    ADDTOGROUP=3   ; Transit-only links
61
62    ; classify links based on presence/absence of tolls
63    LINKCLASS=1 ; no toll
64
65    IF (LI.TOLL > 0)
66      LINKCLASS = 2 ; with toll
67      T0 = LI.LINKCOST
68
69
70      IF (LI.TOLLTYPE = 1)
71        LW.DELAY  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)   ; CONVERT SECONDS TO
    (MINS)
72        LW.SERVR  = 1 / (LW.DELAY / 60)                    ; CONVERT (MINS) DELAY
    TO HOURS AND INVERT (VPH PER LANE)
73        LW.ARRIVR = 0
74      ELSE
75        LW.DELAY  = 0
76        LW.SERVR  = 1000000                                ; PREVENT DIVIDE BY
    ZERO FOR RAMP BOOTHS
77        LW.ARRIVR = 0
78      ENDIF
79
80    ENDIF
81
82    IF (LI.TOLL_ACC > 0)
83      LINKCLASS=3 ; Toll Plaza Acceleration link
84      LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000
              ; (MINS) congested COST toll acceleration links
85    ENDIF
86
87    IF (LI.TOLL_DEC > 0)
88      LINKCLASS=4 ; Toll Plaza Deceleration link
89      LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000)
    / 1000 ; (MINS) congested COST toll deceleration links
90    ENDIF
91
92    IF (LI.TOLL > MAXPLZNO) MAXPLZNO = LI.TOLL
93
94    IF (LI.FACILITY = 76) LINKCLASS=5
95
96  ENDPHASE
97
98  ;=============================================================================
    =========
99  ;=== ILOOP (ASSIGNMENT) PHASE ===
100 ;=============================================================================
    =========
```

```
101   PHASE=ILOOP
102
103   ; NORMAL MODE
104       if (DUMMYZONE(1,I)=0)
105           PATHLOAD PATH=COST, VOL[1]=MI.1.LOV_AM + MI.1.LTRK_AM + MI.1.HTRK_AM,
      PENI=1-3, EXCLUDEGROUP=2-3  ;,
      patho=1,name='lovassignment',allj=f,includecosts=t
106           PATHLOAD PATH=COST, VOL[2]=MI.1.HOV_AM, PENI=1-3, EXCLUDEGROUP=3    ;,
      patho=1,name='hovassignment',allj=f,includecosts=t
107           ;PATHLOAD PATH=COST, VOL[3]=MI.1.3, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
108           ;PATHLOAD PATH=COST, VOL[4]=MI.1.4, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
109           ;PATHLOAD PATH=COST, VOL[1]=MI.1.1+MI.1.3+MI.1.4, PENI=1,
      EXCLUDEGROUP=2-3
110           ;PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1, EXCLUDEGROUP=3
111       endif
112
113   ; DYNASIM GENERATOR MODE
114   ;  PATHLOAD PATH=COST, VOL[1]=MI.1.1,VOL[2]=MI.1.2,VOL[3]=MI.1.3, PENI=1,
      patho=1,name='Assignment',allj=f,includecosts=f
115
116   ENDPHASE
117
118   ;================================================================================
      =========
119   ;=== ADJUST PHASE (WITH REPORTING)===
120   ;================================================================================
      =========
121   ; User Specified Functions
122   ; Delay functions, one for congested travel time, the other for toll links
      convert to time equivalent
123   PHASE=ADJUST
124
125
126     IF (TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (TIME / 60)                ;
      (MPH) CONVERT TIME (MINS) TO HRS
127
128     IF (LI.CARTOLL > 0)
129       IF (LI.TOLL > _TOLLLINKS) _TOLLLINKS = LI.TOLL
130
131       IF (LI.TOLLTYPE = 1)
132         ;LW.ARRIVR = ROUND((V / LI.UROADFACTOR) * LI.CONFAC / LI.PLZALNSMin) ;
      (HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
133         LW.ARRIVR = ROUND((V*TODCONFAC / LI.UROADFACTOR) / LI.PLZALNSMin) ;
      (HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
134         LW.SERVT  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)                  ;
      (MINS) Plaza lane service time in minutes per vehicle
135         LW.SERVR  = 1 / (LW.SERVT / 60)                                   ;
      (HRS)  Plaza lane service rate in vehicle per hour
136
137         IF ( (LW.ARRIVR >= LW.SERVR) && (LW.SERVR > 1000) )
138           LW.ARRIVR = 0.99 * LW.SERVR                                     ;
      (HRS)  Prevent infinite or negative queue
139         ELSEIF (LW.ARRIVR >= LW.SERVR)
140           LW.ARRIVR = 0.95 * LW.SERVR                                     ;
      (HRS)  Prevent infinite or negative queue
141         ENDIF
142
143         IF (LW.SERVR = 0) LW.SERVR = 1000                                 ;
      PREVENT DIVIDE BY ZERO
144
```

