



National Capital Region  
**Transportation Planning Board**

*Metropolitan Washington Council of Governments (COG)  
National Capital Region Transportation Planning Board (TPB)*

# User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.75

---

*Volume 2 of 2: Appendices B (Batch Files), C (Cube  
Voyager Scripts), and D (AEMS and Linesum Control  
Files)*

*December 5, 2018*



## **User's Guide for the COG/TPB Travel Demand Forecasting Model, Version 2.3.75: Volume 2 of 2**

December 5, 2018

### **About the TPB**

The National Capital Region Transportation Planning Board (TPB) is the federally designated metropolitan planning organization (MPO) for metropolitan Washington. It is responsible for developing and carrying out a continuing, cooperative, and comprehensive transportation planning process in the metropolitan area. Members of the TPB include representatives of the transportation agencies of the states of Maryland and Virginia and the District of Columbia, 23 local governments, the Washington Metropolitan Area Transit Authority, the Maryland and Virginia General Assemblies, and nonvoting members from the Metropolitan Washington Airports Authority and federal agencies. The TPB is staffed by the Department of Transportation Planning at the Metropolitan Washington Council of Governments (COG).

### **About COG**

The Metropolitan Washington Council of Governments (COG) is an independent, nonprofit association that brings area leaders together to address major regional issues in the District of Columbia, suburban Maryland, and Northern Virginia. COG's membership is comprised of 300 elected officials from 23 local governments, the Maryland and Virginia state legislatures, and U.S. Congress.

### **Credits**

Director, Department of Transportation Planning (DTP): Kanti Srikanth

Director, Travel Forecasting and Emissions Analysis Program, DTP: Ronald Milone (retired Oct. 2018)

Report Authors:

- This Update: Ray Ngo, Mark Moran, Meseret Seifu, and Feng Xie
- Past Versions: Ron Milone, Mark Moran, Meseret Seifu, Hamid Humeida, and Mary Martchouk

Oversight: COG/TPB Travel Forecasting Subcommittee

### **Acknowledgements**

This publication was funded, in part, by grants from the District of Columbia Department of Transportation, the Maryland Department of Transportation, the Virginia Department of Transportation, the Federal Highway Administration and the Federal Transit Administration. The material herein does not necessarily reflect the views of the sponsoring agencies.

### **Accommodations Policy**

Alternative formats of this document are available upon request. Visit [www.mwcog.org/accommodations](http://www.mwcog.org/accommodations) or call (202) 962-3300 or (202) 962-3213 (TDD).

### **Title VI Nondiscrimination Policy**

The Metropolitan Washington Council of Governments (COG) fully complies with Title VI of the Civil Rights Act of 1964 and related statutes and regulations prohibiting discrimination in all programs and activities. For more information, to file a Title VI related complaint, or to obtain information in another language, visit [www.mwcog.org/nondiscrimination](http://www.mwcog.org/nondiscrimination) or call (202) 962-3300.

El Consejo de Gobiernos del Área Metropolitana de Washington (COG) cumple con el Título VI de la Ley sobre los Derechos Civiles de 1964 y otras leyes y reglamentos en todos sus programas y actividades. Para obtener más información, someter un pleito relacionado al Título VI, u obtener información en otro idioma, visite [www.mwcog.org/nondiscrimination](http://www.mwcog.org/nondiscrimination) o llame al (202) 962-3300.

Copyright © 2018 by the Metropolitan Washington Council of Governments

---

# Appendix B. Batch files

<b>1. Runall</b> .....	<b>B-3</b>
1.1 run_Model_2019_Final.bat.....	B-3
1.2 run_ModelSteps_2019_Final.bat.....	B-3
1.3 run_Model_2021_Final.bat.....	B-4
1.4 run_ModelSteps_2021_Final.bat.....	B-5
1.5 run_Model_2025_Final.bat.....	B-6
1.6 run_ModelSteps_2025_Final.bat.....	B-6
1.7 run_Model_2030_final.bat.....	B-7
1.8 run_ModelSteps_2030_final.bat.....	B-8
1.9 run_Model_2040_Final.bat.....	B-9
1.10 run_ModelSteps_2040_Final.bat.....	B-9
1.11 run_Model_2045_final.bat.....	B-10
1.12 run_ModelSteps_2045_final.bat.....	B-11
<b>2 ‘Pump-Prime’ Iterations</b> .....	<b>B-12</b>
2.1 ArcPy_Walkshed_Process.bat .....	B-12
2.2 set_CPI.bat .....	B-13
2.2 PP_Highway_Build.bat.....	B-13
2.3 PP_Highway_Skims.bat.....	B-13
2.4 PP_Auto_Drivers.bat .....	B-14
<b>3 ‘Standard’ Iterations (1-4)</b> .....	<b>B-15</b>
3.1 Transit_Skim_All_Modes_Parallel.bat.....	B-15
3.2 Transit_Fare.bat .....	B-16
3.3 Trip_Generation.bat .....	B-16
3.4 Trip_Distribution.bat .....	B-17
3.5 Mode_Choice_Parallel.bat.....	B-17
3.6 Auto_Driver.bat .....	B-18
3.7 Time-of-Day.bat.....	B-18
3.8 Highway_Assignment_Parallel.bat.....	B-19
3.9 Average_Link_Speeds.bat - Iterations (2- 4).....	B-19
3.10 Highway_Skims.bat.....	B-19
<b>4 Transit assignment Batch Files</b> .....	<b>B-20</b>
4.1 Transit_Assignment_Parallel.bat .....	B-20
4.2 TranSum.bat.....	B-21
<b>5 Optional/Utility Batch Files</b> .....	<b>B-21</b>

---

5.1	SearchForErrs.bat.....	B-21
5.2	move_temp_files_v6.bat.....	B-22



# 1. Runall

## 1.1 run\_Model\_2019\_Final.bat

```

:: Version 2.3.75
:: 2018-09-11 Tue 10:40 AM

set root=.
set scenar=2019_Final
set runbat=run_ModelSteps_2019_Final.bat
:: Environment variables for (multistep) distributed processing:
:: Environment variables for (intra-step) distributed processing:
::     use MDP = t/f (for true or false)
::     use IDP = t/f (for true or false)
::     Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
set useMdp=t
:: AMsubnode & MDsubnode are used in highway_assignment_parallel.bat/s
set AMsubnode=1-4
set MDsubnode=2-4
:: subnode used in transit fare and transit assignment
:: We no longer use IDP in transit skimming, since it would require 16 cores
set subnode=1-3

:: This command will
:: 1) time the model run (using timethis.exe and the double quotes)
:: 2) redirect standard output and standard error to a file
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the
screen

timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt

:: Open up the file containing the stderr and stdout
if exist %root%\%scenar%\%scenar%_fulloutput.txt start
%root%\%scenar%\%scenar%_fulloutput.txt

:: Look for errors in the reports and output files
call searchForErrs.bat %scenar%
:: Open up the file containing any errors found
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Open up other report files
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start
%root%\%scenar%\i4_Highway_Assignment.rpt
if exist %root%\%scenar%\i4_mc_NL_summary.txt start
%root%\%scenar%\i4_mc_NL_summary.txt
if exist %root%\%scenar%\i4_Assign_Output.net start
%root%\%scenar%\i4_Assign_Output.net
cd %scenar%
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov.txt -tail 1; get-
content i4_ue*AM_hov.txt -tail 1; get-content i4_ue*PM_nonHov.txt -tail 1; get-
content i4_ue*PM_hov.txt -tail 1; get-content i4_ue*MD*txt -tail 1; get-content
i4_ue*NT*txt -tail 1
cd ..
move_temp_files_v6.bat %scenar%

:: Cleanup
set root=
set scenar=
set runbat=

```

```

set useIdp=
set useMdp=
set AMsubnode=
set MDsubnode=
set subnode=

```

## 1.2 run\_ModelSteps\_2019\_Final.bat

```

:: Version 2.3.75
:: 2018-09-11
:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2019
set _alt_=Ver2.3.75_2019_Final
:: Maximum number of user equilibrium iterations used in traffic assignment
:: User should not need to change this. Instead, change _relGap_ (below)
set _maxUeIter_=1000

:: Set transit constraint path and files
:: Current year used to set the constraint = 2020

set _tcpath=

:: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4
for i4)
:: Set the value below

rem ===== Pump Prime Iteration =====

set _iter_=pp
set _prev_=pp
set _relGap_=0.01

REM call ArcPy_Walkshed_Process.bat %1
call Set_CPI.bat %1
call PP_Highway_Build.bat %1
call PP_Highway_Skims.bat %1
call Transit_Skim_All_Modes_Parallel.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call PP_Auto_Drivers.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 1 =====

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 2 =====

```

## Appendix B Batch file

```
set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 3 =====

set _iter_=i3
set _prev_=i2
set _relGap_=0.001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 4 =====

set _iter_=i4
set _prev_=i3
set _relGap_=0.0001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Transit assignment =====
@echo Starting Transit Assignment Step
@date /t & time/t

call Transit_Assignment_Parallel.bat %1
call TranSum.bat %1

@echo End of batch file
@date /t & time/t
:: rem ===== End of batch file =====

REM cd %1
REM copy *.txt MDP_%useMDP%\*.txt
REM copy *.rpt MDP_%useMDP%\*.rpt
REM copy *.log MDP_%useMDP%\*.log
REM CD..

set _year_=
set _alt_=
set _iter_=
```

```
set _prev_=
set _maxUeIter_=
set _relGap_=
```

### 1.3 run\_Model\_2021\_Final.bat

```
:: Version 2.3.75
:: 2018-09-06 Thu 3:30 PM
```

```
set root=.
set scenar=2021_Final
set runbat=run_ModelSteps_2021_Final.bat
:: Environment variables for (multistep) distributed processing:
:: Environment variables for (intrastep) distributed processing:
:: use MDP = t/f (for true or false)
:: use IDP = t/f (for true or false)
:: Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
set useMdp=t
:: AMsubnode & MDsubnode are used in highway_assignment_parallel.bat/s
set AMsubnode=1-4
set MDsubnode=2-4
:: subnode used in transit fare and transit assignment
:: We no longer use IDP in transit skimming, since it would require 16 cores
set subnode=1-3

:: This command will
:: 1) time the model run (using timethis.exe and the double quotes)
:: 2) redirect standard output and standard error to a file
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the
screen

timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt

:: Open up the file containing the stderr and stdout
if exist %root%\%scenar%\%scenar%_fulloutput.txt start
%root%\%scenar%\%scenar%_fulloutput.txt

:: Look four errors in the reports and output files
call searchForErrs.bat %scenar%
:: Open up the file containing any errors found
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Open up other report files
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start
%root%\%scenar%\i4_Highway_Assignment.rpt
if exist %root%\%scenar%\i4_mc_NL_summary.txt start
%root%\%scenar%\i4_mc_NL_summary.txt
if exist %root%\%scenar%\i4_Assign_Output.net start
%root%\%scenar%\i4_Assign_Output.net
cd %scenar%
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov*txt -tail 1; get-
content i4_ue*AM_hov*txt -tail 1; get-content i4_ue*PM_nonHov*txt -tail 1; get-
content i4_ue*PM_hov*txt -tail 1; get-content i4_ue*MD*txt -tail 1; get-content
i4_ue*NT*txt -tail 1
cd ..
move_temp_files_v6.bat %scenar%

:: Cleanup
set root=
set scenar=
set runbat=
set useIdp=
```



## Appendix B Batch file

```
set useMdp=  
set Amsubnode=  
set MDsubnode=  
set subnode=
```

### 1.4 run\_ModelSteps\_2021\_Final.bat

```
:: Version 2.3.75  
:: 2018-09-06 Thu 3:30 PM  
  
:: Version 2.3 TPB Travel Model on 3722 TAZ System  
  
set _year_=2021  
set _alt_=Ver2.3.75_2021_Final  
:: Maximum number of user equilibrium iterations used in traffic assignment  
:: User should not need to change this. Instead, change _relGap_ (below)  
set _maxUeIter_=1000  
  
:: Set transit constraint path and files  
:: Current year used to set the constraint = 2020  
  
set _topath=  
  
:: UE relative gap threshold: Progressive ( $10^{-2}$  for pp-i2,  $10^{-3}$  for i3, &  $10^{-4}$   
for i4)  
:: Set the value below  
  
rem ===== Pump Prime Iteration =====  
  
set _iter_=pp  
set _prev_=pp  
set _relGap_=0.01  
  
REM call ArcPy_Walkshed_Process.bat %1  
call Set_CPI.bat %1  
call PP_Highway_Build.bat %1  
call PP_Highway_Skims.bat %1  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call PP_Auto_Drivers.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 1 =====  
  
set _iter_=i1  
set _prev_=pp  
  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Transit_Fare.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call Mode_Choice_Parallel.bat %1  
call Auto_Driver.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 2 =====  
  
set _iter_=i2
```

```
set _prev_=i1  
  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Transit_Fare.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call Mode_Choice_Parallel.bat %1  
call Auto_Driver.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Average_Link_Speeds.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 3 =====  
  
set _iter_=i3  
set _prev_=i2  
set _relGap_=0.001  
  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Transit_Fare.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call Mode_Choice_Parallel.bat %1  
call Auto_Driver.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Average_Link_Speeds.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 4 =====  
  
set _iter_=i4  
set _prev_=i3  
set _relGap_=0.0001  
  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Transit_Fare.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call Mode_Choice_Parallel.bat %1  
call Auto_Driver.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Average_Link_Speeds.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Transit assignment =====  
@echo Starting Transit Assignment Step  
@date /t & time/t  
  
call Transit_Assignment_Parallel.bat %1  
call TranSum.bat %1  
  
@echo End of batch file  
@date /t & time/t  
:: rem ===== End of batch file =====  
  
REM cd %1  
REM copy *.txt MDP_%useMDP%\*.txt  
REM copy *.rpt MDP_%useMDP%\*.rpt  
REM copy *.log MDP_%useMDP%\*.log  
REM CD..  
  
set _year_=  
set _alt_=  
set _iter_=  
set _prev_=
```

## Appendix B Batch file

```
set _maxUeIter=  
set _relGap=
```

### 1.5 run\_Model\_2025\_Final.bat

```
:: Version 2.3.75  
:: 2018-08-16 Thu 11:28:04  
  
set root=  
set scenar=2025_Final  
set runbat=run_ModelSteps_2025_Final.bat  
:: Environment variables for (multistep) distributed processing:  
:: Environment variables for (intrastep) distributed processing:  
:: use MDP = t/f (for true or false)  
:: use IDP = t/f (for true or false)  
:: Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total  
set useIdp=t  
set useMdp=t  
:: AMsubnode & MDsubnode are used in highway_assignment_parallel.bat/s  
set AMsubnode=1-4  
set MDsubnode=2-4  
:: subnode used in transit fare and transit assignment  
:: We no longer use IDP in transit skimming, since it would require 16 cores  
set subnode=1-3  
  
:: This command will  
:: 1) time the model run (using timethis.exe and the double quotes)  
:: 2) redirect standard output and standard error to a file  
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the  
screen  
  
timethis "runbat %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt  
  
:: Open up the file containing the stderr and stdout  
if exist %root%\%scenar%\%scenar%_fulloutput.txt start  
%root%\%scenar%\%scenar%_fulloutput.txt  
  
:: Look four errors in the reports and output files  
call searchForErrs.bat %scenar%  
:: Open up the file containing any errors found  
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start  
%root%\%scenar%\%scenar%_searchForErrs.txt  
  
:: Open up other report files  
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start  
%root%\%scenar%\i4_Highway_Assignment.rpt  
if exist %root%\%scenar%\i4_mc_NL_summary.txt start  
%root%\%scenar%\i4_mc_NL_summary.txt  
if exist %root%\%scenar%\i4_Assign_Output.net start  
%root%\%scenar%\i4_Assign_Output.net  
cd %scenar%  
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov*txt -tail 1; get-  
content i4_ue*AM_hov*txt -tail 1; get-content i4_ue*PM_nonHov*txt -tail 1; get-  
content i4_ue*PM_hov*txt -tail 1; get-content i4_ue*MD*txt -tail 1; get-content  
i4_ue*NT*txt -tail 1  
cd ..  
move_temp_files_v6.bat %scenar%  
  
:: Cleanup  
set root=  
set scenar=  
set runbat=  
set useIdp=  
set useMdp=  
set AMsubnode=  
set MDsubnode=
```

```
set subnode=
```

### 1.6 run\_ModelSteps\_2025\_Final.bat

```
:: Version 2.3.75  
:: 2018-08-16 Thu 11:27:04  
  
:: Version 2.3 TPB Travel Model on 3722 TAZ System  
  
set _year_=2025  
set _alt_=Ver2.3.75_2025_Final  
:: Maximum number of user equilibrium iterations used in traffic assignment  
:: User should not need to change this. Instead, change _relGap_ (below)  
set _maxUeIter_=1000  
  
:: Set transit constraint path and files  
:: Current year used to set the constraint = 2020  
  
set _tcpath=  
  
:: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4  
for i4)  
:: Set the value below  
  
rem ===== Pump Prime Iteration =====  
  
set _iter_=pp  
set _prev_pp  
set _relGap_=0.01  
  
REM call ArcPy_Walkshed_Process.bat %1  
call Set_CPI.bat %1  
call PP_Highway_Build.bat %1  
call PP_Highway_Skims.bat %1  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call PP_Auto_Drivers.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 1 =====  
  
set _iter_=i1  
set _prev_pp  
  
call Transit_Skim_All_Modes_Parallel.bat %1  
call Transit_Fare.bat %1  
call Trip_Generation.bat %1  
call Trip_Distribution.bat %1  
call Mode_Choice_Parallel.bat %1  
call Auto_Driver.bat %1  
call Time-of-Day.bat %1  
call Highway_Assignment_Parallel.bat %1  
call Highway_Skims.bat %1  
  
:: rem ===== Iteration 2 =====  
  
set _iter_=i2  
set _prev_=i1
```

## Appendix B Batch file

```
call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 3 =====

set _iter_=i3
set _prev_=i2
set _relGap_=0.001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 4 =====

set _iter_=i4
set _prev_=i3
set _relGap_=0.0001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Transit assignment =====
@echo Starting Transit Assignment Step
@date /t & time/t

call Transit_Assignment_Parallel.bat %1
call TranSum.bat %1

@echo End of batch file
@date /t & time/t
:: rem ===== End of batch file =====

REM cd %1
REM copy *.txt MDP_%useMDP%\*.txt
REM copy *.rpt MDP_%useMDP%\*.rpt
REM copy *.log MDP_%useMDP%\*.log
REM CD..

set _year_=
set _alt_=
set _iter_=
set _prev_=
set _maxUeIter_=
set _relGap_=
```

## 1.7 run\_Model\_2030\_final.bat

```
:: Version 2.3.75
:: 2018-09-10 Mon 10:30 AM

set root=.
set scenar=2030_Final
set runbat=run_ModelSteps_2030_Final.bat
:: Environment variables for (multistep) distributed processing:
:: Environment variables for (intrastep) distributed processing:
:: use MDP = t/f (for true or false)
:: use IDP = t/f (for true or false)
:: Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
set useMdp=t
:: AMsubnode & MDsubnode are used in highway_assignment_parallel.bat/s
set AMsubnode=1-4
set MDsubnode=2-4
:: subnode used in transit fare and transit assignment
:: We no longer use IDP in transit skimming, since it would require 16 cores
set subnode=1-3

:: This command will
:: 1) time the model run (using timethis.exe and the double quotes)
:: 2) redirect standard output and standard error to a file
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the
screen

timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt

:: Open up the file containing the stderr and stdout
if exist %root%\%scenar%\%scenar%_fulloutput.txt start
%root%\%scenar%\%scenar%_fulloutput.txt

:: Look four errors in the reports and output files
call searchForErrs.bat %scenar%
:: Open up the file containing any errors found
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Open up other report files
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start
%root%\%scenar%\i4_Highway_Assignment.rpt
if exist %root%\%scenar%\i4_mc_NL_summary.txt start
%root%\%scenar%\i4_mc_NL_summary.txt
if exist %root%\%scenar%\i4_Assign_Output.net start
%root%\%scenar%\i4_Assign_Output.net
cd %scenar%
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov*txt -tail 1; get-
content i4_ue*AM_hov*txt -tail 1; get-content i4_ue*PM_nonHov*txt -tail 1; get-
content i4_ue*PM_hov*txt -tail 1; get-content i4_ue*MD*txt -tail 1; get-content
i4_ue*NT*txt -tail 1
cd ..
move_temp_files_v6.bat %scenar%

:: Cleanup
set root=
set scenar=
set runbat=
set useIdp=
set useMdp=
set AMsubnode=
set MDsubnode=
set subnode=
```

### 1.8 run\_ModelSteps\_2030\_final.bat

```

:: Version 2.3.75
:: 2018-09-10 10:30 AM

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2030
set _alt_=Ver2.3.75_2030_Final
:: Maximum number of user equilibrium iterations used in traffic assignment
:: User should not need to change this. Instead, change _relGap_ (below)
set _maxUeIter_=1000

:: Set transit constraint path and files
:: Current year used to set the constraint = 2020

set _tcpath_=

:: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4
for i4)
:: Set the value below

rem ===== Pump Prime Iteration =====

set _iter_=pp
set _prev_=pp
set _relGap_=0.01

REM call ArcPy_Walkshed_Process.bat %1
call Set_CPI.bat %1
call PP_Highway_Build.bat %1
call PP_Highway_Skims.bat %1
call Transit_Skim_All_Modes_Parallel.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call PP_Auto_Drivers.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 1 =====

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 2 =====

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1

```

```

call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 3 =====

set _iter_=i3
set _prev_=i2
set _relGap_=0.001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 4 =====

set _iter_=i4
set _prev_=i3
set _relGap_=0.0001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Transit assignment =====
@echo Starting Transit Assignment Step
@date /t & time/t

call Transit_Assignment_Parallel.bat %1
call TranSum.bat %1

@echo End of batch file
@date /t & time/t
:: rem ===== End of batch file =====

REM cd %1
REM copy *.txt MDP_%useMDP%\*.txt
REM copy *.rpt MDP_%useMDP%\*.rpt
REM copy *.log MDP_%useMDP%\*.log
REM CD..

set _year_=
set _alt_=
set _iter_=
set _prev_=
set _maxUeIter_=
set _relGap_=

```

## 1.9 run\_Model\_2040\_Final.bat

```

:: Version 2.3.75
:: 2018-09-10 Mon 03:40 PM

set root=.
set scenar=2040_Final
set runbat=run_ModelSteps_2040_Final.bat
:: Environment variables for (multistep) distributed processing:
:: Environment variables for (intrastep) distributed processing:
::   use MDP = t/f (for true or false)
::   use IDP = t/f (for true or false)
::   Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
set useMdp=t
:: AMSubnode & MDSubnode are used in highway_assignment_parallel.bat/s
set AMSubnode=1-4
set MDSubnode=2-4
:: subnode used in transit fare and transit assignment
:: We no longer use IDP in transit skimming, since it would require 16 cores
set subnode=1-3

:: This command will
:: 1) time the model run (using timethis.exe and the double quotes)
:: 2) redirect standard output and standard error to a file
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the
screen

timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt

:: Open up the file containing the stderr and stdout
if exist %root%\%scenar%\%scenar%_fulloutput.txt start
%root%\%scenar%\%scenar%_fulloutput.txt

:: Look four errors in the reports and output files
call searchForErrs.bat %scenar%
:: Open up the file containing any errors found
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Open up other report files
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start
%root%\%scenar%\i4_Highway_Assignment.rpt
if exist %root%\%scenar%\i4_mc_NL_summary.txt start
%root%\%scenar%\i4_mc_NL_summary.txt
if exist %root%\%scenar%\i4_Assign_Output.net start
%root%\%scenar%\i4_Assign_Output.net
cd %scenar%
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov.txt -tail 1; get-
content i4_ue*AM_hov.txt -tail 1; get-content i4_ue*PM_nonHov.txt -tail 1; get-
content i4_ue*PM_hov.txt -tail 1; get-content i4_ue*MD.txt -tail 1; get-content
i4_ue*NT.txt -tail 1
cd ..
move_temp_files_v6.bat %scenar%

:: Cleanup
set root=
set scenar=
set runbat=
set useIdp=
set useMdp=
set AMSubnode=
set MDSubnode=
set subnode=

```

## 1.10 run\_ModelSteps\_2040\_Final.bat

```

:: Version 2.3.75
:: 2018-09-10 Mon 03:40 PM

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2040
set _alt_=Ver2.3.75_2040_Final
:: Maximum number of user equilibrium iterations used in traffic assignment
:: User should not need to change this. Instead, change _relGap_ (below)
set _maxUeIter_=1000

:: Set transit constraint path and files
:: Current year used to set the constraint = 2020

set _tcpath=

:: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4
for i4)
:: Set the value below

rem ===== Pump Prime Iteration =====

set _iter_=pp
set _prev_=pp
set _relGap_=0.01

REM call ArcPy_Walkshed_Process.bat %1
call Set_CPI.bat %1
call PP_Highway_Build.bat %1
call PP_Highway_Skims.bat %1
call Transit_Skim_All_Modes_Parallel.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call PP_Auto_Drivers.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 1 =====

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 2 =====

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1

```

## Appendix B Batch file

```
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 3 =====

set _iter_=i3
set _prev_=i2
set _relGap_=0.001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 4 =====

set _iter_=i4
set _prev_=i3
set _relGap_=0.0001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Transit assignment =====
@echo Starting Transit Assignment Step
@date /t & time/t

call Transit_Assignment_Parallel.bat %1
call TranSum.bat %1

@echo End of batch file
@date /t & time/t
:: rem ===== End of batch file =====

REM cd %1
REM copy *.txt MDP_%useMDP%*.txt
REM copy *.rpt MDP_%useMDP%*.rpt
REM copy *.log MDP_%useMDP%*.log
REM CD..

set _year_=
set _alt_=
set _iter_=
set _prev_=
set _maxUeIter_=
set _relGap_=
```

## 1.11 run\_Model\_2045\_final.bat

```
:: Version 2.3.75
:: 2018-08-09 Thu 15:55:04

set root=.
set scenar=2045_Final
set runbat=run_ModelSteps_2045_Final.bat
:: Environment variables for (multistep) distributed processing:
:: Environment variables for (intrastep) distributed processing:
:: use MDP = t/f (for true or false)
:: use IDP = t/f (for true or false)
:: Number of subnodes: 1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
set useMdp=t
:: AMSubnode & MDSubnode are used in highway_assignment_parallel.bat/s
set AMSubnode=1-4
set MDSubnode=2-4
:: subnode used in transit fare and transit assignment
:: We no longer use IDP in transit skimming, since it would require 16 cores
set subnode=1-3

:: This command will
:: 1) time the model run (using timethis.exe and the double quotes)
:: 2) redirect standard output and standard error to a file
:: 3) Use the tee command so that stderr & stdout are sent both to the file and the
screen

timethis "%runbat% %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt

:: Open up the file containing the stderr and stdout
if exist %root%\%scenar%\%scenar%_fulloutput.txt start
%root%\%scenar%\%scenar%_fulloutput.txt

:: Look four errors in the reports and output files
call searchForErrs.bat %scenar%
:: Open up the file containing any errors found
if exist %root%\%scenar%\%scenar%_searchForErrs.txt start
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Open up other report files
if exist %root%\%scenar%\i4_Highway_Assignment.rpt start
%root%\%scenar%\i4_Highway_Assignment.rpt
if exist %root%\%scenar%\i4_mc_NL_summary.txt start
%root%\%scenar%\i4_mc_NL_summary.txt
if exist %root%\%scenar%\i4_Assign_Output.net start
%root%\%scenar%\i4_Assign_Output.net
cd %scenar%
start powershell.exe -noexit -Command get-content i4_ue*AM_nonHov*txt -tail 1; get-
content i4_ue*AM_hov*txt -tail 1; get-content i4_ue*PM_nonHov*txt -tail 1; get-
content i4_ue*PM_hov*txt -tail 1; get-content i4_ue*MD*txt -tail 1; get-content
i4_ue*NT*txt -tail 1
cd ..
move_temp_files_v6.bat %scenar%

:: Cleanup
set root=
set scenar=
set runbat=
set useIdp=
set useMdp=
set AMSubnode=
set MDSubnode=
set subnode=
```

### 1.12 run\_ModelSteps\_2045\_final.bat

```

:: Version 2.3.75
:: 2018-08-09 Thu 15:50:04

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2045
set _alt_=Ver2.3.75_2045_Final
:: Maximum number of user equilibrium iterations used in traffic assignment
:: User should not need to change this. Instead, change _relGap_ (below)
set _maxUeIter_=1000

:: Set transit constraint path and files
:: Current year used to set the constraint = 2020

set _tspath_=

:: UE relative gap threshold: Progressive (10^-2 for pp-i2, 10^-3 for i3, & 10^-4
for i4)
:: Set the value below

rem ===== Pump Prime Iteration =====

set _iter_=pp
set _prev_=pp
set _relGap_=0.01

REM call ArcPy_Walkshed_Process.bat %1
call Set_CPI.bat %1
call PP_Highway_Build.bat %1
call PP_Highway_Skims.bat %1
call Transit_Skim_All_Modes_Parallel.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call PP_Auto_Drivers.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 1 =====

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 2 =====

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1

```

```

call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 3 =====

set _iter_=i3
set _prev_=i2
set _relGap_=0.001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Iteration 4 =====

set _iter_=i4
set _prev_=i3
set _relGap_=0.0001

call Transit_Skim_All_Modes_Parallel.bat %1
call Transit_Fare.bat %1
call Trip_Generation.bat %1
call Trip_Distribution.bat %1
call Mode_Choice_Parallel.bat %1
call Auto_Driver.bat %1
call Time-of-Day.bat %1
call Highway_Assignment_Parallel.bat %1
call Average_Link_Speeds.bat %1
call Highway_Skims.bat %1

:: rem ===== Transit assignment =====
@echo Starting Transit Assignment Step
@date /t & time/t

call Transit_Assignment_Parallel.bat %1
call TranSum.bat %1

@echo End of batch file
@date /t & time/t
:: rem ===== End of batch file =====

REM cd %1
REM copy *.txt MDP_%useMDP%\*.txt
REM copy *.rpt MDP_%useMDP%\*.rpt
REM copy *.log MDP_%useMDP%\*.log
REM CD..

set _year_=
set _alt_=
set _iter_=
set _prev_=
set _maxUeIter_=
set _relGap_=

```

## 2 'Pump-Prime' Iterations

### 2.1 ArcPy\_Walkshed\_Process.bat

```
@ECHO OFF
IF [%1] == [] goto usage
CD %1

SET orig_dir=%CD%
SET WalkshedDirName=Transit_Walksheds_GIS

:: Python Directory
::   Cube 6.1.0 SP1 comes with ArcGIS 10.1 runtime
::   Cube 6.0.2 comes with ArcGIS 9.3.1 SP2 runtime and is not supported
here
::   Python 2.6 will support ArcPy upto ArcGIS 10.0. Python 2.7 is needed to
support ArcPy with ArcGIS 10.1.

SET python_bindir=0

:: Look in C drive
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python27\ArcGIS10.5 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python27\ArcGIS10.4 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python27\ArcGIS10.3 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python27\ArcGIS10.2 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python27\ArcGIS10.1 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath C:\Python26\ArcGIS10.0 python_bindir

:: Look in D drive
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python27\ArcGIS10.4 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python27\ArcGIS10.4 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python27\ArcGIS10.3 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python27\ArcGIS10.2 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python27\ArcGIS10.1 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath D:\Python26\ArcGIS10.0 python_bindir

:: Look in E drive
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python27\ArcGIS10.4 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python27\ArcGIS10.4 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python27\ArcGIS10.3 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python27\ArcGIS10.2 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python27\ArcGIS10.1 python_bindir
IF %python_bindir%==0 CALL:CheckPythonPath E:\Python26\ArcGIS10.0 python_bindir

:: Python should be found by now
IF %python_bindir%==0 GOTO:error-dependency

:DoSteps
ECHO.
ECHO.
ECHO. Using Python from Directory = %python_bindir%
CD /D %orig_dir%
ECHO.
ECHO.

:: Create directories

ECHO.
ECHO.1) Creating Subdirectories ...
ECHO.

IF NOT EXIST "inputs\%WalkshedDirName%\input" MKDIR
"inputs\%WalkshedDirName%\input"
```

```
IF NOT EXIST "inputs\%WalkshedDirName%\output" MKDIR
"inputs\%WalkshedDirName%\output"

:: Change Working Directory

DEL /F /Q /S inputs\%WalkshedDirName%\*
CD /D inputs\%WalkshedDirName%

:: Prepare Inputs (if *.Lin files exist then use PT else use TRNBUILD)

ECHO.
ECHO.2) Preparing Inputs ...
ECHO.

if exist "..\ModelAM.Lin" (
    ECHO. using PT line files
    start /w Voyager.exe
    ..\..\..\scripts\MWCOG_Prepare_Inputs_to_Walkshed_Process_PT.s /start -Pvoya
    -S"%orig_dir%\inputs\%WalkshedDirName%"
    if errorlevel 2 goto error
) else (
    ECHO. using TRNBUILD line files
    start /w Voyager.exe
    ..\..\..\scripts\MWCOG_Prepare_Inputs_to_Walkshed_Process_TRNBUILD.s /start -Pvoya
    -S"%orig_dir%\inputs\%WalkshedDirName%"
    if errorlevel 2 goto error
)

:: Create Walksheds

ECHO.
ECHO.3) Launching ArcPy-based Walkshed Process ...
ECHO.

%python_bindir%\python.exe ..\..\..\scripts\MWCOG_ArcPy_Walkshed_Process.py
if errorlevel 1 goto error

:: Copy AreaWalk.txt file

ECHO.
ECHO.4) Copying AreaWalk.txt / PercentWalk.txt File(s) ...
ECHO.

:: Backup existing copies as "Old"

IF EXIST ..\AreaWalk.txt COPY /Y ..\AreaWalk.txt
..\AreaWalk_Old.txt
IF EXIST ..\PercentWalk.txt COPY /Y ..\PercentWalk.txt
..\PercentWalk_Old.txt

:: Now install and overwrite existing copies

IF EXIST output\AreaWalk.txt COPY /Y output\AreaWalk.txt ..\
if errorlevel 1 goto error

IF EXIST output\PercentWalk.txt COPY /Y output\PercentWalk.txt ..\
if errorlevel 1 goto error

:: Copy Walkshed MXD

ECHO.
ECHO.5) Copying ArcGIS MXD File ...
ECHO.

COPY /Y ..\..\..\scripts\MWCOG_ArcPy_Walkshed_Process_TEMPLATE.mxd
MWCOG_ArcPy_Walksheds_%1.mxd
if errorlevel 1 goto error
```



```

:: Change to Original Directory
CD /D %orig_dir%
ECHO.
ECHO.

:: Done

ECHO. Process Complete!
GOTO end

:-----
:-----
:-----

:error
ECHO.
ECHO.
ECHO. ERROR: Error in Walkshed Process
ECHO.
ECHO.
PAUSE
:end
CD..
GOTO:EOF

:usage
ECHO.
ECHO. Error: No Folder Name Provided
ECHO. This batch file requires a folder name.
ECHO.
PAUSE
GOTO:EOF

:error-dependency
ECHO.
ECHO.
ECHO ERROR: DEPENDENCIES NOT SATISFIED (Do you have Cube 6.1.0 SP1 or above (with
ArcGIS runtime)?)
ECHO.
ECHO.
goto error

:-----
:--- BEGIN Function CheckPythonPath
:-----

:CheckPythonPath      -- This function checks and sets python path
:                    -- %~1: Path containing Python.exe
:                    -- %~2: Variable containing status
SET myPath=0

ECHO    Searching for Python in Path %~1
IF EXIST %~1 (
    CD /D %~1
    IF EXIST python.exe (
        SET myPath=%~1
        ECHO    Found Python in Path %~1
    ) ELSE ( SET myPath=0 )
) ELSE ( SET myPath=0 )

SET "%~2=%myPath%"

GOTO:EOF
:-----
:--- END Function CheckPythonPath
:-----

```

## 2.2 set\_CPI.bat

```

cd %1

REM  CPI Establishment

if exist voya*.*      del voya*.*
if exist set_CPI.rpt del set_CPI.rpt
start /w Voyager.exe ..\scripts\set_CPI.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
    if exist voya*.prn copy voya*.prn set_CPI.rpt /y

if exist voya*.*      del voya*.*
if exist set_factors.rpt del set_factors.rpt
start /w Voyager.exe ..\scripts\set_factors.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
    if exist voya*.prn copy voya*.prn set_factors.rpt /y
goto end

:error
REM Processing Error.....
PAUSE
:end
cd..

```

## 2.2 PP\_Highway\_Build.bat

```

cd %1

REM  Highway Network Building

if exist voya*.*      del voya*.*
if exist AreaType_File.rpt del AreaType_File.rpt
start /w Voyager.exe ..\scripts\AreaType_File.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn AreaType_File.rpt /y

if exist voya*.*      del voya*.*
if exist highway_build_toll.rpt del V2.3_highway_build.rpt
start /w Voyager.exe ..\scripts\V2.3_highway_build.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn V2.3_highway_build.rpt /y
if exist temp.net      del temp.net

goto end

:error
REM Processing Error.....
PAUSE
:end
cd..

```

## 2.3 PP\_Highway\_Skims.bat

```

:: 2018-03-30 RN Add lines to execute a check on whether transit stations are
accessible

```

## Appendix B Batch file

```
CD %1
set _iterOrder_=initial

REM Highway Skims

:: COPY ZONEHWY.NET TEMPORARILY TO PPHWY.NET

if exist ZONEHWY.NET COPY ZONEHWY.NET PP_HWY.NET /y

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Highway_Skims_am.rpt del
%_iter%_%_iterOrder%_Highway_Skims_am.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_am.s /start -Pvoya -S.%\1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Highway_Skims_am.rpt /y

ping -n 11 127.0.0.1 > nul

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Highway_Skims_md.rpt del
%_iter%_%_iterOrder%_Highway_Skims_md.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_md.s /start -Pvoya -S.%\1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Highway_Skims_md.rpt /y

:: Additional Steps per the Nested Logit
:: modnet.bat / Highway_Skims_Mod.bat / JoinSkims.bat ===

REM Utility - Convert dummy centroid connectors

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_ModNet.rpt del %_iter%_%_iterOrder%_ModNet.rpt
start /w Voyager.exe ..\scripts\modnet.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_ModNet.rpt /y

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Highway_Skims_mod_am.rpt del
%_iter%_%_iterOrder%_Highway_Skims_mod_am.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_mod_am.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Highway_Skims_Mod_am.rpt
/y

:: Check whether transit stations are accessible
if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_CheckStationAccess.rpt del
%_iter%_%_iterOrder%_CheckStationAccess.rpt
start /w Voyager.exe ..\scripts\CheckStationAccess.s /start -Pvoya -S.%\1
if errorlevel 2 (echo STATION CENTROIDS WITHOUT SKIMS. PLEASE CHECK THE NETWORK &&
goto stationerr)
if errorlevel 3 (echo STATION CENTROIDS WITHOUT SKIMS. PLEASE CHECK THE NETWORK &&
goto stationerr)
if exist voya*.prn copy voya*.prn CheckStationAccess.rpt /y

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Highway_Skims_mod_md.rpt del
%_iter%_%_iterOrder%_Highway_Skims_mod_md.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_mod_md.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Highway_Skims_Mod_md.rpt
/y

:: ----- Save initial highway skims to special names for later checking
```

```
if exist pp_am_SOV.SKM copy pp_am_SOV.SKM pp_am_SOV_Initial.SKM /y
if exist pp_md_SOV.SKM copy pp_md_SOV.SKM pp_md_SOV_Initial.SKM /y
if exist pp_am_HOV2.SKM copy pp_am_HOV2.SKM pp_am_HOV2_Initial.SKM /y
if exist pp_md_HOV2.SKM copy pp_md_HOV2.SKM pp_md_HOV2_Initial.SKM /y
if exist pp_am_HOV3.SKM copy pp_am_HOV3.SKM pp_am_HOV3_Initial.SKM /y
if exist pp_md_HOV3.SKM copy pp_md_HOV3.SKM pp_md_HOV3_Initial.SKM /y

if exist pp_am_SOV_mod.SKM copy pp_am_SOV_mod.SKM pp_am_SOV_mod_Initial.SKM /y
if exist pp_md_SOV_mod.SKM copy pp_md_SOV_mod.SKM pp_md_SOV_mod_Initial.SKM /y
if exist pp_am_HOV2_mod.SKM copy pp_am_HOV2_mod.SKM pp_am_HOV2_mod_Initial.SKM /y
if exist pp_md_HOV2_mod.SKM copy pp_md_HOV2_mod.SKM pp_md_HOV2_mod_Initial.SKM /y
if exist pp_am_HOV3_mod.SKM copy pp_am_HOV3_mod.SKM pp_am_HOV3_mod_Initial.SKM /y
if exist pp_md_HOV3_mod.SKM copy pp_md_HOV3_mod.SKM pp_md_HOV3_mod_Initial.SKM /y

:: ----- the PP_?.SKM files will be overwritten after the skimming
:: ----- of the PP Highway assignment network

REM Utility - Join Highway Skims

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_JoinSkims.rpt del
%_iter%_%_iterOrder%_JoinSkims.rpt
start /w Voyager.exe ..\scripts\joinskims.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_JoinSkims.rpt /y

:: DELETE TEMPORARY pphwy.net, THIS WILL BE CREATED AFTER the PP HIGHWAY ASSIGNMENT

rem if exist PP_HWY.NET del PP_HWY.NET

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Remove_PP_Speed.rpt del
%_iter%_%_iterOrder%_Remove_PP_Speed.rpt
start /w Voyager.exe ..\scripts\Remove_PP_Speed.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Remove_PP_Speed.rpt /y

goto end

:stationerr
PAUSE&EXIT

:error
REM Processing Error....
PAUSE
:end
CD..
set _iterOrder_=
```

## 2.4 PP\_Auto\_Drivers.bat

```
CD %1

REM Pump Prime Auto Driver Trips

if exist voya*.* del voya*.*
if exist %_iter%_%_iterOrder%_Auto_Drivers.rpt del %_iter%_%_iterOrder%_Auto_Drivers.rpt

start /w Voyager.exe ..\scripts\PP_Auto_Drivers.s /start -Pvoya -S.%\1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter%_%_iterOrder%_Auto_Drivers.rpt /y
```

## Appendix B Batch file

```
..\software\extrtab    %_iter_%_Auto_Drivers.rpt
copy extrtab.out      %_iter_%_Auto_Drivers.tab /y
del extrtab.out
```

```
goto end
:error
REM Processing Error....
PAUSE
:end
CD..
```

### 3 'Standard' Iterations (1-4)

#### 3.1 Transit\_Skim\_All\_Modes\_Parallel.bat

```
:: Transit Skimming for All Submodes
:: updated 4/27/07 copy sta_tpp.bse from inputs to output subdir.
:: updated 6/15/11 runs walkacc process for pp iteration only
:: updated 5/11/16 Update autoacc and local-bus in-vehicle speed degradation
CD %1

:: Delete previous iteration highway skim files for Transit Skimming (if files
exist)

if exist am_sov_mod.skm del am_sov_mod.skm
if exist md_sov_mod.skm del md_sov_mod.skm

:: Set up current iteration highway skim files for transit Skimming

if exist %_prev_%_am_sov_mod.skm copy %_prev_%_am_sov_mod.skm am_sov_mod.skm /y
if exist %_prev_%_md_sov_mod.skm copy %_prev_%_md_sov_mod.skm md_sov_mod.skm /y

if exist voya*.* del voya*.*
if exist parker.rpt del parker.rpt
start /w Voyager.exe ..\scripts\parker.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn parker.rpt /y

if %_iter_%==pp goto runwalk
goto skipwalk

:runwalk

::copy transit lines and support files from the inputs subdir.
copy inputs\*.TB /y
copy inputs\mfarel.a1 /y

::develop walk access links.
if exist voya*.* del voya*.*
if exist walkacc.rpt del walkacc.rpt
start /w Voyager.exe ..\scripts\walkacc.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn walkacc.rpt /y

::adjust local bus run times by applying bus speed degradation factors.
if exist voya*.* del voya*.*
if exist Adjust_Runtime.rpt del Adjust_Runtime.rpt
start /w Voyager.exe ..\scripts\Adjust_Runtime.s /start -Pvoya -S..\%1
```

```
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn Adjust_Runtime.rpt /y

:skipwalk
if exist voya*.* del voya*.*
if exist autoacc5.rpt del autoacc5.rpt
start /w Voyager.exe ..\scripts\autoacc5.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn autoacc5.rpt /y

rem ----- Do some cleaning up -----
:: del /F ..\%1\hov2m%_prev_%am.skm
:: del /F ..\%1\hov2m%_prev_%op.skm
:: del /F ..\%1\hov3m%_prev_%am.skm
:: del /F ..\%1\hov3m%_prev_%op.skm
:: del /F ..\%1\ttpl1*.*

CD..

:: =====
:: = Transit Skimming Section =
:: =====

:: Transit Network Building (Final) Commuter Rail

if %useMDP%==t goto Parallel_Processing
if %useMDP%==T goto Parallel_Processing
@echo Start Transit Skims
REM If only one CPU, run the four skims sequentially

START /wait Transit_Skim_LineHaul_Parallel.bat %1 CR

REM Transit Network Building (Final) Metrorail
START /wait Transit_Skim_LineHaul_Parallel.bat %1 MR

REM Transit Network Building (Final) All Bus
START /wait Transit_Skim_LineHaul_Parallel.bat %1 AB

REM Transit Network Building (Final) Bus+MetroRail
START /wait Transit_Skim_LineHaul_Parallel.bat %1 BM

goto Transit_Skims_Are_Done

:Parallel_Processing
@echo Start Transit Skim - Parallel

START Transit_Skim_LineHaul_Parallel.bat %1 CR

REM Transit Network Building (Final) Metrorail
START Transit_Skim_LineHaul_Parallel.bat %1 MR

REM Transit Network Building (Final) All Bus
START Transit_Skim_LineHaul_Parallel.bat %1 AB

REM Transit Network Building (Final) Bus+MetroRail
START /wait Transit_Skim_LineHaul_Parallel.bat %1 BM

:Transit_Skims_Are_Done

CD %1

goto checkIfDone

:waitForMC
@ping -n 11 127.0.0.1

:checkIfDone
```

## Appendix B Batch file

```
@REM Check file existence to ensure that there are no errors
if exist Transit_Skims_CR.err echo Error in Transit_Skims_CR && goto error
if exist Transit_Skims_MR.err echo Error in Transit_Skims_MR && goto error
if exist Transit_Skims_AB.err echo Error in Transit_Skims_AB && goto error
if exist Transit_Skims_BM.err echo Error in Transit_Skims_BM && goto error

@REM Check to ensure that each of the batch processes have finished successfully, if
not wait.
if not exist Transit_Skims_CR.done goto waitForMC
if not exist Transit_Skims_MR.done goto waitForMC
if not exist Transit_Skims_AB.done goto waitForMC
if not exist Transit_Skims_BM.done goto waitForMC

REM @type CR.txt
REM @type MR.txt
REM @type AB.txt
REM @type BM.txt

REM CD %1
REM Transit Network Accessibility File developement (For Demographic Models)

if exist voya*. * del voya*. *
if exist %_iter_%_TRANSIT_Accessibility.RPT del %_iter_%_TRANSIT_Accessibility.RPT
start /w Voyager.exe ..\scripts\transit_Accessibility.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_TRANSIT_Accessibility.RPT /y
goto end
:error
REM Processing Error.....
PAUSE
:end

CD..
```

### 3.2 Transit\_Fare.bat

```
::-----
:: Version 2.3 Transit Fare Process
::-----

CD %1

if exist voya*. * del voya*. *
if exist %_iter_%_prefarV23.rpt del %_iter_%_prefarV23.rpt
start /w Voyager.exe ..\scripts\prefarV23.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_prefarV23.rpt /y

if exist voya*. * del voya*. *
if exist %_iter_%_Metrorail_skims.rpt del %_iter_%_Metrorail_skims.rpt
start /w Voyager.exe ..\scripts\Metrorail_skims.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Metrorail_skims.rpt /y

if exist voya*. * del voya*. *
if exist %_iter_%_MFARE1.rpt del %_iter_%_MFARE1.rpt
start /w Voyager.exe ..\scripts\MFARE1.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_MFARE1.rpt /y
```

```
if exist voya*. * del voya*. *
if exist %_iter_%_MFARE2.rpt del %_iter_%_MFARE2.rpt
Cluster.exe MFCOG %subnode% start exit
start /w Voyager.exe ..\scripts\MFARE2.s /start -Pvoya -S..\%1
if errorlevel 1 goto error ; Moved from line 37 to correctly report errors on
MFARE2 report file
Cluster.exe MFCOG %subnode% close exit
echo %ERRORLEVEL% ; Added to report Errorlevel return code
if exist voya*.prn copy voya*.prn %_iter_%_MFARE2.rpt /y
```

```
if exist voya*. * del voya*. *
if exist %_iter_%_Assemble_Skims_MR.rpt del %_iter_%_Assemble_Skims_MR.rpt
start /w Voyager.exe ..\scripts\Assemble_Skims_MR.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Assemble_Skims_MR.rpt /y
```

```
if exist voya*. * del voya*. *
if exist %_iter_%_Assemble_Skims_BM.rpt del %_iter_%_Assemble_Skims_BM.rpt
start /w Voyager.exe ..\scripts\Assemble_Skims_BM.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Assemble_Skims_BM.rpt /y
```

```
if exist voya*. * del voya*. *
if exist %_iter_%_Assemble_Skims_AB.rpt del %_iter_%_Assemble_Skims_AB.rpt
start /w Voyager.exe ..\scripts\Assemble_Skims_AB.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Assemble_Skims_AB.rpt /y
```

```
if exist voya*. * del voya*. *
if exist %_iter_%_Assemble_Skims_CR.rpt del %_iter_%_Assemble_Skims_CR.rpt
start /w Voyager.exe ..\scripts\Assemble_Skims_CR.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Assemble_Skims_CR.rpt /y
```

goto end

```
:error
REM Processing Error.....
PAUSE
:end
```

CD..