```
145         LW.DELAY = ROUND(1 / ((LW.SERVR - LW.ARRIVR) / 60) * 1000) / 1000   ;
    CONVERT HOURLY RATES TO (MINS) OF DELAY
146       ENDIF
147
148       _REV  = (ROUND(V * LI.PCTTRUCKS) * 3 * LI.CARTOLL) + (ROUND(V * (1 -
    LI.PCTTRUCKS)) * LI.CARTOLL)
149       ;_UTIL = (V / (LW.SERVR / (LI.UROADFACTOR * LI.CONFAC)))
150       _UTIL = (V / (LW.SERVR / (LI.UROADFACTOR / TODCONFAC)))
151
152       _SVCSECONDS = '00' + LTRIM(STR(LI.SVCSECONDS,2,0))
153       _SVCSECLEN  = STRLEN(_SVCSECONDS)-1
154       _SVCSECONDS = SUBSTR(_SVCSECONDS,_SVCSECLEN,2)
155
156       PRINT LIST = LI.TOLL(7.0), LI.TOLLTYPE(5.0), LI.A(6.0), LI.B(6.0), " ",
157                    LI.PLAZADESC(30C), LI.PLZALNSMIN(4.0), "   $", LI.CARTOLL(5.2),
158                    LI.SVCMINUTES(6.0), ":", _SVCSECONDS(2.0C), "  ",
    (LI.PCTTRUCKS * 100)(7.2),
159                    V(10.0), LW.DELAY(6.2), LI.LINKCOST(6.2), " $", _REV(8.0C),
    _UTIL(7.3), LW.ARRIVR(7.2), LW.SERVR(7.2) printo=1
160
161       TOLLVOL[ITERATION] = TOLLVOL[ITERATION] + V
162       TOLLREVENUE[ITERATION] = TOLLREVENUE[ITERATION] + _REV
163     ENDIF
164
165     IF (LI.TOLL_ACC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2
    * 1000) / 1000           ;(MINS) congested COST toll acceleration links
166     IF (LI.TOLL_DEC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED))
    / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links
167
168     IF (LI.FACILITY=90-99) TOLLVMT[ITERATION] = TOLLVMT[ITERATION] + (LI.DISTANCE
    * V)
169
170     IF (LINKNO=1)
171       PRINT LIST = '\n','\n'," ITERATION ",ITERATION(2.0),
    "  SUMMARY REPORT ON TOLL LINKS -- CTOLL VALUE =",ctoll(6.3), printo=1
172       PRINT LIST = '\n','\n',
    "                                                    No.
    ServTim  %Heavy           Delay Toll    Plan", printo=1
173       PRINT LIST =          "  Class Type  ...Nodes..    ......... Name
    .........    Lanes   Toll   (mm:ss)  Trucks    Volume  (Min) (Min)   Rev.
    Util.", printo=1
174       PRINT LIST = '\n','\n'
175     ENDIF
176
177     IF (LINKNO=NUMLINKS)
178       PRINT LIST = '\n','\n',"
    *---------------------*************************** TOTAL VOLUME FOR THIS
    ITERATION IS     ",TOLLVOL[ITERATION](10.0C), printo=1
179       PRINT LIST =          "     *-TOLL REVENUE SUMMARY-**********************
    TOTAL TOLL REVENUE FOR THIS ITERATION IS    $",TOLLREVENUE[ITERATION](10.0C),
    printo=1
180       PRINT LIST =          "
    *---------------------**********************  TOLL FACILITY VMT FOR THIS
    ITERATION IS     ",TOLLVMT[ITERATION](10.0C), printo=1
181       PRINT LIST =          "     *----------------------*****************
    AVERAGE REVENUE PER VMT FOR THIS ITERATION IS    $",(TOLLREVENUE[ITERATION] /
    TOLLVMT[ITERATION])(14.3C), printo=1
182     ENDIF
183
184   ENDPHASE
185
186
```

```
187    ENDRUN
188
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\Cube\TOD\TSHWY00F.PRN" MSG='EE Midday Period Highway Assignment'
FILEO PRINTO[1] = "{SCENARIO_DIR}\Output\TSHWY00H.PRN"
FILEO TURNVOLO[1] = "{SCENARIO_DIR}\Output\TSHWY00B.TRN",
FORMAT=DBF

FILEO NETO = "{SCENARIO_DIR}\OUTPUT\HWYLOAD_MD_{ALT}{YEAR}_EE.NET"
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\HNET_XY.DBF"
FILEI TURNPENI = "{SCENARIO_DIR}\INPUT\TURNTOD_{YEAR}{ALT}.PEN"
FILEI NETI = "{SCENARIO_DIR}\OUTPUT\SA_HNET_{YEAR}.NET"
FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\EE_TOD_{ALT}{YEAR}.MAT"
DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4




ARRAY TOLLVOL=1, TOLLREVENUE=1, TOLLVMT=1
PAR MAXITERS=1, RELATIVEGAP=0,  GAP=0.001, AAD=0, RAAD=0, RMSE=0
;RELATIVEGAP=0.07    GAP=0.0005,
PAR COMBINE=EQUI, ENHANCE=2 ; KDK
TURNS N=1-99999
ZONEMSG=100
CTOLL=.06 ;{CTOLL}

; look up deceleration rate based on approach speed
LOOKUP,
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = '30 4',
    '70 6.2'

LOOKUP LOOKUPI=1,
       NAME=DUMMYZONE, ;COUNTY LOOKUP
         LOOKUP[1]=N, RESULT=DUMMY,
         FAIL[1]=0,FAIL[2]=0,FAIL[3]=0,INTERPOLATE=N, LIST=N

FUNCTION  TC[1] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll links
FUNCTION  TC[2] = ( .005 * LI.CARTOLL * 60); + LI.SVCMINUTES + (LI.SVCSECONDS / 60)             ;(MINS) time for toll links
FUNCTION  TC[3] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll acceleration links
FUNCTION  TC[4] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll deceleration links
FUNCTION  TC[5] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll TRUCK ONLY links

FUNCTION  COST[1] = TIME

;(MINS) congested COST for non-toll links
FUNCTION  COST[2] = TIME + ROUND((1 / (LW.SERVR - LW.ARRIVR)) * 60 * 1000) / 1000                  ;(MINS) congested COST for toll links
FUNCTION  COST[3] = TIME + ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000           ;(MINS) congested COST toll acceleration links MINS
FUNCTION  COST[4] = TIME + ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links MINS
FUNCTION  COST[5] = TIME
```

```
    ;(MINS) congested COST for non-toll TRUCK ONLY links
49
50  PROCESS PHASE=LINKREAD
51
52     TODCONFAC={MDCAPFAC} ; different for each period
53
54    ; basics
55    T0 = LI.TOTALCOST
56    ;C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR / LI.CONFAC
57    C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR * TODCONFAC
58    IF (LI.TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (LI.TIME / 60)   ; CONVERT TIME
    (MINS) TO HOURS
59
60    IF (LI.FACILITY=49)    ADDTOGROUP=1   ; TRUCK RESTRICTED LOCAL ROADS
61    IF (LI.FACILITY=80-89) ADDTOGROUP=2   ; HOV LANES
62    IF (LI.FACILITY=68-69)    ADDTOGROUP=3   ; Transit-only links
63
64    ; classify links based on presence/absence of tolls
65    LINKCLASS=1 ; no toll
66
67    IF (LI.TOLL > 0)
68      LINKCLASS = 2 ; with toll
69      T0 = LI.LINKCOST
70
71
72      IF (LI.TOLLTYPE = 1)
73        LW.DELAY  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)   ; CONVERT SECONDS TO
    (MINS)
74        LW.SERVR  = 1 / (LW.DELAY / 60)                    ; CONVERT (MINS) DELAY
    TO HOURS AND INVERT (VPH PER LANE)
75        LW.ARRIVR = 0
76      ELSE
77        LW.DELAY  = 0
78        LW.SERVR  = 1000000                                ; PREVENT DIVIDE BY
    ZERO FOR RAMP BOOTHS
79        LW.ARRIVR = 0
80      ENDIF
81
82    ENDIF
83
84    IF (LI.TOLL_ACC > 0)
85      LINKCLASS=3 ; Toll Plaza Acceleration link
86      LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000
              ; (MINS) congested COST toll acceleration links
87    ENDIF
88
89    IF (LI.TOLL_DEC > 0)
90      LINKCLASS=4 ; Toll Plaza Deceleration link
91      LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000)
    / 1000 ; (MINS) congested COST toll deceleration links
92    ENDIF
93
94    IF (LI.TOLL > MAXPLZNO) MAXPLZNO = LI.TOLL
95
96    IF (LI.FACILITY = 76) LINKCLASS=5
97
98  ENDPHASE
99
100 ;================================================================================
    =========
101 ;=== ILOOP (ASSIGNMENT) PHASE ===
```

```
;================================================================================
=========
PHASE=ILOOP

; NORMAL MODE
    if (DUMMYZONE(1,I)=0)
        PATHLOAD PATH=COST, VOL[1]=MI.1.LOV_MID + MI.1.LTRK_MID + MI.1
.HTRK_MID, PENI=1-3, EXCLUDEGROUP=2-3  ;,
patho=1,name='lovassignment',allj=f,includecosts=t
        PATHLOAD PATH=COST, VOL[2]=MI.1.HOV_MID, PENI=1-3, EXCLUDEGROUP=3
;,  patho=1,name='hovassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[3]=MI.1.3, PENI=1-3, EXCLUDEGROUP=1-3  ;,
patho=1,name='ltkassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[4]=MI.1.4, PENI=1-3, EXCLUDEGROUP=1-3  ;,
patho=1,name='ltkassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[1]=MI.1.1+MI.1.3+MI.1.4, PENI=1,
EXCLUDEGROUP=2-3
        ;PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1, EXCLUDEGROUP=3
    endif

; DYNASIM GENERATOR MODE
;  PATHLOAD PATH=COST, VOL[1]=MI.1.1,VOL[2]=MI.1.2,VOL[3]=MI.1.3, PENI=1,
patho=1,name='Assignment',allj=f,includecosts=f

ENDPHASE

;================================================================================
=========
;=== ADJUST PHASE (WITH REPORTING)===
;================================================================================
=========
; User Specified Functions
; Delay functions, one for congested travel time, the other for toll links
convert to time equivalent
PHASE=ADJUST


  IF (TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (TIME / 60)                ;
(MPH) CONVERT TIME (MINS) TO HRS

  IF (LI.CARTOLL > 0)
    IF (LI.TOLL > _TOLLLINKS) _TOLLLINKS = LI.TOLL

    IF (LI.TOLLTYPE = 1)
      ;LW.ARRIVR = ROUND((V / LI.UROADFACTOR) * LI.CONFAC / LI.PLZALNSMin) ;
(HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
      LW.ARRIVR = ROUND((V*TODCONFAC / LI.UROADFACTOR) / LI.PLZALNSMin) ;
(HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
      LW.SERVT  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)                  ;
(MINS) Plaza lane service time in minutes per vehicle
      LW.SERVR  = 1 / (LW.SERVT / 60)                                   ;
(HRS)  Plaza lane service rate in vehicle per hour

      IF ( (LW.ARRIVR >= LW.SERVR) && (LW.SERVR > 1000) )
        LW.ARRIVR = 0.99 * LW.SERVR                                     ;
(HRS)  Prevent infinite or negative queue
      ELSEIF (LW.ARRIVR >= LW.SERVR)
        LW.ARRIVR = 0.95 * LW.SERVR                                     ;
(HRS)  Prevent infinite or negative queue
      ENDIF