### 3.3 Trip\_Generation.bat

```
::-----
:: Version 2.3 Trip Generation Process --
::-----

CD %1

if exist voya*. * del voya*. *
if exist %_iter_%_Demo_Models.rpt del %_iter_%_Demo_Models.rpt
start /w Voyager.exe ..\scripts\Demo_Models.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Demo_Models.rpt /y
```

## Appendix B Batch file

```
if exist voya*.* del voya*.*
if exist %_iter_%_Trip_Generation.rpt del %_iter_%_Trip_Generation.rpt
start /w Voyager.exe ..\scripts\Trip_Generation.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Trip_Generation.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Trip_Generation_Summary.rpt del
%_iter_%_Trip_Generation_Summary.rpt
start /w Voyager.exe ..\scripts\Trip_Generation_Summary.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Trip_Generation_Summary.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Truck_Com_Trip_Generation.rpt del
%_iter_%_Truck_Com_Trip_Generation.rpt
start /w Voyager.exe ..\scripts\Truck_Com_Trip_Generation.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Truck_Com_Trip_Generation.rpt /y

goto end

:error
REM Processing Error.....
PAUSE
:end

CD..
```

### 3.4 Trip\_Distribution.bat

```
::-----
:: Version 2.3 Trip Distribution Process
::-----

CD %1

if exist voya*.* del voya*.*
if exist %_iter_%_Prepare_Ext_Auto_End.s.rpt del %_iter_%_Prepare_Ext_Auto_End.s.rpt
start /w Voyager.exe ..\scripts\Prepare_Ext_Auto_End.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Prepare_Ext_Auto_End.s.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Prepare_Ext_ComTruck_End.s.rpt del
%_iter_%_Prepare_Ext_ComTruck_End.s.rpt
start /w Voyager.exe ..\scripts\Prepare_Ext_ComTruck_End.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Prepare_Ext_ComTruck_End.s.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Trip_Distribution_External.rpt del
%_iter_%_Trip_Distribution_External.rpt
start /w Voyager.exe ..\scripts\Trip_Distribution_External.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Trip_Distribution_External.rpt /y
..\software\extrtab %_iter_%_Trip_Distribution_External.rpt
copy extrtab.out %_iter_%_Trip_Distribution_External.tab /y
```

```
del extrtab.out

if exist voya*.* del voya*.*
if exist %_iter_%_Prepare_Internal_End.s.rpt del %_iter_%_Prepare_Internal_End.s.rpt
start /w Voyager.exe ..\scripts\Prepare_Internal_End.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Prepare_Internal_End.s.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Trip_Distribution_Internal.rpt del
%_iter_%_Trip_Distribution_Internal.rpt
start /w Voyager.exe ..\scripts\Trip_Distribution_Internal.s /start -Pvoya -S..\%1
if errorlevel 2 goto error REM changed from 1 to 2 due to Cube 6.4 warning of
unbalanced Ps & As
if exist voya*.prn copy voya*.prn %_iter_%_Trip_Distribution_Internal.rpt /y
..\software\extrtab %_iter_%_Trip_Distribution_Internal.rpt
copy extrtab.out %_iter_%_Trip_Distribution_Internal.tab /y
del extrtab.out

goto end

:error
REM Processing Error.....
PAUSE
:end

CD..
```

### 3.5 Mode\_Choice\_Parallel.bat

```
@rem Nested Logit Mode Choice Model Application
@echo off
CD %1

@rem Copy iteration-specific inputs to generic names

if exist %_iter_%_hbw_NL.ptt copy %_iter_%_hbw_NL.ptt HBW_INCOME.PTT /y
if exist %_iter_%_hbs_NL.ptt copy %_iter_%_hbs_NL.ptt HBS_INCOME.PTT /y
if exist %_iter_%_hbo_NL.ptt copy %_iter_%_hbo_NL.ptt HBO_INCOME.PTT /y
if exist %_iter_%_nhw_NL.ptt copy %_iter_%_nhw_NL.ptt NHW_INCOME.PTT /y
if exist %_iter_%_nho_NL.ptt copy %_iter_%_nho_NL.ptt NHO_INCOME.PTT /y

if exist %_prev_%_hwy_AM.SKM copy %_prev_%_hwy_AM.SKM HWYAM.SKM /y
if exist %_prev_%_hwy_OP.SKM copy %_prev_%_hwy_OP.SKM HWYOP.SKM /y

if exist %_iter_%_TRNAM_CR.SKM copy %_iter_%_TRNAM_CR.SKM TRNAM_CR.SKM /y
if exist %_iter_%_TRNAM_AB.SKM copy %_iter_%_TRNAM_AB.SKM TRNAM_AB.SKM /y
if exist %_iter_%_TRNAM_MR.SKM copy %_iter_%_TRNAM_MR.SKM TRNAM_MR.SKM /y
if exist %_iter_%_TRNAM_BM.SKM copy %_iter_%_TRNAM_BM.SKM TRNAM_BM.SKM /y

if exist %_iter_%_TRNOP_CR.SKM copy %_iter_%_TRNOP_CR.SKM TRNOP_CR.SKM /y
if exist %_iter_%_TRNOP_AB.SKM copy %_iter_%_TRNOP_AB.SKM TRNOP_AB.SKM /y
if exist %_iter_%_TRNOP_MR.SKM copy %_iter_%_TRNOP_MR.SKM TRNOP_MR.SKM /y
if exist %_iter_%_TRNOP_BM.SKM copy %_iter_%_TRNOP_BM.SKM TRNOP_BM.SKM /y

if %useMDP%==t goto Parallel_Processing
if %useMDP%==T goto Parallel_Processing

REM If only one CPU, run the five purposes sequentially
@echo Starting Mode Choice
@date /t & time/t

START /high /wait CALL ../MC_purp.bat %1 NHO
```

## Appendix B Batch file

```
START /high /wait CALL ../MC_purp.bat %1 HBS
START /high /wait CALL ../MC_purp.bat %1 NHW
START /high /wait CALL ../MC_purp.bat %1 HBO
START /high /wait CALL ../MC_purp.bat %1 HBW

goto Mode_Choice_is_Done

:Parallel_Processing
@echo Starting Mode Choice - Parallel Processing
@date /t & time/t

START /high CALL ../MC_purp.bat %1 NHO
@ping -n 11 127.0.0.1
START /high CALL ../MC_purp.bat %1 HBS
@ping -n 11 127.0.0.1
START /high CALL ../MC_purp.bat %1 NHW
@ping -n 11 127.0.0.1
START /high CALL ../MC_purp.bat %1 HBO
@ping -n 11 127.0.0.1
START /high /wait CALL ../MC_purp.bat %1 HBW

goto checkIfDone

:waitForMC
@ping -n 11 127.0.0.1

:checkIfDone

@REM Check file existence to ensure that there are no errors
if exist HBO.err echo Error in HBO MC && goto error
if exist HBS.err echo Error in HBS MC && goto error
if exist HBW.err echo Error in HBW MC && goto error
if exist NHO.err echo Error in NHO MC && goto error
if exist NHW.err echo Error in NHW MC && goto error

@REM Check to ensure that each of the batch processes have finished successfully, if
not wait.
if not exist HBO.done goto waitForMC
if not exist HBS.done goto waitForMC
if not exist HBW.done goto waitForMC
if not exist NHO.done goto waitForMC
if not exist NHW.done goto waitForMC

:Mode_Choice_is_Done
@rem - This step is to collect all the output from the MC to the log file.
@type HBW.txt
@type HBS.txt
@type HBO.txt
@type NHW.txt
@type NHO.txt

@echo Finished Mode Choice
@date /t & time/t

@rem COPY GENERIC MODE CHOICE OUTPUT FILES
@rem TO INTERATION-SPECIFIC NAMES

if exist HBW_NL_MC.MTT copy HBW_NL_MC.MTT %_iter_%_HBW_NL_MC.MTT /y
if exist HBS_NL_MC.MTT copy HBS_NL_MC.MTT %_iter_%_HBS_NL_MC.MTT /y
if exist HBO_NL_MC.MTT copy HBO_NL_MC.MTT %_iter_%_HBO_NL_MC.MTT /y
if exist NHW_NL_MC.MTT copy NHW_NL_MC.MTT %_iter_%_NHw_NL_MC.MTT /y
if exist NHO_NL_MC.MTT copy NHO_NL_MC.MTT %_iter_%_NHO_NL_MC.MTT /y

@ping -n 11 127.0.0.1
```

```
if exist voya*.* del voya*.*
if exist %_iter_%_MC_NL_SUMMARY.rpt del %_iter_%_MC_NL_SUMMARY.rpt

start /w Voyager.exe ../scripts\mc_NL_summary.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_mc_NL_summary.rpt /y
if exist voya*.prn copy voya*.prn temp.rpt /y
../software\extrtab temp.rpt
if exist extrtab.out copy extrtab.out %_iter_%_mc_NL_summary.tab /y
if exist extrtab.out del extrtab.out
if exist temp.rpt del temp.rpt
if exist *.tbl copy *.tbl %_iter_%_mc_NL_summary.txt /y
if exist *.tbl del *.tbl

goto end

:error
@echo Error in Mode Choice
@rem Processing Error....
PAUSE
:end
CD..
```

### 3.6 Auto\_Driver.bat

```
CD %1

REM Auto Driver Trips

if exist voya*.* del voya*.*
if exist %_iter_%_mc_Auto_Drivers.rpt del %_iter_%_mc_Auto_Drivers.rpt
start /w Voyager.exe ../scripts\mc_Auto_Drivers.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_mc_Auto_Drivers.rpt /y
if exist %_iter_%_mc_Auto_Drivers.rpt copy %_iter_%_mc_Auto_Drivers.rpt temp.dat /y
../software\extrtab temp.dat
if exist extrtab.out copy extrtab.out %_iter_%_mc_Auto_Drivers.tab /y
if exist extrtab.out del extrtab.out
if exist temp.out del temp.out

goto end

:error
REM Processing Error....
PAUSE
:end
CD..
```

### 3.7 Time-of-Day.bat

```
CD %1
CD %1
REM -- Time of Day Process ---

REM -----
REM Modeled Auto Driver Time-of-Day Trips
REM -----

if exist voya*.* del voya*.*
```

## Appendix B Batch file

```
if exist %_iter_%Time-of-Day.rpt del %_iter_%Time-of-Day.rpt
start /w Voyager.exe ..\scripts\Time-of-Day.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
copy voya*.prn %_iter_%Time-of-Day.rpt /y
copy %_iter_%Time-of-Day.rpt temp.dat /y
copy ..\software\extrtab temp.dat
copy extrtab.out %_iter_%Time-of-Day.tab /y
del temp.dat

REM -----
REM Truck and Exogenous Time-of-Day Trips
REM -----

if exist voya*.* del voya*.*
if exist %_iter_%Misc_Time-of-Day.rpt del %_iter_%Misc_Time-of-Day.rpt
start /w Voyager.exe ..\scripts\Misc_Time-of-Day.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
copy voya*.prn %_iter_%Misc_Time-of-Day.rpt /y
copy %_iter_%Misc_Time-of-Day.rpt temp.dat /y
copy ..\software\extrtab temp.dat
copy extrtab.out %_iter_%Misc_Time-of-Day.tab /y
del extrtab.out
del temp.dat

REM -----
REM Prepare trips for highway assignment
REM -----

if exist voya*.* del voya*.*
if exist %_iter_%Prepare_Trip_Tables_for_Assignment.rpt del
%_iter_%Prepare_Trip_Tables_for_Assignment.rpt
start /w Voyager.exe ..\scripts\Prepare_Trip_Tables_for_Assignment.s /start -Pvoya
-S..\%1
if errorlevel 1 goto error
copy voya*.prn %_iter_%Prepare_Trip_Tables_for_Assignment.rpt /y
copy %_iter_%Prepare_Trip_Tables_for_Assignment.rpt temp.dat
/
copy ..\software\extrtab temp.dat
copy extrtab.out %_iter_%Prepare_Trip_Tables_for_Assignment.tab /y
del extrtab.out
del temp.dat

goto end

:error
REM Processing Error....
PAUSE
:end
CD..
```

### 3.8 Highway\_Assignment\_Parallel.bat

```
CD %1

REM Highway Assignment

if exist voya*.* del voya*.*
if exist %_iter_%Highway_Assignment.rpt del %_iter_%Highway_Assignment.rpt

Cluster.exe AM %AMsubnode% start exit
Cluster.exe MD %MDsubnode% start exit
```

```
start /w Voyager.exe ..\scripts\Highway_Assignment_Parallel.s /start -Pvoya -S..\%1
if errorlevel 2 goto error ; Moved from below Cluster + Changed to 2 (due to crash
on Cube 6.4.1)
Cluster.exe AM %AMsubnode% close exit
Cluster.exe MD %MDsubnode% close exit
```

```
copy Voya*.prn %_iter_%Highway_Assignment.rpt /y

goto end
:error
REM Processing Error....
PAUSE
:end
CD..
```

### 3.9 Average\_Link\_Speeds.bat - Iterations (2- 4)

```
CD %1

REM Average Link Speeds

:: Write loaded links file from assignment to new file
:: current iteration speeds will be removed and rewritten with averaged speeds below

if exist %_iter_%HWY.net copy %_iter_%HWY.net %_iter_%Assign_Output.net /y
if exist %_iter_%HWY.net copy %_iter_%HWY.net %_iter_%HWY.tem1 /y

if exist voya*.* del voya*.*
if exist %_iter_%Average_Link_Speeds.rpt del %_iter_%Average_Link_Speeds.rpt
start /w Voyager.exe ..\scripts\Average_Link_Speeds.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%Average_Link_Speeds.rpt /y

:: Now copy over the original Loaded file with revised file containing new/avg
speeds
:: Note: the original file from assignment is maintained as
%_iter_%Assigned_%HWY.net

if exist %_iter_%Averaged_HWY.net copy %_iter_%Averaged_HWY.net %_iter_%HWY.net
/y

goto end
:error
REM Processing Error....
PAUSE
:end
CD..
```

### 3.10 Highway\_Skims.bat

```
CD %1

REM Highway Skims

if exist voya*.* del voya*.*
if exist %_iter_%Highway_Skims_am.rpt del %_iter_%Highway_Skims_am.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_am.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%Highway_Skims_am.rpt /y
```

## Appendix B Batch file

```
if exist voya*.* del voya*.*
if exist %_iter_%_Highway_Skims_md.rpt del %_iter_%_Highway_Skims_md.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_md.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Highway_Skims_md.rpt /y

: Additional Steps per the Nested Logit
:: modnet.bat / Highway_Skims_Mod.bat / JoinSkims.bat ===

REM Utility - Convert dummy centroid connectors

if exist voya*.* del voya*.*
if exist %_iter_%_ModNet.rpt del %_iter_%_ModNet.rpt
start /w Voyager.exe ..\scripts\modnet.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_ModNet.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Highway_Skims_mod_am.rpt del %_iter_%_Highway_Skims_mod_am.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_mod_am.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Highway_Skims_Mod_am.rpt /y

if exist voya*.* del voya*.*
if exist %_iter_%_Highway_Skims_mod_md.rpt del %_iter_%_Highway_Skims_mod_md.rpt
start /w Voyager.exe ..\scripts\Highway_Skims_mod_md.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Highway_Skims_Mod_md.rpt /y

REM Utility - Join Highway Skims

if exist voya*.* del voya*.*
if exist %_iter_%_JoinSkims.rpt del %_iter_%_JoinSkims.rpt
start /w Voyager.exe ..\scripts\joinskims.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn copy voya*.prn %_iter_%_JoinSkims.rpt /y

goto end
:error
REM Processing Error....
PAUSE
:end
CD..
```

## 4 Transit assignment Batch Files

### 4.1 Transit\_Assignment\_Parallel.bat

```
CD %1

:: Combine Mode Choice Output for Transit Assignment

if exist voya*.* del voya*.*
if exist %_iter_%_Combine_Tables_For_TrAssign_Parallel.RPT del
%_iter_%_Combine_Tables_For_TrAssign_Parallel.RPT

Cluster.exe MFCOG %subnode% start exit
```

```
start /w voyager.exe ..\Scripts\Combine_Tables_For_TrAssign_Parallel.s /start -Pvoya
-S..\%1
Cluster.exe MFCOG %subnode% close exit

if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Combine_Tables_For_TrAssign_Parallel.RPT
/y
CD..

:: =====
:: = Transit Assignment Section =
:: =====

:: Transit Assignment Commuter Rail

if %useMDP%==t goto Parallel_Processing
if %useMDP%==T goto Parallel_Processing
@echo Start Transit Assignments

START /wait Transit_Assignment_LineHaul_Parallel.bat %1 CR

:: Transit Assignment Metrorail
START /wait Transit_Assignment_LineHaul_Parallel.bat %1 MR

:: Transit Assignment All Bus
START /wait Transit_Assignment_LineHaul_Parallel.bat %1 AB

:: Transit Assignment Bus and Metrorail
START /wait Transit_Assignment_LineHaul_Parallel.bat %1 BM

goto Transit_Assignments_Are_Done

:Parallel_Processing
@echo Start Transit Assignments - Parallel

START Transit_Assignment_LineHaul_Parallel.bat %1 CR
@ping -n 11 127.0.0.1

:: Transit Assignment Metrorail
START Transit_Assignment_LineHaul_Parallel.bat %1 MR
@ping -n 11 127.0.0.1

:: Transit Assignment All Bus
START Transit_Assignment_LineHaul_Parallel.bat %1 AB
@ping -n 11 127.0.0.1

:: Transit Assignment Bus and Metrorail
START /wait Transit_Assignment_LineHaul_Parallel.bat %1 BM

:Transit_Assignments_Are_Done

CD %1
goto checkIfDone

:waitForMC
@ping -n 11 127.0.0.1

:checkIfDone

@REM Check file existence to ensure that there are no errors
if exist Transit_Assignment_CR.err echo Error in Transit_Assignment_CR && goto error
if exist Transit_Assignment_MR.err echo Error in Transit_Assignment_MR && goto error
if exist Transit_Assignment_AB.err echo Error in Transit_Assignment_AB && goto error
if exist Transit_Assignment_BM.err echo Error in Transit_Assignment_BM && goto error

@REM Check to ensure that each of the batch processes have finished successfully, if
not wait.
if not exist Transit_Assignment_CR.done goto waitForMC
```



## Appendix B Batch file

```
if not exist Transit_Assignment_MR.done goto waitForMC
if not exist Transit_Assignment_AB.done goto waitForMC
if not exist Transit_Assignment_BM.done goto waitForMC

goto end
:error
REM Processing Error.....
PAUSE
:end
cd..
```

### 4.2 TranSum.bat

```
:: TranSum.bat
:: To be run from the root directory (e.g., E:\modelRuns\fy13\Ver2.3.46)

REM Change to the Transum folder, under the scenario-specific folder
REM Output report files will be stored in the Transum folder
REM The Transum folder starts out empty, since station_names.dbf is stored in
Controls
CD %1\Transum

REM Copy the lineSum control files from the Controls folder to the Transum folder
copy ..\..\Controls\LineSum_*.ctl

REM Consolidate peak and off-peak volumes from transit assignment
..\..\software\LineSum.exe LineSum_Volume.ctl
if %ERRORLEVEL% == 1 goto error

REM Metrorail station access (does not include transfers)
..\..\software\LineSum.exe lineSum_MR_access.ctl
if %ERRORLEVEL% == 1 goto error

REM Metrorail line summaries
..\..\software\LineSum.exe lineSum_MR_line.ctl
if %ERRORLEVEL% == 1 goto error

goto end
:error
REM Processing Error.....
PAUSE
:end
::CD..
CD..
CD..
```

## 5 Optional/Utility Batch Files

### 5.1 SearchForErrs.bat

```
:: tms3 0:\model_dev\fy14\Ver2.3.56\searchForErrs.bat
:: 2014-04-23 Wed 19:24:09

:: Analyst: msm
```

```
:: This batch file will scan travel model reports and output files for errors
:: and other anomalies. It is designed to work with the Ver. 2.3 travel model.
:: This is to be run from the root dir, e.g., E:\modelRuns\fy12\Ver2.3.37_conf
:: It is called from the "run model" batch file, so it can make use of the
environment
:: variables in that file
:: Usage: searchForErrs.bat <name of scenario folder>
:: e.g., searchForErrs.bat 2007
::
:: Updates
:: DATE PERSON CHANGE
:: 2009-01-07 msm Searches for different versions of TP+/Voyager
:: User must supply expected version, e.g., 6.1.0

:: if searchForErrs.txt exists, copy it to new file with the same name and the date
appended to it
if exist %root%\%scenar%\%scenar%_searchForErrs.txt call datename.bat
%root%\%scenar%\%scenar%_searchForErrs.txt

:: Print header records in search output file (searchForErrs.txt)
echo ***** Searching for errors and anomalies after a travel model run
***** > %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo Program name: searchForErrs.bat >> %root%\%scenar%\%scenar%_searchForErrs.txt
2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l

:: Search *fulloutput.txt
echo ***** Searching *fulloutput.txt >> %root%\%scenar%\%scenar%_searchForErrs.txt
2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo *** Searching for cases where a file could not be found >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
(findstr /i /p /c:"Could Not Find" %1\*fulloutput.txt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
(findstr /i /p /c:"cannot find" %1\*fulloutput.txt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l

:: Search report files
echo ***** Searching report files (*.rpt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo *** Searching for evidence that TP+ (TPMAIN) is running, instead of Voyager
(PILOT) >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
(findstr /i /p /c:"TPMAIN" %1\*.rpt) >> %root%\%scenar%\%scenar%_searchForErrs.txt
2>&l
echo *** Searching for evidence of LINKO nodes that do not have XY values >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
(findstr /i /p /c:"The following LINKO nodes do not have XY values" %1\*.rpt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l

echo *** Searching for different versions of Cube Voyager >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo *** Expected version: 6.1.0 >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
findstr /i /p /r /c:"\[[0-9]\.[0-9]\.[0-9]\]" %1\*.rpt > tppVoyVers.txt
(findstr /i /p /r /v /c:"\[[6]\.[1]\.[0]\]" tppVoyVers.txt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&l

echo *** Searching for return codes greater than 1 >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&l
```

## Appendix B Batch file

```
(findstr /i /p /r /c:"ReturnCode = [2-9]" %1\*.rpt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1

echo *** Searching for fatal errors >> %root%\%scenar%\%scenar%_searchForErrs.txt
2>&1
(findstr /i /p /r /c:"F([0-9]*):" %1\*.rpt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1

echo *** Searching for warnings >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
(findstr /i /p /r /c:"W([0-9]*):" %1\*.rpt) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1

:: Search output files
echo ***** If this file (AT_override.TXT) exists, >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo *** there may be area-type overrides >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo *** (a 2nd value of zero => no override) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
(type %1\inputs\AT_override.TXT) >>
%root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
echo. >> %root%\%scenar%\%scenar%_searchForErrs.txt 2>&1
```

### 5.2 move\_temp\_files\_v6.bat

```
@echo off
:: move_temp_files_v6.bat
:: 2012-12-12 msm Batch file created
:: 2013-01-21 msm Update: Restore report files assoc. w/ SFB iters pp, i1, i2, and
i3
:: 2013-03-04 msm Update: Restore TXT, TAB files assoc. w/ SFB iters pp, i1, i2, and
i3
:: 2016-06-10 dqn Update: Restore files for the HOV2+ skims substitution technique
(HOT-lane modeling)
:: 2017-04-13 dqn Replace Cywin's head and tail functions to Powershell
:: Usage (launched from root folder): move_temp_files.bat <output_folder_name>
::
:: When running the TPB Version 2.3.48 Travel Model, in addition to the useful
:: output files, such as i4_Assign_Output.net, a number of temporary ("temp")
:: files are created, such as i3_Assign_Output.net. This program will move
:: these temp files to a folder under the output folder. For example, "2030_base"
:: will now include a subfolder called "temp_files". After the batch file has
:: been run, the user may use Windows Explorer to manually delete the folders
:: containing the temp files.

:: The majority of temp files are those associated with the first four
:: speed-feedback (SFB) iterations of the model (pp, i1, i2, and i3).
:: Some of the "i3" files should be maintained, however, so this batch file
:: will restore five of the i3 files.

:: Robocopy (robust copy) is a powerful command and "robocopy /MOV" allows the
:: command to mimic the Windows move command (and even create a log file), but
:: "robocopy /MOV" is very slow, so we have refrained from using it.

:: Time delay, x, where x = (seconds of delay) - 1 Thus 6 => 5 sec. delay
set time_delay=6

echo(
echo Change working folder to the output folder
ping -n %time_delay% 127.0.0.1 > nul
```

```
cd %1

:: Calculate number of and size of files BEFORE moving temp files
powershell -noprofile -Command "Get-Childitem | Measure-Object -property Length -
Sum | select-object @{n='Total size of the files before moving temp files (MB)';e={
'{0:N2}' -f ($_.Sum / 1MB)}}, @{n='Files Count';e={$_.Count}} | Out-File
tot_output_files1.txt

echo(
echo Create the subfolder to hold the temp files
ping -n %time_delay% 127.0.0.1 > nul
if not exist temp_files (mkdir temp_files)

echo(
echo Move temp files associated with SFB iterations pp, i1, i2, and i3
ping -n %time_delay% 127.0.0.1 > nul
for %%H in (pp i1 i2 i3) do (
move %%H* .\temp_files
ping -n %time_delay% 127.0.0.1 > nul
)

echo(
echo Restore the i3 files that are used for truck trip gen, trip distr., and MC
ping -n %time_delay% 127.0.0.1 > nul
for %%H in (i3_SKIMTOT.TXT i3_AM_SOV.SKM i3_MD_SOV.SKM i3_HWY_AM.SKM i3_HWY_OP.SKM)
do (
move .\temp_files\%%H .
)

echo(
echo Restore files for the HOV3+, HOV2+ skims substitution technique (HOT-lane
modeling)
ping -n %time_delay% 127.0.0.1 > nul
for %%H in (pp i1 i2 i3) do (
move .\temp_files\%%H_am_hov3_mc.skm .
move .\temp_files\%%H_md_hov3_mc.skm .
move .\temp_files\%%H_am_hov2_mc.skm .
move .\temp_files\%%H_md_hov2_mc.skm .
move .\temp_files\%%H_am_sov_mc.skm .
move .\temp_files\%%H_md_sov_mc.skm .
)

echo(
echo Restore report files associated with SFB iterations pp, i1, i2, and i3
ping -n %time_delay% 127.0.0.1 > nul
for %%H in (pp i1 i2 i3) do (
move .\temp_files\%%H*.rpt .
)

echo(
echo Restore TXT,TAB files associated with SFB iterations pp, i1, i2, and i3
ping -n %time_delay% 127.0.0.1 > nul
for %%H in (pp i1 i2 i3) do (
move .\temp_files\%%H*.txt .
move .\temp_files\%%H*.tab .
)

echo(
echo Move temp files with the following pattern(s): *.tem?, temp.*, temp*.net,
transit.temp.*
echo
ping -n %time_delay% 127.0.0.1 > nul
@echo on
move *.tem? .\temp_files
move temp.* .\temp_files
move temp*.net .\temp_files
move transit.temp.* .\temp_files
@echo off
```

## Appendix B Batch file

---

```
echo(
echo Move temp files with the following pattern(s): *.skf, *.def
ping -n %time_delay% 127.0.0.1 > nul
@echo on
move *.skf          .\temp_files
move *.def          .\temp_files
move *.lkloop      .\temp_files
@echo off

goto end
:error
REM Processing Error....
PAUSE
:end

:: Calculate number of and size of files BEFORE moving temp files
powershell -nologo -Command "Get-Childitem | Measure-Object -property Length -
Sum | select-object @{n='Total size of the files after moving temp files (MB)';e={
'{0:N2}' -f ($_.Sum / 1MB)}}, @{n='Files Count';e={$_.Count}} | Out-File
tot_output_files2.txt

echo(
echo ***** Number of and size of output files BEFORE moving temp files
type tot_output_files1.txt

echo(
echo ***** Number of and size of output files AFTER moving temp files
type tot_output_files2.txt

:: Change working folder back to the root folder
CD..
```



---

## Appendix C. Cube Voyager Scripts

1	Adjust_Runtime.s.....	C-1
2	AreaType_File.s .....	C-2
3	Assemble_Skims_AB.s.....	C-4
4	Assemble_Skims_BM.s.....	C-5
5	Assemble_Skims_CR.s.....	C-6
6	Assemble_Skims_MR.s.....	C-6
7	Autoacc5.s .....	C-7
8	Average_Link_Speeds.s.....	C-13
9	CheckStationAccess.s.....	C-15
10	Combine_Tables_For_TrAssign_Parallels.....	C-16
11	Demo_Models.s.....	C-16
12	Highway_Assignment_Parallels .....	C-26
13	Highway_Skims_am.s.....	C-42
14	Highway_Skims_md.s.....	C-45
15	Highway_Skims_mod_am.s .....	C-47
16	Highway_Skims_mod_md.s .....	C-49
17	joinskims.s.....	C-50
18	MC_Auto_Drivers.s .....	C-51
19	MC_Constraint_V23.s .....	C-52
20	MC_NL_Summary.s.....	C-60
21	Metrorail_skims.s.....	C-65
22	MFARE1.S.....	C-66
23	MFARE2.S.....	C-67
24	Misc_Time-of-Day.S .....	C-72
25	modnet.s .....	C-76
26	MWCOG_Prepare_Inputs_to_Walkshed_Process_PT.s.....	C-76
27	MWCOG_Prepare_Inputs_to_Walkshed_Process_TRNBUILD.s.....	C-79
28	Parker.s .....	C-82

29	pathTrace.s .....	C-83
30	PP_Auto_Drivers.s.....	C-84
31	prefarV23.s .....	C-87
32	Prepare_Ext_Auto_Ends.s .....	C-91
33	Prepare_Ext_ComTruck_Ends.s.....	C-93
34	Prepare_Internal_Ends.s.....	C-95
35	Prepare_Trip_Tables_for_Assignment.s .....	C-100
36	Refine_Station_File.s .....	C-105
37	Remove_PP_Speed.s .....	C-105
38	Set_CPLS .....	C-105
39	Set_Factors.s.....	C-106
40	Time-of-Day.s .....	C-110
41	Transit_Accessibility.s .....	C-114
42	Transit_Assignment_AB.s.....	C-115
43	Transit_Assignment_BM.s.....	C-117
44	Transit_Assignment_CR.s.....	C-120
45	Transit_Assignment_MR.s.....	C-122
46	Transit_Skims_AB.s .....	C-124
47	Transit_Skims_BM.s .....	C-127
48	Transit_Skims_CR.s .....	C-130
49	Transit_Skims_MR.s .....	C-133
50	Trip_Distribution_External.s .....	C-136
51	Trip_Distribution_Internal.s .....	C-145
52	Trip_Generation.s.....	C-152
53	Trip_Generation_Summary.s.....	C-165
54	Truck_Com_Trip_Generation.s.....	C-173
55	unbuild_net.s .....	C-177
56	V2.3_Highway_Build.s .....	C-177
57	walkacc.s .....	C-180

# 1 Adjust\_Runtime.s

```
*del ModelAM.TP
*del Mode6AM.TP
*del Mode8AM.TP
*del Mode1OP.TP
*del Mode6OP.TP
*del Mode8OP.TP

;=====
; First loop through AM/OP and Mode 1,6,8 Line files
; To make sure the line with RUNTIMES are as expected (ABORT if that's not the case)
;
;=====

Loop PRD = 1,2 ;Loop through 1>AM and 2>OP periods
  Loop LBMODE = 1,3 ; Loop through local Bus modes 1>Mode 1, 2>Mode 6 and 3>
  Mode 8

  IF (PRD = 1) PER = 'AM'
  IF (PRD = 2) PER = 'OP'

  IF (LBMODE = 1) MOD = 1
  IF (LBMODE = 2) MOD = 6
  IF (LBMODE = 3) MOD = 8

  RUN PGM=MATRIX
  ZONES = 1

  FILEI RECI="inputs\MODE@MOD@PER@.TB",Fields=1-3

  StrONE = substr(inci,1,7),
  Str1 = substr(inci,8,24),
  StrRUN = substr(inci,32,8),
  StrRTV = substr(inci,40,3),
  StrCma = substr(inci,43,1),
  Str2 = substr(inci,44,150)
  ;
  ; First make sure that the RUNTIME value is where we expect-
  ; If the"ONEWAY=" String is in cols 1-7 then we expect that "RUNTIME=" is in
  cols 32-39 and a ',' is in col 43.
  ; Otherwise ABORT! Note: We will allow condition that StrRUN,StrRTV and StrCma
  are all blank, to allow 'RT=' in Mode strings
  ; The ltrim() function gets rid of the leading spaces of a string.

  IF ((StrONE == 'ONEWAY=') && (ltrim(StrRUN) = '' && ltrim(StrRTV) = '' &&
  ltrim(StrCma) = '')) ;
    GOTO SkipCheck
  ENDIF

  IF ((StrONE == 'ONEWAY=') && (StrRUN != 'RUNTIME=' && StrCma != ',')) ; If
  'Runtime=' and 'comma' strings are not in the expected place
    List=' ONEWAY= is IN cols 1-7 but RUNTIME= is NOT IN cols 32-39 and/or comma
  is NOT IN col 43, so I QUIT! '
    LIST='ONEWAY= RUNTIME= ,'
    LIST=StrONE,Str1,StrRUN,StrRTV,StrCma,Str2
    ABORT
  ENDIF
```

```
:SkipCheck

ENDRUN
ENDLOOP
ENDLOOP

;=====
;
; OK, If there are no ABORTS above then all is well- now read through the
; AM/OP and Mode 1,6,8 line files and factor the RUNTIME values as appropriate
; with the DBF Lookup
;
;=====
;
ModeledYear = '%_Year_%'

Loop PRD = 1,2 ;Loop through 1>AM and 2>OP periods
  Loop LBMODE = 1,3 ; Loop through local Bus modes 1>Mode 1, 2>Mode 6 and 3>
  Mode 8

  IF (PRD = 1) PER = 'AM'
  IF (PRD = 2) PER = 'OP'

  IF (LBMODE = 1)
    MOD = 1
    InOut = 'IB'
  ELSEIF (LBMODE = 2)
    MOD = 6
    InOut = 'IB'
  ELSEIF (LBMODE = 3)
    MOD = 8
    InOut = 'OB'
  ENDIF

  RUN PGM=MATRIX
  ZONES = 1

  ;; Read Bus Factors as a lookup, indexed to Model Year
  FileI LOOKUPI[1]= "inputs\Bus_Factor_File.dbf"
  LOOKUP LOOKUPI=1, NAME=BFTR,
  LOOKUP[1] = Year, Result=AMIBFTR,
  LOOKUP[2] = Year, Result=AMOBFTR,
  LOOKUP[3] = Year, Result=OPIBFTR,
  LOOKUP[4] = Year, Result=OPOBFTR,
  INTERPOLATE=N, FAIL= 1,1,1, LIST=Y ; if "ModeledYear" is not in the
  range of modeled years listed in the lookup table,

  ; "FAIL=1,1,1" ensures bus run time will remain the
  same.

  AMIB_FTR = BFTR(1,@ModeledYear@)
  AMOB_FTR = BFTR(2,@ModeledYear@)
  OPIB_FTR = BFTR(3,@ModeledYear@)
  OPOB_FTR = BFTR(4,@ModeledYear@)

  ;inputs\MODE@MOD@PER@.TB",Fields=1-3
  FILEI RECI="inputs\MODE@MOD@PER@.TB",Fields=1-3

  StrONE = substr(inci,1,7),
  Str1 = substr(inci,8,24),
  StrRUN = substr(inci,32,8),
  StrRTV = substr(inci,40,3),
```

## Appendix C Cube Voyager Scripts

```

StrCma = substr(Rec1,43,1),
Str2 = substr(Rec1,44,150)

Factor = @Per@@InOut@_Ftr
FILEO PRINTO[1] = "MODE@MOD@PER@.TB"

IF (StrONE == 'ONEWAY=' && StrRUN== 'RUNTIME=' && StrCma == ',') ; If Runtime=
and comma exists in the expected place
  RUNTIME_IN = val(StrRTV) ; convert RUNTIME 'string' into numeric
variable
  RUNTIME_OUT = RUNTIME_IN * Factor ; factored Factor numeric value
  Print LIST=StrONE,Str1,StrRUN,RUNTIME_OUT(4.1),StrCma,Str2,'
; Base RUNTIME= ',RUNTIME_IN,' Time Factor: ',Factor(5.3),' Year: ', '@ModeledYear@',
Printo = 1
,Printo =1
ELSE
  Print LIST=StrONE,Str1,StrRUN,StrRTV,StrCma,Str2, Printo = 1
ENDIF

ENDRUN

ENDLOOP
ENDLOOP

```

## 2 AreaType\_File.s

```

; AreaType_File.S
;
;=====
; Accumulate zonal HHs and Jobs around each TAZ based on 1.0 mile
; straightline distances between centroids, and then develop area types
; for each TAZ
;=====
;
;
; Define Inputs Files:
nodefile='inputs\node.dbf'
LUFile = 'inputs\zone.DBF'
ATover = 'inputs\AT_override.txt'

; Output Files:
TAZxys = 'TAZ_XYs.dbf'
FloatLU = 'Floating_LU.dbf'
ATFile = 'AreaType_File.dbf'

RUN PGM=MATRIX

ZONES=1

FILEI DBI[1] = "@nodefile@", sort = N
FILEO RECO[1] = "@TAZxys@", Fields = N(5),X(10),Y(10)

LOOP L= 1,dbi.1.NUMRECORDS
  x=DBIReadRecord(1,L)
  if (DI.1.N <= 3722 )
    ro.N = di.1.N ; Node Number
    ro.X = di.1.X ; X-Coordinate (feet NAD/83)
    ro.Y = di.1.Y ; Y-Coordinate (feet NAD/83)
    WRITE RECO=1 ;
  endif
ENDLOOP
ENDRUN

```

```

;-----
RUN PGM=MATRIX
ZONES=1

FILEO RECO[1] = "@FloatLU@",
fields = TAZ(5),HH00(10),POP00(10),EMP00(10),AREA00(10.4),
HH10(10),POP10(10),EMP10(10),AREA10(10.4)

FileI LOOKUPI[1] = "@TAZxys@"
LOOKUP LOOKUPI=1, NAME=tazxys,
LOOKUP[1] = N, RESULT=x, ;
a2
LOOKUP[2] = N, RESULT=y, ;
a2
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@LUFILE@"
LOOKUP LOOKUPI=2, NAME=Landuse,
LOOKUP[1] = taz, RESULT= HH,
LOOKUP[2] = taz, RESULT= TOTPOP,
LOOKUP[3] = taz, RESULT= TOTEMP,
LOOKUP[4] = taz, RESULT= landarea,
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

ARRAY HH00 =3722, HH10 =3722,
POP00 =3722, POP10 =3722,
EMP00 =3722, EMP10 =3722,
AREA00 =3722, AREA10 =3722

LOOP L = 1,3675 ; Loop through each zone
  Xi = tazxys(1,L)
  Yi = tazxys(2,L)
  IF (Xi = 0.00) Continue

  LOOP M = 1,3675 ; Loop through all proximate zones
    Xj = tazxys(1,M)
    Yj = tazxys(2,M)
    IF (Xj = 0.00) Continue

    Xdiff = abs(Xi-Xj)
    Ydiff = abs(Yi-Yj)

    d_ft = sqrt(xdiff*xdiff + Ydiff*Ydiff)
    d_mi = d_ft/5280.0

    IF (d_mi >= 1.000) Continue
    ;debug1
    If (l=1)
      print form=10 list = l,m,xi,yi,xj,yj,d_ft,d_mi(6.2),
file=debug1.txt
    endif
    ;debug1

    IF (D_mi < 1.000)
      HH10[L] = HH10[L] + Landuse(1,m)
      POP10[L] = POP10[L] + Landuse(2,m)
      EMP10[L] = EMP10[L] + Landuse(3,m)
      AREA10[L] = AREA10[L] + Landuse(4,m)
    ENDIF

    IF (D_mi = 0.000)
      HH00[L] = HH00[L] + Landuse(1,m)
      POP00[L] = POP00[L] + Landuse(2,m)

```



## Appendix C Cube Voyager Scripts

```

        EMP00[L]      = EMP00[L]      + Landuse(3,m)
        AREA00[L]    = AREA00[L]     + Landuse(4,m)
    ENDIF
;debug2
If (L=1)
    print form=8.2 list = l,m, d_mi(6.2),
        HH00[L],POP00[L],EMP00[L],AREA00[L],
        HH10[L],POP10[L],EMP10[L],AREA10[L],
        file=debug2.txt
    endif
;debug2
ENDLOOP

ENDLOOP

LOOP M= 1, 3675
    ro.TAZ          = M

    ro.HH00         = HH00[M]
    ro.POP00        = POP00[M]
    ro.EMP00        = EMP00[M]
    ro.AREA00       = AREA00[M]

    ro.HH10         = HH10[M]
    ro.POP10        = POP10[M]
    ro.EMP10        = EMP10[M]
    ro.AREA10       = AREA10[M]

    WRITE RECO= 1

ENDLOOP
endrun

;=====
; Compute Area Type based on updated 1-mile floating Pop/Emp density
; documented by M. Martchouk on June 10, 2010
;=====

RUN PGM=MATRIX
ZONES=1

FILEO RECO[1] = "@ATfile@",fields =
    TAZ(5), POP_10(10.0), EMP_10(10.0), area_10(10.2),
    POPden(10.2), Empden(10.2),
    POPcode(5), EMPcode(5),
    Atype(5)

FileI LOOKUPI[1] = "@FloatLU@" ; One-mile floating land use
LOOKUP LOOKUPI=1, NAME=PopEmpArea10,
LOOKUP[1] = TAZ, RESULT=POP10, ;
a2
LOOKUP[2] = TAZ, RESULT=EMP10, ;
a2
LOOKUP[3] = TAZ, RESULT=Area10, ;
a2
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;; Read in Area Type override file:
LOOKUP NAME=ATover,
LOOKUP[1] = 1, RESULT=2, ; TAZ & Area Type override no. 1-6
a2
INTERPOLATE=N, FAIL= 0,0,0, LIST=Y, File=@ATover@

ARRAY ATYPEMtx = 7,7 EmpClassDen=6 POPClassDen=6 , atcount= 7

```

```

;; Define Area type code matrix
ATYPEMtx[1][1]=6 ATYPEMtx[1][2]=6 ATYPEMtx[1][3]=5 ATYPEMtx[1][4]=3
ATYPEMtx[1][5]=3 ATYPEMtx[1][6]=3 ATYPEMtx[1][7]=2
ATYPEMtx[2][1]=6 ATYPEMtx[2][2]=5 ATYPEMtx[2][3]=5 ATYPEMtx[2][4]=3
ATYPEMtx[2][5]=3 ATYPEMtx[2][6]=3 ATYPEMtx[2][7]=2
ATYPEMtx[3][1]=6 ATYPEMtx[3][2]=5 ATYPEMtx[3][3]=5 ATYPEMtx[3][4]=3
ATYPEMtx[3][5]=3 ATYPEMtx[3][6]=2 ATYPEMtx[3][7]=2
ATYPEMtx[4][1]=6 ATYPEMtx[4][2]=4 ATYPEMtx[4][3]=4 ATYPEMtx[4][4]=3
ATYPEMtx[4][5]=2 ATYPEMtx[4][6]=2 ATYPEMtx[4][7]=1
ATYPEMtx[5][1]=4 ATYPEMtx[5][2]=4 ATYPEMtx[5][3]=4 ATYPEMtx[5][4]=2
ATYPEMtx[5][5]=2 ATYPEMtx[5][6]=2 ATYPEMtx[5][7]=1
ATYPEMtx[6][1]=4 ATYPEMtx[6][2]=4 ATYPEMtx[6][3]=4 ATYPEMtx[6][4]=2
ATYPEMtx[6][5]=2 ATYPEMtx[6][6]=2 ATYPEMtx[6][7]=1
ATYPEMtx[7][1]=2 ATYPEMtx[7][2]=2 ATYPEMtx[7][3]=2 ATYPEMtx[7][4]=2
ATYPEMtx[7][5]=2 ATYPEMtx[7][6]=1 ATYPEMtx[7][7]=1

;; Define top end of pop, emp. density ranges for classes 1-6
PopClassDen[1] = 750.0
PopClassDen[2] = 1500.0
PopClassDen[3] = 3500.0
PopClassDen[4] = 6000.0
PopClassDen[5] = 10000.0
PopClassDen[6] = 15000.0

EmpClassDen[1] = 100.0
EmpClassDen[2] = 350.0
EmpClassDen[3] = 1500.0
EmpClassDen[4] = 3550.0
EmpClassDen[5] = 13750.0
EmpClassDen[6] = 15000.0

LOOP L = 1,3675 ; Loop through each zone, read one-mile floating land use, area

    _pop          = PopEmpArea10(1,L)
    _emp          = PopEmpArea10(2,L)
    _area         = PopEmpArea10(3,L)

    IF (_area > 0) _popden = Round(_pop/_area) ; calc. densities
    IF (_area > 0) _empden = Round(_emp/_area) ;

    popcode = 1 ; initialize density classes
    empcode = 1 ;

    LOOP M= 1,6 ; slot TAZ into the higher pop/emp density classes as
appropriate

        IF (_popden > PopClassDen[M]) Popcode = M + 1.0
        IF (_empden > EmpClassDen[M]) Empcode = M + 1.0

    ENDLOOP

    IF (popcode < 0 || popcode > 7) abort
    IF (empcode < 0 || empcode > 7) abort

    _Atype = AtypeMtx[PopCode][EmpCode]

;; Impose Area type override if necessary
;; From file: inputs\AT_override.txt which contains two columns, TAZ and
Area Type Override (1-6)

    IF (ATover(1,L) > 0)

```

## Appendix C Cube Voyager Scripts

```

        IF (ATover(1,L) < 0 || ATover(1,L) > 6) abort ;; make sure override
value is reasonable
        _Atype = ATover(1,L)

        ENDIF
;;

atcount[_Atype] = atcount[_Atype] + 1.0
totcnt          = totcnt          + 1.0

Ro.TAZ         = L
Ro.POP_10      = _pop
Ro.EMP_10      = _emp
Ro.area_10     = _area
Ro.POPden      = _popden
Ro.EMPden      = _empden
Ro.POPcode     = POPCode
RO.EMPcode     = EmpCode
RO.Atype       = _Atype
WRITE RECO=1

ENDLOOP

loop kk= 1,6
print list= 'area type ', kk(5), ' TAZ Count is: ', atcount[KK](6.0), file =
AreaType_File.txt
endloop
print list= 'total      ','      ',' TAZ Count is: ', totcnt(6.0),      file =
AreaType_File.txt

endrun

```

### 3 Assemble\_Skims\_AB.s

```

;-----
;Assemble_Skims_AB.s
;MWCOCG Version 2.3 Model
;Assemble Transit Skims by Time Period
; Input Files:
; iteration (%_iter_) = 'i1',...,'i4'
; period (@period@) = 'am'/'op'
;
; Transit Skim Files          = <iteration>_<period>_AB.skm
; Transit Fare Files          = <iteration>_<period>_AB.FAR
; Output File:
; Combined Transit Skims      = <iteration>TRN<Period>_AB.SKM, MO = 1-48,
;-----

;-----
;Loop through each period
;-----

; Read Deflation Factor
READ FILE=TRN_Deflator.txt

LOOP PERIOD=1,2

IF (PERIOD = 1)
TIME_PERIOD = 'AM'
ELSE
TIME_PERIOD = 'OP'
ENDIF

```

```

;-----
;Assemble Skims & Fares into Files for Mode Choice
;-----
RUN PGM=MATRIX
MATI[1]=%_iter_%_@TIME_PERIOD@_WK_AB.SKM
MATI[2]=%_iter_%_@TIME_PERIOD@_WK_AB.FAR
MATI[3]=%_iter_%_@TIME_PERIOD@_DR_AB.SKM
MATI[4]=%_iter_%_@TIME_PERIOD@_DR_AB.FAR
MATI[5]=%_iter_%_@TIME_PERIOD@_KR_AB.SKM
MATI[6]=%_iter_%_@TIME_PERIOD@_KR_AB.FAR
MATO[1]=%_iter_%_TRN@TIME_PERIOD@_AB.SKM, MO = 1-48,
NAME = WWAET, WWLKT, WINIT, WXFRT, WIVTT, WIVCR, WIVXB, WIVMR, WIVN1,
WIVN2, WIVLB, WNXFR, WFARE, WXPEN,
DWAET, DWLKT, DINIT, DXFRT, DIVTT, DIVCR, DIVXB, DIVMR, DIVN1,
DIVN2, DIVLB, DNXFR, DFARE, DXPEN, DACCT, DACCD, DPRKC, DPRKT,
KWAET, KWLKT, KINIT, KXFRT, KIVTT, KIVCR, KIVXB, KIVMR, KIVN1,
KIVN2, KIVLB, KNXFR, KFARE, KKPEN, KACCT, KACCD
MW[1] = MI.1.9 ;---- wlk walk acc time (0.01 min)
MW[2] = MI.1.10 ;---- wlk other walk time (0.01 min)
MW[3] = MI.1.7 ;---- wlk ini.wait time (0.01 min)
MW[4] = MI.1.8 ;---- wlk xfr wait time (0.01 min)
MW[5] = MI.1.3 ;---- wlk ivt-total (0.01 min)
MW[6] = MI.1.4 ;---- wlk ivt-commuter rail (0.01 min)
MW[7] = MI.1.2 ;---- wlk ivt-exp bus (0.01 min)
MW[8] = MI.1.3 ;---- wlk ivt-metrorail (0.01 min)
MW[9] = MI.1.5 ;---- wlk ivt-new rail mode (0.01 min)
MW[10] = MI.1.6 ;---- wlk ivt-new bus mode (0.01 min)
MW[11] = MI.1.1 ;---- wlk ivt-local bus (0.01 min)
MW[12] = MI.1.12 ;---- wlk transfers (0+)
MW[13] = MI.2.1 ;---- wlk fare (2007 cents)
MW[14] = MI.1.11 ;---- wlk added board time (0.01 min)
MW[15] = MI.3.9 ;---- drv walk acc time (0.01 min)
MW[16] = MI.3.10 ;---- drv other walk time (0.01 min)
MW[17] = MI.3.7 ;---- drv ini.wait time (0.01 min)
MW[18] = MI.3.8 ;---- drv xfr wait time (0.01 min)
MW[19] = MI.3.3 ;---- drv ivt-total (0.01 min)
MW[20] = MI.3.4 ;---- drv ivt-commuter rail (0.01 min)
MW[21] = MI.3.2 ;---- drv ivt-exp bus (0.01 min)
MW[22] = MI.3.3 ;---- drv ivt-metrorail (0.01 min)
MW[23] = MI.3.5 ;---- drv ivt-new rail mode (0.01 min)
MW[24] = MI.3.6 ;---- drv ivt-new bus mode (0.01 min)
MW[25] = MI.3.1 ;---- drv ivt-local bus (0.01 min)
MW[26] = MI.3.12 ;---- drv transfers (0+)
MW[27] = MI.4.1 ;---- drv fare (2007 cents)
MW[28] = MI.3.11 ;---- drv added board time (0.01 min)
MW[29] = MI.3.13 ;---- drv acc time (0.01 min)
MW[30] = MI.3.14 ;---- drv acc distance (0.01 mile)
MW[31] = MI.3.16 ;---- drv park cost (2007 cents)
MW[32] = MI.3.15 ;---- drv park time (0.01 min)
MW[33] = MI.5.9 ;---- knr walk acc time (0.01 min)
MW[34] = MI.5.10 ;---- knr other walk time (0.01 min)
MW[35] = MI.5.7 ;---- knr ini.wait time (0.01 min)
MW[36] = MI.5.8 ;---- knr xfr wait time (0.01 min)
MW[37] = MI.5.3 ;---- knr ivt-total (0.01 min)
MW[38] = MI.5.4 ;---- knr ivt-commuter rail (0.01 min)
MW[39] = MI.5.2 ;---- knr ivt-exp bus (0.01 min)
MW[40] = MI.5.3 ;---- knr ivt-metrorail (0.01 min)
MW[41] = MI.5.5 ;---- knr ivt-new rail mode (0.01 min)
MW[42] = MI.5.6 ;---- knr ivt-new bus mode (0.01 min)
MW[43] = MI.5.1 ;---- knr ivt-local bus (0.01 min)
MW[44] = MI.5.12 ;---- knr transfers (0+)
MW[45] = MI.6.1 ;---- knr fare (2007 cents)
MW[46] = MI.5.11 ;---- knr added board time (0.01 min)
MW[47] = MI.5.13 ;---- knr acc time (0.01 min)
MW[48] = MI.5.14 ;---- knr acc distance (0.01 mile)

```

JLOOP

## Appendix C Cube Voyager Scripts

```

; assemble total IVTT

MW[05] = MW[06]+MW[07]+MW[08]+MW[09]+MW[10]+MW[11]
MW[19] = MW[20]+MW[21]+MW[22]+MW[23]+MW[24]+MW[25]
MW[37] = MW[38]+MW[39]+MW[40]+MW[41]+MW[42]+MW[43]

; zero-out fares for IVTT=0

IF (MW[05]=0 ) MW[13]=0
IF (MW[19]=0 ) MW[27]=0
IF (MW[37]=0 ) MW[45]=0

; deflate parking costs to 2007

MW[31] = @DEFLATIONFTR*MW[31]

ENDJLOOP

ENDRUN

ENDLOOP ;---- PERIOD ----

```

## 4 Assemble\_Skims\_BM.s

```

-----
;Assemble_Skims_BM.s
;MwCOG Version 2.3 Model
;Assemble Transit Skims by Time Period
; Input Files:
; iteration (% iter %) = 'i1',...,'i4'
; period (@period@) = 'am'/'op'
;
; Transit Skim Files = <iteration>_<period>_BM.skm
; Transit Fare Files = <iteration>_<period>_BM.FAR
; Output File:
; Combined Transit Skims = <iteration>TRN<Period>_BM.SKM, MO = 1-48,
-----

;-----
;Loop through each period
;-----
; Read Deflation Factor
READ FILE=TRN_Deflator.txt

LOOP PERIOD=1,2

IF (PERIOD = 1)
TIME_PERIOD = 'AM'
ELSE
TIME_PERIOD = 'OP'
ENDIF

;-----
;Assemble Skims & Fares into Files for Mode Choice
;-----
RUN PGM=MATRIX
MATI[1]=% iter % @TIME_PERIOD@_WK_BM.SKM
MATI[2]=% iter % @TIME_PERIOD@_WK_BM.FAR
MATI[3]=% iter % @TIME_PERIOD@_DR_BM.SKM
MATI[4]=% iter % @TIME_PERIOD@_DR_BM.FAR
MATI[5]=% iter % @TIME_PERIOD@_KR_BM.SKM
MATI[6]=% iter % @TIME_PERIOD@_KR_BM.FAR
MATO[1]=% iter % TRN@TIME_PERIOD@_BM.SKM, MO = 1-48,
NAME = WWAET, WWLKT, WINIT, WXFRT, WIVTT, WIVCR, WIVXB, WIVMR, WIVN1,

```

```

WIVN2, WIVLB, WNXFR, WFARE, WXPEN,
DWAET, DWLKT, DINIT, DXFRT, DIVTT, DIVCR, DIVXB, DIVMR, DIVN1,
DIVN2, DIVLB, DNKFR, DFARE, DXPEN, DACCT, DACC, DPRKC, DPRKT,
KWAET, KWLKT, KINIT, KXFRT, KIVTT, KIVCR, KIVXB, KIVMR, KIVN1,
KIVN2, KIVLB, KNKFR, KFARE, KXPEN, KACCT, KACCD
MW[1] = MI.1.9 ;---- wlk walk acc time (0.01 min)
MW[2] = MI.1.10 ;---- wlk other walk time (0.01 min)
MW[3] = MI.1.7 ;---- wlk ini.wait time (0.01 min)
MW[4] = MI.1.8 ;---- wlk xfr wait time (0.01 min)
MW[5] = MI.1.3 ;---- wlk ivt-total (0.01 min)
MW[6] = MI.1.4 ;---- wlk ivt-commuter rail(0.01 min)
MW[7] = MI.1.2 ;---- wlk ivt-exp bus (0.01 min)
MW[8] = MI.1.3 ;---- wlk ivt-metrorail (0.01 min)
MW[9] = MI.1.5 ;---- wlk ivt-new rail mode(0.01 min)
MW[10] = MI.1.6 ;---- wlk ivt-new bus mode (0.01 min)
MW[11] = MI.1.1 ;---- wlk ivt-local bus (0.01 min)
MW[12] = MI.1.12 ;---- wlk transfers (0+)
MW[13] = MI.2.1 ;---- wlk fare (2007 cents)
MW[14] = MI.1.11 ;---- wlk added board time (0.01 min)
MW[15] = MI.3.9 ;---- drv walk acc time (0.01 min)
MW[16] = MI.3.10 ;---- drv other walk time (0.01 min)
MW[17] = MI.3.7 ;---- drv ini.wait time (0.01 min)
MW[18] = MI.3.8 ;---- drv xfr wait time (0.01 min)
MW[19] = MI.3.3 ;---- drv ivt-total (0.01 min)
MW[20] = MI.3.4 ;---- drv ivt-commuter rail(0.01 min)
MW[21] = MI.3.2 ;---- drv ivt-exp bus (0.01 min)
MW[22] = MI.3.3 ;---- drv ivt-metrorail (0.01 min)
MW[23] = MI.3.5 ;---- drv ivt-new rail mode(0.01 min)
MW[24] = MI.3.6 ;---- drv ivt-new bus mode (0.01 min)
MW[25] = MI.3.1 ;---- drv ivt-local bus (0.01 min)
MW[26] = MI.3.12 ;---- drv transfers (0+)
MW[27] = MI.4.1 ;---- drv fare (2007 cents)
MW[28] = MI.3.11 ;---- drv added board time (0.01 min)
MW[29] = MI.3.13 ;---- drv acc time (0.01 min)
MW[30] = MI.3.14 ;---- drv acc distance (0.01 mile)
MW[31] = MI.3.16 ;---- drv park cost (2007 cents)
MW[32] = MI.3.15 ;---- drv park time (0.01 min)
MW[33] = MI.5.9 ;---- knr walk acc time (0.01 min)
MW[34] = MI.5.10 ;---- knr other walk time (0.01 min)
MW[35] = MI.5.7 ;---- knr ini.wait time (0.01 min)
MW[36] = MI.5.8 ;---- knr xfr wait time (0.01 min)
MW[37] = MI.5.3 ;---- knr ivt-total (0.01 min)
MW[38] = MI.5.4 ;---- knr ivt-commuter rail (0.01 min)
MW[39] = MI.5.2 ;---- knr ivt-exp bus (0.01 min)
MW[40] = MI.5.3 ;---- knr ivt-metrorail (0.01 min)
MW[41] = MI.5.5 ;---- knr ivt-new rail mode(0.01 min)
MW[42] = MI.5.6 ;---- knr ivt-new bus mode (0.01 min)
MW[43] = MI.5.1 ;---- knr ivt-local bus (0.01 min)
MW[44] = MI.5.12 ;---- knr transfers (0+)
MW[45] = MI.6.1 ;---- knr fare (2007 cents)
MW[46] = MI.5.11 ;---- knr added board time (0.01 min)
MW[47] = MI.5.13 ;---- knr acc time (0.01 min)
MW[48] = MI.5.14 ;---- knr acc distance (0.01 mile)

```

JLOOP

```

; assemble total IVTT

MW[05] = MW[06]+MW[07]+MW[08]+MW[09]+MW[10]+MW[11]
MW[19] = MW[20]+MW[21]+MW[22]+MW[23]+MW[24]+MW[25]
MW[37] = MW[38]+MW[39]+MW[40]+MW[41]+MW[42]+MW[43]

; zero-out fares for IVTT=0

IF (MW[05]=0 ) MW[13]=0
IF (MW[19]=0 ) MW[27]=0
IF (MW[37]=0 ) MW[45]=0

```

## Appendix C Cube Voyager Scripts

```
; deflate parking costs to 2007

MW[31] = @DEFLATIONFTR*MW[31]

ENDJLOOP

ENDRUN

ENDLOOP ;---- PERIOD ----
```

## 5 Assemble\_Skims\_CR.s

```
-----
;Assemble Skims_CR.s
;MwCOG Version 2.3 Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter %) = 'i1',...,'i4'
;   period (@period@) = 'am'/'op'
;
; Transit Skim Files      = <iteration>_<period>_CR.skm
; Transit Fare Files     = <iteration>_<period>_CR.FAR
; Output File:
;   Combined Transit Skims = <iteration>TRN<Period>_CR.SKM, MO = 1-32,
-----

;-----
;Loop through each period
;-----
; Read Deflation Factor
READ FILE=TRN_Deflator.txt

LOOP PERIOD=1,2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
ELSE
  TIME_PERIOD = 'OP'
ENDIF

;-----
;Assemble Skims & Fares into Files for Mode Choice
;-----
RUN PGM=MATRIX
MATI[1]=%_iter_%_@TIME_PERIOD@_WK_CR.SKM
MATI[2]=%_iter_%_@TIME_PERIOD@_WK_CR.FAR
MATI[3]=%_iter_%_@TIME_PERIOD@_DR_CR.SKM
MATI[4]=%_iter_%_@TIME_PERIOD@_DR_CR.FAR
MATO[1]=%_iter_%_TRN@TIME_PERIOD@_CR.SKM, MO = 1-32,
NAME = WWAET, WWLKT, WINIT, WXFRT, WIVTT, WIVCR, WIVXB, WIVMR, WIVN1,
WIVN2, WIVLB, WNXFR, WFARE, WXPEN,
DWAET, DWLKT, DINIT, DXFRT, DIVTT, DIVCR, DIVXB, DIVMR, DIVN1,
DIVN2, DIVLB, DNxFR, DFARE, DXPEN, DACCT, DACCD, DPRKC, DPRKT

MW[1] = MI.1.9 ;---- wlk walk acc time (0.01 min)
MW[2] = MI.1.10 ;---- wlk other walk time (0.01 min)
MW[3] = MI.1.7 ;---- wlk ini.wait time (0.01 min)
MW[4] = MI.1.8 ;---- wlk xfr wait time (0.01 min)
MW[5] = MI.1.3 ;---- wlk ivt-total (0.01 min)
MW[6] = MI.1.4 ;---- wlk ivt-commuter rail (0.01 min)
MW[7] = MI.1.2 ;---- wlk ivt-exp bus (0.01 min)
MW[8] = MI.1.3 ;---- wlk ivt-metrorail (0.01 min)
MW[9] = MI.1.5 ;---- wlk ivt-new rail mode (0.01 min)
MW[10] = MI.1.6 ;---- wlk ivt-new bus mode (0.01 min)
MW[11] = MI.1.1 ;---- wlk ivt-local bus (0.01 min)
```

```
MW[12] = MI.1.12 ;---- wlk transfers (0+)
MW[13] = MI.2.1 ;---- wlk fare (2007 cents)
MW[14] = MI.1.11 ;---- wlk added board time (0.01 min)
MW[15] = MI.3.9 ;---- drv walk acc time (0.01 min)
MW[16] = MI.3.10 ;---- drv other walk time (0.01 min)
MW[17] = MI.3.7 ;---- drv ini.wait time (0.01 min)
MW[18] = MI.3.8 ;---- drv xfr wait time (0.01 min)
MW[19] = MI.3.3 ;---- drv ivt-total (0.01 min)
MW[20] = MI.3.4 ;---- drv ivt-commuter rail (0.01 min)
MW[21] = MI.3.2 ;---- drv ivt-exp bus (0.01 min)
MW[22] = MI.3.3 ;---- drv ivt-metrorail (0.01 min)
MW[23] = MI.3.5 ;---- drv ivt-new rail mode (0.01 min)
MW[24] = MI.3.6 ;---- drv ivt-new bus mode (0.01 min)
MW[25] = MI.3.1 ;---- drv ivt-local bus (0.01 min)
MW[26] = MI.3.12 ;---- drv transfers (0+)
MW[27] = MI.4.1 ;---- drv fare (2007 cents)
MW[28] = MI.3.11 ;---- drv added board time (0.01 min)
MW[29] = MI.3.13 ;---- drv acc time (0.01 min)
MW[30] = MI.3.14 ;---- drv acc distance (0.01 mile)
MW[31] = MI.3.16 ;---- drv park cost (2007 cents)
MW[32] = MI.3.15 ;---- drv park time (0.01 min)
JLOOP
```

```
; assemble total IVTT

MW[05] = MW[06]+MW[07]+MW[08]+MW[09]+MW[10]+MW[11]
MW[19] = MW[20]+MW[21]+MW[22]+MW[23]+MW[24]+MW[25]
```

```
; zero-out fares for IVTT=0
```

```
IF (MW[05]=0 ) MW[13]=0
IF (MW[19]=0 ) MW[27]=0
```

```
; deflate parking costs to 2007
```

```
MW[31] = @DEFLATIONFTR*MW[31]
```

```
ENDJLOOP
```

```
ENDRUN
```

```
ENDLOOP ;---- PERIOD ----
```

## 6 Assemble\_Skims\_MR.s

```
-----
;Assemble_Skims_MR.s
;MwCOG Version 2.3 Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter %) = 'i1',...,'i4'
;   period (@period@) = 'am'/'op'
;
; Transit Skim Files      = <iteration>_<period>_MR.skm
; Transit Fare Files     = <iteration>_<period>_MR.FAR
; Output File:
;   Combined Transit Skims = <iteration>TRN<Period>_MR.SKM, MO = 1-48,
-----

;-----
;Loop through each period
;-----
; Read Deflation Factor
READ FILE=TRN_Deflator.txt
```

## Appendix C Cube Voyager Scripts

```

LOOP PERIOD=1,2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
ELSE
  TIME_PERIOD = 'OP'
ENDIF

;-----
;Assemble Skims & Fares into Files for Mode Choice
;-----

RUN PGM=MATRIX
MATI[1]=%_iter%_@TIME_PERIOD@_WK_MR.SKM
MATI[2]=%_iter%_@TIME_PERIOD@_WK_MR.FAR
MATI[3]=%_iter%_@TIME_PERIOD@_DR_MR.SKM
MATI[4]=%_iter%_@TIME_PERIOD@_DR_MR.FAR
MATI[5]=%_iter%_@TIME_PERIOD@_KR_MR.SKM
MATI[6]=%_iter%_@TIME_PERIOD@_KR_MR.FAR
MATO[1]=%_iter%_@TRN@TIME_PERIOD@_MR.SKM, MO = 1-48,
NAME = WWAET, WWLKT, WINIT, WXFRT, WIVTT, WIVCR, WIVXB, WIVMR, WIVN1,
WIVN2, WIVLB, WNXFR, WFARE, WXPEN,
DWAET, DWLKT, DINIT, DXFRT, DIVTT, DIVCR, DIVXB, DIVMR, DIVN1,
DIVN2, DIVLB, DNxFR, DFARE, DXPEN, DACCT, DACCD, DPRKC, DPRKT,
KWAET, KWLKT, KINIT, KXFRT, KIVTT, KIVCR, KIVXB, KIVMR, KIVN1,
KIVN2, KIVLB, KNXFR, KFARE, KXPEN, KACCT, KACCD
MW[1] = MI.1.9 ;---- wlk walk acc time (0.01 min)
MW[2] = MI.1.10 ;---- wlk other walk time (0.01 min)
MW[3] = MI.1.7 ;---- wlk ini.wait time (0.01 min)
MW[4] = MI.1.8 ;---- wlk xfr wait time (0.01 min)
MW[5] = MI.1.3 ;---- wlk ivt-total (0.01 min)
MW[6] = MI.1.4 ;---- wlk ivt-commuter rail (0.01 min)
MW[7] = MI.1.2 ;---- wlk ivt-exp bus (0.01 min)
MW[8] = MI.1.3 ;---- wlk ivt-metrorail (0.01 min)
MW[9] = MI.1.5 ;---- wlk ivt-new rail mode (0.01 min)
MW[10] = MI.1.6 ;---- wlk ivt-new bus mode (0.01 min)
MW[11] = MI.1.1 ;---- wlk ivt-local bus (0.01 min)
MW[12] = MI.1.12 ;---- wlk transfers (0+)
MW[13] = MI.2.1 ;---- wlk fare (2007 cents)
MW[14] = MI.1.11 ;---- wlk added board time (0.01 min)
MW[15] = MI.3.9 ;---- drv walk acc time (0.01 min)
MW[16] = MI.3.10 ;---- drv other walk time (0.01 min)
MW[17] = MI.3.7 ;---- drv ini.wait time (0.01 min)
MW[18] = MI.3.8 ;---- drv xfr wait time (0.01 min)
MW[19] = MI.3.3 ;---- drv ivt-total (0.01 min)
MW[20] = MI.3.4 ;---- drv ivt-commuter rail (0.01 min)
MW[21] = MI.3.2 ;---- drv ivt-exp bus (0.01 min)
MW[22] = MI.3.3 ;---- drv ivt-metrorail (0.01 min)
MW[23] = MI.3.5 ;---- drv ivt-new rail mode (0.01 min)
MW[24] = MI.3.6 ;---- drv ivt-new bus mode (0.01 min)
MW[25] = MI.3.1 ;---- drv ivt-local bus (0.01 min)
MW[26] = MI.3.12 ;---- drv transfers (0+)
MW[27] = MI.4.1 ;---- drv fare (2007 cents)
MW[28] = MI.3.11 ;---- drv added board time (0.01 min)
MW[29] = MI.3.13 ;---- drv acc time (0.01 min)
MW[30] = MI.3.14 ;---- drv acc distance (0.01 mile)
MW[31] = MI.3.16 ;---- drv park cost (2007 cents)
MW[32] = MI.3.15 ;---- drv park time (0.01 min)
MW[33] = MI.5.9 ;---- knr walk acc time (0.01 min)
MW[34] = MI.5.10 ;---- knr other walk time (0.01 min)
MW[35] = MI.5.7 ;---- knr ini.wait time (0.01 min)
MW[36] = MI.5.8 ;---- knr xfr wait time (0.01 min)
MW[37] = MI.5.3 ;---- knr ivt-total (0.01 min)
MW[38] = MI.5.4 ;---- knr ivt-commuter rail (0.01 min)
MW[39] = MI.5.2 ;---- knr ivt-exp bus (0.01 min)
MW[40] = MI.5.3 ;---- knr ivt-metrorail (0.01 min)
MW[41] = MI.5.5 ;---- knr ivt-new rail mode (0.01 min)
MW[42] = MI.5.6 ;---- knr ivt-new bus mode (0.01 min)
MW[43] = MI.5.1 ;---- knr ivt-local bus (0.01 min)

```

```

MW[44] = MI.5.12 ;---- knr transfers (0+)
MW[45] = MI.6.1 ;---- knr fare (2007 cents)
MW[46] = MI.5.11 ;---- knr added board time (0.01 min)
MW[47] = MI.5.13 ;---- knr acc time (0.01 min)
MW[48] = MI.5.14 ;---- knr acc distance (0.01 mile)

```

JLOOP

; assemble total IVTT

```

MW[05] = MW[06]+MW[07]+MW[08]+MW[09]+MW[10]+MW[11]
MW[19] = MW[20]+MW[21]+MW[22]+MW[23]+MW[24]+MW[25]
MW[37] = MW[38]+MW[39]+MW[40]+MW[41]+MW[42]+MW[43]

```

; zero-out fares for IVTT=0

```

IF (MW[05]=0 ) MW[13]=0
IF (MW[19]=0 ) MW[27]=0
IF (MW[37]=0 ) MW[45]=0

```

; deflate parking costs to 2007

```

MW[31] = @DEFLATIONFTR@*MW[31]

```

ENDJLOOP

ENDRUN

ENDLOOP ;---- PERIOD ----

## 7 Autoacc5.s

```

*del voya*.prn
; AutoAcc5.s - auto access link development - based on AutoAcc4.for from AECOM
; 2010-10-22 Previously, only bus PNR links were built to bus PNR & bus KNR paths.
; Now, we have created bus KNR access links from TAZ to bus stop node,
; instead of TAZ to PNR node (rjm/msm)
; 2012-12-03 File names have been made more consistent (msm)
; -----
; 2016-03-29(rjm):
; * Corrected problem causing erroneous PNR connections from the "CBD TAZ"
; Changed the line:
; TAZCBDDist = dcbd/52.8
; To this:
; IF (I=@NCBD@)
; TAZCBDDist = 0.001
; ELSE
; TAZCBDDist = dcbd/52.8
; ENDIF
; -----
; * Disabled "BackD" & "BackPC" params which are unused
; -----
; * Changed "CBD TAZ" (@NCBD@) from 35 to 8, per 3722 TAZ system
; -----
; Dimensions:
;
TAZSTASize = 7000
IZsize = 3675
FrstStaCen = 5001
Stasize = 1000

```

## Appendix C Cube Voyager Scripts

```

;;Input Files:
AMSkimFile = 'am_sov_mod.skm'
OPSkimFile = 'md_sov_mod.skm'
;
Sta_File = 'inputs\Station.dbf' ; Std. Station file
StaAccFile = 'inputs\StaAcc.dbf' ; Station mode-station type-max access dist.
lookup
JurisFile = 'inputs\Jur.dbf' ; juris code- juris group lookup
PentFile = 'inputs\Pen.dbf' ; TAZ in Pentagon's 'slug' shed
TNodeFile = 'TAZ_xys.dbf' ; TAZ XY Crd. file
Zonefile = 'inputs\ZONE.dbf' ; zonal land use file w/ jur code
;
; Output Files:
M_Pnr_AM = 'met_am_pnr.asc' ;unit 21
M_Knr_AM = 'met_am_knr.asc' ; 22
C_Pnr_AM = 'com_am.asc' ; 23
B_Pnr_AM = 'bus_am_pnr.asc' ; 24 renamed file
B_Knr_AM = 'bus_am_knr.asc' ; new file
L_Pnr_AM = 'lrt_am_pnr.asc' ; 25
N_Pnr_AM = 'new_am_pnr.asc' ; 26
L_Knr_AM = 'lrt_am_knr.asc' ; 43
N_Knr_AM = 'new_am_knr.asc' ; 44

M_Pnr_OP = 'met_op_pnr.asc' ;unit 21
M_Knr_OP = 'met_op_knr.asc' ; 22
C_Pnr_OP = 'com_op.asc' ; 23
B_Pnr_OP = 'bus_op_pnr.asc' ; 24 renamed file
B_Knr_OP = 'bus_op_knr.asc' ; new file
L_Pnr_OP = 'lrt_op_pnr.asc' ; 25
N_Pnr_OP = 'new_op_pnr.asc' ; 26
L_Knr_OP = 'lrt_op_knr.asc' ; 43
N_Knr_OP = 'new_op_knr.asc' ; 44
;
AutoAll = 'auto_all.asc' ; 40
;
; Params:
;;BackD = 1000.00 ; disabled param/ not used
;;BackPC = 0.30 ; disabled param/ not used
Divpc = 1.30 ; Diversion tolerance of the ratio: [TAZ/PNR Dist +
PNR/CBDTAZ Dist] / TAZ/CBDTAZ Dist
NCBD = 8 ; Representative TAZ of the region's CBD - TAZ 8
(Ellipse)

RUN PGM=MATRIX

ZONES=@TAZStaSize@
FILEI DBI[1] = "@Sta_File@"
FILEI DBI[2] = "@ZoneFile@"
FILEI DBI[3] = "@TNODEFILE@"
FILEI DBI[4] = "@StaAccFile@"

FileI LOOKUPI[1] = "@PentFile@"
FileI LOOKUPI[2] = "@JurisFile@"

FILEI MATI[1] = @AMSKIMFile@
FILEI MATI[2] = @OPSKIMFile@

MW[101] = mi.1.1 mw[102] = mi.1.2 ; am time, dist
MW[201] = mi.2.1 mw[202] = mi.2.2 ; op time, dist

FILEO PRINTO[1] = @AutoAll@

FILEO PRINTO[2] = @M_Knr_AM@
FILEO PRINTO[3] = @M_Knr_OP@

```

```

FILEO PRINTO[4] = @M_Pnr_AM@
FILEO PRINTO[5] = @M_Pnr_OP@

FILEO PRINTO[6] = @C_Pnr_AM@
FILEO PRINTO[7] = @C_Pnr_OP@

FILEO PRINTO[8] = @B_Pnr_AM@
FILEO PRINTO[9] = @B_Pnr_OP@

FILEO PRINTO[10] = @L_Pnr_AM@
FILEO PRINTO[11] = @L_Pnr_OP@
FILEO PRINTO[12] = @L_Knr_AM@
FILEO PRINTO[13] = @L_Knr_OP@

FILEO PRINTO[14] = @N_Pnr_AM@
FILEO PRINTO[15] = @N_Pnr_OP@
FILEO PRINTO[16] = @N_Knr_AM@
FILEO PRINTO[17] = @N_Knr_OP@

FILEO PRINTO[18] = @B_Knr_AM@
FILEO PRINTO[19] = @B_Knr_OP@

ARRAY Type=c1 MM = @STASize@,
STAPARK = @STASize@,
STAUSE = @STASize@,
MODE = 14

ARRAY NCT = @STASize@,
STAT = @STASize@,
STAP = @STASize@,
STAN1 = @STASize@,
STAC = @STASize@,
STAZ = @STASize@,
STAX = @STASize@,
STAY = @STASize@,
STAD = @STASize@,
ST_J = @STASize@

;SNAME = @STASize@ ;c27
;STAN2 = @STASize@
;STAN3 = @STASize@
;STAN4 = @STASize@
;STAPCAP = @STASize@
;STAC = @STASize@
;STAZ = @STASize@
;STAPKCost = @STASize@
;STAOFCost = @STASize@
;STAPKShad = @STASize@
;STAOFShad = @STASize@
;FirstYr = @STASize@

ARRAY JurCode = @IZSIZE@,
JurGrp = @IZSize@,
JurAcc = @IZSize@,
PentTAZ = @IZSize@,
TazX = @IZSize@,
TazY = @IZSize@,
AccDIST = 14,
AccCode = 14

;; Lookup list of origin TAZ's in the 'slug shed' of the Pentagon
Lookup Lookupi=1, name = PentNodes,
Lookup[1] = PentNode, result=Seqn, Interpolate=N, List=Y , fail=0,0,0

;; Lookup equivalence of Juris codes (0-23) and Access Groups

```

## Appendix C Cube Voyager Scripts

```

Lookup Lookupi=2, name = JurAcceqv,
      Lookup[1] = Jur_Code, result=AccGrp, Interpolate=N, List=Y,
fail=0,0,0

; Fill Access Code/distance 'lookup' Array
LOOP K = 1,dbi.4.NUMRECORDS
  x = DBIReadRecord(4,k)
  idx = dbi.4.recno

  Mode[idx] = di.4.Mode
  AccCode[idx] = di.4.AccCode
  AccDist[idx] = di.4.AccDist
ENDLOOP

; Fill in Station Array
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  idx = dbi.1.recno
  STACx = di.1.STAC
  STAZx = di.1.STAZ
  MMx = di.1.MM

  IF (MMx = 'M' || MMx = 'C')
    Ino = STACx
  ELSE
    Ino = STAZx
  ENDIF

  MM[idx] = di.1.MM
  NCT[idx] = di.1.NCT
  STAPARK[idx] = di.1.STAPARK
  STAUSE[idx] = di.1.STAUSE
  STAT[idx] = di.1.STAT
  STAZ[idx] = di.1.STAZ
  STAC[idx] = di.1.STAC
  STAN1[idx] = di.1.STAN1
  STAP[idx] = di.1.STAP
  STAX[idx] = di.1.STAX
  STAY[idx] = di.1.STAY
  ST_J[idx] = Ino

  STACnt = dbi.1.NUMRECORDS
ENDLOOP

IF (I=1) ;---;
Loop fdx = 1,STACnt
  ;; put in default driv

  ;; Add Acc. dist to Station Array with lookup array
  STAD[fdx] = 0
  Loop L = 1,dbi.4.NUMRECORDS ; 13
    IF (MM[fdx] = Mode[L] && NCT[fdx] = AccCode[L]) STAD[fdx] =
AccDist[L]
  ENDLOOP
  IF (STAPARK[fdx] != 'Y') STAD[fdx] = 300
  IF (STAUSE[fdx] != 'Y') STAD[fdx] = 0

  IF (NCT[fdx] = 8)
    Pentsta = STAC[fdx]
    Pentnode = STAT[fdx]
  ENDIF

  ;; write out transit XYs for used nodes

```

```

IF (MM[fdx] = 'M' || MM[fdx] = 'C')
  IF (STAUSE[fdx] = 'Y')
    print list = STAT[fdx](6), STAX[fdx](10), STAY[fdx](10), ' ; Final
index: ',fdx(5), File= extral.XY
  ENDIF
  IF (STAPARK[fdx] = 'Y')
    print list = STAP[fdx](6), STAX[fdx](10), STAY[fdx](10), ' ; Final
index: ',fdx(5), File= extra2.XY
  ENDIF
  ENDIF
  IF (MM[fdx] = 'B')
    IF (STAUSE[fdx] = 'Y')
      print list = STAN1[fdx](6), STAX[fdx](10), STAY[fdx](10), ' ;
Final index: ',fdx(5),File= extra3.XY
    ENDIF
  ENDIF

;;debug1 - echo print out station data
if (STAX[fdx] > 0)
  print form= 5.0 list =
    ' fdx: ', fdx(4),
    ' MM[fdx] ', MM[fdx],
    ' NCT[fdx] ', NCT[fdx],
    ' STAPARK[fdx] ', STAPARK[fdx],
    ' STAUSE[fdx] ', STAUSE[fdx],
    ' STAT[fdx] ', STAT[fdx],
    ' STAZ[fdx] ', STAZ[fdx],
    ' STAC[fdx] ', STAC[fdx],
    ' STAN1[fdx] ', STAN1[fdx],
    ' STAP[fdx] ', STAP[fdx],
    ' STAX[fdx] ', STAX[fdx](10),
    ' STAY[fdx] ', STAY[fdx](10),
    ' ST_J[fdx] ', ST_j[fdx],
    ' STAD[fdx] ', STAD[fdx],file= debug1.asc
  endif
;; End debug1

ENDLOOP

;
; Fill in TAZ Array - jurCodes
LOOP K = 1,dbi.2.NUMRECORDS
  x = DBIReadRecord(2,k)

  tdx = di.2.TAZ
  IF (tdx <= @IZSize@)
    JurCode[tdx] = di.2.JurCode ; std juris code (0-23)
  ENDIF
ENDLOOP

; Fill in TAZ Array - X,Ys
LOOP K = 1,dbi.3.NUMRECORDS
  x = DBIReadRecord(3,k)

  tdx = di.3.N
  IF (tdx <= @IZSize@)
    TAZX[tdx] = di.3.X
    TAZY[tdx] = di.3.Y

    IF (tdx = @NCBD@) ;
      CBDX = di.3.X ; X crd of CBD Taz
      CBDY = di.3.Y ; Y crd of CBD TAZ
    ENDIF
    print list = tdx, tazx[tdx],tazy[tdx],pentnode, file= tazsys.dbg

  PentTAZ[tdx]= PentNodes(1,tdx)
ENDIF

```

## Appendix C Cube Voyager Scripts

```

ENDLOOP
  print list = 'CBD TAZ X,Y = ', @NCBD@,' ' cbdx,' ', cbdy ,' Pent Sta Node=
', pentsta, file= cbd.dbg
ENDIF      ;---;

;-----
;; Now begin zonal I-Loop with binary matrices
;-----

IF (I <= @IZSize@)      ; if 'I's are internal TAZs'
IF (TAZX[I] > 0)        ; if 'I's are 'Used'

LOOP STADX =1,StaCnt    ; STADX LOOP

CurrJ= ST_j[stadx]

IJur      = jurcode[I]
IJurAcc   = JurAcceqv(1,IJur) ; Origin TAZ- juris group code 1-4 (determines river
crossings)

JTAZ      = STAZ[stadx]
JJur      = Jurcode[JTAZ]
JStaAcc   = JurAcceqv(1,JJur) ; Stat.TAZ- juris group code 1-4 (determines river
crossings)

;1 1 0 1 0 - original crossing array
;2 0 1 0 1
;3 1 0 1 0
;4 0 1 1 1

      IF ((IJurAcc = 1 && JStaAcc = 1) || (IJurAcc = 1 && JStaAcc = 3) || ;1 1 0
1 0      (IJurAcc = 2 && JStaAcc = 2) || (IJurAcc = 2 && JStaAcc = 4) || ;2 0 1
0 1      (IJurAcc = 3 && JStaAcc = 1) || (IJurAcc = 3 && JStaAcc = 3) || ;3 1 0
1 0      (IJurAcc = 4 && JStaAcc = 2) || (IJurAcc = 4 && JStaAcc = 3) || ;4 0 1
1 1      (IJurAcc = 4 && JStaAcc = 4) ) ; If station doesn't cross river ;

;; debug 4
IF (I= 35 )
  print list = 'i: ', i,' St_j: ',ST_j[stadx],' Sta Cen: ',STAC[stadx],'
StaPark: ',STAP[stadx],' Ijuracc: ',Ijuracc ,' JSTAacc: ',JStaAcc, file =debug4.asc
ENDIF
;; debug4

;; Clear all variables in the Jloop
amdists = 0
amtime  = 0
amspd   = 0

opdist  = 0
optime  = 0
opspd   = 0

Xdiff   = 0
Ydiff   = 0
xback   = 0
xi       = 0
xj       = 0

```

```

dcbd     = 0
TAZCBDDist = 0
STACBDDist = 0
TSdist   = 0
TAZSTADist = 0
xdiv     = 0
xback    = 0

;-----
JLOOP ; process J stations for each I-TAZ
;-----

;; Skip all j's not equal to current station/taz
IF (j != CurrJ) CONTINUE

  amdists = max(10.0,(mw[102][j] * 10.00))
  amtime  = mw[101][j]
  amspd   = 0.0
  IF (AMtime > 0) amspd =0.60 * amdists/AMtime
  IF (AMtime = 0)
    amspd = 25
    amdists = 50
  endif

  opdist  = max(10.0,(mw[202][j] * 10.00))
  optime  = mw[201][j]
  opspd   = 0.0
  IF (optime > 0) opspd =0.60 * opdist/optime
  IF (optime = 0)
    opspd = 25
    opdist = 50
  endif

;-----
;; Print out special AM/OP Pentagon KNR Access links
;-----
  IF (PentTAZ[I] > 0 && j = pentsta)

    Print Printo=1 list = ' SUPPLINK N=',I(5),'-',Pentnode(5),'
DIST=',AMDIST(6),
      ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)

    Print Printo=2 list = ' SUPPORT N=',I(5),'-',Pentnode(5),'
DIST=',AMDIST(6),
      ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)

    Print Printo=3 list = ' SUPPORT N=',I(5),'-',Pentnode(5),'
DIST=',OPDIST(6),
      ' ONEWAY=Y MODE=11 SPEED=',OPSPD(4)
  ENDIF
;; end AM/OP Pentagon Links-----
;; ENDJLOOP
;;

;-----
;; Calculate TAZ-CBD, Sta-CBD, TAZ-Sta distances & diversion ratio
;-----

  xback = 0
  xi     = abs(TAZx[I] - CBDX )
  xj     = abs(TAZy[I] - CBDY )
  dcbd   = sqrt(xi*xi+xj*xj)
  ;; TAZCBDDist = dcbd/52.8

```

Code (disabled)

< Original



## Appendix C Cube Voyager Scripts

```

Code      IF (I=@NCBD@) ; < Updated
          TAZCBDDist = 0.001 ; <
          ELSE ; <
          TAZCBDDist = dcbd/52.8 ; <
          ENDFIF ; <

          Xi = abs(STAX[STADX] - CBDX)
          Xj = abs(STAY[STADX] - CBDY)
          dscbd = sqrt(Xi*Xi+Xj*Xj)
          STACBDDist = dscbd/52.8

          xi = abs(TAZx[I] - STAX[STADX])
          xj = abs(TAZy[I] - STAY[STADX])
          dtsta = sqrt(xi*xi+xj*xj)
          TAZSTADist = round(dtsta/52.8)
          xdiv = 0.0
          if(TAZCBDDIST > 0.0) xdiv= (STACBDDist +
TAZSTADist)/TAZCBDDIST
          if(xdiv > @divpc@) xback=1

          ;; debug 7
          ; if ((i= 241 && stap[stadx] =7310) || (i= 397 && stap[stadx] =7523)
||
          ; (i= 483 && stap[stadx] =7302) || (i= 491 && stap[stadx] =7803)
||
          ; (i= 499 && stap[stadx] =8004) || (i= 680 && stap[stadx] =8008)
||
          ; (i= 746 && stap[stadx] =7543) || (i= 753 && stap[stadx] =7340)
||
          ; (i= 878 && stap[stadx] =8007) || (i= 964 && stap[stadx] =8034)
||
          ; (i= 1217 && stap[stadx] =7545) || (i= 1425 && stap[stadx] =7363)
||
          ; (i= 1935 && stap[stadx] =8210))

          IF (i = 20-38)
              print form = 8.2 list = 'ITAZ: ', i, ' JTAZ: ', j,
              ', ' , MM[stadx], ' ', ' STATION: ', stat[STADX], '
STAPARK: ', staP[STADX],
              ' IJURACC ', IJURACC, ' JSTAACC ', JSTAACC, ' TAZCBDDist:
', tazcbddist,
              ' ', ' STACBDDist: ', STACBDDist, ' TAZSTADist:
', TAZSTADist, ' Z-S dist.max: ', STAd[stadx], ' div.ratio: ', xdiv(6.4), file
=debug7.asc
              endif
          ;; debug7

          IF (xback = 0 && TAZStaDist <= STAD[stadx]) ; If diversion factor
and TAZ-station distance is acceptable

          ;;-----
          ;; Print out Standard Auto Access Links -----
          ;;-----

          ;;-----
          IF (MM[STADX] = 'M' && STAPARK[STADX] = 'Y') ;;; print
am/op metro PNR links

              Print Printo=1 list = ' SUPPLINK N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=4 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

```

```

          ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=5 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', opDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', opSPD(4)
          ENDFIF
          ;;; end print am/op
metro PNR links
          -----
          ;;-----

          -----
          IF ((MM[STADX] = 'M' && STAPARK[STADX] != 'Y' &&
NCT[STADX] !=9 && TAZSTADist <= 300.0) ||
              (MM[STADX] = 'M' && STAPARK[STADX] = 'Y' &&
TAZSTADist <= 300.0))
              ;;; print am/op metro KNR links

              Print Printo=1 list = ' SUPPLINK N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=2 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=3 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', opDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', opSPD(4)
          ENDFIF
          ;;; end print am/op
metro KNR links
          -----
          ;;-----

          IF (MM[STADX] = 'C' && STAPARK[STADX] = 'Y') ;;; print
am/op Comm Rail PNR links

              Print Printo=1 list = ' SUPPLINK N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=6 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO=7 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', opDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', opSPD(4)
          ENDFIF
          ;;; end print am/op
comm Rail PNR links
          -----
          ;;-----

          IF (MM[STADX] = 'B' && STAPARK[STADX] = 'Y') ;;; print
am/op Bus PNR links

              Print Printo=1 list = ' SUPPLINK N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

              Print PRINTO= 8 list = ' SUPPORT N=', I(5), '-
', stap[STADX] (5), ' DIST=', AMDIST(6),
              ' ONEWAY=Y MODE=11 SPEED=', AMSPD(4)

```

## Appendix C Cube Voyager Scripts

```

                Print PRINTO= 9 list = ' SUPPORT N=',I(5),'-
',stap[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
BUS PNR links
-----
;;-----
;;-----
                IF (MM[STADX] = 'B' && STAPARK[STADX] = 'Y') ;; print
am/op Bus KNR links
                Print Printo=1 list = ' SUPPLINK N=',I(5),'-
',stan1[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO= 18 list = ' SUPPORT N=',I(5),'-
',stan1[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO= 19 list = ' SUPPORT N=',I(5),'-
',stan1[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
BUS KNR links
-----
;;-----
;;-----
                IF (MM[STADX] = 'L' && STAPARK[STADX] = 'Y') ;; print
am/op Light Rail PNR links
                Print Printo=1 list = ' SUPPLINK N=',I(5),'-
',stap[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=10 list = ' SUPPORT N=',I(5),'-
',stap[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=11 list = ' SUPPORT N=',I(5),'-
',stap[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
Light Rail PNR links
-----
;;-----
;;-----
                IF (MM[STADX] = 'L' && STAPARK[STADX] !='Y' &&
NCT[STADX] !=9 && TAZSTADist < 300.0 ) ;; print am/op Light Rail KNR links
                Print Printo=1 list = ' SUPPLINK N=',I(5),'-
',stat[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=12 list = ' SUPPORT N=',I(5),'-
',stat[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=13 list = ' SUPPORT N=',I(5),'-
',stat[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
                ;; end print am/op
Comm KNR links
-----
;;-----
;;-----
                IF (MM[STADX] = 'N' && STAPARK[STADX] = 'Y') ;; print
am/op New PNR links
                Print Printo=1 list = ' SUPPLINK N=',I(5),'-
',stap[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=14 list = ' SUPPORT N=',I(5),'-
',stap[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=15 list = ' SUPPORT N=',I(5),'-
',stap[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
New PNR links
-----
;;-----
;;-----
                IF (MM[STADX] = 'N' && STAPARK[STADX] !='Y' &&
NCT[STADX] !=9 && TAZSTADist < 300.0 ) ;; print am/op New KNR links
                Print Printo=1 list = ' SUPPLINK N=',I(5),'-
',stat[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=16 list = ' SUPPORT N=',I(5),'-
',stat[STADX] (5),' DIST=',AMDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',AMSPD(4)
                Print PRINTO=17 list = ' SUPPORT N=',I(5),'-
',stat[STADX] (5),' DIST=',opDIST(6),
                ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
            ENDIF
                ;; end print am/op
New KNR links
-----
;;-----
;;-----
                ENDIF
                ;   endif diversion factor and TAZ-Sta
distance is acceptable
ENDJLOOP
ENDIF
                ;   endif station doesn't cross river
ENDLOOP
                ; STATION (STADX) Loop
ENDIF
                ;   endif 'I's are Used
ENDIF
                ;   endif 'I's are internal
ENDRUN
*copy extral.xy+extra2.XY+extra3.xy extra.xy
*del extral.xy

```

```
*del extra2.xy
*del extra3.xy
```

## 8 Average\_Link\_Speeds.s

```
;; Average the restrained speeds on highway links using MSA
;; 8/5/2011 Corrected NTPCTadt factor from 35.0 to 15.0.
VDF_File = '..\support\hwy_assign_Conical_VDF.s'      ;; Volume Delay
Functions file
Iter = '%_iter_%'
Prev = '%_prev_%'

AMPctadt = 41.7
PMPctadt = 29.4
MDPctadt = 17.7
NTPctadt = 15.0

IF (iter = 'pp') itrno = 0
IF (iter = 'i1') itrno = 1
IF (iter = 'i2') itrno = 2
IF (iter = 'i3') itrno = 3
IF (iter = 'i4') itrno = 4
IF (iter = 'i5') itrno = 5
IF (iter = 'i6') itrno = 6

;; Remove VOLUME,VMT,SPEED-relate variables from a copy of original loaded links
file
RUN PGM=NETWORK
NETI[1] = @iter@_HWY.tem1
NETO = @iter@_HWY.tem2,
      exclude=@iter@AMVOL, @iter@PMVOL, @iter@MDVOL,
@iter@NTVOL,@iter@24Vol,
      @iter@AMVMT, @iter@PMVMT, @iter@MDVMT, @iter@NTVMT,
      @iter@AMFFSPD, @iter@PMFFSPD, @iter@MDFFSPD, @iter@NTFFSPD,

@iter@AMHRLKCAP,@iter@PMHRLKCAP,@iter@MDHRLKCAP,@iter@NTHRLKCAP,

@iter@AMHRLNCAP,@iter@PMHRLNCAP,@iter@MDHRLNCAP,@iter@NTHRLNCAP,
      @iter@AMVVC, @iter@PMVVC, @iter@MDVVC, @iter@NTVVC,
      @iter@AMVDF, @iter@PMVDF, @iter@MDVDF, @iter@NTVDF,
      @iter@AMSPD, @iter@PMSPD, @iter@MDSPD, @iter@NTSPD

ENDRUN

RUN PGM=NETWORK
NETI[1] = @iter@_HWY.tem2      ;; original LL file with speeds
removed
NETI[2] = @prev@_HWY.net      ;; previous iteration LL file w/
final speeds
NETI[3] = @iter@_Assign_Output.net ;; current iteration LL file w/
traffic assigned speeds
NETO = @iter@_Averaged_HWY.net

@prev@AMVOL = LI.2.@prev@AMVOL
@prev@MDVOL = LI.2.@prev@MDVOL
@prev@PMVOL = LI.2.@prev@PMVOL
@prev@NTVOL = LI.2.@prev@NTVOL

@iter@AMVOL = LI.3.@iter@AMVOL
@iter@MDVOL = LI.3.@iter@MDVOL
@iter@PMVOL = LI.3.@iter@PMVOL
@iter@NTVOL = LI.3.@iter@NTVOL

;; Define averaging proportions based on iteration no.
```

```
IF (@itrno@ = 1)
  @prev@_VOL_Shr = 0.000
  @iter@_VOL_Shr = 1.000
ELSEIF (@itrno@ = 2)
  @prev@_VOL_Shr = 0.500
  @iter@_VOL_Shr = 0.500
ELSEIF (@itrno@ = 3)
  @prev@_VOL_Shr = 0.666
  @iter@_VOL_Shr = 0.334
ELSEIF (@itrno@ = 4)
  @prev@_VOL_Shr = 0.750
  @iter@_VOL_Shr = 0.250
ELSEIF (@itrno@ = 5)
  @prev@_VOL_Shr = 0.800
  @iter@_VOL_Shr = 0.200
ELSEIF (@itrno@ = 6)
  @prev@_VOL_Shr = 0.833
  @iter@_VOL_Shr = 0.167
ENDIF

;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Rmps
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@

@iter@AMVOL = @prev@AMVOL * @prev@_VOL_Shr + @iter@AMVOL * @iter@_VOL_Shr ;
Final AM Link Volume
@iter@PMVOL = @prev@PMVOL * @prev@_VOL_Shr + @iter@PMVOL * @iter@_VOL_Shr ;
Final PM Link Volume
@iter@MDVOL = @prev@MDVOL * @prev@_VOL_Shr + @iter@MDVOL * @iter@_VOL_Shr ;
Final MD Link Volume
@iter@NTVOL = @prev@NTVOL * @prev@_VOL_Shr + @iter@NTVOL * @iter@_VOL_Shr ;
Final NT Link Volume
@iter@24VOL = @iter@AMVOL + @iter@MDVOL + @iter@PMVOL + @iter@NTVOL ;
Final 24hr Link Volume

@iter@AMVMT = @iter@AMVOL * distance
; Final AM link VMT
@iter@PMVMT = @iter@PMVOL * distance
; Final PM link VMT
@iter@MDVMT = @iter@MDVOL * distance
; Final MD link VMT
@iter@NTVMT = @iter@NTVOL * distance
; Final NT link VMT
@iter@24VMT = (@iter@AMVol + @iter@MDVol + @iter@PMVol + @iter@NTVol)* distance
; Final daily VMT

@iter@AMFFSPD =SPEEDFOR (AMLANE,SPDCCLASS) ;
Freeflow AM speed
@iter@PMFFSPD =SPEEDFOR (PMLANE,SPDCCLASS) ;
Freeflow PM speed
@iter@MDFFSPD =SPEEDFOR (OPLANE,SPDCCLASS) ;
Freeflow MD speed
```

## Appendix C Cube Voyager Scripts

```

@iter@NTFFSPD =SPEEDFOR(OPLANE,SPDCLASS) ;
Freeflow NT speed ;

AMHRLKCAP=CAPACITYFOR(AMLANE,CAPCLASS) ;
Hrly Link capacity ;
PMHRLKCAP=CAPACITYFOR(PMLANE,CAPCLASS) ;
Hrly Link capacity ;
MDHRLKCAP=CAPACITYFOR(OPLANE,CAPCLASS) ;
Hrly Link capacity ;
NTHRLKCAP=CAPACITYFOR(OPLANE,CAPCLASS) ;
Hrly Link capacity ;

AMHRLNAP=CAPACITYFOR(1,CAPCLASS) ;
Hrly Lane capacity ;
PMHRLNAP=CAPACITYFOR(1,CAPCLASS) ;
Hrly Lane capacity ;
MDHRLNAP=CAPACITYFOR(1,CAPCLASS) ;
Hrly Lane capacity ;
NTHRLNAP=CAPACITYFOR(1,CAPCLASS) ;
Hrly Lane capacity ;

@iter@AMVC=(@iter@AMVOL*(@AMPctadt@/100.0)/AMHRLKCAP) ;
AM VC ratio ;
@iter@PMVC=(@iter@PMVOL*(@PMPctadt@/100.0)/PMHRLKCAP) ;
PM VC ratio ;
@iter@MDVC=(@iter@MDVOL*(@MDPctadt@/100.0)/MDHRLKCAP) ;
MD VC ratio ;
@iter@NTVC=(@iter@NTVOL*(@NTPctadt@/100.0)/NTHRLKCAP) ;
NT VC ratio ;

@iter@AMVDF = VCRV((Ftype + 1), @iter@AMVC) ;
AM VDF ;
@iter@PMVDF = VCRV((Ftype + 1), @iter@PMVC) ;
PM VDF ;
@iter@MDVDF = VCRV((Ftype + 1), @iter@MDVC) ;
MD VDF ;
@iter@NTVDF = VCRV((Ftype + 1), @iter@NTVC) ;
NT VDF ;

@iter@AMSPD = @iter@AMFFSPD ;
AM restrained speed ;
@iter@PMSPD = @iter@PMFFSPD ;
PM restrained speed ;
@iter@MDSPD = @iter@MDFFSPD ;
MD restrained speed ;
@iter@NTSPD = @iter@NTFFSPD ;
NT restrained speed ;

if (@iter@AMVDF > 0) @iter@AMSPD = @iter@AMFFSPD / @iter@AMVDF ;
AM restrained speed ;
if (@iter@PMVDF > 0) @iter@PMSPD = @iter@PMFFSPD / @iter@PMVDF ;
PM restrained speed ;
if (@iter@MDVDF > 0) @iter@MDSPD = @iter@MDFFSPD / @iter@MDVDF ;
MD restrained speed ;
if (@iter@NTVDF > 0) @iter@NTSPD = @iter@NTFFSPD / @iter@NTVDF ;
NT restrained speed ;

_ATYPE=SPDCLASS%10 ;
Area Type ;
_cnt = 1.0 ;

;; debug section - select some links to check with ;;
IF (li.1.1.a =1-13,23000-23100,33000-33200)
  print form=5.2 list = a(6), b(6),
    ' AM_Prev_Vol > ', _@prev@AMVol,
    ' AM_Prev_Shr > ', _@prev@VOL_Shr,
    ' AM_Curr_Vol > ', _@iter@AMVol,
    ' AM_Curr_Shr > ', _@iter@VOL_Shr,
    ' AMAvgVOL > ', _@iter@AMVOL,
    ' AMLnkCap > ', AMHRLKCAP,
    ' AMVC > ', @iter@AMVC,
    ' AMVDF > ', @iter@AMVDF,
    ' AMSpd > ', @iter@AMSPD,
    ' PM_Prev_Vol > ', _@prev@PMVol,
    ' PM_Prev_Shr > ', _@prev@VOL_Shr,
    ' PM_Curr_Vol > ', _@iter@PMVol,
    ' PM_Curr_Shr > ', _@iter@VOL_Shr,
    ' PMAvgVOL > ', _@iter@PMVOL,
    ' PMLnkCap > ', PMHRLKCAP,
    ' PMVC > ', @iter@PMVC,
    ' PMVDF > ', @iter@PMVDF,
    ' PMSpd > ', @iter@PMSPD,
    ' MD_Prev_Vol > ', _@prev@MDVol,
    ' MD_Prev_Shr > ', _@prev@VOL_Shr,
    ' MD_Curr_Vol > ', _@iter@MDVol,
    ' MD_Curr_Shr > ', _@iter@VOL_Shr,
    ' MDAvgVOL > ', _@iter@MDVOL,
    ' MDLnkCap > ', MDHRLKCAP,
    ' MDVC > ', @iter@MDVC,
    ' MDVDF > ', @iter@MDVDF,
    ' MDSpd > ', @iter@MDSPD,
    ' NT_Prev_Vol > ', _@prev@NTVol,
    ' NT_Prev_Shr > ', _@prev@VOL_Shr,
    ' NT_Curr_Vol > ', _@iter@NTVol,
    ' NT_Curr_Shr > ', _@iter@VOL_Shr,
    ' NTAvgVOL > ', _@iter@NTVOL,
    ' NTLnkCap > ', NTHRLKCAP,
    ' NTVC > ', @iter@NTVC,
    ' NTVDF > ', @iter@NTVDF,
    ' NTSpd > ', @iter@NTSPD,

    file= @iter@_Average_Link_Speeds.txt

  ENDIF

;;compute WEIGHTED restrained and freeflow SPEEDS for Aggregate summaries

_AMWRSPD =ROUND(@iter@AMVMT * @iter@AMSPD)
_AMWFFSPD=ROUND(@iter@AMVMT * @iter@AMFFSPD)

_PMWRSPD =ROUND(@iter@PMVMT * @iter@PMSPD)
_PMWFFSPD=ROUND(@iter@PMVMT * @iter@PMFFSPD)

_MDWRSPD =ROUND(@iter@MDVMT * @iter@MDSPD)
_MDWFFSPD=ROUND(@iter@MDVMT * @iter@MDFFSPD)

_NTWRSPD =ROUND(@iter@NTVMT * @iter@NTSPD)
_NTWFFSPD=ROUND(@iter@NTVMT * @iter@NTFFSPD)

=====
;; AM X-Tabs
=====
;; Crosstab AM VMT,Weighted Restrained Speed, Weighted FF Speed by JUR and FTYPE
CROSSTAB VAR=@iter@AMVMT, AMWRSPD, AMWFFSPD, FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= AMWRSPD/@iter@AMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= AMWFFSPD/@iter@AMVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab AM VMT,Weighted Restrained Speed, Weighted FF Speed by ATYPE and FTYPE