      IF (LW.SERVR = 0) LW.SERVR = 1000                                 ;
```

```
    PREVENT DIVIDE BY ZERO
146
147         LW.DELAY = ROUND(1 / ((LW.SERVR - LW.ARRIVR) / 60) * 1000) / 1000   ;
    CONVERT HOURLY RATES TO (MINS) OF DELAY
148       ENDIF
149
150       _REV  = (ROUND(V * LI.PCTTRUCKS) * 3 * LI.CARTOLL) + (ROUND(V * (1 -
    LI.PCTTRUCKS)) * LI.CARTOLL)
151       ;_UTIL = (V / (LW.SERVR / (LI.UROADFACTOR * LI.CONFAC)))
152       _UTIL = (V / (LW.SERVR / (LI.UROADFACTOR / TODCONFAC)))
153
154       _SVCSECONDS = '00' + LTRIM(STR(LI.SVCSECONDS,2,0))
155       _SVCSECLEN  = STRLEN(_SVCSECONDS)-1
156       _SVCSECONDS = SUBSTR(_SVCSECONDS,_SVCSECLEN,2)
157
158       PRINT LIST = LI.TOLL(7.0), LI.TOLLTYPE(5.0), LI.A(6.0), LI.B(6.0), " ",
159                    LI.PLAZADESC(30C), LI.PLZALNSMIN(4.0), "   $", LI.CARTOLL(5.2),
160                    LI.SVCMINUTES(6.0), ":", _SVCSECONDS(2.0C), "  ",
    (LI.PCTTRUCKS * 100)(7.2),
161                    V(10.0), LW.DELAY(6.2), LI.LINKCOST(6.2), " $", _REV(8.0C),
    _UTIL(7.3), LW.ARRIVR(7.2), LW.SERVR(7.2) printo=1
162
163       TOLLVOL[ITERATION] = TOLLVOL[ITERATION] + V
164       TOLLREVENUE[ITERATION] = TOLLREVENUE[ITERATION] + _REV
165     ENDIF
166
167     IF (LI.TOLL_ACC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2
    * 1000) / 1000           ;(MINS) congested COST toll acceleration links
168     IF (LI.TOLL_DEC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED))
    / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links
169
170     IF (LI.FACILITY=90-99) TOLLVMT[ITERATION] = TOLLVMT[ITERATION] + (LI.DISTANCE
    * V)
171
172     IF (LINKNO=1)
173       PRINT LIST = '\n','\n',"                         ITERATION ",ITERATION(2.0),
    "  SUMMARY REPORT ON TOLL LINKS -- CTOLL VALUE =",ctoll(6.3), printo=1
174       PRINT LIST = '\n','\n',
    "                                                    No.
    ServTim  %Heavy           Delay Toll    Plan", printo=1
175       PRINT LIST =          "  Class Type  ...Nodes..    ......... Name
    .........    Lanes   Toll   (mm:ss)  Trucks    Volume  (Min) (Min)   Rev.
    Util.", printo=1
176       PRINT LIST = '\n','\n'
177     ENDIF
178
179     IF (LINKNO=NUMLINKS)
180       PRINT LIST = '\n','\n',"
    *----------------------*************************** TOTAL VOLUME FOR THIS
    ITERATION IS     ",TOLLVOL[ITERATION](10.0C), printo=1
181       PRINT LIST =          "     *-TOLL REVENUE SUMMARY-**********************
    TOTAL TOLL REVENUE FOR THIS ITERATION IS    $",TOLLREVENUE[ITERATION](10.0C),
    printo=1
182       PRINT LIST =          "
    *----------------------**********************  TOLL FACILITY VMT FOR THIS
    ITERATION IS     ",TOLLVMT[ITERATION](10.0C), printo=1
183       PRINT LIST =          "     *----------------------*****************
    AVERAGE REVENUE PER VMT FOR THIS ITERATION IS    $",(TOLLREVENUE[ITERATION] /
    TOLLVMT[ITERATION])(14.3C), printo=1
184     ENDIF
185
186   ENDPHASE
```

```
187
188
189    ENDRUN
190
```

```
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\Cube\TOD\TSHWY00I.PRN" MSG='EEPM Period Highway Assignment'
FILEO PRINTO[1] = "{SCENARIO_DIR}\Output\TSHWY00J.PRN"
FILEO TURNVOLO[1] = "{SCENARIO_DIR}\Output\TSHWY00C.TRN",
FORMAT=DBF

FILEO NETO = "{SCENARIO_DIR}\OUTPUT\HWYLOAD_PM_{ALT}{YEAR}_EE.NET"
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\HNET_XY.DBF"
FILEI TURNPENI = "{SCENARIO_DIR}\INPUT\TURNTOD_{YEAR}{ALT}.PEN"
FILEI NETI = "{SCENARIO_DIR}\OUTPUT\SA_HNET_{YEAR}.NET"
FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\EE_TOD_{ALT}{YEAR}.MAT"
DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4




ARRAY TOLLVOL=1, TOLLREVENUE=1, TOLLVMT=1
PAR MAXITERS=1, RELATIVEGAP=0,  GAP=0.001, AAD=0, RAAD=0, RMSE=0
;RELATIVEGAP=0.07    GAP=0.0005,
PAR COMBINE=EQUI, ENHANCE=2 ; KDK
TURNS N=1-99999
ZONEMSG=100
CTOLL=.06 ;{CTOLL}


; look up deceleration rate based on approach speed
LOOKUP,
 INTERPOLATE=Y, LIST=Y, NAME=DECEL,
        LOOKUP[1]=1,RESULT=2,
R = '30 4',
    '70 6.2'

LOOKUP LOOKUPI=1,
       NAME=DUMMYZONE, ;COUNTY LOOKUP
         LOOKUP[1]=N, RESULT=DUMMY,
         FAIL[1]=0,FAIL[2]=0,FAIL[3]=0,INTERPOLATE=N, LIST=N

FUNCTION  TC[1] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll links
FUNCTION  TC[2] = (.005 * LI.CARTOLL * 60); + LI.SVCMINUTES + (LI.SVCSECONDS / 60)            ;(MINS) time for toll links
FUNCTION  TC[3] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll acceleration links
FUNCTION  TC[4] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time toll deceleration links
FUNCTION  TC[5] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^ LI.BPREXPONENT)))   ;(MINS) congested time for non-toll TRUCK ONLY links

FUNCTION  COST[1] = TIME

;(MINS) congested COST for non-toll links
FUNCTION  COST[2] = TIME + ROUND((1 / (LW.SERVR - LW.ARRIVR)) * 60 * 1000) / 1000                ;(MINS) congested COST for toll links
FUNCTION  COST[3] = TIME + ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000          ;(MINS) congested COST toll acceleration links MINS
FUNCTION  COST[4] = TIME + ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60)
```

```
      / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links MINS
   50 FUNCTION  COST[5] = TIME