```

## Appendix C Cube Voyager Scripts

```

CROSSTAB VAR=@iter@AMVMT, AMWRSPD, AMWFFSPD, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= AMWRSPD/@iter@AMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= AMWFFSPD/@iter@AMVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab AM VMT,Weighted Restrained Speed, Weighted FF Speed by AM V/C and
FTYPE
CROSSTAB VAR=@iter@AMVMT, AMWRSPD, AMWFFSPD, FORM=12cs,
ROW=@iter@AMVC, RANGE=0-2-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= AMWRSPD/@iter@AMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= AMWFFSPD/@iter@AMVMT, FORM=12.2cs ; AVG FINAL SPD

;;
;;=====
;; PM X-Tabs
;;=====
;; Crosstab PM VMT,Weighted Restrained Speed, Weighted FF Speed by JUR and FTYPE
CROSSTAB VAR=@iter@PMVMT, PMWRSPD, PMWFFSPD, FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= PMWRSPD/@iter@PMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= PMWFFSPD/@iter@PMVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab PM VMT,Weighted Restrained Speed, Weighted FF Speed by ATYPE and FTYPE
CROSSTAB VAR=@iter@PMVMT, PMWRSPD, PMWFFSPD, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= PMWRSPD/@iter@PMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= PMWFFSPD/@iter@PMVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab PM VMT,Weighted Restrained Speed, Weighted FF Speed by AM V/C and
FTYPE
CROSSTAB VAR=@iter@PMVMT, PMWRSPD, PMWFFSPD, FORM=12cs,
ROW=@iter@PMVC, RANGE=0-2-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= PMWRSPD/@iter@PMVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= PMWFFSPD/@iter@PMVMT, FORM=12.2cs ; AVG FINAL SPD

;;
;;=====
;; MD X-Tabs
;;=====
;; Crosstab MD VMT,Weighted Restrained Speed, Weighted FF Speed by JUR and FTYPE
CROSSTAB VAR=@iter@MDVMT, MDWRSPD, MDWFFSPD, FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= MDWRSPD/@iter@MDVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= MDWFFSPD/@iter@MDVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab MD VMT,Weighted Restrained Speed, Weighted FF Speed by ATYPE and FTYPE
CROSSTAB VAR=@iter@MDVMT, MDWRSPD, MDWFFSPD, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= MDWRSPD/@iter@MDVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= MDWFFSPD/@iter@MDVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab MD VMT,Weighted Restrained Speed, Weighted FF Speed by AM V/C and
FTYPE
CROSSTAB VAR=@iter@MDVMT, MDWRSPD, MDWFFSPD, FORM=12cs,
ROW=@iter@MDVC, RANGE=0-2-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= MDWRSPD/@iter@MDVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= MDWFFSPD/@iter@MDVMT, FORM=12.2cs ; AVG FINAL SPD

```

```

;;
;;=====
;; NT X-Tabs
;;=====
;; Crosstab NT VMT,Weighted Restrained Speed, Weighted FF Speed by JUR and FTYPE
CROSSTAB VAR=@iter@NTVMT, NTWRSPD, NTWFFSPD, FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= NTWRSPD/@iter@NTVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= NTWFFSPD/@iter@NTVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab NT VMT,Weighted Restrained Speed, Weighted FF Speed by ATYPE and FTYPE
CROSSTAB VAR=@iter@NTVMT, NTWRSPD, NTWFFSPD, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= NTWRSPD/@iter@NTVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= NTWFFSPD/@iter@NTVMT, FORM=12.2cs ; AVG FINAL SPD

;; Crosstab NT VMT,Weighted Restrained Speed, Weighted FF Speed by AM V/C and
FTYPE
CROSSTAB VAR=@iter@NTVMT, NTWRSPD, NTWFFSPD, FORM=12cs,
ROW=@iter@NTVC, RANGE=0-2-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP= NTWRSPD/@iter@NTVMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP= NTWFFSPD/@iter@NTVMT, FORM=12.2cs ; AVG FINAL SPD

;;
;;=====
;; DAILY X-Tabs
;;=====
;; Crosstab DAILY VMT by JUR and FTYPE
CROSSTAB VAR=@iter@24VMT, FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6

;; Crosstab DAILY VMT by ATYPE and FTYPE
CROSSTAB VAR=@iter@24VMT, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6

ENDRUN

```

## 9 CheckStationAccess.s

```

*del voya*.prn
;=====
; CheckStationAccess.s
; Purpose: To check whether transit stations are accessible

; RM, RN
; Date: 2018-03-30
;
;=====
RUN PGM=MATRIX

ZONES=8000
FILEI LOOKUPI[1] = 'inputs\node.dbf'
FILEI MATI[1] = '%_iter_%_am_sov_mod.skm'

```

## Appendix C Cube Voyager Scripts

```

LOOKUP LOOKUPI = 1, NAME = NODES, LOOKUP[1] = N, RESULT = N, INTERPOLATE = N, FAIL =
0,0,0, LIST = Y

MW[1] = MI.1.1
TIMESUM = ROWSUM(1)
IF (I = 5000 - 8000)
    NODEVAL = NODES(1, I)
    IF (NODEVAL>0 && TIMESUM = 0)
        PRINT LIST = 'STATION NUMBER ', NODEVAL(5), ' HAS NO SKIM BUILT TO IT'
        ABORT MSG = STATION CENTROIDS WITHOUT SKIMS. PLEASE CHECK THE
NETWORK.
    ENDIF
ENDIF
ENDRUN

```

## 10 Combine\_Tables\_For\_TrAssign\_Parallel.s

```

; COMBINE TRIPS FOR ALL PURPOSES INTO ONE FOR EACH SUB TRANSIT MODE
distribute intrastep=%useIdp% multistep=%useMdp%

RUN PGM=MATRIX
distributeIntrastep processId='MWC0G', ProcessList=%subnode%
MATI[1] = '%_iter_%_HBS_NL_MC.MTT' ;AECOM HBS Mode Choice file (Input)
MATI[2] = '%_iter_%_HBO_NL_MC.MTT' ;AECOM HBO Mode Choice file (Input)
MATI[3] = '%_iter_%_NHW_NL_MC.MTT' ;AECOM NHW Mode Choice file (Input)
MATI[4] = '%_iter_%_NHO_NL_MC.MTT' ;AECOM NHO Mode Choice file (Input)
MATI[5] = '%_iter_%_NHO_NL_MC.MTT' ;AECOM NHO Mode Choice file (Input)

; Note: There are 11 tables on the *.TRP files, not 12, since, for CR, KNR and PNR
are combined
MATO[1]='%_iter_%_AMMS.TRP',MO=04-14,
NAME = WK_CR, WK_BUS, WK_BUS_MR, WK_MR, PNR_KNR_CR, PNR_BUS, KNR_BUS, PNR_BUS_MR,
KNR_BUS_MR, PNR_MR, KNR_MR
MATO[2]='%_iter_%_OPMS.TRP',MO=24-34,
NAME = WK_CR, WK_BUS, WK_BUS_MR, WK_MR, PNR_KNR_CR, PNR_BUS, KNR_BUS, PNR_BUS_MR,
KNR_BUS_MR, PNR_MR, KNR_MR

;PK TRIP MATRICES
MW[1]=MI.1.1 ; AM DR ALONE
MW[2]=MI.1.2 ; AM SR2
MW[3]=MI.1.3 ; AM SR3+
MW[4]=MI.1.4 ; AM WK-CR
MW[5]=MI.1.5 ; AM WK-BUS
MW[6]=MI.1.6 ; AM WK-BU/MR
MW[7]=MI.1.7 ; AM WK-MR
MW[8]=MI.1.8 ; AM PNR-CR, KNR-CR
MW[9]=MI.1.9 ; AM PNR-BUS
MW[10]=MI.1.10 ; AM KNR-BUS
MW[11]=MI.1.11 ; AM PNR-BU/MR
MW[12]=MI.1.12 ; AM KNR-BU/MR
MW[13]=MI.1.13 ; AM PNR-MR
MW[14]=MI.1.14 ; AM KNR-MR

;OP TRIP MATRICES
MW[21]=MI.2.1+MI.3.1+MI.4.1+MI.5.1 ; OP DR ALONE
MW[22]=MI.2.2+MI.3.2+MI.4.2+MI.5.2 ; OP SR2
MW[23]=MI.2.3+MI.3.3+MI.4.3+MI.5.3 ; OP SR3+
MW[24]=MI.2.4+MI.3.4+MI.4.4+MI.5.4 ; OP WK-CR
MW[25]=MI.2.5+MI.3.5+MI.4.5+MI.5.5 ; OP WK-BUS
MW[26]=MI.2.6+MI.3.6+MI.4.6+MI.5.6 ; OP WK-BU/MR

```

```

MW[27]=MI.2.7+MI.3.7+MI.4.7+MI.5.7 ; OP WK-MR
MW[28]=MI.2.8+MI.3.8+MI.4.8+MI.5.8 ; OP PNR-CR, KNR-CR
MW[29]=MI.2.9+MI.3.9+MI.4.9+MI.5.9 ; OP PNR-BUS
MW[30]=MI.2.10+MI.3.10+MI.4.10+MI.5.10 ; OP KNR-BUS
MW[31]=MI.2.11+MI.3.11+MI.4.11+MI.5.11 ; OP PNR-BU/MR
MW[32]=MI.2.12+MI.3.12+MI.4.12+MI.5.12 ; OP KNR-BU/MR
MW[33]=MI.2.13+MI.3.13+MI.4.13+MI.5.13 ; OP PNR-MR
MW[34]=MI.2.14+MI.3.14+MI.4.14+MI.5.14 ; OP KNR-MR

ENDRUN

```

## 11 Demo\_Models.s

```

;=====
; Demo_Models.S
;
; Version 2.3, 3722 TAZ System - Demographic Model
;
; The models have been updated using the 2007 ACS
; Program to Allocation Total Zonal Households among 64 Classes:
; 4 HH Size groups by 4 Income Groups by 4 Veh. Avail. groups
;
; Programmer: Milone
; Date: 09/3/10
;
; Test: BASE
;=====
;
;
; ZONESIZE = 3722 ; No. of TAZs
LastIzn = 3675 ; Last Internal TAZ no.

JrCl = 24 ; No. of Juris. Classes (transformed JURIS.
Code 0-23 becomes 1-24)
ArCl = 6 ; No. of Area Classe (ATypes)
SzCl = 4 ; No. of HH Size Classes
InCl = 4 ; No. of Income Classes
VaCl = 4 ; No. of Veh Avail Classes

ZNFIL IN1 = 'inputs\ZONE.dbf' ; Input Zonal Land Use File
ZNFIL IN2 = 'AreaType_File.dbf' ; Input Zonal Area Type File from network
building

Rept = '%_iter_%_Demo_Models.txt' ; Summary Reports

ZNFIL INa1 = '%_iter_%_AM_WK_MR_JobAcc.dbf' ; Input Jobs accessible within 45
min. by AM WalkAcc Metrorail Only Service
ZNFIL INa2 = '%_iter_%_AM_DR_MR_JobAcc.dbf' ; Input Jobs accessible within 45
min. by AM DriveAcc Metrorail Only Service
ZNFIL INa3 = '%_iter_%_AM_WK_BM_JobAcc.dbf' ; Input Jobs accessible within 45
min. by AM WalkAcc Bus&Metrorail Service
ZNFIL INa4 = '%_iter_%_AM_DR_BM_JobAcc.dbf' ; Input Jobs accessible within 45
min. by AM DriveAcc Bus&Metrorail Service

ZNFIL OU1 = 'HHI1_SV.txt' ; Output Zonal Income 1 HH by Size& VehAv
Classes: i1slv1,i1slv2,...,i1slv4

```

# Appendix C Cube Voyager Scripts

```

ZNFIL02 = 'HHI2 SV.txt' ; Output Zonal Income 2 HH by Size& VehAv
Classes: i2s1v1,i2s1v2,...,i2s4v4
ZNFIL03 = 'HHI3 SV.txt' ; Output Zonal Income 3 HH by Size& VehAv
Classes: i3s1v1,i3s1v2,...,i3s4v4
ZNFIL04 = 'HHI4 SV.txt' ; Output Zonal Income 4 HH by Size& VehAv
Classes: i4s1v1,i4s1v2,...,i4s4v4

ZonalCCHHS = '%_iter_%_Demo_Models_HHbyISV.dbf' ; output zonal HHs by 64 cross-
classes

Ofmt = '(12.2)' ; Format of Output file data Note:
Integer/real Spec. Here!

RUN PGM=MATRIX
ZONES=@ZONESIZE@

pageheight=32767 ; Preclude header breaks

; Set up zone arrays for accumulating I/O variables
;

ARRAY ISZA =@SzCl@, ; Initial Marginal HH Totals by size
levels
IINA =@InCl@, ; Initial Marginal HH Totals by income
levels
AreaA =@ArCl@, ; Area Type class size
CSZA =@SzCl@, ; Computed Marginal HH Totals by size
levels
CINA =@InCl@, ; Computed Marginal HH Totals by income
levels
CSZAdjA =@SzCl@, ; Marginal HH adjustment ftr by Income
class
CINAdjA =@InCl@, ; Marginal Inc adjustment ftr by HH
size class
P_VA =@VaCl@, ; Veh Avail probabilities
CVAA =@VaCl@, ; Veh Avail Totals
JurA =@JrCl@, ; Juris. HH Totals array
RegSzA =@SzCl@, ; Regional HH by Size array
RegInA =@InCl@, ; Regional HH by Inc array
RegVaA =@VaCl@, ; Regional HH by VeAv array
HH_ArCoopT =@ArCl@

ARRAY CSZINA =@SzCl@,@InCl@ ; HH Size by Income level Matrix,
11,12,13,...,44
ARRAY JurSzA =@JrCl@,@SzCl@ ; Juris. HH by size array
ARRAY JurInA =@JrCl@,@InCl@ ; Juris. HH by Inc array
ARRAY JurVaA =@JrCl@,@VaCl@ ; Juris. HH by VeAv array
ARRAY RegSzInA =@SzCl@,@InCl@ ; Regional Size by Inc array
ARRAY RegVaSzA =@VaCl@,@SzCl@ ; Regional V by Size matrix
ARRAY RegVaInA =@VaCl@,@InCl@ ; Regional V1 by Inc matrix

ARRAY ArSzA =@ArCl@,@SzCl@ ; Area Type by size array
ARRAY ArInA =@ArCl@,@InCl@ ; Area Type by Inc array
ARRAY ArVaA =@ArCl@,@VaCl@ ; Area Type by VeAv array

ARRAY CSZINVAA =@SzCl@,@InCl@,@VaCl@ ; Veh Avail by HH Size by Inc Matrix,
111,112,113,...,444
ARRAY RegSzInvAA =@SzCl@,@InCl@,@VaCl@ ; Regional Size by Inc by vehav array

;=====
; Define Loop-up Tables =
;=====
;
;=====

```

```

; HH Size Distribution from 2000 CTPP
;=====
LOOKUP Name=SZPCTA,
LOOKUP[1] = 1,Result = 2,
LOOKUP[2] = 1,Result = 3,
LOOKUP[3] = 1,Result = 4,
LOOKUP[4] = 1,Result = 5,
Interpolate = N, FAIL=0,0,0,
; Avg HHSIZE PctHH1psn PctHH2psn PctHH3Psn PctHH4+Psn

R=" 1.0, 100.0, 0.0, 0.0, 0.0",
" 1.1, 86.7, 10.5, 1.0, 1.8",
" 1.2, 78.2, 15.8, 4.1, 1.9",
" 1.3, 72.7, 20.4, 4.9, 2.0",
" 1.4, 67.1, 24.7, 5.8, 2.4",
" 1.5, 63.0, 27.1, 6.7, 3.2",
" 1.6, 59.0, 28.9, 7.9, 4.2",
" 1.7, 55.2, 30.2, 8.7, 5.9",
" 1.8, 50.9, 31.1, 10.1, 7.9",
" 1.9, 46.7, 31.7, 11.5, 10.1",
" 2.0, 42.8, 32.1, 12.7, 12.4",
" 2.1, 39.0, 32.3, 14.0, 14.7",
" 2.2, 35.5, 32.4, 15.0, 17.1",
" 2.3, 32.2, 32.4, 16.0, 19.4",
" 2.4, 29.1, 32.3, 16.9, 21.7",
" 2.5, 26.3, 32.1, 17.6, 24.0",
" 2.6, 23.8, 31.9, 18.2, 26.1",
" 2.7, 21.5, 31.5, 18.7, 28.3",
" 2.8, 19.4, 31.1, 19.2, 30.3",
" 2.9, 17.4, 30.5, 19.8, 32.3",
" 3.0, 15.6, 29.8, 20.3, 34.3",
" 3.1, 14.0, 28.9, 20.7, 36.4",
" 3.2, 12.6, 27.9, 20.8, 38.7",
" 3.3, 11.3, 26.6, 20.9, 41.2",
" 3.4, 10.2, 25.0, 20.8, 44.0",
" 3.5, 09.2, 23.2, 20.4, 47.2",
" 3.6, 08.3, 21.2, 19.6, 50.9",
" 3.7, 07.5, 18.9, 18.4, 55.2",
" 3.8, 06.7, 15.6, 17.4, 60.3",
" 3.9, 05.9, 11.2, 16.5, 66.4"

;=====
; income level distribution from 2000 CTPP
; adjusted by rjm 9/5/10 per 2007 ACS
;=====
LOOKUP Name=INPCTA,
LOOKUP[1] = 1,Result = 2,
LOOKUP[2] = 1,Result = 3,
LOOKUP[3] = 1,Result = 4,
LOOKUP[4] = 1,Result = 5,
Interpolate = N, FAIL=0,0,0,
; inc level: QRT1 QRT2 QRT3 QRT4 ; proportion of
income ; zonal median inc.
; to regional median

R=" 0, 100.00 0.00 0.00 0.00 ", ; 0 inc ratio
" 1, 88.83 8.19 2.34 0.64 ", ; 0.1 inc ratio
" 2, 80.54 14.73 3.13 1.60 ", ; 0.2 inc ratio
" 3, 73.42 20.29 4.23 2.05 ", ; 0.3 inc ratio
" 4, 65.32 25.44 6.44 2.80 ", ; 0.4 inc ratio
" 5, 56.93 29.97 9.32 3.78 ", ; 0.5 inc ratio
" 6, 48.78 33.41 12.51 5.30 ", ; 0.6 inc ratio
" 7, 41.27 35.85 15.69 7.19 ", ; 0.7 inc ratio
" 8, 34.56 36.96 18.64 9.84 ", ; 0.8 inc ratio
" 9, 28.84 36.84 21.22 13.10 ", ; 0.9 inc ratio
" 10, 24.27 35.69 23.28 16.77 ", ; 1 inc ratio
" 11, 20.63 33.70 24.75 20.92 ", ; 1.1 inc ratio

```

Appendix C Cube Voyager Scripts

```

" 12, 17.89 30.95 25.59 25.56 ", ; 1.2 inc ratio
" 13, 16.00 27.91 25.83 30.27 ", ; 1.3 inc ratio
" 14, 14.63 24.78 25.45 35.15 ", ; 1.4 inc ratio
" 15, 13.72 21.74 24.71 39.83 ", ; 1.5 inc ratio
" 16, 12.99 19.13 23.53 44.35 ", ; 1.6 inc ratio
" 17, 12.23 17.04 22.16 48.57 ", ; 1.7 inc ratio
" 18, 11.39 15.65 20.67 52.29 ", ; 1.8 inc ratio
" 19, 10.50 14.70 19.19 55.61 ", ; 1.9 inc ratio
" 20, 9.71 14.35 17.77 58.17 ", ; 2 inc ratio
" 21, 8.74 14.16 16.60 60.50 ", ; 2.1 inc ratio
" 22, 8.05 14.11 15.46 62.38 ", ; 2.2 inc ratio
" 23, 7.79 14.02 14.54 63.65 ", ; 2.3 inc ratio
" 24, 7.37 13.77 14.08 64.77 ", ; 2.4 inc ratio
" 25, 7.25 13.49 13.60 65.66 ", ; 2.5 inc ratio
" 26, 7.17 12.55 13.54 66.75 ", ; 2.6 inc ratio
" 27, 6.89 12.26 13.34 67.51 ", ; 2.7 inc ratio
" 28, 6.93 11.97 12.74 68.36 ", ; 2.8 inc ratio
" 29, 6.52 11.03 12.90 69.55 ", ; 2.9 inc ratio
" 30, 5.96 10.06 13.19 70.78 ", ; 3 inc ratio
" 31, 5.21 9.27 13.49 72.04 ", ; 3.1 inc ratio
" 32, 5.26 8.78 13.01 72.96 ", ; 3.2 inc ratio
" 33, 4.97 8.30 12.75 73.98 ", ; 3.3 inc ratio
" 34, 4.69 7.64 12.62 75.05 ", ; 3.4 inc ratio
" 35, 4.41 6.96 12.49 76.14 ", ; 3.5 inc ratio
" 36, 3.95 6.27 12.50 77.28 ", ; 3.6 inc ratio
" 37, 3.66 5.56 12.40 78.38 " ; 3.7 inc ratio

;=====
; Initial Joint HH Size x Income Distribution from 2000 CTPP =
;=====
LOOKUP Name=I_SPCTA, LOOKUP[1] = 1,Result = 2,
Interpolate = N, FAIL=0,0,0,
Size_Inc Initial
Class Pct Pct of Size 'X' HHs in Inc group 'Y'
;-----
R=" 11, 45.51 ", ; 1 1
" 12, 29.18 ", ; 1 2
" 13, 18.47 ", ; 1 3
" 14, 6.84 ", ; 1 4
" 21, 18.77 ", ; 2 1
" 22, 22.26 ", ; 2 2
" 23, 29.81 ", ; 2 3
" 24, 29.16 ", ; 2 4
" 31, 16.61 ", ; 3 1
" 32, 20.66 ", ; 3 2
" 33, 31.27 ", ; 3 3
" 34, 31.46 ", ; 3 4
" 41, 13.32 ", ; 4 1
" 42, 19.65 ", ; 4 2
" 43, 32.53 ", ; 4 3
" 44, 34.50 " ; 4 4

;=====
; Final Size and Income adjustments by area type =
; Factors are Unused (set to 1.0) but available if needed =
;=====
LOOKUP Name=AreaSizFtr,
LOOKUP[1] = 1,Result = 2,
LOOKUP[2] = 1,Result = 3,
LOOKUP[3] = 1,Result = 4,
LOOKUP[4] = 1,Result = 5,
Interpolate = N, FAIL=0,0,0,
Area Size1 Size2 Size3 Size4
Type Factor Factor Factor Factor
;-----

```

```

R=" 1, 1.00 1.00 1.00 1.00 ",
" 2, 1.00 1.00 1.00 1.00 ",
" 3, 1.00 1.00 1.00 1.00 ",
" 4, 1.00 1.00 1.00 1.00 ",
" 5, 1.00 1.00 1.00 1.00 ",
" 6, 1.00 1.00 1.00 1.00 ",
" 7, 1.00 1.00 1.00 1.00 "

LOOKUP Name=AreaIncFtr,
LOOKUP[1] = 1,Result = 2,
LOOKUP[2] = 1,Result = 3,
LOOKUP[3] = 1,Result = 4,
LOOKUP[4] = 1,Result = 5,
Interpolate = N, FAIL=0,0,0,
Area Inc1 Inc2 Inc3 Inc4
Type Factor Factor Factor Factor
;-----
R=" 1, 1.00 1.00 1.00 1.00 ",
" 2, 1.00 1.00 1.00 1.00 ",
" 3, 1.00 1.00 1.00 1.00 ",
" 4, 1.00 1.00 1.00 1.00 ",
" 5, 1.00 1.00 1.00 1.00 ",
" 6, 1.00 1.00 1.00 1.00 ",
" 7, 1.00 1.00 1.00 1.00 "

;=====
; Coefficients for the Veh Avail Model - provided as variables instead of lookups
;=====
;; v1_constant= 0 v2_constant= 1.05719498 v3_constant =-2.70675604 v4_constant
=-6.03433686 Estimated Constants
;; v1_constant= 0 v2_constant= 0.4512 v3_constant =-3.1838 v4_constant
=-6.9323 Calibrated Constants/Try 1
;; v1_constant= 0 v2_constant= 0.5173 v3_constant =-3.1112 v4_constant
=-6.8805 Calibrated Constants/Try 2
;; v1_constant= 0 v2_constant= 0.5334 v3_constant =-3.0902 v4_constant
=-6.8599 Calibrated Constants/Try 3
;; v1_constant= 0 v2_constant= 0.5382 v3_constant =-3.0820 v4_constant
=-6.8508 Calibrated Constants/Try 4

;; Estimated Coefficients --updated by M. Martchouk 11/02/10
;; Calibrated constants updated by Milone 11/02/10
v1_constant= 0 v2_constant= 0.5382 v3_constant =-3.0820 v4_constant =-
6.8508
v1_idum1 = 0 v2_idum1 = 0.0 v3_idum1 = 0.0 v4_idum1 =
0.0
v1_idum2 = 0 v2_idum2 = 1.45353047 v3_idum2 = 1.84315742 v4_idum2 =
2.46187933
v1_idum3 = 0 v2_idum3 = 2.25891102 v3_idum3 = 3.42089498 v4_idum3 =
4.62339172
v1_idum4 = 0 v2_idum4 = 2.65576393 v3_idum4 = 3.91630481 v4_idum4 =
5.54022044
v1_hh = 0 v2_hh = 0.16933726 v3_hh = 1.3438729 v4_hh =
1.69095555
v1_TrnAcc = 0 v2_TrnAcc =-1.20E-06 v3_TrnAcc =-2.04E-06 v4_TrnAcc =-
2.37E-06
v1_Atype = 0 v2_Atype = 0.20915613 v3_Atype = 0.47716419 v4_Atype =
0.77921942
v1_DcDum = 0 v2_DcDum =-0.94482292 v3_DcDum =-1.39768896 v4_DcDum =-
1.52940323

```



## Appendix C Cube Voyager Scripts

```

=====
; End of Lookups- Now read the input files
=
=====
;
; read Zonal land use files into Z-File
;
ZDATI[1] = @ZNFIL_ IN1@ ;; variables in DBF file: TAZ, HH, HHPOP, JURCODE, HHINCIDX
; Zonal Area Type File
ZDATI[2] = @ZNFIL_ IN2@ ;; variables in DBF file: TAZ, ATYPE
; Zonal Transit Access. Files
ZDATI[3] = @ZNFIL_ INa1@ ; TAZ, emp45
ZDATI[4] = @ZNFIL_ INa2@ ; TAZ, emp45
ZDATI[5] = @ZNFIL_ INa3@ ; TAZ, emp45
ZDATI[6] = @ZNFIL_ INa4@ ; TAZ, emp45
; Jobs within 45 min by AM Transit (Metrorail), use the Maximum Accessibility
; of all the AM Metrorail related path options
TrnAcc = MAX(zi.3.emp45, zi.4.emp45, zi.5.emp45, zi.6.emp45)
;
; establish variables
;
HH = zi.1.HH[I]
HHPOP = zi.1.HHPOP[I]
IncRat = zi.1.HHINCIDX[I]
Atype = zi.2.ATYPE[I]
IF (I > @LastIzn@) Atype=6 ; temporarily assign externals
; so input value ('0') doesn't
violate array dimensions
;; TrnAcc = zi.3.TrnAcc[I]
IF( HH>HHPOP)
HH=HHPOP
ENDIF
HH_IP_Total = HH_IP_Total + HH ; Input HH Total (to check
O/P Total)
; Compute HH Size rounded to nearest 1/10th (K.Vaughn fix)
If (HH == 0)
AvHHSz = 1.0
Else
AvHHSz10ths = Round(HHPOP/HH * 10.0)
; compute Avg HH Size in tenths
AvHHSztrue = AvHHSz10ths/10.0
; compute Avg HH Size actual
AvHHSz = MIN(AvHHSztrue,3.9)
;
Endif
; Compute Juris. index 1-24 / compute DC dummy code for VA model
Jdx = zi.1.JURCODE + 1
IF (zi.1.JURCODE = 0)

```

```

DCDUM = 1
ELSE
DCDUM = 0
ENDIF
; Accumulate jurisdiction level & total land use values
;
;-----
;Begin Matrix Work Now ...
;-----
; Clear all initial/computed arrays, establish initial marginal controls
Loop sz = 1, @SzCl@
Loop in = 1, @InCl@
CSZINA[sz][in] = 0 ; initial matrix cell value
EndLoop
EndLoop
Loop IDX=1,@SzCl@
ISZA[IDX] = 0
CSZA[IDX] = 0
ISZA[IDX] = HH * (SZPCTA(IDX,AvHHSz)/100.0)
EndLOOP
Loop IDX=1,@InCl@
IInA[IDX] = 0
CInA[IDX] = 0
IInA[IDX] = HH * (INPCTA(IDX,IncRat)/100.0)
EndLOOP
; ** Debug 1 On **
; * if (I==1)
; * Print List = I(5),HHPOP(10),HH(10.0),Incrat(10.2),
AvHHSz(10.2),file=debug.txt
; * loop idx = 1,4
; * spct =SZPCTA(IDX,AvHHSz)
; * ipct =INPCTA(IDX,IncRat)
; * Print List = HH(10),
AvHHSz(10.2),Incrat(10.2),SPCT,IPCT,ISZA[IDX],IINA[IDX], file=debug1.txt
; * endloop
; * endif
; ** Debug 1 Off**
;
; Setup Initial HH Size by Income Matrix with PUMS seed Pcts
; and accumulate Size, Income marginals
Loop sz = 1, @SzCl@
Loop in = 1, @InCl@
IDX = sz * 10.0 + in ; 2-digit index, 1st=HHsize& 2nd=Inc.level
CSZINA[sz][in] = ISZA[sz] * (I_SPCTA(1,IDX)/100.00) ; initial matrix cell
value
CSZA[sz] = CSZA[sz] + CSZINA[sz][in] ; initial/'control' marginal
size total
CINA[in] = CINA[in] + CSZINA[sz][in] ; initial/'control' marginal
Inc total
; ** Debug 2 On **
; * if (I==1)
; * IF (sz <= 4 && in<=4)
; * print list = ' init matrix: inc: ', in(3),' hhs: ', sz(3),
cszina[idx](7.3) , file=debug2.txt
; * Endif

```

## Appendix C Cube Voyager Scripts

```

;* endif
;*
;*
;** Debug 2 Off**
EndLoop
EndLoop

; Initial matrix now established, now
; begin fratar process
;
;

LOOP FRAT= 1,3
  OddEve = FRAT%2 ; Modulo function to check Odd/Even
  iteration:0=even/nonzero=odd
  IF (OddEve != 0) ; if an odd iteration then adjust cols ...
  ;
  Loop in=1,@InCl@
    IF (CINA[in] == 0 )
      CINADJA[in] = 0
    ELSE
      CINADJA[in] = IINA[in] / CINA[in]
    ENDIF
  EndLoop

  Loop IDX=1,@SzCl@
    CSZA[IDX] = 0
  EndLOOP

  Loop IDX=1,@InCl@
    CINA[IDX] = 0
  EndLOOP

  Loop sz= 1,@SzCl@
    Loop in= 1,@InCl@

      CSZINA[sz][in] = CSZINA[sz][in] * CINADJA[in]
      CSZA[sz] = CSZA[sz] + CSZINA[sz][in] ; computed/current
    marginal size total
      CINA[in] = CINA[in] + CSZINA[sz][in] ; computed/current
    marginal Inc total
  EndLoop
  EndLoop
  ;
  ELSE
    ; begin computing of row (size) adjustments
    ; and apply adjustments to the matrix...
    ;
    Loop sz=1,@SzCl@
      IF (CSZA[sz] == 0 )
        CSZADJA[sz] = 0
      ELSE
        CSZADJA[sz] = ISZA[sz] / CSZA[sz]
      ENDIF
    EndLoop

    Loop IDX=1,@SzCl@
      CSZA[IDX] = 0
    EndLOOP

    Loop IDX=1,@InCl@
      CINA[IDX] = 0
    EndLOOP

    Loop sz= 1,@SzCl@

```

```

    Loop in= 1,@InCl@

      CSZINA[sz][in] = CSZINA[sz][in] * CSZADJA[sz]
      CSZA[sz] = CSZA[sz] + CSZINA[sz][in] ; computed/current
    marginal size total
      CINA[in] = CINA[in] + CSZINA[sz][in] ; computed/current
    marginal Inc total
  EndLoop
  EndLoop
  ENDIF
ENDLOOP

;
=====
; Apply final Size/Income adjustments (if desired) and then
; accumulate final Jurisdictional/ Regional marginals and totals
;
=====
    Loop sz= 1,@SzCl@
      Loop in= 1,@InCl@

        temp = CSZINA[sz][in] * AreaSizFtr(Sz,Atype) *
        AreaIncFtr(In,Atype) ; Apply Final Size/Income Adjustment
        CSZINA[sz][in] = temp ; and store
      back in CSZINA array
        RegSzInA[sz][in] = RegSzInA[sz][in] + CSZINA[sz][in]
        JurSzA[jdx][sz] = JurSzA[jdx][sz] + CSZINA[sz][in]
        JurInA[jdx][in] = JurInA[jdx][in] + CSZINA[sz][in]
        RegSzA[sz] = RegSzA[sz] + CSZINA[sz][in]
        RegInA[in] = RegInA[in] + CSZINA[sz][in]
        ArSzA[Atype][sz] = ArSzA[Atype][sz] + CSZINA[sz][in]
        ArInA[Atype][in] = ArInA[Atype][in] + CSZINA[sz][in]
        AreaA[Atype] = AreaA[Atype] + CSZINA[sz][in]
        JurA[Jdx] = JurA[Jdx] + CSZINA[sz][in]
        SITotal = SITotal + CSZINA[sz][in]
      EndLoop
    EndLoop

;
=====
; Summarize/Print HHs by size groups and HHs by Income groups for zonal checking
;
;
=====
    HH_Sz1 = CSZINA[1][1] + CSZINA[1][2] + CSZINA[1][3] + CSZINA[1][4]
    HH_Sz2 = CSZINA[2][1] + CSZINA[2][2] + CSZINA[2][3] + CSZINA[2][4]
    HH_Sz3 = CSZINA[3][1] + CSZINA[3][2] + CSZINA[3][3] + CSZINA[3][4]
    HH_Sz4 = CSZINA[4][1] + CSZINA[4][2] + CSZINA[4][3] + CSZINA[4][4]
;
    HH_In1 = CSZINA[1][1] + CSZINA[2][1] + CSZINA[3][1] + CSZINA[4][1]
    HH_In2 = CSZINA[1][2] + CSZINA[2][2] + CSZINA[3][2] + CSZINA[4][2]
    HH_In3 = CSZINA[1][3] + CSZINA[2][3] + CSZINA[3][3] + CSZINA[4][3]
    HH_In4 = CSZINA[1][4] + CSZINA[2][4] + CSZINA[3][4] + CSZINA[4][4]
;
;
    Print List=
    I(4),HH_Sz1@ofmt@,HH_Sz2@ofmt@,HH_Sz3@ofmt@,HH_Sz4@ofmt@,file=Est_Zonal_HH_Size.TXT
    Print List=
    I(4),HH_In1@ofmt@,HH_In2@ofmt@,HH_In3@ofmt@,HH_In4@ofmt@,file=Est_Zonal_HH_Inc.TXT
;
=====
; All Done with Size and Income computations - Now apply Veh. Availability Model

```

## Appendix C Cube Voyager Scripts

```

; Loop through size and income cell and further disggregate among veh.av. groups
;=====
Loop sz=1,@SzCl@
  Loop in=1,@InCl@

    P_VA[1] = 0
    P_VA[2] = 0
    P_VA[3] = 0
    P_VA[4] = 0
    IncDum1 = 0
    IncDum2 = 0
    IncDum3 = 0
    IncDum4 = 0
    If (in == 1) IncDum1 = 1
    If (in == 2) IncDum2 = 1
    If (in == 3) IncDum3 = 1
    If (in == 4) IncDum4 = 1

;compute VA utilities
u_1 = v1_constant +
      v1_idum1 * IncDum1 +
      v1_idum2 * IncDum2 +
      v1_idum3 * IncDum3 +
      v1_idum4 * IncDum4 +
      v1_hh * SZ +
      v1_TrnAcc * TrnAcc +
      v1_Atype * AType +
      v1_DcDum * DCDUM

u_2 = v2_constant +
      v2_idum1 * IncDum1 +
      v2_idum2 * IncDum2 +
      v2_idum3 * IncDum3 +
      v2_idum4 * IncDum4 +
      v2_hh * SZ +
      v2_TrnAcc * TrnAcc +
      v2_Atype * AType +
      v2_DcDum * DCDUM

u_3 = v3_constant +
      v3_idum1 * IncDum1 +
      v3_idum2 * IncDum2 +
      v3_idum3 * IncDum3 +
      v3_idum4 * IncDum4 +
      v3_hh * SZ +
      v3_TrnAcc * TrnAcc +
      v3_Atype * AType +
      v3_DcDum * DCDUM

u_4 = v4_constant +
      v4_idum1 * IncDum1 +
      v4_idum2 * IncDum2 +
      v4_idum3 * IncDum3 +
      v4_idum4 * IncDum4 +
      v4_hh * SZ +
      v4_TrnAcc * TrnAcc +
      v4_Atype * AType +
      v4_DcDum * DCDUM

;compute VA probabilities
P_VA[1] = exp(u_1) / (exp(u_1) + exp(u_2) + exp(u_3) +
exp(u_4))
P_VA[2] = exp(u_2) / (exp(u_1) + exp(u_2) + exp(u_3) +
exp(u_4))

```

```

P_VA[3] = exp(u_3) / (exp(u_1) + exp(u_2) + exp(u_3) +
exp(u_4))
P_VA[4] = exp(u_4) / (exp(u_1) + exp(u_2) + exp(u_3) +
exp(u_4))

; apply Veh Avail. probabilities
CSZINVAA[Sz][In][1] = CSZINA[Sz][In] * P_VA[1] ;

CSZINVAA[Sz][In][2] = CSZINA[Sz][In] * P_VA[2] ;

CSZINVAA[Sz][In][3] = CSZINA[Sz][In] * P_VA[3] ;

CSZINVAA[Sz][In][4] = CSZINA[Sz][In] * P_VA[4] ;

EndLoop
EndLoop

; accumulate HHs in Vehicle Available groups (0,1,2+) for current TAZ
; also accumulate regional totals for checking

HHw0Vehs = CSZINVAA[1][1][1] + CSZINVAA[1][2][1] +
CSZINVAA[1][3][1] + CSZINVAA[1][4][1] +
CSZINVAA[2][1][1] + CSZINVAA[2][2][1] +
CSZINVAA[2][3][1] + CSZINVAA[2][4][1] +
CSZINVAA[3][1][1] + CSZINVAA[3][2][1] +
CSZINVAA[3][3][1] + CSZINVAA[3][4][1] +
CSZINVAA[4][1][1] + CSZINVAA[4][2][1] +
CSZINVAA[4][3][1] + CSZINVAA[4][4][1]

HHw1Vehs = CSZINVAA[1][1][2] + CSZINVAA[1][2][2] +
CSZINVAA[1][3][2] + CSZINVAA[1][4][2] +
CSZINVAA[2][1][2] + CSZINVAA[2][2][2] +
CSZINVAA[2][3][2] + CSZINVAA[2][4][2] +
CSZINVAA[3][1][2] + CSZINVAA[3][2][2] +
CSZINVAA[3][3][2] + CSZINVAA[3][4][2] +
CSZINVAA[4][1][2] + CSZINVAA[4][2][2] +
CSZINVAA[4][3][2] + CSZINVAA[4][4][2]

HHw2Vehs = CSZINVAA[1][1][3] + CSZINVAA[1][2][3] +
CSZINVAA[1][3][3] + CSZINVAA[1][4][3] +
CSZINVAA[2][1][3] + CSZINVAA[2][2][3] +
CSZINVAA[2][3][3] + CSZINVAA[2][4][3] +
CSZINVAA[3][1][3] + CSZINVAA[3][2][3] +
CSZINVAA[3][3][3] + CSZINVAA[3][4][3] +
CSZINVAA[4][1][3] + CSZINVAA[4][2][3] +
CSZINVAA[4][3][3] + CSZINVAA[4][4][3]

HHw3Vehs = CSZINVAA[1][1][4] + CSZINVAA[1][2][4] +
CSZINVAA[1][3][4] + CSZINVAA[1][4][4] +
CSZINVAA[2][1][4] + CSZINVAA[2][2][4] +
CSZINVAA[2][3][4] + CSZINVAA[2][4][4] +
CSZINVAA[3][1][4] + CSZINVAA[3][2][4] +
CSZINVAA[3][3][4] + CSZINVAA[3][4][4] +
CSZINVAA[4][1][4] + CSZINVAA[4][2][4] +
CSZINVAA[4][3][4] + CSZINVAA[4][4][4]

HHw2PVehs = HHw2Vehs + HHw3Vehs

Tot_HHw0Vehs = Tot_HHw0Vehs + HHw0Vehs

```

## Appendix C Cube Voyager Scripts

```

Tot_HHw1Vehs = Tot_HHw1Vehs + HHw1Vehs
Tot_HHw2Vehs = Tot_HHw2Vehs + HHw2Vehs
Tot_HHw3Vehs = Tot_HHw3Vehs + HHw3Vehs

Tot_HHw2PVehs = Tot_HHw2PVehs + HHw2PVehs

;=====
; --Print out
; zonal Household file for Mode Choice Model HHs by 0 , 1, 2+ Groups
; and Household file for Mode Choice Model HHs by 0 , 1, 2, 3+ Groups
;=====

;; Print List= I(5),
;; HHw0Vehs(6),HHw1Vehs(6),HHw2PVehs(6),file=@ZNFIL_OU5@

Print List= I(4), HHw0Vehs@ofmt@, HHw1Vehs@ofmt@, HHw2Vehs@ofmt@,
HHw3Vehs@ofmt@,file=Est_Zonal_HH_VehAv.TXT

;=====
; The Calculations are complete for the current zone
; and let's accumulate Veh Av. related Jurisdictional/ Regional marginals and totals
;=====

Loop sz=1,@SzCl@
Loop in=1,@InCl@
Loop Va=1,@VaCl@
RegSzInVaA[Sz][In][Va] = RegSzInVaA[Sz][In][Va] +
CSZINVAA[Sz][In][Va]
JurVaA[Jdx][Va] = JurVaA[Jdx][Va] +
CSZINVAA[Sz][In][Va]
ArVaA[Atype][va] = ArVaA[Atype][va] +
CSZINVAA[Sz][In][Va]
RegVaA[VA] = RegVaA[VA] +
CSZINVAA[Sz][In][Va]
RegVaSzA[va][sz] = RegVaSzA[va][sz] +
CSZINVAA[Sz][In][Va]
RegVaInA[va][in] = RegVaInA[va][in] +
CSZINVAA[Sz][In][Va]
SIVTotal = SIVTotal +
CSZINVAA[Sz][In][Va]
EndLoop
EndLoop
EndLoop

;=====
; Now We're at the end of the Loop
; --Print out input files to Trip Generation
; 4 income based files written in text form TAZ, HH by size&VehAv
siv1,siv2,...,s4v4
;=====

;Income 1 file with HHs by Size and VehAv:
Print List= I(4),
CSZINVAA[1][1][1]@ofmt@, CSZINVAA[1][1][2]@ofmt@,
CSZINVAA[1][1][3]@ofmt@, CSZINVAA[1][1][4]@ofmt@,
CSZINVAA[2][1][1]@ofmt@, CSZINVAA[2][1][2]@ofmt@,
CSZINVAA[2][1][3]@ofmt@, CSZINVAA[2][1][4]@ofmt@,

```

```

CSZINVAA[3][1][1]@ofmt@, CSZINVAA[3][1][2]@ofmt@,
CSZINVAA[3][1][3]@ofmt@, CSZINVAA[3][1][4]@ofmt@,
CSZINVAA[4][1][1]@ofmt@, CSZINVAA[4][1][2]@ofmt@,
CSZINVAA[4][1][3]@ofmt@, CSZINVAA[4][1][4]@ofmt@,file=@ZNFIL_OU1@

;Income 2 file with HHs by Size and VehAv:
Print List= I(4),
CSZINVAA[1][2][1]@ofmt@, CSZINVAA[1][2][2]@ofmt@,
CSZINVAA[1][2][3]@ofmt@, CSZINVAA[1][2][4]@ofmt@,
CSZINVAA[2][2][1]@ofmt@, CSZINVAA[2][2][2]@ofmt@,
CSZINVAA[2][2][3]@ofmt@, CSZINVAA[2][2][4]@ofmt@,
CSZINVAA[3][2][1]@ofmt@, CSZINVAA[3][2][2]@ofmt@,
CSZINVAA[3][2][3]@ofmt@, CSZINVAA[3][2][4]@ofmt@,
CSZINVAA[4][2][1]@ofmt@, CSZINVAA[4][2][2]@ofmt@,
CSZINVAA[4][2][3]@ofmt@, CSZINVAA[4][2][4]@ofmt@,file=@ZNFIL_OU2@

;Income 3 file with HHs by Size and VehAv:
Print List= I(4),
CSZINVAA[1][3][1]@ofmt@, CSZINVAA[1][3][2]@ofmt@,
CSZINVAA[1][3][3]@ofmt@, CSZINVAA[1][3][4]@ofmt@,
CSZINVAA[2][3][1]@ofmt@, CSZINVAA[2][3][2]@ofmt@,
CSZINVAA[2][3][3]@ofmt@, CSZINVAA[2][3][4]@ofmt@,
CSZINVAA[3][3][1]@ofmt@, CSZINVAA[3][3][2]@ofmt@,
CSZINVAA[3][3][3]@ofmt@, CSZINVAA[3][3][4]@ofmt@,
CSZINVAA[4][3][1]@ofmt@, CSZINVAA[4][3][2]@ofmt@,
CSZINVAA[4][3][3]@ofmt@, CSZINVAA[4][3][4]@ofmt@,file=@ZNFIL_OU3@

;Income 4 file with HHs by Size and VehAv:
Print List= I(4),
CSZINVAA[1][4][1]@ofmt@, CSZINVAA[1][4][2]@ofmt@,
CSZINVAA[1][4][3]@ofmt@, CSZINVAA[1][4][4]@ofmt@,
CSZINVAA[2][4][1]@ofmt@, CSZINVAA[2][4][2]@ofmt@,
CSZINVAA[2][4][3]@ofmt@, CSZINVAA[2][4][4]@ofmt@,
CSZINVAA[3][4][1]@ofmt@, CSZINVAA[3][4][2]@ofmt@,
CSZINVAA[3][4][3]@ofmt@, CSZINVAA[3][4][4]@ofmt@,
CSZINVAA[4][4][1]@ofmt@, CSZINVAA[4][4][2]@ofmt@,
CSZINVAA[4][4][3]@ofmt@, CSZINVAA[4][4][4]@ofmt@,file=@ZNFIL_OU4@

;;
;; write out dbf files for HHs by cross-class
;; Define output variables

FILEO RECO[1] = "@ZonalCCHHs@",fields =
I(5),
HHsISV111@ofmt@, HHsISV112@ofmt@, HHsISV113@ofmt@, HHsISV114@ofmt@,
HHsISV211@ofmt@, HHsISV212@ofmt@, HHsISV213@ofmt@, HHsISV214@ofmt@,
HHsISV311@ofmt@, HHsISV312@ofmt@, HHsISV313@ofmt@, HHsISV314@ofmt@,
HHsISV411@ofmt@, HHsISV412@ofmt@, HHsISV413@ofmt@, HHsISV414@ofmt@,
HHsISV121@ofmt@, HHsISV122@ofmt@, HHsISV123@ofmt@, HHsISV124@ofmt@,
HHsISV221@ofmt@, HHsISV222@ofmt@, HHsISV223@ofmt@, HHsISV224@ofmt@,
HHsISV321@ofmt@, HHsISV322@ofmt@, HHsISV323@ofmt@, HHsISV324@ofmt@,
HHsISV421@ofmt@, HHsISV422@ofmt@, HHsISV423@ofmt@, HHsISV424@ofmt@,
HHsISV131@ofmt@, HHsISV132@ofmt@, HHsISV133@ofmt@, HHsISV134@ofmt@,
HHsISV231@ofmt@, HHsISV232@ofmt@, HHsISV233@ofmt@, HHsISV234@ofmt@,
HHsISV331@ofmt@, HHsISV332@ofmt@, HHsISV333@ofmt@, HHsISV334@ofmt@,
HHsISV431@ofmt@, HHsISV432@ofmt@, HHsISV433@ofmt@, HHsISV434@ofmt@,
HHsISV141@ofmt@, HHsISV142@ofmt@, HHsISV143@ofmt@, HHsISV144@ofmt@,
HHsISV241@ofmt@, HHsISV242@ofmt@, HHsISV243@ofmt@, HHsISV244@ofmt@,
HHsISV341@ofmt@, HHsISV342@ofmt@, HHsISV343@ofmt@, HHsISV344@ofmt@,
HHsISV441@ofmt@, HHsISV442@ofmt@, HHsISV443@ofmt@, HHsISV444@ofmt@

;;
;; write out dbf files for HHs by cross class (Corrected 10/30/10)
;;
ro.HHsISV111 = CSZINVAA[1][1][1] ro.HHsISV112 = CSZINVAA[1][1][2] ro.HHsISV113 =
CSZINVAA[1][1][3] ro.HHsISV114 = CSZINVAA[1][1][4]

```

## Appendix C Cube Voyager Scripts

```

ro.HHsISV211 = CSZINVAA[1][2][1]   ro.HHsISV212 = CSZINVAA[1][2][2]   ro.HHsISV213 =
CSZINVAA[1][2][3]   ro.HHsISV214 = CSZINVAA[1][2][4]
ro.HHsISV311 = CSZINVAA[1][3][1]   ro.HHsISV312 = CSZINVAA[1][3][2]   ro.HHsISV313 =
CSZINVAA[1][3][3]   ro.HHsISV314 = CSZINVAA[1][3][4]
ro.HHsISV411 = CSZINVAA[1][4][1]   ro.HHsISV412 = CSZINVAA[1][4][2]   ro.HHsISV413 =
CSZINVAA[1][4][3]   ro.HHsISV414 = CSZINVAA[1][4][4]

ro.HHsISV121 = CSZINVAA[2][1][1]   ro.HHsISV122 = CSZINVAA[2][1][2]   ro.HHsISV123 =
CSZINVAA[2][1][3]   ro.HHsISV124 = CSZINVAA[2][1][4]
ro.HHsISV221 = CSZINVAA[2][2][1]   ro.HHsISV222 = CSZINVAA[2][2][2]   ro.HHsISV223 =
CSZINVAA[2][2][3]   ro.HHsISV224 = CSZINVAA[2][2][4]
ro.HHsISV321 = CSZINVAA[2][3][1]   ro.HHsISV322 = CSZINVAA[2][3][2]   ro.HHsISV323 =
CSZINVAA[2][3][3]   ro.HHsISV324 = CSZINVAA[2][3][4]
ro.HHsISV421 = CSZINVAA[2][4][1]   ro.HHsISV422 = CSZINVAA[2][4][2]   ro.HHsISV423 =
CSZINVAA[2][4][3]   ro.HHsISV424 = CSZINVAA[2][4][4]

ro.HHsISV131 = CSZINVAA[3][1][1]   ro.HHsISV132 = CSZINVAA[3][1][2]   ro.HHsISV133 =
CSZINVAA[3][1][3]   ro.HHsISV134 = CSZINVAA[3][1][4]
ro.HHsISV231 = CSZINVAA[3][2][1]   ro.HHsISV232 = CSZINVAA[3][2][2]   ro.HHsISV233 =
CSZINVAA[3][2][3]   ro.HHsISV234 = CSZINVAA[3][2][4]
ro.HHsISV331 = CSZINVAA[3][3][1]   ro.HHsISV332 = CSZINVAA[3][3][2]   ro.HHsISV333 =
CSZINVAA[3][3][3]   ro.HHsISV334 = CSZINVAA[3][3][4]
ro.HHsISV431 = CSZINVAA[3][4][1]   ro.HHsISV432 = CSZINVAA[3][4][2]   ro.HHsISV433 =
CSZINVAA[3][4][3]   ro.HHsISV434 = CSZINVAA[3][4][4]

ro.HHsISV141 = CSZINVAA[4][1][1]   ro.HHsISV142 = CSZINVAA[4][1][2]   ro.HHsISV143 =
CSZINVAA[4][1][3]   ro.HHsISV144 = CSZINVAA[4][1][4]
ro.HHsISV241 = CSZINVAA[4][2][1]   ro.HHsISV242 = CSZINVAA[4][2][2]   ro.HHsISV243 =
CSZINVAA[4][2][3]   ro.HHsISV244 = CSZINVAA[4][2][4]
ro.HHsISV341 = CSZINVAA[4][3][1]   ro.HHsISV342 = CSZINVAA[4][3][2]   ro.HHsISV343 =
CSZINVAA[4][3][3]   ro.HHsISV344 = CSZINVAA[4][3][4]
ro.HHsISV441 = CSZINVAA[4][4][1]   ro.HHsISV442 = CSZINVAA[4][4][2]   ro.HHsISV443 =
CSZINVAA[4][4][3]   ro.HHsISV444 = CSZINVAA[4][4][4]
WRITE RECO=1

;
=====
; Finally accumulate Size, Inc, Veh.Av variables by area type for reporting
=====

If (I <= @LastIZN@)

HH_S1 = HH_S1 + CSZINA[1][1] + CSZINA[1][2] + CSZINA[1][3] + CSZINA[1][4]
HH_S2 = HH_S2 + CSZINA[2][1] + CSZINA[2][2] + CSZINA[2][3] + CSZINA[2][4]
HH_S3 = HH_S3 + CSZINA[3][1] + CSZINA[3][2] + CSZINA[3][3] + CSZINA[3][4]
HH_S4 = HH_S4 + CSZINA[4][1] + CSZINA[4][2] + CSZINA[4][3] + CSZINA[4][4]

HH_I1 = HH_I1 + CSZINA[1][1] + CSZINA[2][1] + CSZINA[3][1] + CSZINA[4][1]
HH_I2 = HH_I2 + CSZINA[1][2] + CSZINA[2][2] + CSZINA[3][2] + CSZINA[4][2]
HH_I3 = HH_I3 + CSZINA[1][3] + CSZINA[2][3] + CSZINA[3][3] + CSZINA[4][3]
HH_I4 = HH_I4 + CSZINA[1][4] + CSZINA[2][4] + CSZINA[3][4] + CSZINA[4][4]

HH_V1 = HH_V1 + HHw0Vehs
HH_V2 = HH_V2 + HHw1Vehs
HH_V3 = HH_V3 + HHw2Vehs
HH_V4 = HH_V4 + HHw3Vehs

HH_S = HH_S + CSZINA[1][1] + CSZINA[1][2] + CSZINA[1][3] + CSZINA[1][4] +
CSZINA[2][1] + CSZINA[2][2] + CSZINA[2][3] + CSZINA[2][4] +
CSZINA[3][1] + CSZINA[3][2] + CSZINA[3][3] + CSZINA[3][4] +
CSZINA[4][1] + CSZINA[4][2] + CSZINA[4][3] + CSZINA[4][4]

HH_I = HH_I + CSZINA[1][1] + CSZINA[2][1] + CSZINA[3][1] + CSZINA[4][1] +
CSZINA[1][2] + CSZINA[2][2] + CSZINA[3][2] + CSZINA[4][2] +

```

```

CSZINA[1][3] + CSZINA[2][3] + CSZINA[3][3] + CSZINA[4][3] +
CSZINA[1][4] + CSZINA[2][4] + CSZINA[3][4] + CSZINA[4][4]

HH_V = HH_V + HHw0Vehs +
HHw1Vehs +
HHw2Vehs +
HHw3Vehs

Endif

;
=====
; If we're at the last Zone, it's time to printout the listings and we're done.
=====

IF (I=@ZONESIZE@)

Print LIST= ' Demographic Model Report ', file=@Rept@ ;
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@

Print LIST= ' ',file=@Rept@
Print LIST= ' Untransformed - Household Total from the Input File:',
HH_IP_Total(12.0),file=@Rept@ ;
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
PRINT LIST = ' Regional Households by Size and Income Summary ',file=@Rept@
PRINT LIST = ' Size Inc_1 Inc_2 Inc_3 Inc_4 Total
',file=@Rept@
PRINT LIST = '
-- ',file=@Rept@

Print form=12.csv LIST= ' 1
',RegSzInA[1][1],RegSzInA[1][2],RegSzInA[1][3],RegSzInA[1][4],RegSzA[1],file=@Rept@
;
Print form=12.csv LIST= ' 2
',RegSzInA[2][1],RegSzInA[2][2],RegSzInA[2][3],RegSzInA[2][4],RegSzA[2],file=@Rept@
;
Print form=12.csv LIST= ' 3
',RegSzInA[3][1],RegSzInA[3][2],RegSzInA[3][3],RegSzInA[3][4],RegSzA[3],file=@Rept@
;
Print form=12.csv LIST= ' 4+
',RegSzInA[4][1],RegSzInA[4][2],RegSzInA[4][3],RegSzInA[4][4],RegSzA[4],file=@Rept@
;
Print LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegInA[1], RegInA[2], RegInA[3],
RegInA[4], SITotal,file=@Rept@ ;
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@

;
=====

PRINT LIST = ' Jurisdictional Households by Size ',file=@Rept@
PRINT LIST = ' Juris. Size_1 Size_2 Size_3 Size_4 Total
',file=@Rept@
PRINT LIST = '
-- ',file=@Rept@

Print form=12.csv LIST= ' 0_DC
',JurSzA[01][1],JurSzA[01][2],JurSzA[01][3],JurSzA[01][4],JurA[01],file=@Rept@ ;
Print form=12.csv LIST= ' 1_Mtg
',JurSzA[02][1],JurSzA[02][2],JurSzA[02][3],JurSzA[02][4],JurA[02],file=@Rept@ ;

```

## Appendix C Cube Voyager Scripts

```

Print form=12.csv LIST= ' 2 PG
',JurSzA[03][1],JurSzA[03][2],JurSzA[03][3],JurSzA[03][4],JurA[03],file=@Rept@ ;
Print form=12.csv LIST= ' 3 Arl
',JurSzA[04][1],JurSzA[04][2],JurSzA[04][3],JurSzA[04][4],JurA[04],file=@Rept@ ;
Print form=12.csv LIST= ' 4 Alx
',JurSzA[05][1],JurSzA[05][2],JurSzA[05][3],JurSzA[05][4],JurA[05],file=@Rept@ ;
Print form=12.csv LIST= ' 5 Ffx
',JurSzA[06][1],JurSzA[06][2],JurSzA[06][3],JurSzA[06][4],JurA[06],file=@Rept@ ;
Print form=12.csv LIST= ' 6 Ldn
',JurSzA[07][1],JurSzA[07][2],JurSzA[07][3],JurSzA[07][4],JurA[07],file=@Rept@ ;
Print form=12.csv LIST= ' 7 PW
',JurSzA[08][1],JurSzA[08][2],JurSzA[08][3],JurSzA[08][4],JurA[08],file=@Rept@ ;
Print form=12.csv LIST= ' 8 -
',JurSzA[09][1],JurSzA[09][2],JurSzA[09][3],JurSzA[09][4],JurA[09],file=@Rept@ ;
Print form=12.csv LIST= ' 9 Frd
',JurSzA[10][1],JurSzA[10][2],JurSzA[10][3],JurSzA[10][4],JurA[10],file=@Rept@ ;
Print form=12.csv LIST= ' 10 How
',JurSzA[11][1],JurSzA[11][2],JurSzA[11][3],JurSzA[11][4],JurA[11],file=@Rept@ ;
Print form=12.csv LIST= ' 11 AA
',JurSzA[12][1],JurSzA[12][2],JurSzA[12][3],JurSzA[12][4],JurA[12],file=@Rept@ ;
Print form=12.csv LIST= ' 12 Chs
',JurSzA[13][1],JurSzA[13][2],JurSzA[13][3],JurSzA[13][4],JurA[13],file=@Rept@ ;
Print form=12.csv LIST= ' 13 -
',JurSzA[14][1],JurSzA[14][2],JurSzA[14][3],JurSzA[14][4],JurA[14],file=@Rept@ ;
Print form=12.csv LIST= ' 14 Car
',JurSzA[15][1],JurSzA[15][2],JurSzA[15][3],JurSzA[15][4],JurA[15],file=@Rept@ ;
Print form=12.csv LIST= ' 15 Cal
',JurSzA[16][1],JurSzA[16][2],JurSzA[16][3],JurSzA[16][4],JurA[16],file=@Rept@ ;
Print form=12.csv LIST= ' 16 SM
',JurSzA[17][1],JurSzA[17][2],JurSzA[17][3],JurSzA[17][4],JurA[17],file=@Rept@ ;
Print form=12.csv LIST= '
17_KGeo',JurSzA[18][1],JurSzA[18][2],JurSzA[18][3],JurSzA[18][4],JurA[18],file=@Rept
@ ;
Print form=12.csv LIST= ' 18 Fbg
',JurSzA[19][1],JurSzA[19][2],JurSzA[19][3],JurSzA[19][4],JurA[19],file=@Rept@ ;
Print form=12.csv LIST= ' 19 Sta
',JurSzA[20][1],JurSzA[20][2],JurSzA[20][3],JurSzA[20][4],JurA[20],file=@Rept@ ;
Print form=12.csv LIST= ' 20 Spt
',JurSzA[21][1],JurSzA[21][2],JurSzA[21][3],JurSzA[21][4],JurA[21],file=@Rept@ ;
Print form=12.csv LIST= ' 21 Fau
',JurSzA[22][1],JurSzA[22][2],JurSzA[22][3],JurSzA[22][4],JurA[22],file=@Rept@ ;
Print form=12.csv LIST= ' 22 Clk
',JurSzA[23][1],JurSzA[23][2],JurSzA[23][3],JurSzA[23][4],JurA[23],file=@Rept@ ;
Print form=12.csv LIST= ' 23 Jef
',JurSzA[24][1],JurSzA[24][2],JurSzA[24][3],JurSzA[24][4],JurA[24],file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegSzA[1], RegSzA[2], RegSzA[3],
RegSzA[4], SITotal,file=@Rept@ ;
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
;
=====
PRINT LIST= ' Jurisdictional Households by Income ',file=@Rept@
PRINT LIST= ' Juris. Inc_1 Inc_2 Inc_3 Inc_4 Total
',file=@Rept@
PRINT LIST= ' -----
-- ',file=@Rept@

Print form=12.csv LIST= ' 0 DC
',JurInA[01][1],JurInA[01][2],JurInA[01][3],JurInA[01][4],JurA[01],file=@Rept@ ;
Print form=12.csv LIST= ' 1 Mtg
',JurInA[02][1],JurInA[02][2],JurInA[02][3],JurInA[02][4],JurA[02],file=@Rept@ ;
Print form=12.csv LIST= ' 2 PG
',JurInA[03][1],JurInA[03][2],JurInA[03][3],JurInA[03][4],JurA[03],file=@Rept@ ;
Print form=12.csv LIST= ' 3 Arl
',JurInA[04][1],JurInA[04][2],JurInA[04][3],JurInA[04][4],JurA[04],file=@Rept@ ;

```

```

Print form=12.csv LIST= ' 4 Alx
',JurInA[05][1],JurInA[05][2],JurInA[05][3],JurInA[05][4],JurA[05],file=@Rept@ ;
Print form=12.csv LIST= ' 5 Ffx
',JurInA[06][1],JurInA[06][2],JurInA[06][3],JurInA[06][4],JurA[06],file=@Rept@ ;
Print form=12.csv LIST= ' 6 Ldn
',JurInA[07][1],JurInA[07][2],JurInA[07][3],JurInA[07][4],JurA[07],file=@Rept@ ;
Print form=12.csv LIST= ' 7 PW
',JurInA[08][1],JurInA[08][2],JurInA[08][3],JurInA[08][4],JurA[08],file=@Rept@ ;
Print form=12.csv LIST= ' 8 -
',JurInA[09][1],JurInA[09][2],JurInA[09][3],JurInA[09][4],JurA[09],file=@Rept@ ;
Print form=12.csv LIST= ' 9 Frd
',JurInA[10][1],JurInA[10][2],JurInA[10][3],JurInA[10][4],JurA[10],file=@Rept@ ;
Print form=12.csv LIST= ' 10 How
',JurInA[11][1],JurInA[11][2],JurInA[11][3],JurInA[11][4],JurA[11],file=@Rept@ ;
Print form=12.csv LIST= ' 11 AA
',JurInA[12][1],JurInA[12][2],JurInA[12][3],JurInA[12][4],JurA[12],file=@Rept@ ;
Print form=12.csv LIST= ' 12 Chs
',JurInA[13][1],JurInA[13][2],JurInA[13][3],JurInA[13][4],JurA[13],file=@Rept@ ;
Print form=12.csv LIST= ' 13 -
',JurInA[14][1],JurInA[14][2],JurInA[14][3],JurInA[14][4],JurA[14],file=@Rept@ ;
Print form=12.csv LIST= ' 14 Car
',JurInA[15][1],JurInA[15][2],JurInA[15][3],JurInA[15][4],JurA[15],file=@Rept@ ;
Print form=12.csv LIST= ' 15 Cal
',JurInA[16][1],JurInA[16][2],JurInA[16][3],JurInA[16][4],JurA[16],file=@Rept@ ;
Print form=12.csv LIST= ' 16 SM
',JurInA[17][1],JurInA[17][2],JurInA[17][3],JurInA[17][4],JurA[17],file=@Rept@ ;
Print form=12.csv LIST= '
17_KGeo',JurInA[18][1],JurInA[18][2],JurInA[18][3],JurInA[18][4],JurA[18],file=@Rept
@ ;
Print form=12.csv LIST= ' 18 Fbg
',JurInA[19][1],JurInA[19][2],JurInA[19][3],JurInA[19][4],JurA[19],file=@Rept@ ;
Print form=12.csv LIST= ' 19 Sta
',JurInA[20][1],JurInA[20][2],JurInA[20][3],JurInA[20][4],JurA[20],file=@Rept@ ;
Print form=12.csv LIST= ' 20 Spt
',JurInA[21][1],JurInA[21][2],JurInA[21][3],JurInA[21][4],JurA[21],file=@Rept@ ;
Print form=12.csv LIST= ' 21 Fau
',JurInA[22][1],JurInA[22][2],JurInA[22][3],JurInA[22][4],JurA[22],file=@Rept@ ;
Print form=12.csv LIST= ' 22 Clk
',JurInA[23][1],JurInA[23][2],JurInA[23][3],JurInA[23][4],JurA[23],file=@Rept@ ;
Print form=12.csv LIST= ' 23 Jef
',JurInA[24][1],JurInA[24][2],JurInA[24][3],JurInA[24][4],JurA[24],file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegInA[1], RegInA[2], RegInA[3],
RegInA[4], SITotal,file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
;
=====
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
PRINT LIST= ' Regional Households by Vehicles Available and Size Summary
',file=@Rept@
PRINT LIST= ' VeAv Size_1 Size_2 Size_3 Size_4 Total
',file=@Rept@
PRINT LIST= ' -----
-- ',file=@Rept@

Print form=12.csv LIST= ' 1
',RegVaSzA[1][1],RegVaSzA[1][2],RegVaSzA[1][3],RegVaSzA[1][4],RegVaA[1],file=@Rept@
;
Print form=12.csv LIST= ' 2
',RegVaSzA[2][1],RegVaSzA[2][2],RegVaSzA[2][3],RegVaSzA[2][4],RegVaA[2],file=@Rept@
;

```

## Appendix C Cube Voyager Scripts

```

Print form=12.csv LIST= ' 3
',RegVaSzA[3][1],RegVaSzA[3][2],RegVaSzA[3][3],RegVaSzA[3][4],RegVaA[3],file=@Rept@
;
Print form=12.csv LIST= ' 4+
',RegVaSzA[4][1],RegVaSzA[4][2],RegVaSzA[4][3],RegVaSzA[4][4],RegVaA[4],file=@Rept@
;
Print LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegSzA[1], RegSzA[2], RegSzA[3],
RegSzA[4], SITotal,file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@

Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
PRINT LIST= ' Regional Households by Vehicles Available Groups 1, 2, 3&4 ', '\n',
' Hhs w/ 0 Vehs: ', Tot_HHw0Vehs(12.0), '\n',
' Hhs w/ 1 Vehs: ', Tot_HHw1Vehs(12.0), '\n',
' Hhs w/ 2+Vehs: ', Tot_HHw2PVehs(12.0), '\n', file=@Rept@
;
=====
Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@
PRINT LIST= ' Regional Households by Vehicles Available and Income Summary
',file=@Rept@
PRINT LIST= ' VeAv Inc_1 Inc_2 Inc_3 Inc_4 Total
',file=@Rept@
PRINT LIST= ' -----
-- ',file=@Rept@

Print form=12.csv LIST= ' 1
',RegVaInA[1][1],RegVaInA[1][2],RegVaInA[1][3],RegVaInA[1][4],RegVaA[1],file=@Rept@
;
Print form=12.csv LIST= ' 2
',RegVaInA[2][1],RegVaInA[2][2],RegVaInA[2][3],RegVaInA[2][4],RegVaA[2],file=@Rept@
;
Print form=12.csv LIST= ' 3
',RegVaInA[3][1],RegVaInA[3][2],RegVaInA[3][3],RegVaInA[3][4],RegVaA[3],file=@Rept@
;
Print form=12.csv LIST= ' 4+
',RegVaInA[4][1],RegVaInA[4][2],RegVaInA[4][3],RegVaInA[4][4],RegVaA[4],file=@Rept@
;
Print form=12.csv LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegInA[1], RegInA[2], RegInA[3],
RegInA[4], SITotal,file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@

;
=====
PRINT LIST= ' Jurisdictional Households by Vehicles Available ',file=@Rept@
PRINT LIST= ' Juris. Veh_0 Veh_1 Veh_2 Veh_3+ Total
',file=@Rept@
PRINT LIST= ' -----
-- ',file=@Rept@

Print form=12.csv LIST= ' 0_DC
',JurVaA[01][1],JurVaA[01][2],JurVaA[01][3],JurVaA[01][4],JurA[01],file=@Rept@ ;
Print form=12.csv LIST= ' 1_Mtg
',JurVaA[02][1],JurVaA[02][2],JurVaA[02][3],JurVaA[02][4],JurA[02],file=@Rept@ ;
Print form=12.csv LIST= ' 2_PG
',JurVaA[03][1],JurVaA[03][2],JurVaA[03][3],JurVaA[03][4],JurA[03],file=@Rept@ ;
Print form=12.csv LIST= ' 3_Arl
',JurVaA[04][1],JurVaA[04][2],JurVaA[04][3],JurVaA[04][4],JurA[04],file=@Rept@ ;

```

```

Print form=12.csv LIST= ' 4 Alx
',JurVaA[05][1],JurVaA[05][2],JurVaA[05][3],JurVaA[05][4],JurA[05],file=@Rept@ ;
Print form=12.csv LIST= ' 5 Ffx
',JurVaA[06][1],JurVaA[06][2],JurVaA[06][3],JurVaA[06][4],JurA[06],file=@Rept@ ;
Print form=12.csv LIST= ' 6 Ldn
',JurVaA[07][1],JurVaA[07][2],JurVaA[07][3],JurVaA[07][4],JurA[07],file=@Rept@ ;
Print form=12.csv LIST= ' 7_PW
',JurVaA[08][1],JurVaA[08][2],JurVaA[08][3],JurVaA[08][4],JurA[08],file=@Rept@ ;
Print form=12.csv LIST= ' 8_
',JurVaA[09][1],JurVaA[09][2],JurVaA[09][3],JurVaA[09][4],JurA[09],file=@Rept@ ;
Print form=12.csv LIST= ' 9 Frd
',JurVaA[10][1],JurVaA[10][2],JurVaA[10][3],JurVaA[10][4],JurA[10],file=@Rept@ ;
Print form=12.csv LIST= ' 10_How
',JurVaA[11][1],JurVaA[11][2],JurVaA[11][3],JurVaA[11][4],JurA[11],file=@Rept@ ;
Print form=12.csv LIST= ' 11_AA
',JurVaA[12][1],JurVaA[12][2],JurVaA[12][3],JurVaA[12][4],JurA[12],file=@Rept@ ;
Print form=12.csv LIST= ' 12_ChS
',JurVaA[13][1],JurVaA[13][2],JurVaA[13][3],JurVaA[13][4],JurA[13],file=@Rept@ ;
Print form=12.csv LIST= ' 13_
',JurVaA[14][1],JurVaA[14][2],JurVaA[14][3],JurVaA[14][4],JurA[14],file=@Rept@ ;
Print form=12.csv LIST= ' 14_Car
',JurVaA[15][1],JurVaA[15][2],JurVaA[15][3],JurVaA[15][4],JurA[15],file=@Rept@ ;
Print form=12.csv LIST= ' 15_Cal
',JurVaA[16][1],JurVaA[16][2],JurVaA[16][3],JurVaA[16][4],JurA[16],file=@Rept@ ;
Print form=12.csv LIST= ' 16_SM
',JurVaA[17][1],JurVaA[17][2],JurVaA[17][3],JurVaA[17][4],JurA[17],file=@Rept@ ;
Print form=12.csv LIST= '
17_KGeo',JurVaA[18][1],JurVaA[18][2],JurVaA[18][3],JurVaA[18][4],JurA[18],file=@Rept@
;
Print form=12.csv LIST= ' 18_Fbg
',JurVaA[19][1],JurVaA[19][2],JurVaA[19][3],JurVaA[19][4],JurA[19],file=@Rept@ ;
Print form=12.csv LIST= ' 19_Sta
',JurVaA[20][1],JurVaA[20][2],JurVaA[20][3],JurVaA[20][4],JurA[20],file=@Rept@ ;
Print form=12.csv LIST= ' 20_Spt
',JurVaA[21][1],JurVaA[21][2],JurVaA[21][3],JurVaA[21][4],JurA[21],file=@Rept@ ;
Print form=12.csv LIST= ' 21_Fau
',JurVaA[22][1],JurVaA[22][2],JurVaA[22][3],JurVaA[22][4],JurA[22],file=@Rept@ ;
Print form=12.csv LIST= ' 22_Clk
',JurVaA[23][1],JurVaA[23][2],JurVaA[23][3],JurVaA[23][4],JurA[23],file=@Rept@ ;
Print form=12.csv LIST= ' 23_Jef
',JurVaA[24][1],JurVaA[24][2],JurVaA[24][3],JurVaA[24][4],JurA[24],file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print form=12.csv LIST= ' Total ',RegVaA[1], RegVaA[2], RegVaA[3],
RegVaA[4], SITotal,file=@Rept@ ;

Print LIST= ' ',file=@Rept@
Print LIST= ' ',file=@Rept@

PRINT LIST= ' Estimated Households By Size Level by Area Type ', '\n',
file=@Rept@

PRINT LIST= ' Area Tp Hhs_Size1 Hhs_Size2 Hhs_Size3
Hhs_Size4 Total ',file=@Rept@
PRINT LIST= ' -----
-- ',file=@Rept@

Print form=12.csv LIST= ' 1
ArSzA[1][1], ArSzA[1][2], ArSzA[1][3],
ArSzA[1][4], AreaA[1],file=@Rept@ ;
Print form=12.csv LIST= ' 2
',ArSzA[2][1], ArSzA[2][2], ArSzA[2][3],
ArSzA[2][4], AreaA[2],file=@Rept@ ;
Print form=12.csv LIST= ' 3
',ArSzA[3][1], ArSzA[3][2], ArSzA[3][3],
ArSzA[3][4], AreaA[3],file=@Rept@ ;
Print form=12.csv LIST= ' 4
',ArSzA[4][1], ArSzA[4][2], ArSzA[4][3],
ArSzA[4][4], AreaA[4],file=@Rept@ ;
Print form=12.csv LIST= ' 5
',ArSzA[5][1], ArSzA[5][2], ArSzA[5][3],
ArSzA[5][4], AreaA[5],file=@Rept@ ;

```

## Appendix C Cube Voyager Scripts

```

Print form=12.csv LIST= '      6  ',ArSzA[6][1], ArSzA[6][2] , ArSzA[6][3],
ArSzA[6][4] , AreaA[6],file =@Rept@ ;
Print LIST= '      ',file=@Rept@
Print form=12.csv LIST= ' Sum ', RegSzA[1], RegSzA[2] , RegSzA[3],
RegSzA[4] , SITotal, file =@Rept@ ;
Print LIST= '      ','\n',file=@Rept@

```

```

PRINT LIST = ' Estimated Households By Income Level by Area Type ','\n',
file=@Rept@

```

```

PRINT LIST = ' Area_Tp Income_1 Income_2 Income_3
Income_4 Total ',file=@Rept@
PRINT LIST = ' -----
----- ',file=@Rept@
Print form=12.csv LIST= ' 1 ',ArInA[1][1], ArInA[1][2] , ArInA[1][3],
ArInA[1][4] , AreaA[1],file =@Rept@ ;
Print form=12.csv LIST= ' 2 ',ArInA[2][1], ArInA[2][2] , ArInA[2][3],
ArInA[2][4] , AreaA[2],file =@Rept@ ;
Print form=12.csv LIST= ' 3 ',ArInA[3][1], ArInA[3][2] , ArInA[3][3],
ArInA[3][4] , AreaA[3],file =@Rept@ ;
Print form=12.csv LIST= ' 4 ',ArInA[4][1], ArInA[4][2] , ArInA[4][3],
ArInA[4][4] , AreaA[4],file =@Rept@ ;
Print form=12.csv LIST= ' 5 ',ArInA[5][1], ArInA[5][2] , ArInA[5][3],
ArInA[5][4] , AreaA[5],file =@Rept@ ;
Print form=12.csv LIST= ' 6 ',ArInA[6][1], ArInA[6][2] , ArInA[6][3],
ArInA[6][4] , AreaA[6],file =@Rept@ ;
Print LIST= '      ',file=@Rept@
Print form=12.csv LIST= ' Sum ', RegInA[1], RegInA[2] , RegInA[3],
RegInA[4] , SITotal, file =@Rept@ ;
Print LIST= '      ','\n',file=@Rept@

```

```

PRINT LIST = ' Estimated Households By Vehicle Availability Level by Area Type
','\n', file=@Rept@

```

```

PRINT LIST = ' Area_Tp 0 Vehs.Av. 1 Veh.Av. 2 Vehs.Av. 3+
Vehs.Av. Total ',file=@Rept@
PRINT LIST = ' -----
----- ',file=@Rept@
Print form=12.csv LIST= ' 1 ',ArVaA[1][1], ArVaA[1][2] , ArVaA[1][3],
ArVaA[1][4] , AreaA[1],file =@Rept@ ;
Print form=12.csv LIST= ' 2 ',ArVaA[2][1], ArVaA[2][2] , ArVaA[2][3],
ArVaA[2][4] , AreaA[2],file =@Rept@ ;
Print form=12.csv LIST= ' 3 ',ArVaA[3][1], ArVaA[3][2] , ArVaA[3][3],
ArVaA[3][4] , AreaA[3],file =@Rept@ ;
Print form=12.csv LIST= ' 4 ',ArVaA[4][1], ArVaA[4][2] , ArVaA[4][3],
ArVaA[4][4] , AreaA[4],file =@Rept@ ;
Print form=12.csv LIST= ' 5 ',ArVaA[5][1], ArVaA[5][2] , ArVaA[5][3],
ArVaA[5][4] , AreaA[5],file =@Rept@ ;
Print form=12.csv LIST= ' 6 ',ArVaA[6][1], ArVaA[6][2] , ArVaA[6][3],
ArVaA[6][4] , AreaA[6],file =@Rept@ ;
Print LIST= '      ',file=@Rept@
Print form=12.csv LIST= ' Sum ', RegVaA[1], RegVaA[2] , RegVaA[3],
RegVaA[4] , SIVTotal, file =@Rept@ ;
Print LIST= '      ','\n',file=@Rept@

```

```

ENDIF ; -end of printing section

```

```

;
;
;
ENDRUN

```

## 12 Highway\_Assignment\_Parallel.s

```

/*

```

```

-----
Highway_Assignment.s - Version 2.3 / 3722 TAZ traffic assignment
(File renamed to highway_assignment.s) Developed from the
assignment process from V2.2 CTP2step_Highway_Assignment.S (1/7/11 rjm)

```

```

Added added "Vol[6]" to the NonHOV assignment "V=..." statement (RM)

```

```

-----
Updated to write out user market-specific period volumes as well as
total period volumes- This is done for iteration 4 only (JC,RM 10/18/11)
-----

```

Four time-of-day trip tables are used:

AM peak period	3 Hrs. (6 AM - 9 AM)	AM
Midday period	6 Hrs. (9 AM - 3 PM)	MD
PM peak period	4 Hrs. (3 PM - 7 PM)	PM
Night period	11 Hrs. (7 PM - 6 AM)	NT

The AM and PM periods are considered "peak"  
The MD and NT periods are considered "off peak"

Tables on input trip table file:

- 1- SOV
- 2- HOV2-Occ
- 3- HOV3+Occ
- 4- Commercial Vehicles
- 5- Medium/Heavy Truck
- 6- Airport Auto Driver

Structure of the script:

Step 1: Execute peak-period traffic assignments (AM & PM)

- NonHOV3+ traffic assignment
- HOV3+ traffic assignment

Step 2: Execute off-peak-period traffic assignments (midday/MD & evening/OP)

- Off-peak (midday & evening) traffic assignment

Step 3: Calculate restrained speed/perform MSA volume averaging

- Loop thru 1 (AM) and 2 (PM); Each pk per. includes NonHOV3+ and HOV3+
- Loop thru 3 (midday, NT) and 4 (evening/off-peak, OP)

Step 4: Summarize 24-hour VMT of current AM, PM, MD & NT assignments

Traffic assignment is done on a period-specific basis (not peak hour), so hourly capacities are converted to period-specific capacities. By contrast, all period-specific speeds actually represent the peak hour of the given period.

Period-specific trip tables representing more than one hour are assigned, but link capacities are specified in vehicles per hour. A peak-hour factor (PHF), which is the percent of traffic in the peak hour of the period, is used to relate the hourly capacities to the multiple-hour trip tables. See Barton-Aschman Associates, Inc. and Cambridge Systematics, Inc., Model Validation and Reasonableness Checking Manual, February 1997, pp. 78-81.

Environment Variables (set in the "run\_ModelSteps" batch file)

- \_iter\_ Iteration indicator = 'pp','il' - 'i4'
- \_relGap\_ Relative gap (e.g., 10^-3 = 0.001)
- \_maxUeIter\_ Maximum number of user equilibrium iterations

2011-02-11 msm V/C ratio tabulation now goes from 0 to 5 (was 0 to 2), i.e., "0-5-0.1"



## Appendix C Cube Voyager Scripts

```

2011-07-25 rm Added iteration report text file to be written in the the converge
phase
--the following reports are written:
"UE Iteration Report_NonHOV_%_iter_%_@Prd@.txt" -Peak nonHOV
assignments
"UE Iteration Report_HOV_%_iter_%_@Prd@.txt" -Peak HOV
assignments
"UE Iteration Report_%_iter_%_@Prd@.txt"
2012-11-07 msm Standardized names of relative gap report files
"ue_iteration_report_%_iter_%_@prd@_nonHov.txt" -Peak period
(A|PM) nonHOV assignment
"ue_iteration_report_%_iter_%_@prd@_hov.txt" -Peak period
(A|PM) HOV assignment
"ue_iteration_report_%_iter_%_@prd@.txt" -Off-pk per
(MD|NT) assignment
2012-12-04 msm Standardized names of relative gap report files
"%_iter_%_ue_iteration_report@prd@_nonHov.txt" -Peak period
(A|PM) nonHOV assignment
"%_iter_%_ue_iteration_report@prd@_hov.txt" -Peak period
(A|PM) HOV assignment
"%_iter_%_ue_iteration_report@prd@.txt" -Off-pk per
(MD|NT) assignment

2013-02-22 RJM Added time penalty (timepen) to the path impedance function. the
timepen variable is
defined in the V2.3_Highway_Build.s script.

2016-05-10 DNQ Modified the PRINT LIST for the @prd@CHK.LKLOOP files:
"TIMEPEN(5.2)"" becomes "LI.TIMEPEN(5.2)""
to fix the error ? in the printed file (Cube 6.4.1 warns, not Cube
6.1 SP1)
2017-03-16 DNQ Modified to prohibit airport trips using HOV3+ and correct LIMIT
code 1 ->0
*/

/* **** Set up tokens in Voyager Pilot step **** */

PAGEHEIGHT=32767 ; preclude insertion of page headers

; useIdp = t (true) or f (false); this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=%useMdp%

; Choose traffic assignment type, using "enhance=" keyword
; enhance=0 Frank-Wolfe
; enhance=1 Conjugate Frank-Wolfe
; enhance=2 Bi-conjugate Frank-Wolfe
assignType=2

;;;*****
;;; Step 1: Execute peak-period traffic assignments (AM & PM)
;;; AM nonHOV, HOV and PM nonHOV and HOV Assignemnts
;;;*****

itr = '%_iter_%' ;;

; The Input Network Depends on the previous Iteration network

;; IF (itr = 'pp')
;; INPNET = 'ZONEHWY.NET'
;; ELSE
;; INPNET = '%_prev_%_HWY.NET'
;;ENDIF

INPNET = 'ZONEHWY.NET'

DistributeMULTISTEP ProcessID='AM', ProcessNum=1

```

```

PRD = 'AM' ;
PCTADT = 41.7 ; %_AMPF_% AM PHF (% of traffic in pk hr of period)

CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)

in_tmin = '..\support\toll_minutes.txt' ;; read in toll
minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf' ;; AM Toll Factors
by Veh. Type
in_PMTfac = 'inputs\PM_Tfac.dbf' ;; PM Toll Factors
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf' ;; MD Toll Factors
by Veh. Type
in_NTTfac = 'inputs\NT_Tfac.dbf' ;; NT Toll Factors
by Veh. Type

in_capSpd = '..\support\hwy_assign_capSpeedLookup.s' ;; FT x AT Speed &
Capacity lookup
VDF_File = '..\support\hwy_assign_Conical_VDF.s' ;; Volume Delay
Functions file
;

;;;*****
;;; Step 1.1: Assign AM NonHOV3+ trip tables only
;;; (SOV, HOV2, CV, TRUCK & AIRPORT PASSENGER TRIPS)
;;;*****

RUN PGM=HIGHWAY ; NonHOV3+ traffic assignment
distributeIntrastep processId='AM', ProcessList=%AMsubnode%
FILEI NETI = @INPNET@ ; TP+ Network
;
; The input trip table has 6 Vehicle Tables:
; 1 - 1-Occ Auto Drivers
; 2 - 2-Occ Auto Drivers
; 3 - 3+Occ Auto Drivers
; 4 - Commercial Vehicles
; 5 - Trucks
; 6 - Airport Pass. Auto Driver Trips

FILEI MATI=%_iter_%_@prd@.VTT ;
;
FILEO NETO=TEMP1@PRD@.NET ; Output loaded network of current iter/time
prd.
PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
PARAMETERS RELATIVEGAP=%_relGap_% ; Set a relative gap tolerance
PARAMETERS MAXITERS=%_maxUeIter_% ; We control on relative gap. This is
backup criterion

;-----$
; Read in LOS'E' Capacities and Freeflow Speeds $
;-----$
READ FILE = @in_capSpd@
;
;-----$
; Read in Toll Parameters: $
;-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;

```

## Appendix C Cube Voyager Scripts

```

INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] =      "@in_Pmtfac@"
LOOKUP LOOKUPI=2,      NAME=PM_Tfac,
  LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
  LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
  LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
  LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
  LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
  LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] =      "@in_MDtfac@"
LOOKUP LOOKUPI=3,      NAME=MD_Tfac,
  LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
  LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
  LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
  LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
  LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
  LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] =      "@in_NtTfac@"
LOOKUP LOOKUPI=4,      NAME=NT_Tfac,
  LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
  LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;
  LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
  LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
  LOOKUP[5]= TOLLGrp, result=NTTRKTFTR, ;
  LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;
;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
;
;
LOOKUP NAME=VCRV,
  lookup[1] = 1,result = 2, ;Centroids old VCRV1
  lookup[2] = 1,result = 3, ;Fwys old VCRV2
  lookup[3] = 1,result = 4, ;MajArts old VCRV3
  lookup[4] = 1,result = 5, ;MinArts old VCRV4
  lookup[5] = 1,result = 6, ;Colls old VCRV5
  lookup[6] = 1,result = 7, ;Expways old VCRV6
  lookup[7] = 1,result = 8, ;Ramps old VCRV2
  FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@

FUNCTION { ; Congested Time
(TC)specification:
  V = VOL[1] + VOL[2] + VOL[4] + VOL[5] + VOL[6]
  TC[1]= T0*VCRV(1,V/C) ; TC(LINKCLASS) =
  TC[2]= T0*VCRV(2,V/C) ; Uncongested Time(T0) *
  TC[3]= T0*VCRV(3,V/C) ; Volume Delay Funtion(VDF)Value
  TC[4]= T0*VCRV(4,V/C) ; VDF function is based on ((V/C)
  TC[5]= T0*VCRV(5,V/C) ; Note: the LINKCLASS is defined
  TC[6]= T0*VCRV(6,V/C) ; during the LINKREAD phase below.
  TC[7]= T0*VCRV(7,V/C) ; during the LINKREAD phase below.
}
;
;
CAPFAC=@CAPFAC@ ;
; MAXITERS=3 ;
; GAP = 0.0 ;
; AAD = 0.0 ;

; RMSE = 0.0 ;
; RAAD = 0.0 ;

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC@ ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
  LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV TOLLS
in 2007 cents
  LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2 occ TOLLS
in 2007 cents
; ; LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents
  LW.CV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; CV TOLLS
in 2007 cents
  LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; Truck TOLLS
in 2007 cents
  LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(6,LI.TOLLGRP) ; AP Pax TOLLS
in 2007 cents

; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by vehicle
type here:
; ; LW.SOV@PRD@IMP = T0 + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ;SOV IMP
; ; LW.HV2@PRD@IMP = T0 + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ;HOV 2 IMP
; ; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
; ; LW.CV@PRD@IMP = T0 + (LW.CV@PRD@TOLL/100.0)* CV@PRD@EQM ;CV IMP
; ; LW.TRK@PRD@IMP = T0 + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ;Truck IMP
; ; LW.APX@PRD@IMP = T0 + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM ;APAX IMP
; ;
; ; IF (LI.@PRD@TOLL > 0)
; ; PRINT LIST = 'iteration: ',iteration(3), ' A: ',A(7),' B: ',B(7),
; ; ' DISTANCE: ',LI.DISTANCE(6.2),
; ; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
; ; ' FFSPEED: ', SPEED(5.2),
; ; ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
; ; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
; ; ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
; ; ' T0: ', T0(5.2),
; ; ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
; ; file = @prd@CHK.LKREAD
; ; ENDFIF
; ; ENDFIF

; $
;
; The highway network is coded with limit codes from 0 to 9
; LimitCode addGrp Definition
;-----$
; ; 0 1 All vehicles accepted
; ; 2 2 Only HOV2 (or greater) vehicles accepted only
; ; 3 3 Only HOV3 vehicles accepted only
; ; 4 4 Med,Hvy Trks not accepted, all other traffic is accepted
; ; 5 5 Airport Passenger Veh. Trips
; ; 6-8 6 (Unused)
; ; 9 7 No vehicles are accepted at all
;
IF (LI.@PRD@LIMIT==0)
ADDTOGROUP=1

```

## Appendix C Cube Voyager Scripts

```

ELSEIF (LI.@PRD@LIMIT==2)
  ADDTOGROUP=2
ELSEIF (LI.@PRD@LIMIT==3)
  ADDTOGROUP=3
ELSEIF (LI.@PRD@LIMIT==4)
  ADDTOGROUP=4
ELSEIF (LI.@PRD@LIMIT==5)
  ADDTOGROUP=5
ELSEIF (LI.@PRD@LIMIT==6-8)
  ADDTOGROUP=6
ELSEIF (LI.@PRD@LIMIT==9)
  ADDTOGROUP=7
ENDIF

IF (LI.FTYPE = 0) ; LinkClass related to TC[?] above
  LINKCLASS = 1 ;
ELSEIF (LI.FTYPE = 1) ;
  LINKCLASS= 2 ;
ELSEIF (LI.FTYPE = 2) ;
  LINKCLASS= 3 ;
ELSEIF (LI.FTYPE = 3) ;
  LINKCLASS= 4 ;
ELSEIF (LI.FTYPE = 4) ;
  LINKCLASS= 5 ;
ELSEIF (LI.FTYPE = 5) ;
  LINKCLASS= 6 ;
ELSEIF (LI.FTYPE = 6) ;
  LINKCLASS= 7 ;
ENDIF

ENDPHASE

PHASE=ILOOP

  IF (i=FirstZone)
    LINKLOOP
      ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
      vehicle type here:
      LW.SOV@PRD@IMP = TIME + LI.TIMEPEN + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM
;SOV IMP
      LW.HV2@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM
;HOV 2 IMP
      LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP
;-->> LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP
      LW.CV@PRD@IMP = TIME + LI.TIMEPEN + (LW.CV@PRD@TOLL/100.0)* CV@PRD@EQM
;CV IMP
      LW.TRK@PRD@IMP = TIME + LI.TIMEPEN + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM
;Truck IMP
      LW.APX@PRD@IMP = TIME + LI.TIMEPEN + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM
;APAX IMP

      IF (LI.@PRD@TOLL > 0)
        PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
          ' DISTANCE: ',LI.DISTANCE(6.2),
          ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
          ' FFSPEED: ', SPEED(5.2),
          '@PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
          ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
          ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
          ' T0: ', T0(5.2),
          ' TIME: ', TIME(5.2),
          ' TIMEPEN: ', LI.TIMEPEN(5.2),
          ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
          ' file = @prd@CHK.LKLOOP
      ENDIF
    ENDLINKLOOP
  ENDIF

```

```

ENDIF

; Multi-user class or multiclass assignment implemented through volume sets (vol[#])

  PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[1]=MI.1.1 ; SOV
veh
  PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGROUP=3,5,6,7, VOL[2]=MI.1.2 ; HOV 2
;; PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGROUP=5,6,7, VOL[3]=MI.1.3 ; HOV 3
  PATHLOAD PATH=LW.CV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[4]=MI.1.4 ; CVs
  PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGROUP=2,3,4,5,6,7,VOL[5]=MI.1.5 ;
Trucks
  PATHLOAD PATH=LW.APX@PRD@IMP, EXCLUDEGROUP=6,7,3, VOL[6]=MI.1.6 ;
Airport

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
  Fileo Printo[1] = "%_iter_%_ue_iteration_report@prd@_nonHov.txt"
  Print List= "Iter: ", Iteration(3.0)," Gap: ",GAP(16.15)," Relative Gap:
",RGAP(16.15), PRINTO=1
  if (rgap < rgapcutoff)
    balance=1
  endif
ENDPHASE

ENDRUN
;;*****
;; Step 1.2: Assign AM HOV3+ only
;;*****

;;Turnpen = 'inputs\turnpen.pen' ; turn penalty file

  RUN PGM=HIGHWAY ; HOV3+ traffic assignment
  distributeIntrastep processId='AM', ProcessList=%AMsubnode%
  FILEI NETI = TEMP1@PRD@.NET ; TP+ Network
  ;; TURNPENI = @TURNPEN@ ; HOV turn penalty at Gallows
  Road Ramp
  ;
  ; The input trip table has 6 Vehicle Tables:
  ; 1 - 1-Occ Auto Drivers
  ; 2 - 2-Occ Auto Drivers
  ; 3 - 3+Occ Auto Drivers
  ; 4 - Commercial Vehicles
  ; 5 - Trucks
  ; 6 - Airport Pass. Auto Driver Trips

  FILEI MATI=%_iter_%@prd@.VTT ;
  ;
  FILEI NETO=TEMP2@PRD@.NET ; Output loaded network of current iter/time
  prd.
  PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
  PARAMETERS RELATIVEGAP=%_relGap_% ; Set a relative gap tolerance
  PARAMETERS MAXITERS=%_maxUeIter_% ; We control on relative gap. This is
  backup criterion
  ;
  ;-----$
  ; Read in LOS'E' Capacities and Freeflow Speeds $
  ;-----$
  READ FILE = @in_capSpd@

; $
;-----$
; Read in Toll Parameters: $

```

## Appendix C Cube Voyager Scripts

```

;-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_Amtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_PMtfac@"
LOOKUP LOOKUPI=2, NAME=PM_Tfac,
LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] = "@in_Mdtfac@"
LOOKUP LOOKUPI=3, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] = "@in_NTtfac@"
LOOKUP LOOKUPI=4, NAME=NT_Tfac,
LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=NTTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Ramps old VCRV2
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@

FUNCTION {
Congested Time (TC)specification:
V = VOL[3]
TC[1]= T0*VCRV(1,((V+LW.V_1)/C)) ; TC(LINKCLASS) =
TC[2]= T0*VCRV(2,((V+LW.V_1)/C)) ; Uncongested Time(T0) *
TC[3]= T0*VCRV(3,((V+LW.V_1)/C)) ; Volume Delay Funtion(VDF) Value

```

```

TC[4]= T0*VCRV(4,((V+LW.V_1)/C)) ; VDF function is based on (V+LI.V_1)/C
TC[5]= T0*VCRV(5,((V+LW.V_1)/C)) ; Note: the LINKCLASS is defined
TC[6]= T0*VCRV(6,((V+LW.V_1)/C)) ; during the LINKREAD phase below.
TC[7]= T0*VCRV(7,((V+LW.V_1)/C)) ; during the LINKREAD phase below.
}
;
;
CAPFAC=@CAPFAC@ ;
;MAXITERS=3 ;
;GAP = 0.0 ;
;AAD = 0.0 ;
;RMSE = 0.0 ;
;RAAD = 0.0 ;

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC@ ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
T1 = LI.TIME_1
LW.V_1 = LI.V_1
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents

;;; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by vehicle
type here:
;;; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
;;;
;;; IF (LI.@PRD@TOLL > 0)
;;; PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
;;; ' DISTANCE: ',LI.DISTANCE(6.2),
;;; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
;;; ' FFSPEED: ', SPEED(5.2),
;;; '@PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
;;; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
;;; ' LW.HV3@PRD@TOLL: ', LW.HV3@PRD@TOLL(5.2),
;;; ' T0: ', T0(5.2),
;;; ' LW.HV3@PRD@IMP', LW.HV3@PRD@IMP(5.2),
;;; file = @prd@CHK.LKREAD
;;; ENDF
;;;
ENDIF

;$
;
; The highway network is coded with limit codes from 0 to 9
; LimitCode addGrp Definition
; -----
; 0 1 All vehicles accepted
; 2 2 Only HOV2 (or greater) vehicles accepted only
; 3 3 Only HOV3 vehicles accepted only
; 4 4 Med,Hvy Trks not accepted, all other traffic is accepted
; 5 5 Airport Passenger Veh. Trips
; 6-8 6 (Unused)
; 9 7 No vehicles are accepted at all
;
IF (LI.@PRD@LIMIT==0)
ADDTOGROUP=1
ELSEIF (LI.@PRD@LIMIT==2)
ADDTOGROUP=2