      ;(MINS) congested COST for non-toll TRUCK ONLY links
   51
   52 PROCESS PHASE=LINKREAD
   53
   54    TODCONFAC={PMCAPFAC} ; different for each period
   55
   56
   57
   58   ; basics
   59   T0 = LI.TOTALCOST
   60   ;C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR / LI.CONFAC
   61   C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR * TODCONFAC
   62   IF (LI.TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (LI.TIME / 60)   ; CONVERT TIME
      (MINS) TO HOURS
   63
   64   IF (LI.FACILITY=49)    ADDTOGROUP=1   ; TRUCK RESTRICTED LOCAL ROADS
   65   IF (LI.FACILITY=80-89) ADDTOGROUP=2   ; HOV LANES
   66   IF (LI.FACILITY=68-69)    ADDTOGROUP=3   ; Transit-only links
   67
   68   ; classify links based on presence/absence of tolls
   69   LINKCLASS=1 ; no toll
   70
   71
   72
   73   IF (LI.TOLL > 0)
   74     LINKCLASS = 2 ; with toll
   75     T0 = LI.LINKCOST
   76
   77
   78     IF (LI.TOLLTYPE = 1)
   79       LW.DELAY  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)   ; CONVERT SECONDS TO
      (MINS)
   80       LW.SERVR  = 1 / (LW.DELAY / 60)                    ; CONVERT (MINS) DELAY
      TO HOURS AND INVERT (VPH PER LANE)
   81       LW.ARRIVR = 0
   82     ELSE
   83       LW.DELAY  = 0
   84       LW.SERVR  = 1000000                                ; PREVENT DIVIDE BY
      ZERO FOR RAMP BOOTHS
   85       LW.ARRIVR = 0
   86     ENDIF
   87
   88   ENDIF
   89
   90   IF (LI.TOLL_ACC > 0)
   91     LINKCLASS=3 ; Toll Plaza Acceleration link
   92     LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000
                ; (MINS) congested COST toll acceleration links
   93   ENDIF
   94
   95   IF (LI.TOLL_DEC > 0)
   96     LINKCLASS=4 ; Toll Plaza Deceleration link
   97     LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000)
      / 1000 ; (MINS) congested COST toll deceleration links
   98   ENDIF
   99
  100   IF (LI.TOLL > MAXPLZNO) MAXPLZNO = LI.TOLL
  101
  102   IF (LI.FACILITY = 76) LINKCLASS=5
```

```
103
104   ENDPHASE
105
106   ;================================================================================
      =========
107   ;=== ILOOP (ASSIGNMENT) PHASE ===
108   ;================================================================================
      =========
109   PHASE=ILOOP
110
111   ; NORMAL MODE
112       if (DUMMYZONE(1,I)=0)
113           PATHLOAD PATH=COST, VOL[1]=MI.1.LOV_PM + MI.1.LTRK_PM + MI.1.HTRK_PM,
      PENI=1-3, EXCLUDEGROUP=2-3  ;,
      patho=1,name='lovassignment',allj=f,includecosts=t
114           PATHLOAD PATH=COST, VOL[2]=MI.1.HOV_PM, PENI=1-3, EXCLUDEGROUP=3    ;,
      patho=1,name='hovassignment',allj=f,includecosts=t
115           ;PATHLOAD PATH=COST, VOL[3]=MI.1.3, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
116           ;PATHLOAD PATH=COST, VOL[4]=MI.1.4, PENI=1-3, EXCLUDEGROUP=1-3  ;,
      patho=1,name='ltkassignment',allj=f,includecosts=t
117           ;PATHLOAD PATH=COST, VOL[1]=MI.1.1+MI.1.3+MI.1.4, PENI=1,
      EXCLUDEGROUP=2-3
118           ;PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1, EXCLUDEGROUP=3
119       endif
120
121   ; DYNASIM GENERATOR MODE
122   ;  PATHLOAD PATH=COST, VOL[1]=MI.1.1,VOL[2]=MI.1.2,VOL[3]=MI.1.3, PENI=1,
      patho=1,name='Assignment',allj=f,includecosts=f
123
124   ENDPHASE
125
126   ;================================================================================
      =========
127   ;=== ADJUST PHASE (WITH REPORTING)===
128   ;================================================================================
      =========
129   ; User Specified Functions
130   ; Delay functions, one for congested travel time, the other for toll links
      convert to time equivalent
131   PHASE=ADJUST
132
133
134     IF (TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (TIME / 60)                ;
      (MPH) CONVERT TIME (MINS) TO HRS
135
136     IF (LI.CARTOLL > 0)
137       IF (LI.TOLL > _TOLLLINKS) _TOLLLINKS = LI.TOLL
138
139       IF (LI.TOLLTYPE = 1)
140         ;LW.ARRIVR = ROUND((V / LI.UROADFACTOR) * LI.CONFAC / LI.PLZALNSMin) ;
      (HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
141         LW.ARRIVR = ROUND((V*TODCONFAC / LI.UROADFACTOR) / LI.PLZALNSMin) ;
      (HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
142         LW.SERVT  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)                  ;
      (MINS) Plaza lane service time in minutes per vehicle
143         LW.SERVR  = 1 / (LW.SERVT / 60)                                   ;
      (HRS)  Plaza lane service rate in vehicle per hour
144
145         IF ( (LW.ARRIVR >= LW.SERVR) && (LW.SERVR > 1000) )
146           LW.ARRIVR = 0.99 * LW.SERVR                                     ;
      (HRS)  Prevent infinite or negative queue
```