```

## Appendix C Cube Voyager Scripts

```

ELSEIF (LI.@PRD@LIMIT==3)
  ADDTOGROUP=3
ELSEIF (LI.@PRD@LIMIT==4)
  ADDTOGROUP=4
ELSEIF (LI.@PRD@LIMIT==5)
  ADDTOGROUP=5
ELSEIF (LI.@PRD@LIMIT==6-8)
  ADDTOGROUP=6
ELSEIF (LI.@PRD@LIMIT==9)
  ADDTOGROUP=7
ENDIF

IF (LI.FTYPE = 0) ; LinkClass related to TC[?] above
  LINKCLASS = 1 ;
ELSEIF (LI.FTYPE = 1) ;
  LINKCLASS= 2 ;
ELSEIF (LI.FTYPE = 2) ;
  LINKCLASS= 3 ;
ELSEIF (LI.FTYPE = 3) ;
  LINKCLASS= 4 ;
ELSEIF (LI.FTYPE = 4) ;
  LINKCLASS= 5 ;
ELSEIF (LI.FTYPE = 5) ;
  LINKCLASS= 6 ;
ELSEIF (LI.FTYPE = 6) ;
  LINKCLASS= 7 ;
ENDIF

ENDPHASE

PHASE=ILOOP

IF (i=FirstZone)
  LINKLOOP
  ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
  vehicle type here:
  LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP

  IF (LI.@PRD@TOLL > 0)
    PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
    ' DISTANCE: ',LI.DISTANCE(6.2),
    ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
    ' FFSPEED: ', SPEED(5.2),
    '@PRD@_TFAC(1,LI.TOLLGRP) : ',@PRD@_TFAC(1,LI.TOLLGRP) (5.1),
    ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
    ' LW.HV3@PRD@TOLL: ', LW.HV3@PRD@TOLL(5.2),
    ' T0: ', T0(5.2),
    ' TIME: ', TIME(5.2),
    ' TIMEPEN: ', LI.TIMEPEN(5.2),
    ' LW.HV3@PRD@IMP', LW.HV3@PRD@IMP(5.2),
    file = @prd@CHK.LKLOOP
  ENDIF

  ENDLINKLOOP
ENDIF
;
;
; There is only one volume set, so this is not a multi-user class or multiclass
assignm.

PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGROUP=5,6,7, VOL[3]=MI.1.3 ; HOV 3

ENDPHASE

```

```

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
  Fileo Printo[1] = "%_iter_%_ue_iteration_report@prd@hov.txt"
  Print List= "Iter: ", Iteration(3.0)," Gap: ",GAP(16.15)," Relative Gap:
",RGAP(16.15), PRINTO=1
  if (rgap < rgapcutoff)
    balance=1
  endif
ENDPHASE

ENDRUN
ENDDistributeMULTISTEP

PRD = 'PM' ;
PCTADT = 29.4 ; %_AMPF_% AM PHF (% of traffic in pk hr of period)

CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)

in_tmin = '..\support\toll_minutes.txt' ; read in toll
minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf' ; AM Toll Factors
by Veh. Type
in_PMTfac = 'inputs\PM_Tfac.dbf' ; PM Toll Factors
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf' ; MD Toll Factors
by Veh. Type
in_NTTfac = 'inputs\NT_Tfac.dbf' ; NT Toll Factors
by Veh. Type

in_capSpd = '..\support\hwy_assign_capSpeedLookup.s' ; FT x AT Speed &
Capacity lookup
VDF_File = '..\support\hwy_assign_Conical_VDF.s' ; Volume Delay
Functions file
;
;*****
;; Step 1.3: Assign PM NonHOV3+ trip tables only
;; (SOV, HOV2, CV, TRUCK & AIRPORT PASSENGER TRIPS)
;*****

RUN PGM=HIGHWAY ; NonHOV3+ traffic assignment
distributeIntrastep processId='MD', ProcessList=%MDsubnode%
FILEI NETI = @INPNET@ ; TP+ Network
;
; The input trip table has 6 Vehicle Tables:
; 1 - 1-Occ Auto Drivers
; 2 - 2-Occ Auto Drivers
; 3 - 3+Occ Auto Drivers
; 4 - Commercial Vehicles
; 5 - Trucks
; 6 - Airport Pass. Auto Driver Trips

FILEI MATI=%_iter_%_@prd@.VTT ;
;
FILEO NETO=TEMP1_@PRD@.NET ; Output loaded network of current iter/time
prd.
PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
PARAMETERS RELATIVEGAP=%_relGap_% ; Set a relative gap tolerance
PARAMETERS MAXITERS=%_maxUeIter_% ; We control on relative gap. This is
backup criterion

;-----$

```

## Appendix C Cube Voyager Scripts

```

; Read in LOS'E' Capacities and Freeflow Speeds $
;-----$
READ FILE = @in_capSpd@
;
;-----$
; Read in Toll Parameters: $
;-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_PMtfac@"
LOOKUP LOOKUPI=2, NAME=PM_Tfac,
LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] = "@in_MDtfac@"
LOOKUP LOOKUPI=3, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] = "@in_NTtfac@"
LOOKUP LOOKUPI=4, NAME=NT_Tfac,
LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=NTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Ramps old VCRV2
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@

```

```

FUNCTION { ; Congested Time
(TC)specification:
V = VOL[1] + VOL[2] + VOL[4] + VOL[5] + VOL[6]
TC[1]= T0*VCRV(1,V/C) ; TC(LINKCLASS) =
TC[2]= T0*VCRV(2,V/C) ; Uncongested Time(T0) *
TC[3]= T0*VCRV(3,V/C) ; Volume Delay Funtion(VDF)Value
TC[4]= T0*VCRV(4,V/C) ; VDF function is based on ((V/C)
TC[5]= T0*VCRV(5,V/C) ; Note: the LINKCLASS is defined
TC[6]= T0*VCRV(6,V/C) ; during the LINKREAD phase below.
TC[7]= T0*VCRV(7,V/C) ; during the LINKREAD phase below.
}
;
; CAPFAC=@CAPFAC@ ;
; MAXITERS=3 ;
; GAP = 0.0 ;
; AAD = 0.0 ;
; RMSE = 0.0 ;
; RAAD = 0.0 ;

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC@ ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV TOLLS
in 2007 cents
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2 occ TOLLS
in 2007 cents
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents
LW.CV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; CV TOLLS
in 2007 cents
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; Truck TOLLS
in 2007 cents
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(6,LI.TOLLGRP) ; AP Pax TOLLS
in 2007 cents
; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by vehicle
type here:
; LW.SOV@PRD@IMP = T0 + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ;SOV IMP
; LW.HV2@PRD@IMP = T0 + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ;HOV 2 IMP
; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
; LW.CV@PRD@IMP = T0 + (LW.CV@PRD@TOLL/100.0)* CV@PRD@EQM ;CV IMP
; LW.TRK@PRD@IMP = T0 + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ;Truck IMP
; LW.APX@PRD@IMP = T0 + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM ;APAX IMP
;
; IF (LI.@PRD@TOLL > 0)
; PRINT LIST = 'iteration: ',iteration(3), ' A: ',A(7), ' B: ',B(7),
; ' DISTANCE: ',LI.DISTANCE(6.2),
; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
; ' FFSPEED: ', SPEED(5.2),
; ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
; ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
; ' T0: ', T0(5.2),
; ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
; file = @prd@CHK.LKREAD
; ENDDIF
;
ENDIF

```

## Appendix C Cube Voyager Scripts

```

;S
;
; The highway network is coded with limit codes from 0 to 9
; LimitCode addGrp Definition
; -----
; 0 1 All vehicles accepted
; 2 2 Only HOV2 (or greater) vehicles accepted only
; 3 3 Only HOV3 vehicles accepted only
; 4 4 Med,Hvy Trks not accepted, all other traffic is accepted
; 5 5 Airport Passenger Veh. Trips
; 6-8 6 (Unused)
; 9 7 No vehicles are accepted at all
;
IF (LI.@PRD@LIMIT==0)
  ADDTOGROUP=1
ELSEIF (LI.@PRD@LIMIT==2)
  ADDTOGROUP=2
ELSEIF (LI.@PRD@LIMIT==3)
  ADDTOGROUP=3
ELSEIF (LI.@PRD@LIMIT==4)
  ADDTOGROUP=4
ELSEIF (LI.@PRD@LIMIT==5)
  ADDTOGROUP=5
ELSEIF (LI.@PRD@LIMIT==6-8)
  ADDTOGROUP=6
ELSEIF (LI.@PRD@LIMIT==9)
  ADDTOGROUP=7
ENDIF

IF (LI.FTYPE = 0) ; LinkClass related to TC[?] above
  LINKCLASS = 1 ;
ELSEIF (LI.FTYPE = 1) ;
  LINKCLASS= 2 ;
ELSEIF (LI.FTYPE = 2) ;
  LINKCLASS= 3 ;
ELSEIF (LI.FTYPE = 3) ;
  LINKCLASS= 4 ;
ELSEIF (LI.FTYPE = 4) ;
  LINKCLASS= 5 ;
ELSEIF (LI.FTYPE = 5) ;
  LINKCLASS= 6 ;
ELSEIF (LI.FTYPE = 6) ;
  LINKCLASS= 7 ;
ENDIF

ENDPHASE

PHASE=ILOOP

IF (i=FirstZone)
  LINKLOOP
  ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
  vehicle type here:
  LW.SOV@PRD@IMP = TIME + LI.TIMEPEN + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM
;SOV IMP
  LW.HV2@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM
;HOV 2 IMP
  LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP
  LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP
  LW.CV@PRD@IMP = TIME + LI.TIMEPEN + (LW.CV@PRD@TOLL/100.0)* CV@PRD@EQM
;CV IMP
  LW.TRK@PRD@IMP = TIME + LI.TIMEPEN + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM
;Truck IMP

LW.APX@PRD@IMP = TIME + LI.TIMEPEN + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM
;APAX IMP

IF (LI.@PRD@TOLL > 0)
  PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
  ' DISTANCE: ',LI.DISTANCE(6.2),
  ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
  ' FFSPEED: ', SPEED(5.2),
  '@PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
  ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
  ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
  ' T0: ', T0(5.2),
  ' TIME: ', TIME(5.2),
  ' TIMEPEN: ', LI.TIMEPEN(5.2),
  ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
  file = @prd@CHK.LKLOOP
ENDIF
ENDLINKLOOP

ENDIF

; Multi-user class or multiclass assignment implemented through volume sets (vol[#])

veh
  PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[1]=MI.1.1 ; SOV
; ;
  PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGROUP=3,5,6,7, VOL[2]=MI.1.2 ; HOV 2
; ;
  PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGROUP=5,6,7, VOL[3]=MI.1.3 ; HOV 3
; ;
  PATHLOAD PATH=LW.CV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[4]=MI.1.4 ; CVs
; ;
  PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGROUP=2,3,4,5,6,7, VOL[5]=MI.1.5 ;
Trucks
  PATHLOAD PATH=LW.APX@PRD@IMP, EXCLUDEGROUP=6,7,3, VOL[6]=MI.1.6 ;
Airport

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
Fileo Printo[1] = "%_iter_%_ue_iteration_report@prd@_nonHov.txt"
Print List= "Iter: ", Iteration(3.0)," Gap: ",GAP(16.15)," Relative Gap:
",RGAP(16.15), PRINTO=1
if (rgap < rgapcutoff)
  balance=1
endif
ENDPHASE

ENDRUN
; ;*****
; ; Step 1.4: Assign PM HOV3+ only
; ;*****

; ;Turnpen = 'inputs\turnpen.pen' ; turn penalty file

RUN PGM=HIGHWAY ; HOV3+ traffic assignment
distributeIntrastep processId='MD', ProcessList=%MDsubnode%
FILEI NETI = TEMPI@PRD@NET ; TP+ Network
; ; TURNPENI = @TURNPEN@ ; HOV turn penalty at Gallows
Road Ramp
;
; The input trip table has 6 Vehicle Tables:
; 1 - 1-Occ Auto Drivers
; 2 - 2-Occ Auto Drivers
; 3 - 3+Occ Auto Drivers
; 4 - Commercial Vehicles
; 5 - Trucks
; 6 - Airport Pass. Auto Driver Trips

```

## Appendix C Cube Voyager Scripts

```

FILEI MATI=%_iter%_@prd@.VTT ;
;
FILEO NETO=TEMP2_@PRD@.NET ; Output loaded network of current iter/time
prd.
PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
PARAMETERS RELATIVEGAP=%_relGap% ; Set a relative gap tolerance
PARAMETERS MAXITERS=%_maxUeIter% ; We control on relative gap. This is
backup criterion
;
-----$
; Read in LOS'E' Capacities and Freeflow Speeds $
-----$
READ FILE = @in_capSpd@

;$
-----$
; Read in Toll Parameters: $
-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_PMtfac@"
LOOKUP LOOKUPI=2, NAME=PM_Tfac,
LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] = "@in_MDtfac@"
LOOKUP LOOKUPI=3, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] = "@in_NTtfac@"
LOOKUP LOOKUPI=4, NAME=NT_Tfac,
LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=NTTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;
-----$
; VDF (Volume Delay Function) establishment: $
-----$

```

```

; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Ramps old VCRV2
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@

FUNCTION {
Congested Time (TC)specification:
V = VOL[3]
TC[1]= T0*VCRV(1,((V+LW.V_1)/C)) ; TC(LINKCLASS) =
TC[2]= T0*VCRV(2,((V+LW.V_1)/C)) ; Uncongested Time(T0) *
TC[3]= T0*VCRV(3,((V+LW.V_1)/C)) ; Volume Delay Funtion(VDF)Value
TC[4]= T0*VCRV(4,((V+LW.V_1)/C)) ; VDF function is based on (V+LI.V_1)/C
TC[5]= T0*VCRV(5,((V+LW.V_1)/C)) ; Note: the LINKCLASS is defined
TC[6]= T0*VCRV(6,((V+LW.V_1)/C)) ; during the LINKREAD phase below.
TC[7]= T0*VCRV(7,((V+LW.V_1)/C)) ; during the LINKREAD phase below.
}
;
;
CAPFAC=@CAPFAC@ ;
;MAXITERS=3 ;
;GAP = 0.0 ;
;AAD = 0.0 ;
;RMSE = 0.0 ;
;RAAD = 0.0 ;

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
T1 = LI.TIME_1
LW.V_1 = LI.V_1
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents

; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by vehicle
type here:
; ; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ; HOV 3+IMP
; ;
; ; IF (LI.@PRD@TOLL > 0)
; ; PRINT LIST = 'iteration: ',iteration(3), ' A: ',A(7),' B: ',B(7),
; ; ' DISTANCE: ',LI.DISTANCE(6.2),
; ; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
; ; ' FFSPEED: ', SPEED(5.2),
; ; ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
; ; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
; ; ' LW.HV3@PRD@TOLL: ', LW.HV3@PRD@TOLL(5.2),
; ; ' T0: ', T0(5.2),
; ; ' LW.HV3@PRD@IMP', LW.HV3@PRD@IMP(5.2),
; ; file = @prd@CHK.LKREAD
; ; ENDF
; ;
; ; ENDF
; ;

```





## Appendix C Cube Voyager Scripts

```

FILEO NETO=temp2_@PRD@.net          ; Output loaded network of current iter/time
prd. FOR OFF PEAK
PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
PARAMETERS RELATIVEGAP=%_relGap%   ; Set a relative gap tolerance
PARAMETERS MAXITERS=%_maxUeiter%   ; We control on relative gap. This is
backup criterion
;
;-----$
;   Read in LOS'E' Capacities and Freeflow Speeds   $
;-----$
READ FILE = @in_capSpd@

;$
;-----$
;   Read in Toll Parameters:                         $
;-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] =           "@in_Amtfac@"
LOOKUP LOOKUPI=1,          NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] =           "@in_PMtfac@"
LOOKUP LOOKUPI=2,          NAME=PM_Tfac,
LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] =           "@in_MDtfac@"
LOOKUP LOOKUPI=3,          NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] =           "@in_NTtfac@"
LOOKUP LOOKUPI=4,          NAME=NT_Tfac,
LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=NTTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;
;-----$
;   VDF (Volume Delay Function) establishment:       $
;-----$
; Note: curves updated 2/16/06 rjm/msm
LOOKUP NAME=VCRV,

```

```

lookup[1] = 1,result = 2, ;Centroids   old VCRV1
lookup[2] = 1,result = 3, ;Fwys       old VCRV2
lookup[3] = 1,result = 4, ;MajArts   old VCRV3
lookup[4] = 1,result = 5, ;MinArts   old VCRV4
lookup[5] = 1,result = 6, ;Collis    old VCRV5
lookup[6] = 1,result = 7, ;Expways   old VCRV6
lookup[7] = 1,result = 8, ;Ramps     old VCRV2
FAIL=0.00,0.00,0.00, INTERPOLATE=T, file=@VDF_File@

```

```

FUNCTION {
; Congested Time (TC) specification:
TC[1]= T0*VCRV(1,VC) ; TC(LINKCLASS) =
TC[2]= T0*VCRV(2,VC) ; Uncongested Time(T0) *
TC[3]= T0*VCRV(3,VC) ; Volume Delay Funtion(VDF) Value
TC[4]= T0*VCRV(4,VC) ; VDF function is based on VC
TC[5]= T0*VCRV(5,VC) ; Note: the LINKCLASS is defined
TC[6]= T0*VCRV(6,VC) ; during the LINKREAD phase below.
TC[7]= T0*VCRV(7,VC) ; during the LINKREAD phase below.
}
;
;
CAPFAC=@CAPFAC@ ;
;MAXITERS=3 ;
;GAP = 0.0 ;
;AAD = 0.0 ;
;RMSE = 0.0 ;
;RAAD = 0.0 ;

```

```

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC@ ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

```

```

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV TOLLS
in 2007 cents
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2 occ TOLLS
in 2007 cents
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents
LW.CV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; CV TOLLS
in 2007 cents
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; Truck TOLLS
in 2007 cents
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(6,LI.TOLLGRP) ; AP Pax TOLLS
in 2007 cents

```

```

;;; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
vehicle type here:
;;; LW.SOV@PRD@IMP = T0 + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ; SOV IMP
;;; LW.HV2@PRD@IMP = T0 + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ; HOV 2 IMP
;;; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ; HOV 3+IMP
;;; LW.CV@PRD@IMP = T0 + (LW.CV@PRD@TOLL /100.0)* CV@PRD@EQM ; CV IMP
;;; LW.TRK@PRD@IMP = T0 + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ; Truck IMP
;;; LW.APX@PRD@IMP = T0 + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM ; APAX IMP
;;;
;;; IF (LI.@PRD@TOLL > 0)
;;; PRINT LIST = 'iteration: ',iteration(3), ' A: ',A(7), ' B: ',B(7),
;;; ' DISTANCE: ',LI.DISTANCE(6.2),
;;; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
;;; ' FFSPEED: ', SPEED(5.2),
;;; ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
;;; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),

```

## Appendix C Cube Voyager Scripts

```

;;;      ' LW.SOV@PRD@TOLL: ',          LW.SOV@PRD@TOLL(5.2),
;;;      ' T0: ',                      T0(5.2),
;;;      ' LW.SOV@PRD@IMP',            LW.SOV@PRD@IMP(5.2),
;;;      file = @prd@CHK.LKREAD
;;;      ENDIF
;;;
ENDIF

;$
;
; The highway network is coded with limit codes from 0 to 9
; LimitCode addGrp Definition
; -----
; 0 1 All vehicles accepted
; 2 2 Only HOV2 (or greater) vehicles accepted only
; 3 3 Only HOV3 vehicles accepted only
; 4 4 Med,Hvy Trks not accepted, all other traffic is accepted
; 5 5 Airport Passenger Veh. Trips
; 6-8 6 (Unused)
; 9 7 No vehicles are accepted at all
;
IF (LI.@PRD@LIMIT==0)
  ADDTOGROUP=1
ELSEIF (LI.@PRD@LIMIT==2)
  ADDTOGROUP=2
ELSEIF (LI.@PRD@LIMIT==3)
  ADDTOGROUP=3
ELSEIF (LI.@PRD@LIMIT==4)
  ADDTOGROUP=4
ELSEIF (LI.@PRD@LIMIT==5)
  ADDTOGROUP=5
ELSEIF (LI.@PRD@LIMIT==6-8)
  ADDTOGROUP=6
ELSEIF (LI.@PRD@LIMIT==9)
  ADDTOGROUP=7
ENDIF

IF (LI.FTYPE = 0) ; LinkClass related to TC[?] above
  LINKCLASS = 1 ;
ELSEIF (LI.FTYPE = 1) ;
  LINKCLASS = 2 ;
ELSEIF (LI.FTYPE = 2) ;
  LINKCLASS = 3 ;
ELSEIF (LI.FTYPE = 3) ;
  LINKCLASS = 4 ;
ELSEIF (LI.FTYPE = 4) ;
  LINKCLASS = 5 ;
ELSEIF (LI.FTYPE = 5) ;
  LINKCLASS = 6 ;
ELSEIF (LI.FTYPE = 6) ;
  LINKCLASS = 7 ;
ENDIF

ENDPHASE

PHASE=ILOOP

IF (i=FirstZone)
  LINKLOOP
  ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
  vehicle type here:
  LW.SOV@PRD@IMP = TIME + LI.TIMEPEN + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM
;SOV IMP
  LW.HV2@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM
;HOV 2 IMP
  LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP

```

```

      LW.CV@PRD@IMP = TIME + LI.TIMEPEN + (LW.CV@PRD@TOLL /100.0)* CV@PRD@EQM
;CV IMP
      LW.TRK@PRD@IMP = TIME + LI.TIMEPEN + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM
;Truck IMP
      LW.APX@PRD@IMP = TIME + LI.TIMEPEN + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM
;APAX IMP

IF (LI.@PRD@TOLL > 0)
  PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
  ' DISTANCE: ',LI.DISTANCE(6.2),
  ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
  ' FFSPEED: ', SPEED(5.2),
  ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP) (5.1),
  ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
  ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
  ' T0: ', T0(5.2),
  ' TIME: ', TIME(5.2),
  ' TIMEPEN: ', LI.TIMEPEN(5.2),
  ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
  file = @prd@CHK.LKLOOP
ENDIF

ENDLINKLOOP

ENDIF

; Multi-user class or multiclass assignment implemented through volume sets (vol[#])
PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[1]=MI.1.1 ; SOV
veh
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGROUP=3,5,6,7, VOL[2]=MI.1.2 ; HOV 2
PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGROUP=5,6,7, VOL[3]=MI.1.3 ; HOV 3
PATHLOAD PATH=LW.CV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[4]=MI.1.4 ; CVs
PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGROUP=2,3,4,5,6,7,VOL[5]=MI.1.5 ;
Trucks
PATHLOAD PATH=LW.APX@PRD@IMP, EXCLUDEGROUP=6,7,3, VOL[6]=MI.1.6 ;
Airport

;$

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
  Fileo Printo[1] = "%_iter_%_ue_iteration_report_@prd@.txt"
  Print List= "Iter: ", Iteration(3.0)," _Gap: ",GAP(16.15)," Relative Gap:
",RGAP(16.15), PRINTO=1
  if (rgap < rgapcutoff)
    balance=1
  endif
ENDPHASE

ENDRUN

ENDDistributeMULTISTEP

PRD = 'NT' ;
PCTADT = 15.0 ; %_NTPF_% NT PHF (% of traffic in pk hr of period)

CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)
; Turnpen = 'inputs\turnpen.pen' ; Turn penalty

RUN PGM=HIGHWAY ; Off-peak (midday & evening) traffic assignment
distributeIntrastep processId='MD', ProcessList=%MDsubnode$

```

## Appendix C Cube Voyager Scripts

```

FILEI NETI = @INPNET@ ; TP+ Network
;; TURNPENI = @TURNPENI@ ; HOV turn penalty at Gallows Road
Ramp
;
; The input trip table has 6 Vehicle Tables:
; 1 - 1-Occ Auto Drivers
; 2 - 2-Occ Auto Drivers
; 3 - 3+Occ Auto Drivers
; 4 - Commercial Vehicles
; 5 - Trucks
; 6 - Airport Pass. Auto Driver Trips

FILEI MATTI=%_iter%_@prd@.VTT ;
;
FILEO NETO=temp2_@PRD@.net ; Output loaded network of current iter/time
prd. FOR OFF PEAK
PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
PARAMETERS RELATIVEGAP=%_relGap% ; Set a relative gap tolerance
PARAMETERS MAXITERS=%_maxUeIter% ; We control on relative gap. This is
backup criterion
;
;-----$
; Read in LOS'E' Capacities and Freeflow Speeds $
;-----$
READ FILE = @in_capSpd@

;$
;-----$
; Read in Toll Parameters: $
;-----$
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_PMtfac@"
LOOKUP LOOKUPI=2, NAME=PM_Tfac,
LOOKUP[1]= TOLLGrp, result=PMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=PMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=PMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=PMCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=PMTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=PMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] = "@in_MDtfac@"
LOOKUP LOOKUPI=3, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[4] = "@in_NTtfac@"
LOOKUP LOOKUPI=4, NAME=NT_Tfac,
LOOKUP[1]= TOLLGrp, result=NTSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=NTHV2TFTR, ;

```

```

LOOKUP[3]= TOLLGrp, result=NTHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=NTCOMTFTR, ;
LOOKUP[5]= TOLLGrp, result=NTTRKTFTR, ;
LOOKUP[6]= TOLLGrp, result=NTAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Ramps old VCRV2
FAIL=0.00,0.00,0.00, INTERPOLATE=T, file=@VDF_File@

FUNCTION { ; Congested Time (TC) specification:
TC[1]= T0*VCRV(1,VC) ; TC(LINKCLASS) =
TC[2]= T0*VCRV(2,VC) ; Uncongested Time(T0) *
TC[3]= T0*VCRV(3,VC) ; Volume Delay Funtion(VDF) Value
TC[4]= T0*VCRV(4,VC) ; VDF function is based on VC
TC[5]= T0*VCRV(5,VC) ; Note: the LINKCLASS is defined
TC[6]= T0*VCRV(6,VC) ; during the LINKREAD phase below.
TC[7]= T0*VCRV(7,VC) ; during the LINKREAD phase below.
}
;
;
CAPFAC=@CAPFAC@ ;
;MAXITERS=3 ;
;GAP = 0.0 ;
;AAD = 0.0 ;
;RMSE = 0.0 ;
;RAAD = 0.0 ;

PHASE=LINKREAD
C = CAPACITYFOR(LI.@PRD@LANE,LI.CAPCLASS) * @CAPFAC@ ; Convert hourly
capacities to period-specific
SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCCLASS)
T0 = (LI.DISTANCE/SPEED)*60.0
; Since there is no "DISTANCE =" statement, this assumes that DISTANCE is
avail. on input network

IF (ITERATION = 0)
; Define link level tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV TOLLS
in 2007 cents
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2 occ TOLLS
in 2007 cents
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV 3+occ TOLLS
in 2007 cents
LW.CV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; CV TOLLS
in 2007 cents
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; Truck TOLLS
in 2007 cents
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(6,LI.TOLLGRP) ; AP Fax TOLLS
in 2007 cents

;; ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
vehicle type here:

```

## Appendix C Cube Voyager Scripts

```

;;; LW.SOV@PRD@IMP = T0 + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ;SOV IMP
;;; LW.HV2@PRD@IMP = T0 + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ;HOV 2 IMP
;;; LW.HV3@PRD@IMP = T0 + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
;;; LW.CV@PRD@IMP = T0 + (LW.CV@PRD@TOLL /100.0)* CV@PRD@EQM ;CV IMP
;;; LW.TRK@PRD@IMP = T0 + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ;Truck IMP
;;; LW.APX@PRD@IMP = T0 + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM ;APAX IMP
;;;
;;; IF (LI.@PRD@TOLL > 0)
;;; PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
;;; ' DISTANCE: ',LI.DISTANCE(6.2),
;;; ' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
;;; ' FFSPEED: ', SPEED(5.2),
;;; ' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
;;; ' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
;;; ' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
;;; ' T0: ', T0(5.2),
;;; ' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
;;; file = @prd@CHK.LKREAD
;;; ENDDIF
;;;
ENDIF

;$
;
; The highway network is coded with limit codes from 0 to 9
; LimitCode addGrp Definition
; -----
; 1 1 All vehicles accepted
; 2 2 Only HOV2 (or greater) vehicles accepted only
; 3 3 Only HOV3 vehicles accepted only
; 4 4 Med,Hvy Trks not accepted, all other traffic is accepted
; 5 5 Airport Passenger Veh. Trips
; 6-8 6 (Unused)
; 9 7 No vehicles are accepted at all
;
;
IF (LI.@PRD@LIMIT==0)
ADDTOGROUP=1
ELSEIF (LI.@PRD@LIMIT==2)
ADDTOGROUP=2
ELSEIF (LI.@PRD@LIMIT==3)
ADDTOGROUP=3
ELSEIF (LI.@PRD@LIMIT==4)
ADDTOGROUP=4
ELSEIF (LI.@PRD@LIMIT==5)
ADDTOGROUP=5
ELSEIF (LI.@PRD@LIMIT==6-8)
ADDTOGROUP=6
ELSEIF (LI.@PRD@LIMIT==9)
ADDTOGROUP=7
ENDIF

IF (LI.FTYPE = 0) ; LinkClass related to TC[?] above
LINKCLASS = 1 ;
ELSEIF (LI.FTYPE = 1) ;
LINKCLASS = 2 ;
ELSEIF (LI.FTYPE = 2) ;
LINKCLASS = 3 ;
ELSEIF (LI.FTYPE = 3) ;
LINKCLASS = 4 ;
ELSEIF (LI.FTYPE = 4) ;
LINKCLASS = 5 ;
ELSEIF (LI.FTYPE = 5) ;
LINKCLASS = 6 ;
ELSEIF (LI.FTYPE = 6) ;
LINKCLASS = 7 ;
ENDIF

```

```

ENDPHASE

PHASE=ILOOP

IF (i=FirstZone)
LINKLOOP
; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by
vehicle type here:
LW.SOV@PRD@IMP = TIME + LI.TIMEPEN + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM
;SOV IMP
LW.HV2@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM
;HOV 2 IMP
LW.HV3@PRD@IMP = TIME + LI.TIMEPEN + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM
;HOV 3+IMP
LW.CV@PRD@IMP = TIME + LI.TIMEPEN + (LW.CV@PRD@TOLL /100.0)* CV@PRD@EQM
;CV IMP
LW.TRK@PRD@IMP = TIME + LI.TIMEPEN + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM
;Truck IMP
LW.APX@PRD@IMP = TIME + LI.TIMEPEN + (LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM
;APAX IMP

IF (LI.@PRD@TOLL > 0)
PRINT LIST = 'iteration: ',iteration(3),' A: ',A(7),' B: ',B(7),
'DISTANCE: ',LI.DISTANCE(6.2),
' LI.@PRD@TOLL: ', LI.@PRD@TOLL(5.2),
' FFSPEED: ', SPEED(5.2),
' @PRD@_TFAC(1,LI.TOLLGRP): ',@PRD@_TFAC(1,LI.TOLLGRP)(5.1),
' SV@PRD@EQM: ', SV@PRD@EQM(5.1),
' LW.SOV@PRD@TOLL: ', LW.SOV@PRD@TOLL(5.2),
' T0: ', T0(5.2),
' LW.SOV@PRD@IMP', LW.SOV@PRD@IMP(5.2),
file = @prd@CHK.LKLOOP
ENDIF

ENDLINKLOOP

ENDIF

; Multi-user class or multiclass assignment implemented through volume sets (vol[#])

PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[1]=MI.1.1 ; SOV
veh
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGROUP=3,5,6,7, VOL[2]=MI.1.2 ; HOV 2
PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGROUP=5,6,7, VOL[3]=MI.1.3 ; HOV 3
PATHLOAD PATH=LW.CV@PRD@IMP, EXCLUDEGROUP=2,3,5,6,7, VOL[4]=MI.1.4 ; CVs
PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGROUP=2,3,4,5,6,7,VOL[5]=MI.1.5 ;
Trucks
PATHLOAD PATH=LW.APX@PRD@IMP, EXCLUDEGROUP=6,7,3, VOL[6]=MI.1.6 ;
Airport

;$

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
Fileo Printo[1] = "%_iter_%_iteration_report@prd.txt"
Print List = "Iter: ", Iteration(3.0)," Gap: ",GAP(16.15)," Relative Gap:
",RGAP(16.15), PRINTO=1
if (rgap < rgapcutoff)
balance=1
endif
ENDPHASE

```

## Appendix C Cube Voyager Scripts

```

ENDRUN

Wait4Files Files=AM1.script.end, CheckReturnCode=T, PrintFiles=Merge,
DelDistribFiles=T

;
; END OF MIDDAY and OFF PEAK ASSIGNMENT
;
;;*****
;; Step 3: Calculate restrained final Volumes, speeds, V/Cs (No MSA)
;;*****
;;*****
;; Step 3.1: Loop thru 1 (AM) and 2 (PM)
;;*****

LOOP PERIOD = 1,2 ; Loop thru 1 (AM) and 2 (PM); Each pk per. includes NonHOV3+
and HOV3+

IF (PERIOD==1)
    PRD = 'AM' ;
    PCTADT = 41.7 ;
ELSE
    PRD = 'PM' ;
    PCTADT = 29.4 ;
ENDIF
;
; CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)

RUN PGM=HWYNET ; Calculate restrained speed/perform MSA volume
averaging
FILEI NETI=temp2_@PRD@.net ; input network from highway assignment
FILEO NETO=temp_@prd@.net, ; output/@PRD@ network with updated speeds
EXCLUDE=V_1,TIME_1,VC_1,V1_1, V2_1, V3_1, V4_1,V5_1, V6_1,
VT_1,V1T_1,V2T_1,V3T_1,V4T_1,V5T_1,V6T_1,
CSPD_1,VDT_1,VHT_1,
V_2,TIME_2,VC_2,V1_2, V2_2,V3_2,V4_2,V5_2,V6_2,
VT_2,V1T_2,V2T_2,V3T_2,V4T_2,V5T_2,V6T_2,
WRSPD,WFFSPD
;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Rmps
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@
;
; to keep stratified vehicular volume
; only in Iteration 4
;
IF ('%_iter_%' = 'I4')
    %_iter_%@PRD@SOV = V1_1
    %_iter_%@PRD@HV2 = V2_1
    %_iter_%@PRD@HV3 = V_2
    %_iter_%@PRD@CV = V4_1
    %_iter_%@PRD@TRK = V5_1

```

```

    %_iter_%@PRD@APX = V6_1
ENDIF
;
; %_iter_%@prd@VOL = V_1 + V_2 ; Final
AM/PM Link Volume ; Final
; %_iter_%@prd@VMT = %_iter_%@prd@VOL * distance ; Final
AM/PM link VMT ;
; %_iter_%@prd@FFSPD =SPEEDFOR(@prd@LANE,SPDCLASS) ;
Freeflow speed ;
; @prd@HRLKCAP=CAPACITYFOR(@prd@LANE,CAPCLASS) ; Hrly
Link capacity ; Hrly
; @prd@HRLNCA=CAPACITYFOR(1,CAPCLASS) ;
Lane capacity ;
; %_iter_%@prd@VC=(%_iter_%@prd@VOL*(@pctadt@/100.0)/@prd@HRLKCAP) ; AM/PM
VC ratio ;
; %_iter_%@prd@VDF = VCRV((Ftype + 1), %_iter_%@prd@VC) ; AM/PM
VDF ;
; if (%_iter_%@prd@VDF > 0) %_iter_%@prd@SPD = %_iter_%@prd@FFSPD /
; %_iter_%@prd@VDF ; AM/PM speed (No queuing)
; ATYPE=SPDCLASS%10 ; Area
Type
; _cnt = 1.0
;
;
;
; compute WEIGHTED restrained and freeflow SPEEDS for Aggregate summaries

WRSPD =ROUND(%_iter_%@prd@VMT * %_iter_%@prd@SPD)
WFFSPD=ROUND(%_iter_%@prd@VMT * %_iter_%@prd@FFSPD)

; Crosstab VMT,Wrspd,Wffspd, by FTYPE and JUR
CROSSTAB VAR=%_iter_%@prd@VMT,Wrspd,Wffspd,_CNT,FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter_%@prd@VMT, FORM=12.2cs ; AVG INITIAL SPD
COMP=Wffspd/%_iter_%@prd@VMT, FORM=12.2cs ; AVG FINAL SPD

; Crosstab %_iter_%@prd@VMT,WOSPD,WNSPD,_CNT2 by ATYPE and FTYPE
CROSSTAB VAR=%_iter_%@prd@VMT,Wrspd,Wffspd,_CNT, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter_%@prd@VMT, FORM=12.2cs ; AVG INITIAL SPD
COMP=Wffspd/%_iter_%@prd@VMT, FORM=12.2cs ; AVG FINAL SPD

; Crosstab VMT,WOSPD,WNSPD,WFFSPD,_CNT2 by EVC and FTYPE
CROSSTAB VAR=%_iter_%@prd@VMT,Wrspd,Wffspd,_CNT, FORM=12cs,
ROW=%_iter_%@prd@VC, RANGE=0-5-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter_%@prd@VMT, FORM=12.2cs ; AVG INITIAL SPD
COMP=Wffspd/%_iter_%@prd@VMT, FORM=12.2cs ; Freeflow Speed

; PRINT TO check

print LIST=A(5),' ',B(5),DISTANCE(7.2),' ',@PCTADT@(4.3),' ',@prd@LANE(2.0),' ',
@prd@HRLKCAP(5.0),' ',@prd@HRLNCA(5.0),' ',
%_iter_%@prd@VOL(8.2),' ',
%_iter_%@prd@ffspd(5.1),' ',%_iter_%@prd@VC(6.4),' ',%_iter_%@prd@VDF(6.4),' ',
ftype(3.0),' ',ATYPE(3.0),' ',%_iter_%@prd@SPD(5.1),
FILE=%_iter_%@prd@_load_link.asc

;;

ENDRUN
ENDLOOP ; Loop thru 1 (AM) and 2 (PM); Each pk per. includes NonHOV3+
and HOV3+

```

## Appendix C Cube Voyager Scripts

```

;;*****
;; Step 3.2: Loop thru 3 (MD) and 4 (OP)
;;*****
LOOP PERIOD = 3,4 ; Loop thru 1 (midday, MD) and 2 (evening/off-peak, OP)
IF (PERIOD=3)
    PRD = 'MD' ;
    PCTADT = 17.7
ELSE
    PRD = 'NT' ;
    PCTADT = 15.0
ENDIF
;
CAPFAC=1/(PCTADT/100) ; Capacity Factor = 1/(PCTADT/100)

RUN PGM=HWYNET ; Calculate restrained speed/perform MSA volume averaging
FILEI NETI=temp2_@PRD@.net ; input network from highway assignment
FILEO NETO=temp_@PRD@.net, ; output/@PRD@ network with updated speeds
    EXCLUDE=V_1,TIME_1,VC_1,V1_1, V2_1, V3_1, V4_1,V5_1,V6_1,
    VT_1,V1T_1,V2T_1,V3T_1,V4T_1,V5T_1,V6T_1,
    CSPD_1,VDT_1,VHT_1,WRSPD,WFFSPD
;
;-----$
; VDF (Volume Delay Function) establishment: $
;-----$
; Note: curves updated 2/16/06 rjm/msm
;
LOOKUP NAME=VCRV,
lookup[1] = 1,result = 2, ;Centroids old VCRV1
lookup[2] = 1,result = 3, ;Fwys old VCRV2
lookup[3] = 1,result = 4, ;MajArts old VCRV3
lookup[4] = 1,result = 5, ;MinArts old VCRV4
lookup[5] = 1,result = 6, ;Colls old VCRV5
lookup[6] = 1,result = 7, ;Expways old VCRV6
lookup[7] = 1,result = 8, ;Rmps
FAIL=0.00,0.00,0.00, INTERPOLATE=T,file=@VDF_File@
;
; to keep stratified vehicular volume
; only in Iteration 4
;
IF ('%_iter%' = 'I4')
    %_iter%@PRD@SOV = V1_1
    %_iter%@PRD@HV2 = V2_1
    %_iter%@PRD@HV3 = V3_1
    %_iter%@PRD@CV = V4_1
    %_iter%@PRD@TRK = V5_1
    %_iter%@PRD@APX = V6_1
ENDIF
;
;
%_iter%@PRD@VOL = V_1
; Final Link Volume
%_iter%@PRD@VMT = %_iter%@PRD@VOL * distance
; Final Link VMT
%_iter%@PRD@FFSPD =SPEEDFOR(@PRD@LANE,SPCLASS)
; Freeflow speed
@PRD@HRLKCAP=CAPACITYFOR(@PRD@LANE,CAPCLASS)
; Hrly LINK capacity
@PRD@HRLNCP=CAPACITYFOR(1,CAPCLASS)
; Hrly LANE capacity
%_iter%@PRD@VC=(%_iter%@PRD@VOL*(pctadt%/100.0)/@PRD@HRLKCAP)
; Period VC ratio
%_iter%@PRD@VDF = VCRV((Ftype + 1), %_iter%@PRD@VC)
; Period VDF value

```

```

if (%_iter%@PRD@VDF > 0) %_iter%@PRD@SPD = %_iter%@PRD@FFSPD / %_iter%@PRD@VDF
; Restrained Link speed(no Queuing delay)
ATYPE=SPDCLASS%10
; area type
_cnt = 1.0
; counter

;;
; compute WEIGHTED restrained and freeflow SPEEDS for Aggregate summaries

WRSPD =ROUND(%_iter%@PRD@VMT * %_iter%@PRD@SPD)
WFFSPD=ROUND(%_iter%@PRD@VMT * %_iter%@PRD@FFSPD)

; Crosstab VMT,Wrspd,Wffspd, by FTYPE and JUR
CROSSTAB VAR=%_iter%@PRD@VMT,Wrspd,Wffspd,_CNT,FORM=12cs,
ROW=JUR, RANGE=0-23-1,,0-23,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter%@PRD@VMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP=Wffspd/%_iter%@PRD@VMT, FORM=12.2cs ; AVG FINAL SPD

; Crosstab %_iter%@PRD@VMT,WOSPD,WNSPD,_CNT2 by ATYPE and FTYPE
CROSSTAB VAR=%_iter%@PRD@VMT,Wrspd,Wffspd,_CNT, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,,1-7,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter%@PRD@VMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP=Wffspd/%_iter%@PRD@VMT, FORM=12.2cs ; AVG FINAL SPD

; Crosstab VMT,WOSPD,WNSPD,WFFSPD,_CNT2 by EVC and FTYPE
CROSSTAB VAR=%_iter%@PRD@VMT,Wrspd,Wffspd,_CNT, FORM=12cs,
ROW=%_iter%@PRD@VC, RANGE=0-5-0.1,,1-99,
COL=FTYPE, RANGE=1-6-1,1-6,
COMP=Wrspd/%_iter%@PRD@VMT, FORM=12.2cs, ; AVG INITIAL SPD
COMP=Wffspd/%_iter%@PRD@VMT, FORM=12.2cs ; Freeflow Speed

; PRINT TO check

print LIST=A(5),' ',B(5),DISTANCE(7.2),' ',@PCTADT@(4.3),' ',@PRD@LANE(2.0),' ',
@PRD@HRLKCAP(5.0),' ',@PRD@HRLNCP(5.0),' ',
%_iter%@PRD@VOL(8.2),' ',
%_iter%@PRD@ffspd(5.1),' ',%_iter%@PRD@VC(6.4),' ',%_iter%@PRD@VDF(6.4),' ',
ftype(3.0),' ',ATYPE(3.0),' ',%_iter%@PRD@SPD(5.1),
FILE=%_iter%@PRD@_load_link.asc

;;
ENDRUN
ENDLOOP ; Loop thru 1 (midday, MD) and 2 (evening/off-peak, OP)

;;*****
;; Step 4: Summarize 24-hour VMT of current AM, PM, MD & NT assignments
;;*****
RUN PGM=HWYNET ; Summarize 24-hour VMT of current AM, PM, MD & OP assignments
FILEI NETI[1]=temp_AM.net
FILEI NETI[2]=temp_MD.net
FILEI NETI[3]=temp_PM.net
FILEI NETI[4]=temp_NT.net
FILEO NETO =%_iter% HWY.NET,
    EXCLUDE=OLDVOL1,NEWVOL1,OLDVOL2,NEWVOL2,OLDVOL3,NEWVOL3,
    OLDVOL4,NEWVOL4,OLDVOL5,NEWVOL5,
    OLDSPD1,OLDSPD2,OLDSPD3,OLDSPD4,OLDSPD5,%_iter%_24VMT,
    CSPD_2,VDT_2,VHT_2

%_iter%amspd = LI.1.%_iter%amspd
%_iter%mdspd = LI.2.%_iter%mdspd

```

## Appendix C Cube Voyager Scripts

```

%_iter_%_pmspd = LI.3.%_iter_%_pmspd
%_iter_%_ntspd = LI.4.%_iter_%_ntspd
;
;
VOLAM = LI.1.%_iter_%_AMVOL
VOLMD = LI.2.%_iter_%_MDVOL
VOLPM = LI.3.%_iter_%_PMVOL
VOLNT = LI.4.%_iter_%_NTVOL
; COMPUTE FINAL DAILY VOLUME ON ALL LINKS
%_iter_%24VOL = VOLAM + VOLMD + VOLPM + VOLNT ; Total Daily Volume
; COMPUTE FINAL DAILY VMT ON ALL NON-CENTROID LINKS
IF (FTYPE = 0)
%_iter_%24VMT = 0
ELSE
%_iter_%24VMT = %_iter_%24VOL * DISTANCE ; Total Daily VMT
ENDIF
;
;
IF (FTYPE=1-6)
TVOL00=ROUND((VOLAM + VOLMD + VOLPM + VOLNT)/1000.0) ; total hwy vol in 000s
TVMT00=TVOL00*DISTANCE ; total hwy VMT in 000s
ELSE
TVOL00=0
TVMT00=0
ENDIF
;
; IF (FTYPE=1-6 && COUNT > 0 || (AMLIMIT = 2-3 || PMLIMIT=2-3 || NTLIMIT=2-3))
; TVolEST=TVol00 ; total hwy vol in 000s
; TVolObs=count ; total hwy vol in 000s
; TVMTEST=TVMT00 ; total hwy vol in 000s
; TVMTOBS=count*DISTANCE ; total hwy VMT in 000s
; ELSE
; Tvmtest=0
; TVMTObs=0 ; total hwy VMT in 000s
;ENDIF
;
comp atype=spdclass%10 ; area type code 1-7
; ; its the first digit of spdclass var
; Crosstab TVMTEST,TVMTOBS by ATYPE and FTYPE
; CROSSTAB VAR=TVMTEST,TVMTOBS, FORM=8cs,
; ROW=ATYPE, RANGE=1-7-1,1-7,
; COL=FTYPE, RANGE=0-6-1,0-6,
; COMP=TVMTEST-TVMTOBS, FORM=8cs, ; Difference (est-obs)
; COMP=TVMTEST/TVMTOBS, FORM=8.2cs ; Ratio (est/obs)
;
; Crosstab TVMTEST,TVMTOBS by Jurisdiction and FTYPE
; CROSSTAB VAR=TVMTEST,TVMTOBS, FORM=8cs,
; ROW=JUR, RANGE=0-23-1,0-23,
; COL=FTYPE, RANGE=0-6-1,0-6,
; COMP=TVMTEST-TVMTOBS, FORM=8cs, ; Difference (est-obs)
; COMP=TVMTEST/TVMTOBS, FORM=8.2cs ; Ratio (est/obs)
;
; Crosstab TVMTEST,TVMTOBS by Screenline and FTYPE
; CROSSTAB VAR=TVolEST,TVolOBS, FORM=8cs,
; ROW=SCREEN, RANGE=1-38-1,1-38,
; COL=FTYPE, RANGE=0-6-1,0-6,
; COMP=TVolEST-TVolOBS, FORM=8cs, ; Difference (est-obs)
; COMP=TVolEST/TVolOBS, FORM=8.2cs ; Ratio (est/obs)
;
;-----
; DAILY X-Tabs
;-----

```

```

; Crosstab DAILY VMT by ATYPE and FTYPE
CROSSTAB VAR=%_iter_%24VMT, FORM=12cs,
ROW=ATYPE, RANGE=1-7-1,1-7,
COL=FTYPE, RANGE=1-6-1,1-6
; Crosstab Total VMT by Jurisdiction and FTYPE
CROSSTAB VAR=%_iter_%24VMT, FORM=12cs,
ROW=JUR, RANGE=0-23-1,0-23,
COL=FTYPE, RANGE=0-6-1,0-6

```

ENDRUN

## 13 Highway\_Skims\_am.s

```

;//////////////////////////////////////
; Highway_Skims.S //
; MWCOC Version 2.3 Model //
; //
; Build AM Peak/Off-Peak Highway Skims //
; the Current Iteration Assignment //
; AM and Off-Pk Skims are built in 2 separate HWYLOAD //
; programs. //
; Three files are created, per SOV, HOV2, and HOV3 paths.//
; //
; 1) Time (xx.xx minutes) //
; 2) Distance (implied tenths of mi., xx.xx) //
; 3) Toll (in 2007 cents, xx.xx) //
;
; 6/30/03 MODIFICATIONS FOR IMPROVED TOLL MODELING MADE rjm
;
; 1/25/08 Changes made to create special changes to mode choice skims
; 1/31/08 generalized toll used in pathtracing changed to be mode-specific
; e.g. MW[3] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0,
; ..was changed to> MW[3] =PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS=0,
;
; MW[6] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0, ;
; ..was changed to> MW[6] =PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS=0, ;
;
; MW[9] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0, ;
; ..was changed to> MW[9] =PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS=0, ;
;
; 4/25/08 Modifications for Truck model wga/rm
; Note Time is not rounded (to whole mintes) any more
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to the
impedance calculation
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to both
impedance * time skim calculation
;
;//////////////////////////////////////
;
; Environment Variables:
; _iter_ (Iteration indicator = 'pp','i1'-'i4')
;
;
; pageheight=32767 ; Preclude header breaks
NETIN = '%_iter_%_hwy.net'
;
; Output special truck skim only for off-peak conditions
LOOP Period=1,1 ; We are looping through the skimming process
; twice: (1) for the AM Peak & (2) the Midday

```



## Appendix C Cube Voyager Scripts

```

in_tmin = '..\support\toll_minutes.txt'
minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf'
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf'
by Veh. Type

IF (Period=1) ; AM Highway Skim tokens
PRD = 'AM'
MATOUT1 = '%_iter_%_am_sov.skm'
MATOUT2 = '%_iter_%_am_hov2.skm'
MATOUT3 = '%_iter_%_am_hov3.skm'

MATOUTMC1 = '%_iter_%_am_sov_MC.skm'
MATOUTMC2 = '%_iter_%_am_hov2_MC.skm'
MATOUTMC3 = '%_iter_%_am_hov3_MC.skm'

MYID = '%_iter_%_AM skims'

TT = ' ; '
MATOUT4 = ' '
SKMTOT = ' '

ELSEIF (Period=2) ; MD Highway Skim tokens
PRD = 'MD'
MATOUT1 = '%_iter_%_md_sov.skm'
MATOUT2 = '%_iter_%_md_hov2.skm'
MATOUT3 = '%_iter_%_md_hov3.skm'

MATOUTMC1 = '%_iter_%_md_sov_MC.skm'
MATOUTMC2 = '%_iter_%_md_hov2_MC.skm'
MATOUTMC3 = '%_iter_%_md_hov3_MC.skm'

TT = ' ; '
MATOUT4 = '%_iter_%_md_truck.skm'
SKMTOT = '%_iter_%_skimtot.txt'

MYID = '%_iter_%_MD skims'
ENDIF

RUN PGM=HIGHWAY
;
NETI =@NETIN@ ; Pk Prd TP+ network
MATO[1]=@MATOUT1@, MO=1,2,3,13 ; LOV skims: time, dist, total tolls, VP tolls
(default output precision is 2 decimal places)
MATO[2]=@MATOUT2@, MO=4,5,6,16 ; HOV2 skims: time, dist, total tolls, VP tolls
MATO[3]=@MATOUT3@, MO=7,8,9,19 ; HOV3+ skims: time, dist, total tolls, VP tolls
@TT@ MATO[4]=@MATOUT4@, MO=10 ; Truck skims

ID=@MYID@
;-
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_MDtfac@"

```

```

LOOKUP LOOKUPI=2, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;-

PHASE=LINKREAD
SPEED = LI.%_iter_%@PRD@SPD ;Restrained speed (min)
IF (SPEED = 0)
TI = 0
ELSE
TI = (LI.DISTANCE / SPEED * 60.0) + LI.TIMEPEN
ENDIF
;-

; Define AM /MD link level TOTAL tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ;
SOV TOTAL TOLLS in 2007 cents ;
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ;
HOV 2 occ TOTAL TOLLS in 2007 cents ;
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ;
HOV 3+occ TOTAL TOLLS in 2007 cents ;
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ;
Truck TOTAL TOLLS in 2007 cents ;
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ;
AP Pax TOTAL TOLLS in 2007 cents ;

LW.SOV@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(1,LI.TOLLGRP) ;
SOV VarPr TOLLS in 2007 cents ;
LW.HV2@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(2,LI.TOLLGRP) ;
HOV 2 occ VarPr TOLLS in 2007 cents ;
LW.HV3@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(3,LI.TOLLGRP) ;
HOV 3+occ VarPr TOLLS in 2007 cents ;
LW.TRK@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(4,LI.TOLLGRP) ;
Truck VarPr TOLLS in 2007 cents ;
LW.APX@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(5,LI.TOLLGRP) ;
AP Pax VarPr TOLLS in 2007 cents ;

; Define AM /MD IMPEDANCE by vehicle type here:
LW.SOV@PRD@IMP= T1 + ((LW.SOV@PRD@TOLL/100.0) * SV@PRD@EQM);SOV IMP
LW.HV2@PRD@IMP= T1 + ((LW.HV2@PRD@TOLL/100.0) * H2@PRD@EQM);HOV 2 IMP
LW.HV3@PRD@IMP= T1 + ((LW.HV3@PRD@TOLL/100.0) * H3@PRD@EQM);HOV 3+IMP
LW.TRK@PRD@IMP= T1 + ((LW.TRK@PRD@TOLL/100.0) * TK@PRD@EQM);Truck IMP
LW.APX@PRD@IMP= T1 + ((LW.APX@PRD@TOLL/100.0) * AP@PRD@EQM);APAX IMP

;
; Define the three path types here:
;
; limit codes used:
; 1=no prohibitions
; 2=prohibit 1/occ autos,trucks
; 3=prohibit 1&2occ autos,trucks
; 4=prohibit trucks
; 5=prohibit non-airport access trips
; 6-8=unused
; 9=prohibit all traffic use

IF (LI.@PRD@LIMIT = 2,3,5-9) ADDTOGROUP=1 ; SOV prohibited links
IF (LI.@PRD@LIMIT = 3,5-9) ADDTOGROUP=2 ; HOV2 prohibited links
IF (LI.@PRD@LIMIT = 5-9) ADDTOGROUP=3 ; HOV3+ prohibited links
IF (LI.@PRD@LIMIT = 4) ADDTOGROUP=4 ; Truck prohibited links

;
ENDPHASE
;

```

## Appendix C Cube Voyager Scripts

```

; Now do the path skimming, per the three path types. Time, distance,
; and Toll skims created. Scaling to the desired specified below.
; All skims are based on minimum time paths.
;
; Note that override values of 0 will be inserted for disconnected ijs
; (i.e. cells associated with 'unused' zones and intrazonal cells).
; I don't like the TP+ default value of 1,000,000 for these situations
;
; 1/25/08 added skim tabs created:
; (t13,t16,t19) tolls on variably priced facilities only

PHASE=ILOOP

PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGRP=1, ; SOV paths
MW[1] =PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[2] =PATHTRACE(DIST), NOACCESS=0, ; w/LIMIT=2,3,5-9
MW[3] =PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS=0, ;
MW[13]=PATHTRACE(LW.SOV@PRD@TOLL_VP), NOACCESS=0 ;

PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGRP=2, ; HOV2 paths
MW[4] =PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[5] =PATHTRACE(DIST), NOACCESS=0, ; w/LIMIT=3,5-9
MW[6] =PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS=0, ;
MW[16]=PATHTRACE(LW.HV2@PRD@TOLL_VP), NOACCESS=0 ;

PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGRP=3, ; HOV3+ paths
MW[7] =PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[8] =PATHTRACE(DIST), NOACCESS=0, ; w/LIMIT=5-9
MW[9] =PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS=0, ;
MW[19]=PATHTRACE(LW.HV3@PRD@TOLL_VP), NOACCESS=0 ;

@TT@ PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGRP=1,4, ; Truck paths
@TT@ MW[10]=PATHTRACE(TIME), NOACCESS=0

;-----
; scaling, rounding of skim tables done here!!
;-----

mw[2] = ROUND(MW[2]*10) ; FACTOR/ROUND DIST.
mw[5] = ROUND(MW[5]*10) ; SKIMS TO IMPLICIT
mw[8] = ROUND(MW[8]*10) ; 1/10THS OF MILES

mw[3] = ROUND(MW[3]) ; ROUND Total TOLL
mw[6] = ROUND(MW[6]) ; SKIMS TO 2007
mw[9] = ROUND(MW[9]) ; WHOLE CENTS

mw[13] = ROUND(MW[13]) ; ROUND Variable priced TOLL
mw[16] = ROUND(MW[16]) ; SKIMS TO 2007
mw[19] = ROUND(MW[19]) ; WHOLE CENTS

;
;-----
; Print selected rows of skim files
; for checking.
;-----

IF (i = 1-2) ; for select rows (Is)
printrow MW=1-3, j=1-3722 ; print work matrices 1-3
ENDIF ; row value to all Js.
ENDPHASE
ENDRUN

;-----
; Finally create special Mode Choice skims here

```

```

; The mode choice skims will be the same as the above skims unless VP toll lanes
; are used; in that case time will include the VP toll time equivalent
; and the toll value will be the toll on non-VP toll lanes ONLY
;
; Also create zonal truck access file per the @TT@ statements for the OP per. only
;-----

RUN PGM=MATRIX

READ FILE = @in_tmin@ ; read toll time eqv param file
; -- INPUT SKIMS --
MATI[1] = @MATOUT1@ ; SOV skims (tm,dst,total
toll, VP toll)
MATI[2] = @MATOUT2@ ; HOV2 skims (tm,dst,total
toll, VP toll)
MATI[3] = @MATOUT3@ ; HOV3+skims (tm,dst,total
toll, VP toll)
@TT@ MATI[4] = @MATOUT4@ ; read in trk skim (op per
only)
@TT@ MW[99] = MI.4.1
; For the skim total, put a large value in unconnected O/D pairs
@TT@ JLOOP
@TT@ IF (MW[99] = 0) MW[99] = 100000
@TT@ ENDJLOOP
@TT@ REPORT MARGINREC = Y, FILE = @SKMTOT@, FORM=15, LIST=J(5),R99,C99

; -- OUTPUT SKIMS --
MATO[1] = @MATOUTMC1@,MO=101,12,103 ; SOV skims (tm&toll tm eqv,dst,non-VP
toll component)
MATO[2] = @MATOUTMC2@,MO=201,22,203 ; HOV2 skims (tm&toll tm eqv,dst,non-VP
toll component)
MATO[3] = @MATOUTMC3@,MO=301,32,303 ; HOV3+skims (tm&toll tm eqv,dst,non-VP
toll component)

; read in input skims from above
MW[11] = MI.1.1 ; SOV time
MW[12] = MI.1.2 ; SOV distance
MW[13] = MI.1.3 ; SOV total toll
MW[14] = MI.1.4 ; SOV Var.priced toll component (if VP toll facility used)

MW[21] = MI.2.1 ; HOV2 time
MW[22] = MI.2.2 ; HOV2 distance
MW[23] = MI.2.3 ; HOV2 total toll
MW[24] = MI.2.4 ; HOV2 Var.priced toll component (if VP toll facility used)

MW[31] = MI.3.1 ; HOV3+ time
MW[32] = MI.3.2 ; HOV3+ distance
MW[33] = MI.3.3 ; HOV3+ total toll
MW[34] = MI.3.4 ; HOV3+ Var.priced toll component (if VP toll facility used)

; now compute special time and toll values to be used in the mode choice process
; which are normally 1/time, 2/distance, and 3/tolls; the new skims will be:
; 1/ time + the toll time equivalent on VP facilities only
; 2/ distance (as before)
; 3/ tolls on non-VP tolled facilities ONLY

;Mode Choice model Hwy time:
MW[101] = MW[11] + ((MW[14]/100.0) * SV@PRD@EQM);
MW[201] = MW[21] + ((MW[24]/100.0) * H2@PRD@EQM);
MW[301] = MW[31] + ((MW[34]/100.0) * H3@PRD@EQM);

;Mode Choice model Hwy TOLL:
MW[103] = MW[13] - MW[14]
MW[203] = MW[23] - MW[24]
MW[303] = MW[33] - MW[34]

```

## Appendix C Cube Voyager Scripts

```

MW[103] = MAX(0,MW[103])
MW[203] = MAX(0,MW[203])
MW[303] = MAX(0,MW[303])
ENDRUN

;*ping -n 11 127.0.0.1 > nul

; end of truck access section
ENDLOOP

```

## 14 Highway\_Skims\_md.s

```

//////////////////////////////////////
; Highway_Skims.S //;
; MWCOCG Version 2.3 Model //;
; //;
; Build AM Peak/Off-Peak Highway Skims //;
; the Current Iteration Assignment //;
; AM and Off-Pk Skims are built in 2 separate HWYLOAD //;
; programs. //;
; Three files are created, per SOV, HOV2, and HOV3 paths.//;
; //;
; 1) Time (xx.xx minutes) //;
; 2) Distance (implied tenths of mi., xx.xx) //;
; 3) Toll (in 2007 cents, xx.xx) //;
; //;
; 6/30/03 MODIFICATIONS FOR IMPROVED TOLL MODELING MADE rjm
;
; 1/25/08 Changes made to create special changes to mode choice skims
; 1/31/08 generalized toll used in pathtracing changed to be mode-specific
; e.g. MW[3] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0,
; ..was changed to> MW[3] =PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS=0,
;
; MW[6] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0, ;
; ..was changed to> MW[6] =PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS=0, ;
;
; MW[9] =PATHTRACE(LI.@PRD@TOLL), NOACCESS=0, ;
; ..was changed to> MW[9] =PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS=0, ;
;
; 4/25/08 Modifications for Truck model wga/rm
; Note Time is not rounded (to whole mints) any more
;
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to the
impedance calculation
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to both
impedance * time skim calculation
;
//////////////////////////////////////
;
; Environment Variables:
; _iter_ (Iteration indicator = 'pp','i1'-'i4')
;
pageheight=32767 ; Preclude header breaks
NETIN = '%_iter_%_hwy.net'

; Output special truck skim only for off-peak conditions

LOOP Period=2,2 ; We are looping through the skimming process
; twice: (1) For the AM Peak & (2) the Midday

```

```

in_tmin = '..\support\toll_minutes.txt' ;; read in toll
minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf' ;; AM Toll Factors
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf' ;; MD Toll Factors
by Veh. Type

IF (Period=1) ; AM Highway Skim tokens
PRD = 'AM'
MATOUT1 = '%_iter_%_am_sov.skm'
MATOUT2 = '%_iter_%_am_hov2.skm'
MATOUT3 = '%_iter_%_am_hov3.skm'

MATOUTMC1 = '%_iter_%_am_sov_MC.skm'
MATOUTMC2 = '%_iter_%_am_hov2_MC.skm'
MATOUTMC3 = '%_iter_%_am_hov3_MC.skm'

MYID = '%_iter_%_AM skims'

TT = ';'
MATOUT4 = ''
SKMTOT = ''

ELSEIF (Period=2) ; MD Highway Skim tokens
PRD = 'MD'
MATOUT1 = '%_iter_%_md_sov.skm'
MATOUT2 = '%_iter_%_md_hov2.skm'
MATOUT3 = '%_iter_%_md_hov3.skm'

MATOUTMC1 = '%_iter_%_md_sov_MC.skm'
MATOUTMC2 = '%_iter_%_md_hov2_MC.skm'
MATOUTMC3 = '%_iter_%_md_hov3_MC.skm'

TT = ''
MATOUT4 = '%_iter_%_md_truck.skm'
SKMTOT = '%_iter_%_skimtot.txt'

MYID = '%_iter_%_MD skims'
ENDIF

RUN PGM=HIGHWAY
;
;
NETI =@NETIN@ ; Pk Prd TP+ network
MATO[1]=@MATOUT1@, MO=1,2,3,13 ; LOV skims: time, dist, total tolls, VP tolls
(default output precision is 2 decimal places)
MATO[2]=@MATOUT2@, MO=4,5,6,16 ; HOV2 skims: time, dist, total tolls, VP tolls
MATO[3]=@MATOUT3@, MO=7,8,9,19 ; HOV3+ skims: time, dist, total tolls, VP tolls
@TT@ MATO[4]=@MATOUT4@, MO=10 ; Truck skims

ID=@MYID@
;-
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMTfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_MDTfac@"
LOOKUP LOOKUPI=2, NAME=MD_Tfac,

```

## Appendix C Cube Voyager Scripts

```

LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;
;
PHASE=LINKREAD
SPEED = LI.%_iter_@PRD@SPD ;Restrained speed (min)
IF (SPEED = 0)
  T1 = 0
ELSE
  T1 = (LI.DISTANCE / SPEED * 60.0) + LI.TIMEPEN
ENDIF
;
; Define AM /MD link level TOTAL tolls by vehicle type here:
SOV LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ;
TOTAL TOLLS in 2007 cents
HOV 2 LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ;
TOTAL TOLLS in 2007 cents
HOV 3+occ LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ;
TOTAL TOLLS in 2007 cents
Truck LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ;
TOTAL TOLLS in 2007 cents
AP Pax LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ;
TOTAL TOLLS in 2007 cents
;
SOV LW.SOV@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(1,LI.TOLLGRP) ;
VarPr TOLLS in 2007 cents
HOV 2 LW.HV2@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(2,LI.TOLLGRP) ;
VarPr TOLLS in 2007 cents
HOV 3+occ LW.HV3@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(3,LI.TOLLGRP) ;
VarPr TOLLS in 2007 cents
Truck LW.TRK@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(4,LI.TOLLGRP) ;
VarPr TOLLS in 2007 cents
AP Pax LW.APX@PRD@TOLL_VP = LI.@PRD@TOLL_VP * @PRD@_TFAC(5,LI.TOLLGRP) ;
VarPr TOLLS in 2007 cents
;
; Define AM /MD IMPEDANCE by vehicle type here:
LW.SOV@PRD@IMP= T1 + ((LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM);SOV IMP
LW.HV2@PRD@IMP= T1 + ((LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM);HOV 2 IMP
LW.HV3@PRD@IMP= T1 + ((LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM);HOV 3+IMP
LW.TRK@PRD@IMP= T1 + ((LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM);Truck IMP
LW.APX@PRD@IMP= T1 + ((LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM);APAX IMP
;
; Define the three path types here:
;
;
; limit codes used:
; 1=no prohibitions
; 2=prohibit 1/occ autos,trucks
; 3=prohibit 1&2occ autos,trucks
; 4=prohibit trucks
; 5=prohibit non-airport access trips
; 6-8=unused
; 9=prohibit all traffic use
;
IF (LI.@PRD@LIMIT = 2,3,5-9) ADDTOGROUP=1 ; SOV prohibited links
IF (LI.@PRD@LIMIT = 3,5-9) ADDTOGROUP=2 ; HOV2 prohibited links
IF (LI.@PRD@LIMIT = 5-9) ADDTOGROUP=3 ; HOV3+ prohibited links
IF (LI.@PRD@LIMIT = 4) ADDTOGROUP=4 ; Truck prohibited links
;
ENDPHASE
;
; Now do the path skimming, per the three path types. Time, distance,

```

```

; and Toll skims created. Scaling to the desired specified below.
; All skims are based on minimum time paths.
;
; Note that override values of 0 will be inserted for disconnected ijs
; (i.e. cells associated with 'unused' zones and intrazonal cells).
; I don't like the TP+ default value of 1,000,000 for these situations
;
; 1/25/08 added skim tabs created:
; (t13,t16,t19) tolls on variably priced facilities only
;
PHASE=ILOOP
;
PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGRP=1, ; SOV paths
MW[1] =PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[2] =PATHTRACE (DIST), NOACCESS=0, ; w/LIMIT=2,3,5-9
MW[3] =PATHTRACE (LW.SOV@PRD@TOLL), NOACCESS=0, ;
MW[13]=PATHTRACE (LW.SOV@PRD@TOLL_VP), NOACCESS=0 ;
;
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGRP=2, ; HOV2 paths
MW[4] =PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[5] =PATHTRACE (DIST), NOACCESS=0, ; w/LIMIT=3,5-9
MW[6] =PATHTRACE (LW.HV2@PRD@TOLL), NOACCESS=0, ;
MW[16]=PATHTRACE (LW.HV2@PRD@TOLL_VP), NOACCESS=0 ;
;
PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGRP=3, ; HOV3+ paths
MW[7] =PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[8] =PATHTRACE (DIST), NOACCESS=0, ; w/LIMIT=5-9
MW[9] =PATHTRACE (LW.HV3@PRD@TOLL), NOACCESS=0, ;
MW[19]=PATHTRACE (LW.HV3@PRD@TOLL_VP), NOACCESS=0 ;
;
@TT@ PATHLOAD PATH=LW.TRK@PRD@IMP, EXCLUDEGRP=1,4, ; Truck paths
@TT@ MW[10]=PATHTRACE (TIME), NOACCESS=0
;
;-----
; scaling, rounding of skim tables done here!!
;-----
mw[2] = ROUND(MW[2]*10) ; FACTOR/ROUND DIST.
mw[5] = ROUND(MW[5]*10) ; SKIMS TO IMPLICIT
mw[8] = ROUND(MW[8]*10) ; 1/10THS OF MILES
;
mw[3] = ROUND(MW[3]) ; ROUND Total TOLL
mw[6] = ROUND(MW[6]) ; SKIMS TO 2007
mw[9] = ROUND(MW[9]) ; WHOLE CENTS
;
mw[13] = ROUND(MW[13]) ; ROUND Variable priced TOLL
mw[16] = ROUND(MW[16]) ; SKIMS TO 2007
mw[19] = ROUND(MW[19]) ; WHOLE CENTS
;
;
;-----
; Print selected rows of skim files
; for checking.
;-----
IF (i = 1-2) ; for select rows (Is)
printrow MW=1-3, j=1-3722 ; print work matrices 1-3
ENDIF ; row value to all Js.
ENDPHASE
ENDRUN
;
;-----
; Finally create special Mode Choice skims here
; The mode choice skims will be the same as the above skims unless VP toll lanes

```

## Appendix C Cube Voyager Scripts

```

; are used; in that case time will include the VP toll time equivalent
; and the toll value will be the toll on non-VP toll lanes ONLY
;
; Also create zonal truck access file per the @TT@ statements for the OP per. only
;-----
RUN PGM=MATRIX

  READ FILE = @in_tmin@      ; read toll time eqv param file
                             ; -- INPUT SKIMS --
  MATI[1] = @MATOUT1@      ; SOV skims (tm,dst,total
toll, VP toll)
  MATI[2] = @MATOUT2@      ; HOV2 skims (tm,dst,total
toll, VP toll)
  MATI[3] = @MATOUT3@      ; HOV3+skims (tm,dst,total
toll, VP toll)

  @TT@ MATI[4] = @MATOUT4@  ; read in trk skim (op per
only)
  @TT@ MW[99] = MI.4.1
; For the skim total, put a large value in unconnected O/D pairs
  @TT@ JLOOP
  @TT@ IF (MW[99] = 0) MW[99] = 100000
  @TT@ ENDJLOOP
  @TT@ REPORT MARGINREC = Y, FILE = @SKMTOT@, FORM=15, LIST=J(5),R99,C99

  MATO[1] = @MATOUTMC1@,MO=101,12,103 ; SOV skims (tm&toll tm eqv,dst,non-VP
toll component)
  MATO[2] = @MATOUTMC2@,MO=201,22,203 ; HOV2 skims (tm&toll tm eqv,dst,non-VP
toll component)
  MATO[3] = @MATOUTMC3@,MO=301,32,303 ; HOV3+skims (tm&toll tm eqv,dst,non-VP
toll component)

; read in input skims from above
  MW[11] = MI.1.1 ; SOV time
  MW[12] = MI.1.2 ; SOV distance
  MW[13] = MI.1.3 ; SOV total toll
  MW[14] = MI.1.4 ; SOV Var.priced toll component (if VP toll facility used)

  MW[21] = MI.2.1 ; HOV2 time
  MW[22] = MI.2.2 ; HOV2 distance
  MW[23] = MI.2.3 ; HOV2 total toll
  MW[24] = MI.2.4 ; HOV2 Var.priced toll component (if VP toll facility used)

  MW[31] = MI.3.1 ; HOV3+ time
  MW[32] = MI.3.2 ; HOV3+ distance
  MW[33] = MI.3.3 ; HOV3+ total toll
  MW[34] = MI.3.4 ; HOV3+ Var.priced toll component (if VP toll facility used)

; now compute special time and toll values to be used in the mode choice process
; which are normally 1/time, 2/distance, and 3/tolls; the new skims will be:
; 1/ time + the toll time equivalent on VP facilities only
; 2/ distance (as before)
; 3/ tolls on non-VP tolled facilities ONLY

;Mode Choice model Hwy time:
  MW[101] = MW[11] + ((MW[14]/100.0) * SV@PRD@EQM);
  MW[201] = MW[21] + ((MW[24]/100.0) * H2@PRD@EQM);
  MW[301] = MW[31] + ((MW[34]/100.0) * H3@PRD@EQM);

;Mode Choice model Hwy TOLL:
  MW[103] = MW[13] - MW[14]
  MW[203] = MW[23] - MW[24]
  MW[303] = MW[33] - MW[34]

  MW[103] = MAX(0,MW[103])

```

```

  MW[203] = MAX(0,MW[203])
  MW[303] = MAX(0,MW[303])
ENDRUN

; *ping -n 11 127.0.0.1 > nul

; end of truck access section
ENDLOOP

```

## 15 Highway\_Skims\_mod\_am.s

```

;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////;
; Highway_Skims_Mod_am.S //;
; MWCOCG Version 2.3 Model //;
; //;
; Build AM Peak/Midday Highway Skims //;
; the Current Iteration Assignment //;
; AM and Midday Skims are built in 2 separate HWYLOAD //;
; programs. //;
; Three files are created, per SOV, HOV2, and HOV3 paths.//;
; Each file will contain 3 Tables (in Voyager fmt, 2 dec.//;
; 1) Time (whole minutes) //;
; 2) Distance (implied tenths of mi.) //;
; 3) Toll (in 2007 cents) //;
; //;
; 6/30/03 MODIFICATIONS FOR IMPROVED TOLL MODELING MADE rjm
; 2/14/08 generalized toll skimming changed to mode specific skimming
; (See HIGHWAY_SKIMS.S change made on 1/31/08)
; 6/25/10 max zones increased to 7000 per V2.3
; 4/16/11 max zones increased to 7999
; 4/16/11 'PRD=' corrected for period 1 (Set to 'AM' instead of 'MD')
;
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to the
impedance calculation
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to both
impedance * time skim calculation
;
;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////;
;
; Environment Variables:
; _iter_ (Iteration indicator = 'pp','i1'-'i4')
;
;
; pageheight=32767 ; Preclude header breaks
NETIN = '%_iter_%_hwymod.net'

;*****
;*** AM "loop"
;*****

LOOP Period=1,1 ; We are looping through the skimming process
; twice: (1) for the AM Peak & (2) the Off-Peak

in_tmin = '..\support\toll_minutes.txt' ; read in toll minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf' ; AM Toll Factors
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf' ; MD Toll Factors
by Veh. Type

IF (Period=1) ; AM Highway Skim tokens
  PRD = 'AM'
  MATOUT1 = '%_iter_%_am_sov_mod.skm'
  MATOUT2 = '%_iter_%_am_hov2_mod.skm'

```

## Appendix C Cube Voyager Scripts

```

MATOUT3 = '%_iter_%_am_hov3_mod.skm'
MYID = '%_iter_%_AM_skims'
ELSE
    ; MD Highway Skim tokens
    PRD = 'MD'
    MATOUT1 = '%_iter_%_md_sov_mod.skm'
    MATOUT2 = '%_iter_%_md_hov2_mod.skm'
    MATOUT3 = '%_iter_%_md_hov3_mod.skm'
    MYID = '%_iter_%_MD_skims'
ENDIF

RUN PGM=HIGHWAY
zones=7999
;
;
NETI =@NETIN@ ; Pk Prd TP+ network
MATO[1]=@MATOUT1@, MO=1-3;;, LOV skims
MATO[2]=@MATOUT2@, MO=4-6;;, HOV2 skims
MATO[3]=@MATOUT3@, MO=7-9;;, HOV3+ skims
ID=@MYID@
;
READ FILE = @in_tmin@

FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@in_MDTfac@"
LOOKUP LOOKUPI=2, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;
;
PHASE=LINKREAD
SPEED = LI.%_iter_%@PRD@SPD ;Restrained speed (min)
IF (SPEED = 0)
    T1 = 0
ELSE
    T1 = (LI.DISTANCE / SPEED * 60.0) + LI.timepen
ENDIF
;
; Define AM /MD link level tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV
TOLLS in 2007 cents
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2
occ TOLLS in 2007 cents
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV
3+occ TOLLS in 2007 cents
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; Truck
TOLLS in 2007 cents
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; AP Pax
TOLLS in 2007 cents

; Define AM /MD IMPEDANCE by vehicle type here:
LW.SOV@PRD@IMP= T1 + ((LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM);SOV IMP
LW.HV2@PRD@IMP= T1 + ((LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM);HOV 2 IMP
LW.HV3@PRD@IMP= T1 + ((LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM);HOV 3+IMP

```

```

LW.TRK@PRD@IMP= T1 + ((LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM);Truck IMP
LW.APX@PRD@IMP= T1 + ((LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM);APAX IMP
;
; Define the three path types here:
;
;
; limit codes used:
; 1=no prohibitions
; 2=prohibit 1/occ autos,trucks
; 3=prohibit 1&2occ autos,trucks
; 4=prohibit trucks
; 5=prohibit non-airport access trips
; 6-8=unused
; 9=prohibit all traffic use

IF (LI.@PRD@LIMIT = 2,3,5-9) ADDTOGROUP=1 ; SOV prohibited links
IF (LI.@PRD@LIMIT = 3,5-9) ADDTOGROUP=2 ; HOV2 prohibited links
IF (LI.@PRD@LIMIT = 5-9) ADDTOGROUP=3 ; HOV3+ prohibited links
;
ENDPHASE
;
; Now do the path skimming, per the three path types. Time, distance,
; and Toll skims created. Scaling to the desired specified below.
; All skims are based on minimum time paths.
;
; Note that override values of 0 will be inserted for disconnected ijs
; (i.e. cells associated with 'unused' zones and intrazonal cells).
; I don't like the TP+ default value of 1,000,000 for these situations
;
PHASE=ILOOP

PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGRP=1, ; SOV paths
MW[1]=PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[2]=PATHTRACE(DIST), NOACCESS=0, ; w/ LIMIT=2,3,5-9
MW[3]=PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS=0;
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGRP=2, ; HOV2 paths
MW[4]=PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[5]=PATHTRACE(DIST), NOACCESS=0, ; w/ LIMIT=3,5-9
MW[6]=PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS=0;
PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGRP=3, ; HOV3+ paths
MW[7]=PATHTRACE(TIME), NOACCESS=0, ; -excluding links
MW[8]=PATHTRACE(DIST), NOACCESS=0, ; w/ LIMIT=5-9
MW[9]=PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS=0;

;-----
; scaling, rounding of skim tables done here!!
;-----

mw[1] = ROUND(MW[1]) ; ROUND TIME SKIMS
mw[4] = ROUND(MW[4]) ; TO WHOLE MINUTES
mw[7] = ROUND(MW[7]) ;
mw[1] = MIN(MW[1],326.0) ; Impose Max TIME
mw[4] = MIN(MW[4],326.0) ; Impose Max TIME
mw[7] = MIN(MW[7],326.0) ; Impose Max TIME
; ...just in case

mw[2] = ROUND(MW[2]*10) ; FACTOR/ROUND DIST.
mw[5] = ROUND(MW[5]*10) ; SKIMS TO IMPLICIT
mw[8] = ROUND(MW[8]*10) ; 1/10THS OF MILES

mw[3] = ROUND(MW[3]) ; ROUND TOLL
mw[6] = ROUND(MW[6]) ; SKIMS TO 2007
mw[9] = ROUND(MW[9]) ; WHOLE CENTS

;-----
; Print selected rows of skim files
; for checking.
;-----

```

## Appendix C Cube Voyager Scripts

```

IF (i = 1-2) ; for select rows (Is)
  printrow MW=1-3, j=1-7999 ; print work matrices 1-3
ENDIF ; row value to all Js.
ENDPHASE
ENDRUN
ENDLOOP

```

## 16 Highway\_Skims\_mod\_md.s

```

//////////////////////////////////////
; Highway_Skims_Mod_md.S //
; MWCOG Version 2.3 Model //
; //
; Build AM Peak/Midday Highway Skims //
; the Current Iteration Assignment //
; AM and Midday Skims are built in 2 separate HWYLOAD //
; programs. //
; Three files are created, per SOV, HOV2, and HOV3 paths.//
; Each file will contain 3 Tables (in Voyager fmt, 2 dec.//
; 1) Time (whole minutes) //
; 2) Distance (implied tenths of mi.) //
; 3) Toll (in 2007 cents) //
;
; 6/30/03 MODIFICATIONS FOR IMPROVED TOLL MODELING MADE rjm
; 2/14/08 generalized toll skimming changed to mode specific skimming
; (See HIGHWAY_SKIMS.S change made on 1/31/08)
; 6/25/10 max zones increased to 7000 per V2.3
; 4/16/11 max zones increased to 7999
; 4/16/11 'PRD=' corrected for period 1 (Set to 'AM' instead of 'MD')
;
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to the
impedance calculation
; 02/22/13 Added 'timepen' (now a link attribute in the highway net) to both
impedance * time skim calculation
//////////////////////////////////////
;
; Environment Variables:
; _iter_ (Iteration indicator = 'pp','i1'-'i4')
;
; pageheight=32767 ; Preclude header breaks
NETIN = '%_iter_%_hwymod.net'
;
;*****
;*** Midday "loop"
;*****
LOOP Period=2,2 ; We are looping through the skimming process
; twice: (1) for the AM Peak & (2) the Off-Peak
;
in_tmin = '..\support\toll_minutes.txt' ; read in toll minutes equiv file
in_AMTfac = 'inputs\AM_Tfac.dbf' ; AM Toll Factors
by Veh. Type
in_MDTfac = 'inputs\MD_Tfac.dbf' ; MD Toll Factors
by Veh. Type
;
IF (Period=1) ; AM Highway Skim tokens
  PRD = 'AM'
  MATOUT1 = '%_iter_%_am_sov_mod.skm'
  MATOUT2 = '%_iter_%_am_hov2_mod.skm'

```

```

MATOUT3 = '%_iter_%_am_hov3_mod.skm'
MYID = '%_iter_%_AM_skims'
ELSE ; MD Highway Skim tokens
  PRD = 'MD'
  MATOUT1 = '%_iter_%_md_sov_mod.skm'
  MATOUT2 = '%_iter_%_md_hov2_mod.skm'
  MATOUT3 = '%_iter_%_md_hov3_mod.skm'
  MYID = '%_iter_%_MD_skims'
ENDIF

```

```

RUN PGM=HIGHWAY
zones=7999
;
;
NETI =@NETIN@ ; Pk Prd TP+ network
MATO[1]=MATOUT1@, MO=1-3;;, LOV skims
MATO[2]=MATOUT2@, MO=4-6;;, HOV2 skims
MATO[3]=MATOUT3@, MO=7-9;;, HOV3+ skims
ID=@MYID@
;-
READ FILE = @in_tmin@
;
FileI LOOKUPI[1] = "@in_AMtfac@"
LOOKUP LOOKUPI=1, NAME=AM_Tfac,
LOOKUP[1]= TOLLGrp, result=AMSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=AMHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=AMHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=AMTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=AMAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;
FileI LOOKUPI[2] = "@in_MDTfac@"
LOOKUP LOOKUPI=2, NAME=MD_Tfac,
LOOKUP[1]= TOLLGrp, result=MDSOVTFTR, ;
LOOKUP[2]= TOLLGrp, result=MDHV2TFTR, ;
LOOKUP[3]= TOLLGrp, result=MDHV3TFTR, ;
LOOKUP[4]= TOLLGrp, result=MDTRKTFTR, ;
LOOKUP[5]= TOLLGrp, result=MDAPXTFTR, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;-
;
PHASE=LINKREAD
SPEED = LI.%_iter_%@PRD@SPD ;Restrained speed (min)
IF (SPEED = 0)
  T1 = 0
ELSE
  T1 = (LI.DISTANCE / SPEED * 60.0) + LI.timepen
ENDIF
;-
; Define AM /MD link level tolls by vehicle type here:
LW.SOV@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(1,LI.TOLLGRP) ; SOV
TOLLS in 2007 cents
LW.HV2@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(2,LI.TOLLGRP) ; HOV 2
occ TOLLS in 2007 cents
LW.HV3@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(3,LI.TOLLGRP) ; HOV
3+occ TOLLS in 2007 cents
LW.TRK@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(4,LI.TOLLGRP) ; Truck
TOLLS in 2007 cents
LW.APX@PRD@TOLL = LI.@PRD@TOLL * @PRD@_TFAC(5,LI.TOLLGRP) ; AP Pax
TOLLS in 2007 cents
; Define AM /MD IMPEDANCE by vehicle type here:
LW.SOV@PRD@IMP= T1 + ((LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM);SOV IMP
LW.HV2@PRD@IMP= T1 + ((LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM);HOV 2 IMP
LW.HV3@PRD@IMP= T1 + ((LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM);HOV 3+IMP

```

## Appendix C Cube Voyager Scripts

```

LW.TRK@PRD@IMP= T1 + ((LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM);Truck IMP
LW.APX@PRD@IMP= T1 + ((LW.APX@PRD@TOLL/100.0)* AP@PRD@EQM);APAX IMP

;
; Define the three path types here:
;
;
; limit codes used:
; 1=no prohibitions
; 2=prohibit 1/occ autos,trucks
; 3=prohibit 1&2occ autos,trucks
; 4=prohibit trucks
; 5=prohibit non-airport access trips
; 6-8=unused
; 9=prohibit all traffic use

IF (LI.@PRD@LIMIT = 2,3,5-9) ADDTOGROUP=1 ; SOV prohibited links
IF (LI.@PRD@LIMIT = 3,5-9) ADDTOGROUP=2 ; HOV2 prohibited links
IF (LI.@PRD@LIMIT = 5-9) ADDTOGROUP=3 ; HOV3+ prohibited links
;
ENDPHASE
;
; Now do the path skimming, per the three path types. Time, distance,
; and Toll skims created. Scaling to the desired specified below.
; All skims are based on minimum time paths.
;
; Note that override values of 0 will be inserted for disconnected ijs
; (i.e. cells associated with 'unused' zones and intrazonal cells).
; I don't like the TP+ default value of 1,000,000 for these situations
;
PHASE=ILOOP

PATHLOAD PATH=LW.SOV@PRD@IMP, EXCLUDEGRP=1, ; SOV paths
MW[1]=PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[2]=PATHTRACE (DIST), NOACCESS=0, ; w/ LIMIT=2,3,5-9
MW[3]=PATHTRACE (LW.SOV@PRD@TOLL), NOACCESS=0 ;
PATHLOAD PATH=LW.HV2@PRD@IMP, EXCLUDEGRP=2, ; HOV2 paths
MW[4]=PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[5]=PATHTRACE (DIST), NOACCESS=0, ; w/ LIMIT=3,5-9
MW[6]=PATHTRACE (LW.HV2@PRD@TOLL), NOACCESS=0 ;
PATHLOAD PATH=LW.HV3@PRD@IMP, EXCLUDEGRP=3, ; HOV3+ paths
MW[7]=PATHTRACE (TIME), NOACCESS=0, ; -excluding links
MW[8]=PATHTRACE (DIST), NOACCESS=0, ; w/ LIMIT=5-9
MW[9]=PATHTRACE (LW.HV3@PRD@TOLL), NOACCESS=0 ;

;-----
; scaling, rounding of skim tables done here!!
;-----

mw[1] = ROUND(MW[1]) ; ROUND TIME SKIMS
mw[4] = ROUND(MW[4]) ; TO WHOLE MINUTES
mw[7] = ROUND(MW[7]) ;
mw[1] = MIN(MW[1],326.0) ; Impose Max TIME
mw[4] = MIN(MW[4],326.0) ; Impose Max TIME
mw[7] = MIN(MW[7],326.0) ; Impose Max TIME
mw[7] = MIN(MW[7],326.0) ; ...just in case

mw[2] = ROUND(MW[2]*10) ; FACTOR/ROUND DIST.
mw[5] = ROUND(MW[5]*10) ; SKIMS TO IMPLICIT
mw[8] = ROUND(MW[8]*10) ; 1/10THS OF MILES

mw[3] = ROUND(MW[3]) ; ROUND TOLL
mw[6] = ROUND(MW[6]) ; SKIMS TO 2007
mw[9] = ROUND(MW[9]) ; WHOLE CENTS

;-----
; Print selected rows of skim files
; for checking.
;-----

```

```

IF (i = 1-2) ; for select rows (Is)
printrow MW=1-3, j=1-7999 ; print work matrices 1-3
ENDIF ; row value to all Js.
ENDPHASE
ENDRUN
ENDLOOP

```

## 17 joinskims.s

```

; JoinSkims.S - Consolidate highway skims used in Mode Choice Model
; Input skims: ???_iter_@PRD@.skm
; Changed to: ???_iter_@PRD@_MC.skm
; The revised skim reflect
; time (min) + time (min) equivalent of any Variably Priced facility toll such
as ICC/VA Hot lanes
; distance (1/10s of mi),
; tolls (2007 cts) of any FIXED price facility, such as Dulles toll road.
;

pageheight=32767 ; Preclude header breaks

RUN PGM=MATRIX
MATI[1]= %_iter_%_am_sov_MC.skm
MATI[2]= %_iter_%_am_hov2_MC.skm
MATI[3]= %_iter_%_am_hov3_MC.skm

MATI[4]= %_iter_%_md_sov_MC.skm
MATI[5]= %_iter_%_md_hov2_MC.skm
MATI[6]= %_iter_%_md_hov3_MC.skm

FILLMW MW[1] = MI.1.1,2,3
FILLMW MW[4] = MI.2.1,2,3
FILLMW MW[7] = MI.3.1,2,3

FILLMW MW[10] = MI.4.1,2,3
FILLMW MW[13] = MI.5.1,2,3
FILLMW MW[16] = MI.6.1,2,3

MATO[1] = %_iter_%_hwy_am.skm, MO=1-9,
name=SovTime,SOVDst10,SOVToll,
Hv2Time,Hv2Dst10,HV2Toll,
Hv3Time,Hv3Dst10,HV3Toll

MATO[2] = %_iter_%_hwy_op.skm, MO=10-18,
name=SovTime,SOVDst10,SOVToll,
Hv2Time,Hv2Dst10,HV2Toll,
Hv3Time,Hv3Dst10,HV3Toll

ENDRUN

```





## Appendix C Cube Voyager Scripts

```

; Apply Modeled Shares to the Ext1 Auto Drivers in mtx 51-54

MW[51] = (MW[21] * MW[40]) ; Estimated Ext1 1 occ vehicles
MW[52] = (MW[22] * MW[40]) ; Estimated Ext1 2 occ vehicles
MW[53] = (MW[23] * MW[40]) ; Estimated Ext1 3 occ vehicles
MW[54] = (MW[24] * MW[40]) ; Estimated Ext1 4+occ vehicles

; compute add intl and ext1 auto drivers by occ. groups together
; in mtx 61,62,63. Total adrs will be in mtx 70

MW[61] = MW[51] + MW[11] ; Total 1-Occ Total Auto Drivers
MW[62] = MW[52] + MW[12] ; 2-occ
MW[63] = MW[53] + MW[54] + MW[13] ; 3+occ

MW[70] = mw[61] + MW[62] + MW[63]
;
endjloop

JLOOP
; Lets sum up the above to get neat total summaries
Int1_OccAPsn = Int1_OccAPsn + MW[1] ;
Int2_OccAPsn = Int2_OccAPsn + MW[2] ;
Int3POccAPsn = Int3POccAPsn + MW[3] ;
IntAutoPsn = IntAutoPsn + MW[1] + MW[2] + MW[3] ;
;
Int1_OccADrv = Int1_OccADrv + MW[11] ;
Int2_OccADrv = Int2_OccADrv + MW[12] ;
Int3POccADrv = Int3POccADrv + MW[13] ;
IntAutoDrv = IntAutoDrv + MW[11] + MW[12] + MW[13] ;
;
TotalMotorPsn = TotalMotorPsn + MW[20] ;
ExtAutoPsn = ExtAutoPsn + MW[30] ;
ExtAutoDrv = ExtAutoDrv + MW[40] ;
Ext1_OccADrv = Ext1_OccADrv + MW[51] ;
Ext2_OccADrv = Ext2_OccADrv + MW[52] ;
Ext3_OccADrv = Ext3_OccADrv + MW[53] ;
Ext4POccADrv = Ext4POccADrv + MW[54] ;
ExtchkADrv = ExtchkADrv + MW[51] + MW[52] + MW[53] + MW[54] ;
;
Tot1_OccADrv = Tot1_OccADrv + MW[61] ;
Tot2_OccADrv = Tot2_OccADrv + MW[62] ;
Tot3POccADrv = Tot3POccADrv + MW[63] ;
;
TotalAutoDrv = TotalAutoDrv + MW[70] ;
endjloop

IF (I == ZONES)
;
Print LIST='/bt '
LIST='SUMMARY OF ','@PURPOSE@',' ITERATION: ','%_iter_%',' AUTO DRIVER TRIP
RESULTS'
LIST=' '
Print form = 12.2 LIST=' Assumed Avg 3+Veh. Occ.: ','@Avg3P_Occ@
Print form = 12.2 LIST=' Assumed Ext1 Veh Occ. : ','@ExtCarOcc@
LIST=' '
List=' Input Internal Auto Persons '
Print form = 12.0csv List=' 1-Occ.: ', Int1_OccAPsn

```

```

Print form = 12.0csv List=' 2-Occ.: ', Int2_OccAPsn
Print form = 12.0csv List=' 3+Occ.: ', Int3POccAPsn
List=' ----- '
List=' Total ', IntAutoPsn
List=' '
List=' Input / Derived Internal Auto Drivers '
Print form = 12.0csv List=' 1-Occ.: ', Int1_OccADrv
Print form = 12.0csv List=' 2-Occ.: ', Int2_OccADrv
Print form = 12.0csv List=' 3+Occ.: ', Int3POccADrv
List=' ----- '
Print form = 12.0csv List=' Total ', IntAutoDrv
List=' '
Print form = 12.0csv List=' Input Total Motorized Person ', TotalMotorPsn
List=' '
Print form = 12.0csv List=' Input Total External Auto Psn ', ExtAutoPsn
List=' '
Print form = 12.0csv List=' Input/Derived External Auto Drv ', ExtAutoDrv
List=' '
List=' Estimated External Auto Drivers '
Print form = 12.0csv List=' 1-Occ.: ', Ext1_OccADrv
Print form = 12.0csv List=' 2-Occ.: ', Ext2_OccADrv
Print form = 12.0csv List=' 3-Occ.: ', Ext3_OccADrv
Print form = 12.0csv List=' 4+Occ.: ', Ext4POccADrv
List=' ----- '
Print form = 12.0csv List=' Total ', ExtchkADrv
List=' '
List=' Output / Combined Internal/External Auto Drivers '
Print form = 12.0csv List=' 1-Occ.: ', Tot1_OccADrv
Print form = 12.0csv List=' 2-Occ.: ', Tot2_OccADrv
Print form = 12.0csv List=' 3+Occ.: ', Tot3POccADrv
List=' ----- '
Print form = 12.0csv LIST=' Total ', TotalAutoDrv
LIST='==== END OF ','@PURPOSE@',' ITERATION: ','%_iter_%',' AUTO DRV RESULTS ==='
LIST='/et '
ENDIF

MATO=@MC1230CC@,MO=61,62,63 ; output file designation

ENDRUN
ENDLOOP

```

## 19 MC\_Constraint\_V23.s

```

; MC_Constraint.s
; 4/3/12 - Three (3) decimals added to all MATO files intermediate and final
;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
; Mode choice constraint process per the Version 2.3 model
;
; Metrorail trips to/through the core are constrained to a predetermined level
; current year for which constrained levels are set: 2020
; current years for which the constrained levels are imposed: 2030, 2040
;//////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
;
; 4/3/12: Three(3) decimals are maintained in output files
; =====
;
;
; Transit Constraint Process -Applied to modeled mode choice output
; file for forecast years beyond the constraint year.
; The process constrains peak period Metrorail trips heading
; TO or THROUGH the regional core to be constrained
; levels and converts "excess" back to auto person trips.
;
; The process consists of 3 Steps:

```

## Appendix C Cube Voyager Scripts

```

; Step 1. constraint year & post-constraint year peak/off-peak transit trips are
calculated
;       for each purpose using 2007/08 HTS time period factors.
;       (2 Loops for constr./unconstr. mode choice output files)
;
; Step 2. constraint year & post-constraint year peak & total transit trips are
squeezed to
;       a 3x3 (core/va/dc,md). Factors for scaling unconstrained
;       transit trips to constrained transit trips are computed, on
;       an i/j basis. A 'lookup' of constraint factors is produced.
;
; Step 3. Future year constrained zonal trips are computed by applying
;       the constraint factors to the zonal trip tables.
;       The factors yeild constrained transit trips at the zone level.
;       The excess transit trips are converted to auto person trips.
;
;       (5 Loops for each purpose as per the V2.3 model)
;
;-----
; Step 1.
;       Constrained year & future year peak/off-peak transit trips are calculated
;       for each purpose using 2007/08 HTS time period factors.
;-----

TODFile = '..\support\todcomp_2008HTS.dbf' ; Time of Day Factor File

;; define TOD ARRAY parameters in the time of day file
Pur = 5 ; 1/HBW, 2/HBS, 3/HBO, 4/NHW, 5/NHO
Mod = 4 ; 1/Adr, 2/DrAlone 3/CarPoolPsn 4/Transit
Dir = 2 ; 1/H>NH, 2/NH>H
Per = 4 ; 1/AM, 2/MD, 3/PM, 4/NT

LOOP Time = 1, 2 ; Time '1' = constraint year loop/ Time '2' = Future
year loop

IF (Time = 1)
  PATHSPECBW = '%_tcHBW_%' ; path specification of constraint-year HBW transit
trips
  PATHSPECBHS = '%_tcHBS_%' ; path specification of constraint-year HBS transit
trips
  PATHSPECBHO = '%_tcHBO_%' ; path specification of constraint-year HBO transit
trips
  PATHSPECNHW = '%_tcNHW_%' ; path specification of constraint-year NHW transit
trips
  PATHSPECNHO = '%_tcNHO_%' ; path specification of constraint-year NHO transit
trips
  YR = 'con' ; constraint indicator (for file naming)
  title = 'Constraint-Year Metrorail Trip Summary by Time Period '
ELSE
  PATHSPECBW = '%_iter_%_HBW_NL_MC.MTT' ; forecast year should be in current
subdir
  PATHSPECBHS = '%_iter_%_HBS_NL_MC.MTT' ; forecast year should be in current
subdir
  PATHSPECBHO = '%_iter_%_HBO_NL_MC.MTT' ; forecast year should be in current
subdir
  PATHSPECNHW = '%_iter_%_NHW_NL_MC.MTT' ; forecast year should be in current
subdir
  PATHSPECNHO = '%_iter_%_NHO_NL_MC.MTT' ; forecast year should be in current
subdir
  YR = 'ucn' ; unconstrained indicator (for file naming)
  title = 'Future/Post-Constraint Year - UnConstrained Metrorail Trip Summary by
Time Period '

ENDIF

;
; Factors for distributing Daily Transit Trips

```

```

; (HBW,HBS,HBO,NHW,NHO) Among 4 Time Periods:
;
;       - AM peak (6:00 AM - 9:00 AM)
;       - Midday (9:00 AM - 3:00 PM)
;       - PM peak (3:00 PM - 7:00 PM)
;       - Night (All Other hrs )
;
;//////////////////////////////////////
;
; Begin Voyager Step 1
;
;//////////////////////////////////////

RUN PGM=MATRIX
pageheight=32767 ; Preclude header breaks
ZONES=3722
Array TODFtrs =@Pur@,@Mod@,@Dir@,@Per@
FILEI DBI[1] = "@TODFile@"
;=====
;=====
IF (I=1)
; Read in Time of Day factor file and populate TOD factor array
  LOOP K = 1,dbi.1.NUMRECORDS ;;PURP MODE DIR AM MD PM
OP
      x = DBIReadRecord(1,k)
      count = dbi.1.recno
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][1] = di.1.AM
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][2] = di.1.MD
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][3] = di.1.PM
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][4] = di.1.OP
      print list = count, di.1.am, di.1.md, di.1.pm, di.1.op, file = tod.chk
  ENDLLOOP
ENDIF
;=====
;=====

; Read input NL Mode Choice Model Output files to pull out Metrorail trips
MATI[1] = @PATHSPECBW@ ; HBW Metrorail Only, Bus/Metrorail (T6-7,11-14) Trips
MATI[2] = @PATHSPECBHS@ ; HBS Metrorail Only, Bus/Metrorail (T6-7,11-14) Trips
MATI[3] = @PATHSPECBHO@ ; HBO Metrorail Only, Bus/Metrorail (T6-7,11-14) Trips
MATI[4] = @PATHSPECNHW@ ; NBW Metrorail Only, Bus/Metrorail (T6-7,11-14) Trips
MATI[5] = @PATHSPECNHO@ ; NBO Metrorail Only, Bus/Metrorail (T6-7,11-14) Trips

; Specify output Pk, Offpk Total Metrorail trips (t1-3) by purpose
MATO[1] = HBWPKOPALL.@yr@, MO=121,122,120, dec=3*3 ;HBW Pk(AM,PM), Off-Pk(MD,NT),
total(AM,PM,MD,NT) Transit Trips
MATO[2] = HBSPKOPALL.@yr@, MO=221,222,220, dec=3*3 ;HBS Pk(AM,PM), Off-Pk(MD,NT),
total(AM,PM,MD,NT) Transit Trips
MATO[3] = HBOPKOPALL.@yr@, MO=321,322,320, dec=3*3 ;HBO Pk(AM,PM), Off-Pk(MD,NT),
total(AM,PM,MD,NT) Transit Trips
MATO[4] = NHWPKOPALL.@yr@, MO=421,422,420, dec=3*3 ;NHW Pk(AM,PM), Off-Pk(MD,NT),
total(AM,PM,MD,NT) Transit Trips
MATO[5] = NHOPOKOPALL.@yr@, MO=521,522,520, dec=3*3 ;NHO Pk(AM,PM), Off-Pk(MD,NT),
total(AM,PM,MD,NT) Transit Trips

;
; Put Total HBW - NHO Metrorail Transit Trips in MWS 101-105
; These are in P/A format and represent the Home-to-NonHome direction