```
147         ELSEIF (LW.ARRIVR >= LW.SERVR)
148           LW.ARRIVR = 0.95 * LW.SERVR                                   ;
    (HRS)  Prevent infinite or negative queue
149         ENDIF
150
151         IF (LW.SERVR = 0) LW.SERVR = 1000                               ;
    PREVENT DIVIDE BY ZERO
152
153         LW.DELAY = ROUND(1 / ((LW.SERVR - LW.ARRIVR) / 60) * 1000) / 1000   ;
    CONVERT HOURLY RATES TO (MINS) OF DELAY
154       ENDIF
155
156       _REV  = (ROUND(V * LI.PCTTRUCKS) * 3 * LI.CARTOLL) + (ROUND(V * (1 -
    LI.PCTTRUCKS)) * LI.CARTOLL)
157       ;_UTIL = (V / (LW.SERVR / (LI.UROADFACTOR * LI.CONFAC)))
158       _UTIL = (V / (LW.SERVR / (LI.UROADFACTOR / TODCONFAC)))
159
160       _SVCSECONDS = '00' + LTRIM(STR(LI.SVCSECONDS,2,0))
161       _SVCSECLEN  = STRLEN(_SVCSECONDS)-1
162       _SVCSECONDS = SUBSTR(_SVCSECONDS,_SVCSECLEN,2)
163
164       PRINT LIST = LI.TOLL(7.0), LI.TOLLTYPE(5.0), LI.A(6.0), LI.B(6.0), " ",
165                    LI.PLAZADESC(30C), LI.PLZALNSMIN(4.0), "   $", LI.CARTOLL(5.2),
166                    LI.SVCMINUTES(6.0), ":", _SVCSECONDS(2.0C), "  ",
    (LI.PCTTRUCKS * 100)(7.2),
167                    V(10.0), LW.DELAY(6.2), LI.LINKCOST(6.2), " $", _REV(8.0C),
    _UTIL(7.3), LW.ARRIVR(7.2), LW.SERVR(7.2) printo=1
168
169       TOLLVOL[ITERATION] = TOLLVOL[ITERATION] + V
170       TOLLREVENUE[ITERATION] = TOLLREVENUE[ITERATION] + _REV
171     ENDIF
172
173     IF (LI.TOLL_ACC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2
    * 1000) / 1000           ;(MINS) congested COST toll acceleration links
174     IF (LI.TOLL_DEC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED))
    / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links
175
176     IF (LI.FACILITY=90-99) TOLLVMT[ITERATION] = TOLLVMT[ITERATION] + (LI.DISTANCE
    * V)
177
178     IF (LINKNO=1)
179       PRINT LIST = '\n','\n'," ITERATION ",ITERATION(2.0),
    "  SUMMARY REPORT ON TOLL LINKS -- CTOLL VALUE =",ctoll(6.3), printo=1
180       PRINT LIST = '\n','\n',
    "                                                    No.
    ServTim  %Heavy           Delay Toll    Plan", printo=1
181       PRINT LIST =          "  Class Type  ...Nodes..    ......... Name
    .........    Lanes   Toll   (mm:ss)  Trucks    Volume  (Min) (Min)   Rev.
    Util.", printo=1
182       PRINT LIST = '\n','\n'
183     ENDIF
184
185     IF (LINKNO=NUMLINKS)
186       PRINT LIST = '\n','\n',"
    *-------------------------------- TOTAL VOLUME FOR THIS
    ITERATION IS     ",TOLLVOL[ITERATION](10.0C), printo=1
187       PRINT LIST =          "     *-TOLL REVENUE SUMMARY-**********************
    TOTAL TOLL REVENUE FOR THIS ITERATION IS    $",TOLLREVENUE[ITERATION](10.0C),
    printo=1
188       PRINT LIST =          "
    *----------------------********************  TOLL FACILITY VMT FOR THIS
    ITERATION IS     ",TOLLVMT[ITERATION](10.0C), printo=1
```

```
189       PRINT LIST =           "     *----------------------*****************
      AVERAGE REVENUE PER VMT FOR THIS ITERATION IS    $",(TOLLREVENUE[ITERATION] /
      TOLLVMT[ITERATION])(14.3C), printo=1
190     ENDIF
191
192   ENDPHASE
193
194
195   ENDRUN
196
```

```
1   ; Do not change filenames or add or remove FILEI/FILEO statements using an
    editor. Use Cube/Application Manager.
2   RUN PGM=HIGHWAY PRNFILE=
    "C:\FSUTMS\D5\Latest\CFRPM641_09152016.TODFinal\Cube\TOD\TSHWY00H.PRN" MSG='EE
    NT Period Assignment'
3   FILEO PRINTO[1] = "{SCENARIO_DIR}\OUTPUT\TSHWY00N.PRN"
4   FILEO TURNVOLO[1] = "{SCENARIO_DIR}\OUTPUT\TSHWY00E.TRN",
5   FORMAT=DBF
6
7   FILEO NETO = "{SCENARIO_DIR}\OUTPUT\HWYLOAD_NT_{ALT}{YEAR}_EE.NET"
8   FILEI LOOKUPI[1] = "{SCENARIO_DIR}\OUTPUT\HNET_XY.DBF"
9   FILEI TURNPENI = "{SCENARIO_DIR}\INPUT\TURNTOD_{YEAR}{ALT}.PEN"
10  FILEI NETI = "{SCENARIO_DIR}\OUTPUT\SA_HNET_{YEAR}.NET"
11  FILEI MATI[1] = "{SCENARIO_DIR}\OUTPUT\EE_TOD_{ALT}{YEAR}.MAT"
12  DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-%NUMBER_OF_PROCESSORS%
13  ;DistributeINTRASTEP ProcessID="CFRPMdist", ProcessList=1-4
14
15
16
17
18
19
20  ARRAY TOLLVOL=1, TOLLREVENUE=1, TOLLVMT=1
21  PAR MAXITERS=1, RELATIVEGAP=0,  GAP=0.001, AAD=0, RAAD=0, RMSE=0
    ;RELATIVEGAP=0.07    GAP=0.0005,
22  PAR COMBINE=EQUI, ENHANCE=2 ; KDK
23  TURNS N=1-99999
24  ZONEMSG=100
25  CTOLL=.06 ;{CTOLL}
26
27  ; look up deceleration rate based on approach speed
28  LOOKUP,
29   INTERPOLATE=Y, LIST=Y, NAME=DECEL,
30          LOOKUP[1]=1,RESULT=2,
31  R = '30 4',
32      '70 6.2'
33
34  LOOKUP LOOKUPI=1,
35         NAME=DUMMYZONE, ;COUNTY LOOKUP
36           LOOKUP[1]=N, RESULT=DUMMY,
37           FAIL[1]=0,FAIL[2]=0,FAIL[3]=0,INTERPOLATE=N, LIST=N
38
39  FUNCTION  TC[1] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^
    LI.BPREXPONENT)))   ;(MINS) congested time for non-toll links
40  FUNCTION  TC[2] = ( .005 * LI.CARTOLL * 60); + LI.SVCMINUTES + (LI.SVCSECONDS /
    60)             ;(MINS) time for toll links
41  FUNCTION  TC[3] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^
    LI.BPREXPONENT)))   ;(MINS) congested time toll acceleration links
42  FUNCTION  TC[4] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^
    LI.BPREXPONENT)))   ;(MINS) congested time toll deceleration links
43  FUNCTION  TC[5] = T0 * (1 + ({DAMPING} * LI.BPRCOEFFICIENT * (MIN(V/C,4.00) ^
    LI.BPREXPONENT)))   ;(MINS) congested time for non-toll TRUCK ONLY links
44
45  FUNCTION  COST[1] = TIME

    ;(MINS) congested COST for non-toll links
46  FUNCTION  COST[2] = TIME + ROUND((1 / (LW.SERVR - LW.ARRIVR)) * 60 * 1000) /
    1000                  ;(MINS) congested COST for toll links
47  FUNCTION  COST[3] = TIME + ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000
    ) / 1000           ;(MINS) congested COST toll acceleration links MINS
48  FUNCTION  COST[4] = TIME + ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60)
    / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links MINS
```

```
FUNCTION  COST[5] = TIME

;(MINS) congested COST for non-toll TRUCK ONLY links

PROCESS PHASE=LINKREAD

   TODCONFAC={NTCAPFAC} ; different for each period

  ; basics
  T0 = LI.TOTALCOST
  ;C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR / LI.CONFAC
  C  = LI.CAPACITY * LI.NUM_LANES * LI.UROADFACTOR * TODCONFAC
  IF (LI.TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (LI.TIME / 60)   ; CONVERT TIME (MINS) TO HOURS

  IF (LI.FACILITY=49)    ADDTOGROUP=1   ; TRUCK RESTRICTED LOCAL ROADS
  IF (LI.FACILITY=80-89) ADDTOGROUP=2   ; HOV LANES
  IF (LI.FACILITY=68-69)    ADDTOGROUP=3   ; Transit-only links

  ; classify links based on presence/absence of tolls
  LINKCLASS=1 ; no toll

  IF (LI.TOLL > 0)
    LINKCLASS = 2 ; with toll
    T0 = LI.LINKCOST


    IF (LI.TOLLTYPE = 1)
      LW.DELAY  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)   ; CONVERT SECONDS TO (MINS)
      LW.SERVR  = 1 / (LW.DELAY / 60)                    ; CONVERT (MINS) DELAY TO HOURS AND INVERT (VPH PER LANE)
      LW.ARRIVR = 0
    ELSE
      LW.DELAY  = 0
      LW.SERVR  = 1000000                                ; PREVENT DIVIDE BY ZERO FOR RAMP BOOTHS
      LW.ARRIVR = 0
    ENDIF

  ENDIF

  IF (LI.TOLL_ACC > 0)
    LINKCLASS=3 ; Toll Plaza Acceleration link
    LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2 * 1000) / 1000
           ; (MINS) congested COST toll acceleration links
  ENDIF

  IF (LI.TOLL_DEC > 0)
    LINKCLASS=4 ; Toll Plaza Deceleration link
    LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED)) / 60) / 2 * 1000) / 1000 ; (MINS) congested COST toll deceleration links
  ENDIF

  IF (LI.TOLL > MAXPLZNO) MAXPLZNO = LI.TOLL

  IF (LI.FACILITY = 76) LINKCLASS=5

ENDPHASE

;=========================================================================================
```

```
;=== ILOOP (ASSIGNMENT) PHASE ===
;================================================================================
=========
PHASE=ILOOP

; NORMAL MODE
    if (DUMMYZONE(1,I)=0)
        PATHLOAD PATH=COST, VOL[1]=MI.1.LOV_NT + MI.1.LTRK_NT + MI.1.HTRK_NT,
PENI=1-3, EXCLUDEGROUP=2-3  ;,
patho=1,name='lovassignment',allj=f,includecosts=t
        PATHLOAD PATH=COST, VOL[2]=MI.1.HOV_NT, PENI=1-3, EXCLUDEGROUP=3    ;,
patho=1,name='hovassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[3]=MI.1.3, PENI=1-3, EXCLUDEGROUP=1-3  ;,
patho=1,name='ltkassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[4]=MI.1.4, PENI=1-3, EXCLUDEGROUP=1-3  ;,
patho=1,name='ltkassignment',allj=f,includecosts=t
        ;PATHLOAD PATH=COST, VOL[1]=MI.1.1+MI.1.3+MI.1.4, PENI=1,
EXCLUDEGROUP=2-3
        ;PATHLOAD PATH=COST, VOL[2]=MI.1.2, PENI=1, EXCLUDEGROUP=3
    endif

; DYNASIM GENERATOR MODE
;  PATHLOAD PATH=COST, VOL[1]=MI.1.1,VOL[2]=MI.1.2,VOL[3]=MI.1.3, PENI=1,
patho=1,name='Assignment',allj=f,includecosts=f

ENDPHASE

;================================================================================
=========
;=== ADJUST PHASE (WITH REPORTING)===
;================================================================================
=========
; User Specified Functions
; Delay functions, one for congested travel time, the other for toll links
convert to time equivalent
PHASE=ADJUST


  IF (TIME > 0) LW.CGSTSPEED = LI.DISTANCE / (TIME / 60)                ;
(MPH) CONVERT TIME (MINS) TO HRS

  IF (LI.CARTOLL > 0)
    IF (LI.TOLL > _TOLLLINKS) _TOLLLINKS = LI.TOLL

    IF (LI.TOLLTYPE = 1)
      ;LW.ARRIVR = ROUND((V / LI.UROADFACTOR) * LI.CONFAC / LI.PLZALNSMin) ;
(HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
      LW.ARRIVR = ROUND((V*TODCONFAC / LI.UROADFACTOR) / LI.PLZALNSMin) ;
(HRS)  Hourly Volume per toll lane ie. arrival rate in vehicles per hour
      LW.SERVT  = LI.SVCMINUTES + (LI.SVCSECONDS / 60)                  ;
(MINS) Plaza lane service time in minutes per vehicle
      LW.SERVR  = 1 / (LW.SERVT / 60)                                   ;
(HRS)  Plaza lane service rate in vehicle per hour

      IF ( (LW.ARRIVR >= LW.SERVR) && (LW.SERVR > 1000) )
        LW.ARRIVR = 0.99 * LW.SERVR                                     ;
(HRS)  Prevent infinite or negative queue
      ELSEIF (LW.ARRIVR >= LW.SERVR)
        LW.ARRIVR = 0.95 * LW.SERVR                                     ;
(HRS)  Prevent infinite or negative queue
      ENDIF

```

```
146         IF (LW.SERVR = 0) LW.SERVR = 1000                                   ;
    PREVENT DIVIDE BY ZERO
147
148         LW.DELAY = ROUND(1 / ((LW.SERVR - LW.ARRIVR) / 60) * 1000) / 1000   ;
    CONVERT HOURLY RATES TO (MINS) OF DELAY
149       ENDIF
150
151       _REV  = (ROUND(V * LI.PCTTRUCKS) * 3 * LI.CARTOLL) + (ROUND(V * (1 -
    LI.PCTTRUCKS)) * LI.CARTOLL)
152       ;_UTIL = (V / (LW.SERVR / (LI.UROADFACTOR * LI.CONFAC)))
153       _UTIL = (V / (LW.SERVR / (LI.UROADFACTOR / TODCONFAC)))
154
155       _SVCSECONDS = '00' + LTRIM(STR(LI.SVCSECONDS,2,0))
156       _SVCSECLEN  = STRLEN(_SVCSECONDS)-1
157       _SVCSECONDS = SUBSTR(_SVCSECONDS,_SVCSECLEN,2)
158
159       PRINT LIST = LI.TOLL(7.0), LI.TOLLTYPE(5.0), LI.A(6.0), LI.B(6.0), " ",
160                    LI.PLAZADESC(30C), LI.PLZALNSMIN(4.0), "   $", LI.CARTOLL(5.2),
161                    LI.SVCMINUTES(6.0), ":", _SVCSECONDS(2.0C), "  ",
    (LI.PCTTRUCKS * 100)(7.2),
162                    V(10.0), LW.DELAY(6.2), LI.LINKCOST(6.2), " $", _REV(8.0C),
    _UTIL(7.3), LW.ARRIVR(7.2), LW.SERVR(7.2) printo=1
163
164       TOLLVOL[ITERATION] = TOLLVOL[ITERATION] + V
165       TOLLREVENUE[ITERATION] = TOLLREVENUE[ITERATION] + _REV
166     ENDIF
167
168     IF (LI.TOLL_ACC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / {ACCELRATE}) / 60) / 2
    * 1000) / 1000           ;(MINS) congested COST toll acceleration links
169     IF (LI.TOLL_DEC > 0) LW.DELAY = ROUND(((LW.CGSTSPEED / DECEL(1,LW.CGSTSPEED))
    / 60) / 2 * 1000) / 1000 ;(MINS) congested COST toll deceleration links
170
171     IF (LI.FACILITY=90-99) TOLLVMT[ITERATION] = TOLLVMT[ITERATION] + (LI.DISTANCE
    * V)
172
173     IF (LINKNO=1)
174       PRINT LIST = '\n','\n'," ITERATION ",ITERATION(2.0),
    "  SUMMARY REPORT ON TOLL LINKS -- CTOLL VALUE =",ctoll(6.3), printo=1
175       PRINT LIST = '\n','\n',
    "                                                    No.
    ServTim  %Heavy           Delay Toll    Plan", printo=1
176       PRINT LIST =          "  Class Type  ...Nodes..    ......... Name
    .........    Lanes   Toll   (mm:ss)  Trucks    Volume  (Min) (Min)   Rev.
    Util.", printo=1
177       PRINT LIST = '\n','\n'
178     ENDIF
179
180     IF (LINKNO=NUMLINKS)
181       PRINT LIST = '\n','\n',"
    *---------------------************************** TOTAL VOLUME FOR THIS
    ITERATION IS     ",TOLLVOL[ITERATION](10.0C), printo=1
182       PRINT LIST =          "     *-TOLL REVENUE SUMMARY-**********************
    TOTAL TOLL REVENUE FOR THIS ITERATION IS    $",TOLLREVENUE[ITERATION](10.0C),
    printo=1
183       PRINT LIST =          "
    *---------------------**********************  TOLL FACILITY VMT FOR THIS
    ITERATION IS     ",TOLLVMT[ITERATION](10.0C), printo=1
184       PRINT LIST =          "     *----------------------*****************
    AVERAGE REVENUE PER VMT FOR THIS ITERATION IS    $",(TOLLREVENUE[ITERATION] /
    TOLLVMT[ITERATION])(14.3C), printo=1
185     ENDIF
186
```

```
187    ENDPHASE
188
189
190    ENDRUN
191
```