```

# Appendix C Cube Voyager Scripts

```

MW[101] = MI.1.6 + MI.1.7 + MI.1.11 + MI.1.12 + MI.1.13 + MI.1.14 ; Work
Metrorail P/A fmt
MW[102] = MI.2.6 + MI.2.7 + MI.2.11 + MI.2.12 + MI.2.13 + MI.2.14 ; Shop
Metrorail P/A fmt
MW[103] = MI.3.6 + MI.3.7 + MI.3.11 + MI.3.12 + MI.3.13 + MI.3.14 ; Other
Metrorail P/A fmt
MW[104] = MI.4.6 + MI.4.7 + MI.4.11 + MI.4.12 + MI.4.13 + MI.4.14 ; NHWrk
Metrorail P/A fmt
MW[105] = MI.5.6 + MI.5.7 + MI.5.11 + MI.5.12 + MI.5.13 + MI.5.14 ; NHOTH
Metrorail P/A fmt

; develop xpose of the above input tables
; then add xposed tabs to developed total Metrorail tabs in A/P format (MWs 201-
205)
; The transpose represents the NonHome-to-Home direction

MW[11]= MI.1.6.T MW[12]= MI.1.7.T MW[13]= MI.1.11.T MW[14]= MI.1.12.T
MW[15]= MI.1.13.T MW[16]= MI.1.14.T ;Work Metrorail A/P fmt
MW[21]= MI.2.6.T MW[22]= MI.2.7.T MW[23]= MI.2.11.T MW[24]= MI.2.12.T
MW[25]= MI.2.13.T MW[26]= MI.2.14.T ;Shop Metrorail A/P fmt
MW[31]= MI.3.6.T MW[32]= MI.3.7.T MW[33]= MI.3.11.T MW[34]= MI.3.12.T
MW[35]= MI.3.13.T MW[36]= MI.3.14.T ;Other Metrorail A/P fmt
MW[41]= MI.4.6.T MW[42]= MI.4.7.T MW[43]= MI.4.11.T MW[44]= MI.4.12.T
MW[45]= MI.4.13.T MW[46]= MI.4.14.T ;NHWrk Metrorail A/P fmt
MW[51]= MI.5.6.T MW[52]= MI.5.7.T MW[53]= MI.5.11.T MW[54]= MI.5.12.T
MW[55]= MI.5.13.T MW[56]= MI.5.14.T ;NHOTH Metrorail A/P fmt

MW[201]=MW[11] + MW[12] + MW[13] + MW[14] + MW[15] + MW[16] ;
Work total Metrorail A/P fmt
MW[202]=MW[21] + MW[22] + MW[23] + MW[24] + MW[25] + MW[26] ;
Shop total Metrorail A/P fmt
MW[203]=MW[31] + MW[32] + MW[33] + MW[34] + MW[35] + MW[36] ;
Other total Metrorail A/P fmt
MW[204]=MW[41] + MW[42] + MW[43] + MW[44] + MW[45] + MW[46] ;
NHWrk total Metrorail A/P fmt
MW[205]=MW[51] + MW[52] + MW[53] + MW[54] + MW[55] + MW[56] ;
NHOTH total Metrorail A/P fmt
;
; Now we're ready to apply apply TOD factors
;
;
JLOOP

;//////////////////////////////////////
;////////////////////////////////////// AM Trip Calculations - MWs 111-115 ////////////////////////////////////////
;//////////////////////////////////////
;
; 24Hr Trips p m d p 24 Hr Trips p m d p
; in u o i e in u o i e
; H-NH Dir r d r r NH-H Dir r d r r
; | | | | | | | |
MW[111] = (MW[101] * (TODFtrs[1][4][1][1]/100.00) + MW[201] *
(TODFtrs[1][4][2][1]/100.00)) / 2.0 ; AM Period HBW Metrorail Trips *****
MW[112] = (MW[102] * (TODFtrs[2][4][1][1]/100.00) + MW[202] *
(TODFtrs[2][4][2][1]/100.00)) / 2.0 ; AM Period HBS Metrorail Trips *****
MW[113] = (MW[103] * (TODFtrs[3][4][1][1]/100.00) + MW[203] *
(TODFtrs[3][4][2][1]/100.00)) / 2.0 ; AM Period HBO Metrorail Trips *****
MW[114] = (MW[104] * (TODFtrs[4][4][1][1]/100.00) + MW[204] *
(TODFtrs[4][4][2][1]/100.00)) / 2.0 ; AM Period NHW Metrorail Trips *****
MW[115] = (MW[105] * (TODFtrs[5][4][1][1]/100.00) + MW[205] *
(TODFtrs[5][4][2][1]/100.00)) / 2.0 ; AM Period NHO Metrorail Trips *****
;
;//////////////////////////////////////
;////////////////////////////////////// MD Trip Calculations - MWs 211-215 ////////////////////////////////////////
;//////////////////////////////////////
;
; Trips p m d p Trips p m d p
; in u o i e in u o i e
; H-NH Dir r d r r NH-H Dir r d r r
; | | | | | | | |
MW[211] = (MW[101] * (TODFtrs[1][4][1][2]/100.00) + MW[201] *
(TODFtrs[1][4][2][2]/100.00)) / 2.0 ; MD Period HBW Metrorail Trips *****
MW[212] = (MW[102] * (TODFtrs[2][4][1][2]/100.00) + MW[202] *
(TODFtrs[2][4][2][2]/100.00)) / 2.0 ; MD Period HBS Metrorail Trips *****
MW[213] = (MW[103] * (TODFtrs[3][4][1][2]/100.00) + MW[203] *
(TODFtrs[3][4][2][2]/100.00)) / 2.0 ; MD Period HBO Metrorail Trips *****
MW[214] = (MW[104] * (TODFtrs[4][4][1][2]/100.00) + MW[204] *
(TODFtrs[4][4][2][2]/100.00)) / 2.0 ; MD Period NHW Metrorail Trips *****
MW[215] = (MW[105] * (TODFtrs[5][4][1][2]/100.00) + MW[205] *
(TODFtrs[5][4][2][2]/100.00)) / 2.0 ; MD Period NHO Metrorail Trips *****
;
;//////////////////////////////////////
;////////////////////////////////////// PM Trip Calculations - MWs 311-315 ////////////////////////////////////////
;//////////////////////////////////////
;
; Trips p m d p Trips p m d p
; in u o i e in u o i e
; H-NH Dir r d r r NH-H Dir r d r r
; | | | | | | | |
MW[311] = (MW[101] * (TODFtrs[1][4][1][3]/100.00) + MW[201] *
(TODFtrs[1][4][2][3]/100.00)) / 2.0 ; PM Period HBW Metrorail Trips *****
MW[312] = (MW[102] * (TODFtrs[2][4][1][3]/100.00) + MW[202] *
(TODFtrs[2][4][2][3]/100.00)) / 2.0 ; PM Period HBS Metrorail Trips *****
MW[313] = (MW[103] * (TODFtrs[3][4][1][3]/100.00) + MW[203] *
(TODFtrs[3][4][2][3]/100.00)) / 2.0 ; PM Period HBO Metrorail Trips *****
MW[314] = (MW[104] * (TODFtrs[4][4][1][3]/100.00) + MW[204] *
(TODFtrs[4][4][2][3]/100.00)) / 2.0 ; PM Period NHW Metrorail Trips *****
MW[315] = (MW[105] * (TODFtrs[5][4][1][3]/100.00) + MW[205] *
(TODFtrs[5][4][2][3]/100.00)) / 2.0 ; PM Period NHO Metrorail Trips *****
;
;//////////////////////////////////////
;////////////////////////////////////// NT Trip Calculations - MWs 411-415 ////////////////////////////////////////
;//////////////////////////////////////
;
; Trips p m d p Trips p m d p
; in u o i e in u o i e
; H-NH Dir r d r r NH-H Dir r d r r
; | | | | | | | |
MW[411] = (MW[101] * (TODFtrs[1][4][1][4]/100.00) + MW[201] *
(TODFtrs[1][4][2][4]/100.00)) / 2.0 ; NT Period HBW Metrorail Trips *****
MW[412] = (MW[102] * (TODFtrs[2][4][1][4]/100.00) + MW[202] *
(TODFtrs[2][4][2][4]/100.00)) / 2.0 ; NT Period HBS Metrorail Trips *****
MW[413] = (MW[103] * (TODFtrs[3][4][1][4]/100.00) + MW[203] *
(TODFtrs[3][4][2][4]/100.00)) / 2.0 ; NT Period HBO Metrorail Trips *****
MW[414] = (MW[104] * (TODFtrs[4][4][1][4]/100.00) + MW[204] *
(TODFtrs[4][4][2][4]/100.00)) / 2.0 ; NT Period NHW Metrorail Trips *****
MW[415] = (MW[105] * (TODFtrs[5][4][1][4]/100.00) + MW[205] *
(TODFtrs[5][4][2][4]/100.00)) / 2.0 ; NT Period NHO Metrorail Trips *****
;
;
; DONE WITH TIME OF DAY CALCULATIONS AT I/J LEVEL
;
ENDJLOOP

;-----
; Summarize TOTAL Output / calculated Metrorail Trips across time periods for each
purpose for checking (MWs 120,220,...,520)
MW[120] = MW[111] + MW[211] + MW[311] + MW[411] ; Total HBW summed across time
periods
MW[220] = MW[112] + MW[212] + MW[312] + MW[412] ; Total HBS summed across time
periods
MW[320] = MW[113] + MW[213] + MW[313] + MW[413] ; Total HBO summed across time
periods
MW[420] = MW[114] + MW[214] + MW[314] + MW[414] ; Total NBW summed across time
periods
MW[520] = MW[115] + MW[215] + MW[315] + MW[415] ; Total NBO summed across time
periods
;
;-----

```

## Appendix C Cube Voyager Scripts

```

; Summarize PEAK Period (AM&PM) Output / calculated Metrorail Trips for each purpose
(MWs 121,221,...,521)
MW[121] = MW[111]          + MW[311]          ; Peak HBW Metrorail Trips
MW[221] = MW[112]          + MW[312]          ; Peak HBS Metrorail Trips
MW[321] = MW[113]          + MW[313]          ; Peak HBO Metrorail Trips
MW[421] = MW[114]          + MW[314]          ; Peak NBW Metrorail Trips
MW[521] = MW[115]          + MW[315]          ; Peak NBO Metrorail Trips
;
; Summarize Off PK Period (MD&NT) Output / calculated Metrorail Trips for each
purpose (MWs 122,222,...,522)
MW[122] = MW[211]          + MW[411]          ; Off Peak HBW Metrorail Trips
MW[222] = MW[212]          + MW[412]          ; Off Peak HBS Metrorail Trips
MW[322] = MW[213]          + MW[413]          ; Off Peak HBO Metrorail Trips
MW[422] = MW[214]          + MW[414]          ; Off Peak NBW Metrorail Trips
MW[522] = MW[215]          + MW[415]          ; Off Peak NBO Metrorail Trips
;
; Now get regional totals to summarize neatly
Jloop
; accumulate calculated Metrorail trips by period(am,md,pm,nt),
purpose(hbw,hbs,hbo,nhw,nho)
; e.g. 'amhbw' refers to period 'am', and purp 'hbw'

amhbw=amhbw + mw[111]   amhbs=amhbs + mw[112]   amhbo=amhbo + mw[113]   amnhw=amnhw +
mw[114]   amnho = amnho + mw[115]
mdhbw=mdhbw + mw[211]   mdhbs=mdhbs + mw[212]   mdhbo=mdhbo + mw[213]   mdnhw=mdnhw +
mw[214]   mdnho = mdnho + mw[215]
pmhbw=pmhbw + mw[311]   pmhbs=pmhbs + mw[312]   pmhbo=pmhbo + mw[313]   pmnhw=pmnhw +
mw[314]   pmnho = pmnho + mw[315]
nthbw=nthbw + mw[411]   nthbs=nthbs + mw[412]   nthbo=nthbo + mw[413]   ntnhw=ntnhw +
mw[414]   ntnho = ntnho + mw[415]

; accumulate total Metrorail output trips by time period
outam =outam  + MW[111] + MW[112] + MW[113] + MW[114] + MW[115]
outmd =outmd  + MW[211] + MW[212] + MW[213] + MW[214] + MW[215]
outpm =outpm  + MW[311] + MW[312] + MW[313] + MW[314] + MW[315]
outnt =outnt  + MW[411] + MW[412] + MW[413] + MW[414] + MW[415]

outall =outall + MW[111] + MW[112] + MW[113] + MW[114] + MW[115] +
MW[211] + MW[212] + MW[213] + MW[214] + MW[215] +
MW[311] + MW[312] + MW[313] + MW[314] + MW[315] +
MW[411] + MW[412] + MW[413] + MW[414] + MW[415]

; accumulate total input Metrorail trips by purpose, total
inhbw=inhbw + MW[101] ; Input HBW
Metrorail Trips
inhbs=inhbs + MW[102] ; Input HBS
Metrorail Trips
inhbo=inhbo + MW[103] ; Input HBO
Metrorail Trips
innhw=innhw + MW[104] ; Input NHB
Metrorail Trips
innho=innho + MW[105] ; Input NHB
Metrorail Trips
intot=intot + MW[101] + MW[102] + MW[103] + MW[104] + MW[105] ; Input ALL
Metrorail Trips

; accumulate total output Metrorail trips by purpose, total
outhbw=outhbw + MW[120] ; Output HBW
Metrorail Trips

```

```

outhbs=outhbs + MW[220] ; Output HBS
Metrorail Trips
outhbo=outhbo + MW[320] ; Output HBO
Metrorail Trips
outnhw=outnhw + MW[420] ; Output NHB
Metrorail Trips
outnho=outnho + MW[520] ; Output NHB
Metrorail Trips
outtot=outtot + MW[120] + MW[220] + MW[320] + MW[420] + MW[520] ; Output ALL
Metrorail Trips

endjloop

; now write out the totals neatly if at the last zone:
if (i=zones)
; get differences by purpose (output - Input)
dfhbw = outhbw - inhbw;
dfhbs = outhbs - inhbs;
dfhbo = outhbo - inhbo;
dfnhw = outnhw - innhw;
dfnho = outnho - innho;
dftot = outtot - intot;

LIST = '/bt '
LIST = '@title@','\n'
LIST = ' '
list = 'TIME PERIOD HBW HBS HBO NHW NHO
Sum '
list = '----- --- --- --- --- ---
--- '
list = 'AM ',amhbw(12csv), amhbs(12csv), amhbo(12csv), amnhw(12csv),
amnho(12csv), outam(12csv)
list = 'MD ',mdhbw(12csv), mdhbs(12csv), mdhbo(12csv), mdnhw(12csv),
mdnho(12csv), outmd(12csv)
list = 'PM ',pmhbw(12csv), pmhbs(12csv), pmhbo(12csv), pmnhw(12csv),
pmnho(12csv), outpm(12csv)
list = 'NT ',nthbw(12csv), nthbs(12csv), nthbo(12csv), ntnhw(12csv),
ntnho(12csv), outnt(12csv)
list = ' '
list = 'Total ',outhbw(12csv), outhbs(12csv), outhbo(12csv), outnhw(12csv),
outnho(12csv),outall(12csv)
list = ' '
list = ' '
list = 'I/P
Totls',inhbw(12csv), inhbs(12csv), inhbo(12csv), innhw(12csv), innho(12csv), intot(12csv)
list = ' '
list = 'Diff.
',dfhbw(12csv),dfhbs(12csv),dfhbo(12csv),dfnhw(12csv),dfnho(12csv),dftot(12csv)

list = '/et '
endif

;
;-----
;--- END of TRANSIT Time-of-Day Process -----
;---
;-----
ENDRUN
ENDLOOP ; End of time-of -day loop

;//////////////////////////////////////
;
; Step 2
; 2010 & Future year peak & total transit trips are squeezed to

```

## Appendix C Cube Voyager Scripts

```

; a 3x3 (core/va/dc,md). Factors for scaling unconstrained
; transit trips to constrained transit trips are computed, on
; an i/j basis FOR ijs TO AND THROUGH the regional core.
;
; \\\\
; create equiv table
COPY FILE = three.eqv
; Beginning of 3x3 Equivalency Table
D 1=1405-1470,1477-1485,1490-1494,1498-1545 ; ArlNCore [1]
D 1=1546-1610 ; Alex [1]
D 1=1611-2159 ; FFX [1]
D 1=2160-2441 ; LDn [1]
D 1=2442-2554,2556-2628,2630-2819 ; PW [1]
D 1=3604-3653 ; Fau [1]
D 1=3449-3477,3479-3481,3483-3494,3496-3541 ; Stf. [1]
D 1=3654-3662,3663-3675 ; Clk,Jeff. [1]
D 1=3435-3448,3542-3543,3545-3603 ; Fbg,Spots [1]
D 1=3410-3434 ; KG. [1]
D 1=3676-3722 ; Ext(used) [1]
D 1=770,777,2555,2629,3103,3266-3267 ; Unused [1]
D 1=3478,3482,3495,3544 ; Unused [1]
D 2=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; DC CORE [2]
D 2=1471-1476, 1486-1489, 1495-1497 ; ArlCore [2]
D 3=5,48,51,64,66-180,210-281,288-373,382-393 ; DC Noncore [3]
D 3=394-769 ; Montgomery [3]
D 3=771-776,778-1404 ; Prince George [3]
D 3=2820-2949 ; Frd [3]
D 3=3230-3265,3268-3287 ; Car. [3]
D 3=2950-3017 ; How. [3]
D 3=3018-3102,3104-3116 ; AnnAr [3]
D 3=3288-3334 ; Calv [3]
D 3=3335-3409 ; StM [3]
D 3=3117-3229 ; Chs. [3]
; End of 3x3 Equivalency Table
ENDCOPY

RUN PGM=MATRIX
; Read input Files

; Input Constrained Metrorail Trips:
MATI[01] = HBWPKOPALL.con ; t1-3 >HBW Pk,OffPk, Total Metrorail Trips
MATI[02] = HBSKOPALL.con ; t1-3 >HBS Pk,OffPk, Total Metrorail Trips
MATI[03] = HBOPKOPALL.con ; t1-3 >HBO Pk,OffPk, Total Metrorail Trips
MATI[04] = NHWPKOPALL.con ; t1-3 >NHW Pk,OffPk, Total Metrorail Trips
MATI[05] = NHOPOKOPALL.con ; t1-3 >NHO Pk,OffPk, Total Metrorail Trips

; Input Forecast Year /Unconstrained Metrorail Trips:
MATI[06] = HBWPKOPALL.ucn ; t1-3 >HBW Pk,OffPk, Total Metrorail Trips
MATI[07] = HBSKOPALL.ucn ; t1-3 >HBS Pk,OffPk, Total Metrorail Trips
MATI[08] = HBOPKOPALL.ucn ; t1-3 >HBO Pk,OffPk, Total Metrorail Trips
MATI[09] = NHWPKOPALL.ucn ; t1-3 >NHW Pk,OffPk, Total Metrorail Trips
MATI[10] = NHOPOKOPALL.ucn ; t1-3 >NHO Pk,OffPk, Total Metrorail Trips

; Output 3x3 tables
FILEO MATO[1] = tempsqz.dat, MO=1-20
; sequence of squeezed (3x3) output trip tables
; 1- 5 ->> Constrained Peak HBW,HBS,HBO,NHW,NHO Metrorail trips
; 6-10 ->> Constrained Daily HBW,HBS,HBO,NHW,NHO Metrorail trips
; 11-15 ->> Forecast Peak HBW,HBS,HBO,NHW,NHO Metrorail trips
; 16-20 ->> Forecast Daily HBW,HBS,HBO,NHW,NHO Metrorail trips

; Read in Constrained Trips for each purpose (mw 1-10)
MW[01] = MI.1.1 MW[06]=MI.1.3 ; HBW Pk,Total Trips (MWS 1,6)
MW[02] = MI.2.1 MW[07]=MI.2.3 ; HBS Pk,Total Trips (MWS 2,7)
MW[03] = MI.3.1 MW[08]=MI.3.3 ; HBO Pk,Total Trips (MWS 3,8)
MW[04] = MI.4.1 MW[09]=MI.4.3 ; NHW Pk,Total Trips (MWS 4,9)
MW[05] = MI.5.1 MW[10]=MI.5.3 ; NHO Pk,Total Trips (MWS 5,10)

```

```

; Read in Forecasted Transit Trips for each purpose (mw 11-20)
MW[11] = MI.6.1 MW[16]=MI.6.3 ; HBW Pk,Total Trn Trips (MWS 11,16)
MW[12] = MI.7.1 MW[17]=MI.7.3 ; HBS Pk,Total Trn Trips (MWS 12,17)
MW[13] = MI.8.1 MW[18]=MI.8.3 ; HBO Pk,Total Trn Trips (MWS 13,18)
MW[14] = MI.9.1 MW[19]=MI.9.3 ; NHW Pk,Total Trn Trips (MWS 14,19)
MW[15] = MI.10.1 MW[20]=MI.10.3 ; NHO Pk,Total Trn Trips (MWS 15,20)

RENUMBER FILE=three.eqv, MISSINGZI=M, MISSINGZO=W
ENDRUN
;-----
;-----
; create a zonal matrix that indicates 3 superdistrict interchange type
; for example '11' means from SD 1 to SD 1, ..., '33' means from SD 3 to SD 3
RUN PGM=MATRIX
ZONES= 3722
FILEO MATO[1] = superdist.dat, MO=100

IF (I=1405-1470,1477-1485,1490-1494,1498-1545 ) SDi=1 ; ArlNCore
[1]-----
IF (I=1546-1610 ) SDi=1 ; Alex [1]
IF (I=1611-2159 ) SDi=1 ; FFX [1]
IF (I=2160-2441 ) SDi=1 ; LDn [1]
IF (I=2442-2554,2556-2628,2630-2819 ) SDi=1 ; PW [1]
VA NonCore
IF (I=3604-3653 ) SDi=1 ; Fau [1]
IF (I=3449-3477,3479-3481,3483-3494,3496-3541 ) SDi=1 ; Stf. [1]
IF (I=3654-3662,3663-3675 ) SDi=1 ; Clk,Jeff. [1]
IF (I=3435-3448,3542-3543,3545-3603 ) SDi=1 ; Fbg,Spots [1]
IF (I=3410-3434 ) SDi=1 ; KG. [1]
IF (I=3676-3722 ) SDi=1 ; Ext(used) [1]
IF (I=770,777,2555,2629,3103,3266-3267 ) SDi=1 ; Unused [1]
IF (I=3478,3482,3495,3544 ) SDi=1 ; Unused
[1]-----
IF (I=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381) SDi=2 ; DC CORE [2]
Regional
IF (I=1471-1476, 1486-1489, 1495-1497 ) SDi=2 ; ArlCore [2]
Core
IF (I=5,48,51,64,66-180,210-281,288-373,382-393 ) SDi=3 ; DC Noncore
[3]-----
IF (I=394-769 ) SDi=3 ; Montgomery [3]
IF (I=771-776,778-1404 ) SDi=3 ; Prince George [3]
IF (I=2820-2949 ) SDi=3 ; Frd [3]
DC/
IF (I=3230-3265,3268-3287 ) SDi=3 ; Car. [3]
MD Non-Core
IF (I=2950-3017 ) SDi=3 ; How. [3]
IF (I=3018-3102,3104-3116 ) SDi=3 ; AnnAr [3]
IF (I=3288-3334 ) SDi=3 ; Calv [3]
IF (I=3335-3409 ) SDi=3 ; StM [3]
IF (I=3117-3229 ) SDi=3 ; Chs. [3]
[3]-----
jloop
IF (J=1405-1470,1477-1485,1490-1494,1498-1545 ) SDj=1 ; ArlNCore
[1]-----
IF (J=1546-1610 ) SDj=1 ; Alex
[1]
IF (J=1611-2159 ) SDj=1 ; FFX
[1]
IF (J=2160-2441 ) SDj=1 ; LDn
[1]
IF (J=2442-2554,2556-2628,2630-2819 ) SDj=1 ; PW
[1] VA NonCore
IF (J=3604-3653 ) SDj=1 ; Fau
[1]

```

## Appendix C Cube Voyager Scripts

```

IF (J=3449-3477,3479-3481,3483-3494,3496-3541 ) SDj=1 ; Stf.
[1]
IF (J=3654-3662,3663-3675 ) SDj=1 ; Clk,Jeff.
[1]
IF (J=3435-3448,3542-3543,3545-3603 ) SDj=1 ; Fbg,Spots
[1]
IF (J=3410-3434 ) SDj=1 ; KG.
[1]
IF (J=3676-3722 ) SDj=1 ; Ext(unused)
[1]
IF (J=770,777,2555,2629,3103,3266-3267 ) SDj=1 ; Unused
[1]
IF (J=3478,3482,3495,3544 ) SDj=1 ; Unused
[1]-----
IF (J=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381) SDj=2 ; DC CORE
[2] Regional
IF (J=1471-1476, 1486-1489, 1495-1497 ) SDj=2 ; ArlCore
[2] Core
IF (J=5,48,51,64,66-180,210-281,288-373,382-393 ) SDj=3 ; DC Noncore
[3]-----
IF (J=394-769 ) SDj=3 ; Montgomery
[3]
IF (J=771-776,778-1404 ) SDj=3 ; Prince George
[3]
IF (J=2820-2949 ) SDj=3 ; Frd
[3] DC/
IF (J=3230-3265,3268-3287 ) SDj=3 ; Car.
[3] MD Non-Core
IF (J=2950-3017 ) SDj=3 ; How.
[3]
IF (J=3018-3102,3104-3116 ) SDj=3 ; AnnAr
[3]
IF (J=3288-3334 ) SDj=3 ; Calv
[3]
IF (J=3335-3409 ) SDj=3 ; StM
[3]
IF (J=3117-3229 ) SDj=3 ; Chs.
[3]-----

mw[100][j] = Sdi*10 + SDj
IF (MW[100][j] = 0) Abort ;; if ANY ij is not assigned a non-zero interchange
assignment stop
endjloop

;;-----
;;-----
ENDRUN

RUN PGM=MATRIX
; Read input Squeezed
ZONES=3
MATI[1] = tempsqz.dat
; Read in Constraining Metrorail Trips for each purpose (mw 1-10)
MW[1] = MI.1.1 MW[6] =MI.1.6 ; HBW Pk,Total Trips (MW1,6)
MW[2] = MI.1.2 MW[7] =MI.1.7 ; HBS Pk,Total Trips (MW2,7)
MW[3] = MI.1.3 MW[8] =MI.1.8 ; HBO Pk,Total Trips (MW3,8)
MW[4] = MI.1.4 MW[9] =MI.1.9 ; NHW Pk,Total Trips (MW4,9)
MW[5] = MI.1.5 MW[10]=MI.1.10 ; NHO Pk,Total Trips (MW5,10)

; Read in Forecasted Metrorail Trips for each purpose (mw 11-20)
MW[11] = MI.1.11 MW[16] =MI.1.16 ; HBW Pk,Total Trips (MW11,16)
MW[12] = MI.1.12 MW[17] =MI.1.17 ; HBS Pk,Total Trips (MW12,17)
MW[13] = MI.1.13 MW[18] =MI.1.18 ; HBO Pk,Total Trips (MW13,18)
MW[14] = MI.1.14 MW[19] =MI.1.19 ; NHW Pk,Total Trips (MW14,19)
MW[15] = MI.1.15 MW[20] =MI.1.20 ; NHO Pk,Total Trips (MW15,20)

```

```

; Now calculate constrained factors on an ij basis
JLOOP ; Initialize transit constraint factors
HBWConFtr = 1.000 ; HBW ftr
HBSConFtr = 1.000 ; HBS ftr
HBOConFtr = 1.000 ; HBO ftr
NHWConFtr = 1.000 ; NHW ftr
NHOConFtr = 1.000 ; NHO ftr

IF ((I = 1 && J = 2) || ; IF from VA nonCore to Regional Core
(I = 1 && J = 3) || ; or from VA nonCore to DC/MD Non Reg Core
(I = 3 && J = 1) || ; or from MD/DnonCore to VA Non Reg Core
(I = 3 && J = 2)) ; or from MD/DnonCore to Regional Core
; THEN calculate peak constraint factor, by purpose
; Constrained Metrorail trips =
; UnCon. Daily trips - UnCon. Pk Trips + Constrained Pk Trips
MW[21] = (MW[16]-MW[11])+MW[1] ; "Target" Constrained HBW Daily Trips
MW[22] = (MW[17]-MW[12])+MW[2] ; "Target" Constrained HBS Daily Trips
MW[23] = (MW[18]-MW[13])+MW[3] ; "Target" Constrained HBO Daily Trips
MW[24] = (MW[19]-MW[14])+MW[4] ; "Target" Constrained NHW Daily Trips
MW[25] = (MW[20]-MW[15])+MW[5] ; "Target" Constrained NHO Daily Trips

; compute factors for moderating forecasted trips to constrained trips
; also, do not allow factors be greater than 1.00 (some strange cases may
exist)
IF (MW[16]!=0) HBWConFtr = MIN(1.00, (MW[21] / MW[16])) ; HBW
IF (MW[17]!=0) HBSConFtr = MIN(1.00, (MW[22] / MW[17])) ; HBS
IF (MW[18]!=0) HBOConFtr = MIN(1.00, (MW[23] / MW[18])) ; HBO
IF (MW[19]!=0) NHWConFtr = MIN(1.00, (MW[24] / MW[19])) ; NHW
IF (MW[20]!=0) NHOConFtr = MIN(1.00, (MW[25] / MW[20])) ; NHO

; Accumulate Final Constrained Metrorail trips
HBW_FCT = HBW_FCT + ((MW[16]-MW[11])+MW[1]) ; Constrained HBW Daily Trips
HBS_FCT = HBS_FCT + ((MW[17]-MW[12])+MW[2]) ; Constrained HBS Daily Trips
HBO_FCT = HBO_FCT + ((MW[18]-MW[13])+MW[3]) ; Constrained HBO Daily Trips
NHW_FCT = NHW_FCT + ((MW[19]-MW[14])+MW[4]) ; Constrained NHW Daily Trips
NHO_FCT = NHO_FCT + ((MW[20]-MW[15])+MW[5]) ; Constrained NHO Daily Trips

ELSE
HBW_FCT = HBW_FCT + MW[16] ; Constrained HBW Daily Trips
HBS_FCT = HBS_FCT + MW[17] ; Constrained HBS Daily Trips
HBO_FCT = HBO_FCT + MW[18] ; Constrained HBO Daily Trips
NHW_FCT = NHW_FCT + MW[19] ; Constrained NHW Daily Trips
NHO_FCT = NHO_FCT + MW[20] ; Constrained NHO Daily Trips

ENDIF

IJ = I*10+j ; create two digit no where 1st digit=i,2nd=j

; print ij, const pk&total, unconstr pk/total, final total trn trips, ftr
; --one file for each purpose

Print LIST = ij(4),MW[1](8),MW[6](8),
MW[11](8),MW[16](8),MW[21](8),HBWConFtr(10.4),File=tconftr.HBW
Print LIST = ij(4),MW[2](8),MW[7](8),
MW[12](8),MW[17](8),MW[22](8),HBSConFtr(10.4),File=tconftr.HBS
Print LIST = ij(4),MW[3](8),MW[8](8),
MW[13](8),MW[18](8),MW[23](8),HBOConFtr(10.4),File=tconftr.HBO
Print LIST = ij(4),MW[4](8),MW[9](8),
MW[14](8),MW[19](8),MW[24](8),NHWConFtr(10.4),File=tconftr.NHW
Print LIST =
ij(4),MW[5](8),MW[10](8),MW[15](8),MW[20](8),MW[25](8),NHOConFtr(10.4),File=tconftr.
NHO
ENDJLOOP

```

## Appendix C Cube Voyager Scripts

```

IF (I=ZONES)
  Print LIST = ' Control Total HBW Constrained Transit Trips: ',HBW_FCT(10)
  Print LIST = ' Control Total HBS Constrained Transit Trips: ',HBS_FCT(10)
  Print LIST = ' Control Total HBO Constrained Transit Trips: ',HBO_FCT(10)
  Print LIST = ' Control Total NHW Constrained Transit Trips: ',NHW_FCT(10)
  Print LIST = ' Control Total NHO Constrained Transit Trips: ',NHO_FCT(10)
endif
; Now, Let's carry the control totals with us so we can compare with the
; zonal totals, that will be computed in the next step
LOG PREFIX = MATRIX, VAR = HBW_FCT, HBS_FCT, HBO_FCT, NHW_FCT, NHO_FCT
;
;
ENDRUN

////////////////////////////////////
;
; Begin Step 3
;   future year constrained trips are computed by applying
;   the constraint factors to the zonal trip tables.
;   constrained transit trips are produced (i.e., residual auto
;   persons are generated. and LOV,HOV auto person/driver trips
;   are computed using existing distributions on a cell by cell
;   basis.
;
;////////////////////////////////////
LOOP TIME = 1,5 ; Loop through for each purpose
  IF (TIME=1)
    PRP = 'HBW' ; Purpose code
    Control = MATRIX.HBW_FCT ; Transit Control Total
  ELSEIF (TIME=2)
    PRP = 'HBS' ; Purpose Code
    Control = MATRIX.HBS_FCT ; Transit Control Total
  ELSEIF (TIME=3)
    PRP = 'HBO' ; Purpose code
    Control = MATRIX.HBO_FCT ; Transit Control Total
  ELSEIF (TIME=4)
    PRP = 'NHW' ; Purpose code
    Control = MATRIX.NHW_FCT ; Transit Control Total
  ELSEIF (TIME=5)
    PRP = 'NHO' ; Purpose code
    Control = MATRIX.NHO_FCT ; Transit Control Total
  ENDIF

RUN PGM=MATRIX
ZONES = 3722
pageheight=32767 ; Preclude header breaks
;-----
; Tables NL MC model outfile
;
; 1 DR ALONE All transit 4-14
; 2 SR2 Metrorail only 7,13,14
; 3 SR3+ Metrorail related 7,13,14,6,11,12
; 4 WK-CR Auto person 1-3
; 5 WK-BUS Total motorized psn 1-14
; 6 WK-BU/MR Commuter rail 4,8 (may incl bus/Mrail)
; 7 WK-MR Bus only 5,9,10
; 8 PNR-CR Bus only, WMATA Compact area
; 8 KNR-CR
; 9 PNR-BUS
; 10 KNR-BUS
; 11 PNR-BU/MR
; 12 KNR-BU/MR
; 13 PNR-MR
; 14 KNR-MR
;-----
; DEFINE INPUT/OUTPUT FILES HERE:

```

```

MATI[1] = % iter % @prp@_NL_MC.MTT ; UNCONST. MODE CH TRIPS
MATI[2] = superdist.dat ; 3x3 super district ij
indicator
MATO[1] = % iter % @prp@_NL_MC.CON,MO=201-214,dec = 14*3 ; CONSTR. MODE CH TRIPS
FILLMW MW[101]=MI.1.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; Read in Unconstrained
NL MC tabs 1-14

LOOKUP NAME=TCONFTR,
LOOKUP[1]=1,RESULT=7,INTERPOLATE=N,LIST=T,FAIL=0,0,0,FILE=TCONFTR.@prp@

MW[30] = MI.2.1 ; superdistrict ij indicator '11' to '33'
;
; Now Factor transit tables
;
MW[31] = MW[101] + MW[102] + MW[103]
; Initial Auto Person
MW[32] = MW[106] + MW[107] + MW[111] + MW[112] + MW[113] + MW[114]
; Initial/Unconstr. Metrorail
MW[33] = MW[104] + MW[105] + MW[106] + MW[107] +
MW[108] + MW[109] + MW[110] + MW[111] + MW[112] + MW[113] + MW[114]
; Initial/Unconstrained Transit
MW[34] = MW[101] + MW[102] + MW[103] + MW[104] + MW[105] + MW[106] + MW[107]
+
MW[108] + MW[109] + MW[110] + MW[111] + MW[112] + MW[113] + MW[114]
; Initial Total Person

JLOOP
MW[204] = MW[104] ; unaffected Transit
MW[205] = MW[105] ; unaffected Transit
MW[206] = MW[106] * tconftr(1,MW[30]) ; Constrained Metrorail WK-BU/MR
MW[207] = MW[107] * tconftr(1,MW[30]) ; Constrained Metrorail WK-MR
MW[208] = MW[108] ; unaffected Transit
MW[209] = MW[109] ; unaffected Transit
MW[210] = MW[110] ; unaffected Transit
MW[211] = MW[111] * tconftr(1,MW[30]) ; Constrained Metrorail PNR-BU/MR
MW[212] = MW[112] * tconftr(1,MW[30]) ; Constrained Metrorail KNR-BU/MR
MW[213] = MW[113] * tconftr(1,MW[30]) ; Constrained Metrorail PNR-MR
MW[214] = MW[114] * tconftr(1,MW[30]) ; Constrained Metrorail KNR-MR

MW[42] = MW[206] + MW[207] + MW[211] + MW[212] + MW[213] + MW[214]
; Constrained Metrorail
MW[43] = MW[204] + MW[205] + MW[206] + MW[207] +
MW[208] + MW[209] + MW[210] + MW[211] + MW[212] + MW[213] +
MW[214] ; Constrained Transit

MW[99] = MW[32] - MW[42] ; Metrorail 'Residual' /Unconstrained -
Constrained Metrorail
IF (MW[99] < 0.0) ; - Make sure the residual is
MW[99] = 0.0 ; NOT negative
ENDIF ;
;
; Now work on converting 'residual' Metrorail trips among auto modes based on
proration
;
MW[35] = MW[101] + MW[102] + MW[103] ; initial total auto person trips

IF (mw[35] = 0.0) ; if no auto person trips exist, put residual transit
trips into SOVs

MW[201] = MW[99] ; Updated SOV Psn
MW[202] = 0.0 ; Updated HOV2 Psn
MW[203] = 0.0 ; Updated HOV3+ Psn

ELSE ; ; else, add residual trips to occupant groups based on
existing proration

```



## Appendix C Cube Voyager Scripts

```

MW[201] = MW[101] + (MW[99] * (MW[101]/MW[35])) ; Updated SOV Psn
MW[202] = MW[102] + (MW[99] * (MW[102]/MW[35])) ; Updated HOV2 Psn
MW[203] = MW[103] + (MW[99] * (MW[103]/MW[35])) ; Updated HOV3+ Psn
ENDIF

ENDJLOOP
;
;
;
JLOOP
;
; Now Accumulate Initial and Updated Totals /RATES Here:
;
INI_SOV      = INI_SOV      + MW[101]
INI_HOV_2    = INI_HOV_2    + MW[102]
INI_HOV_3    = INI_HOV_3    + MW[103]
INI_WLK_COM  = INI_WLK_COM  + MW[104]
INI_WLK_BUS  = INI_WLK_BUS  + MW[105]
INI_WLK_BMR  = INI_WLK_BMR  + MW[106]
INI_WLK_MR   = INI_WLK_MR   + MW[107]
INI_PNR_COM  = INI_PNR_COM  + MW[108]
INI_PNR_BUS  = INI_PNR_BUS  + MW[109]
INI_KNR_BUS  = INI_KNR_BUS  + MW[110]
INI_PNR_BMR  = INI_PNR_BMR  + MW[111]
INI_KNR_BMR  = INI_KNR_BMR  + MW[112]
INI_PNR_MR   = INI_PNR_MR   + MW[113]
INI_KNR_MR   = INI_KNR_MR   + MW[114]

INI_Metro    = INI_Metro    + MW[106] + MW[107] + MW[111] + MW[112] +
MW[113] + MW[114]

INI_Transit  = INI_Transit  + MW[104] + MW[105] + MW[106] + MW[107] +
MW[108] + MW[109] + MW[110] + MW[111] +
MW[112] + MW[113] + MW[114]

INI_AutoPsn  = INI_AutoPsn  + MW[101] + MW[102] + MW[103]

INI_Person   = INI_Person   + MW[101] + MW[102] + MW[103] + MW[104] +
MW[105] + MW[106] + MW[107] +
MW[108] + MW[109] + MW[110] + MW[111] +
MW[112] + MW[113] + MW[114]

UPD_SOV      = UPD_SOV      + MW[201]
UPD_HOV_2    = UPD_HOV_2    + MW[202]
UPD_HOV_3    = UPD_HOV_3    + MW[203]
UPD_WLK_COM  = UPD_WLK_COM  + MW[204]
UPD_WLK_BUS  = UPD_WLK_BUS  + MW[205]
UPD_WLK_BMR  = UPD_WLK_BMR  + MW[206]
UPD_WLK_MR   = UPD_WLK_MR   + MW[207]
UPD_PNR_COM  = UPD_PNR_COM  + MW[208]
UPD_PNR_BUS  = UPD_PNR_BUS  + MW[209]
UPD_KNR_BUS  = UPD_KNR_BUS  + MW[210]
UPD_PNR_BMR  = UPD_PNR_BMR  + MW[211]
UPD_KNR_BMR  = UPD_KNR_BMR  + MW[212]
UPD_PNR_MR   = UPD_PNR_MR   + MW[213]
UPD_KNR_MR   = UPD_KNR_MR   + MW[214]

UPD_Metro    = UPD_Metro    + MW[206] + MW[207] + MW[211] + MW[212] +
MW[213] + MW[214]

UPD_Transit  = UPD_Transit  + MW[204] + MW[205] + MW[206] + MW[207] +
MW[208] + MW[209] + MW[210] + MW[211] +
MW[212] + MW[213] + MW[214]

UPD_AutoPsn  = UPD_AutoPsn  + MW[201] + MW[202] + MW[203]

```

```

UPD_Person = UPD_Person + MW[201] + MW[202] + MW[203] + MW[204] +
MW[205] + MW[206] + MW[207] +
MW[208] + MW[209] + MW[210] + MW[211] +
MW[212] + MW[213] + MW[214]

ENDJLOOP

; If at end, Get Global Mode differences and regional rates

if (i=zones)

; get differences by purpose (output - Input)

DIF_SOV      = UPD_SOV      - INI_SOV
DIF_HOV_2    = UPD_HOV_2    - INI_HOV_2
DIF_HOV_3    = UPD_HOV_3    - INI_HOV_3
DIF_WLK_COM  = UPD_WLK_COM  - INI_WLK_COM
DIF_WLK_BUS  = UPD_WLK_BUS  - INI_WLK_BUS
DIF_WLK_BMR  = UPD_WLK_BMR  - INI_WLK_BMR
DIF_WLK_MR   = UPD_WLK_MR   - INI_WLK_MR
DIF_PNR_COM  = UPD_PNR_COM  - INI_PNR_COM
DIF_PNR_BUS  = UPD_PNR_BUS  - INI_PNR_BUS
DIF_KNR_BUS  = UPD_KNR_BUS  - INI_KNR_BUS
DIF_PNR_BMR  = UPD_PNR_BMR  - INI_PNR_BMR
DIF_KNR_BMR  = UPD_KNR_BMR  - INI_KNR_BMR
DIF_PNR_MR   = UPD_PNR_MR   - INI_PNR_MR
DIF_KNR_MR   = UPD_KNR_MR   - INI_KNR_MR

DIF_Metro    = UPD_Metro    - INI_Metro
DIF_Transit  = UPD_Transit  - INI_Transit
DIF_AutoPsn  = UPD_AutoPsn  - INI_AutoPsn
DIF_Person   = UPD_Person   - INI_Person

; Calculate transit percentages, initial and updated

IF (INI_Person != 0) INI_TrnPct = INI_Transit/ INI_Person * 100.00 ; input
%TRN
IF (UPD_Person != 0) UPD_TrnPct = UPD_Transit/ UPD_Person * 100.00 ; output
%TRN

DIF_TrnPct = UPD_TrnPct - INI_TrnPct

CONTOTAL = @control@ ; control total from previous step
LIST = '/bt '
LIST = ' @prp@ METRORAIL CONSTRAINT RESULTS- Global Totals by Submode'
LIST = ' Initial and Final Totals by Mode','\n'
LIST = ' '
list = 'MODE ' INITIAL ' ' UPDATED ' 'DIFFERENCE'
list = '----- ' '----- ' '----- '
LIST= ' '
LIST = 'SOV ' INI_SOV(12.2csv), UPD_SOV(12.2csv),
DIF_SOV(12.2csv)
LIST = 'HOV_2 ' INI_HOV_2(12.2csv), UPD_HOV_2(12.2csv),
DIF_HOV_2(12.2csv)
LIST = 'HOV_3 ' INI_HOV_3(12.2csv), UPD_HOV_3(12.2csv),
DIF_HOV_3(12.2csv)
LIST = 'WLK_COM ' INI_WLK_COM(12.2csv), UPD_WLK_COM(12.2csv),
DIF_WLK_COM(12.2csv)
LIST = 'WLK_BUS ' INI_WLK_BUS(12.2csv), UPD_WLK_BUS(12.2csv),
DIF_WLK_BUS(12.2csv)
LIST = 'WLK_BMR ' INI_WLK_BMR(12.2csv), UPD_WLK_BMR(12.2csv),
DIF_WLK_BMR(12.2csv)
LIST = 'WLK_MR ' INI_WLK_MR(12.2csv), UPD_WLK_MR(12.2csv),
DIF_WLK_MR(12.2csv)
LIST = 'PNR_COM ' INI_PNR_COM(12.2csv), UPD_PNR_COM(12.2csv),
DIF_PNR_COM(12.2csv)

```

## Appendix C Cube Voyager Scripts

```

LIST = 'PNR_BUS', INI_PNR_BUS(12.2csv), UPD_PNR_BUS(12.2csv),
DIF_PNR_BUS(12.2csv)
LIST = 'KNR_BUS', INI_KNR_BUS(12.2csv), UPD_KNR_BUS(12.2csv),
DIF_KNR_BUS(12.2csv)
LIST = 'PNR_BMR', INI_PNR_BMR(12.2csv), UPD_PNR_BMR(12.2csv),
DIF_PNR_BMR(12.2csv)
LIST = 'KNR_BMR', INI_KNR_BMR(12.2csv), UPD_KNR_BMR(12.2csv),
DIF_KNR_BMR(12.2csv)
LIST = 'PNR_MR', INI_PNR_MR(12.2csv), UPD_PNR_MR(12.2csv),
DIF_PNR_MR(12.2csv)
LIST = 'KNR_MR', INI_KNR_MR(12.2csv), UPD_KNR_MR(12.2csv),
DIF_KNR_MR(12.2csv)
LIST=' '
LIST = 'TOTAL PERSON:', INI_Person(12.2csv), UPD_Person(12.2csv),
DIF_Person(12.2csv)
LIST=' '
LIST = 'Metrorail', INI_Metro(12.2csv), UPD_Metro(12.2csv),
DIF_Metro(12.2csv)
LIST = 'Transit:', INI_Transit(12.2csv), UPD_Transit(12.2csv),
DIF_Transit(12.2csv)
LIST = 'TRANSIT Control Total', CONTOTAL(12.2csv), '
<-- Based on Squeezed 3x3 Trips'
LIST = 'AutoPerson', INI_AutoPsn(12.2csv), UPD_AutoPsn(12.2csv),
DIF_AutoPsn(12.2csv)
LIST = 'Transit %:', INI_TrnPct(12.2csv), UPD_TrnPct(12.2csv),
DIF_TrnPct(12.2csv)
LIST=' '
list = '/et '
endif

ENDRUN
ENDLOOP

```

## 20 MC\_NL\_Summary.s

```

;-----
; Program Name: MC_NL_Summary.s
; Version 2.3 Model w/ Nested Logit MC model
;
; Summarize final table by purpose & Mode & Submode
; 8/30/11 - Juris. level tables expanded to include auto person trips by occupant
levels
; (SOV, HOV 2+occ. and HOV 3+occ. auto person trips)
;
;
; 65 tables are written out: 6 purposes (HBW,HBS,HBO,NHW,NHO,Total) by 13
modes,submodes:
; 1. Motorized Person
; 2. Transit
; 3. Transit Percentage
; 4. Auto Person
; 5. SOV Person
; 6. HOV2Occ Person
; 7. HOV3+Occ Person
; 8. Auto Driver
; 9. Auto Occupancy
; 10. Commuter Rail
; 11. All Bus
; 12. Bus & Metrorail
; 13. Metrorail Only
;
;
; Environment Variables Used:
; %_iter_%
; %_year_%

```

```

; %_alt_%
;
;-----
; Modes in AECOM MC model Summary modes
; 1 DR ALONE 1 All transit 4-14
; 2 SR2 2 Metrorail only 7,13,14
; 3 SR3+ 3 Metrorail related 7,13,14,6,11,12
; 4 WK-CR 4 Auto person 1-3
; 5 WK-BUS 5 Total motorized psn 1-14
; 6 WK-BU/MR 6 Commuter rail 4,8 (may incl bus/Mrail)
; 7 WK-MR 7 Bus only 5,9,10
; 8 PNR-CR 8 Bus only, WMATA Compact area
; 9 KNR-CR
; 10 PNR-BUS
; 11 KNR-BUS
; 12 PNR-BU/MR
; 13 KNR-BU/MR
; 14 PNR-MR
; 15 KNR-MR
;-----
;-----
; Now summarize total purpose trip tables, by mode
;-----
pageheight=32767 ; Preclude header breaks
HOV3_OCC = 3.50 ; Assumed Occupancy of 3+ Vehicles

RUN PGM=MATRIX

ZONES=3722

MATI[1]= %_iter_%_HBW_NL_MC.MTT
MATI[2]= %_iter_%_HBS_NL_MC.MTT
MATI[3]= %_iter_%_HBO_NL_MC.MTT
MATI[4]= %_iter_%_NHW_NL_MC.MTT
MATI[5]= %_iter_%_NHO_NL_MC.MTT

FILLMW MW[101] = mi.1.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; HBW modal trip tabs
101..114
FILLMW MW[201] = mi.2.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; HBS modal trip tabs
201..214
FILLMW MW[301] = mi.3.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; HBO modal trip tabs
301..314
FILLMW MW[401] = mi.4.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; NHW modal trip tabs
401..414
FILLMW MW[501] = mi.5.1,2,3,4,5,6,7,8,9,10,11,12,13,14 ; NHO modal trip tabs
501..514

MW[601]= MW[101]+MW[201]+MW[301]+MW[401]+MW[501] MW[602]=
MW[102]+MW[202]+MW[302]+MW[402]+MW[502] ; sum
MW[603]= MW[103]+MW[203]+MW[303]+MW[403]+MW[503] MW[604]=
MW[104]+MW[204]+MW[304]+MW[404]+MW[504] ; total purpose
MW[605]= MW[105]+MW[205]+MW[305]+MW[405]+MW[505] MW[606]=
MW[106]+MW[206]+MW[306]+MW[406]+MW[506] ; trips in tabs
MW[607]= MW[107]+MW[207]+MW[307]+MW[407]+MW[507] MW[608]=
MW[108]+MW[208]+MW[308]+MW[408]+MW[508] ; 501..514
MW[609]= MW[109]+MW[209]+MW[309]+MW[409]+MW[509] MW[610]=
MW[110]+MW[210]+MW[310]+MW[410]+MW[510] ;
MW[611]= MW[111]+MW[211]+MW[311]+MW[411]+MW[511] MW[612]=
MW[112]+MW[212]+MW[312]+MW[412]+MW[512] ;
MW[613]= MW[113]+MW[213]+MW[313]+MW[413]+MW[513] MW[614]=
MW[114]+MW[214]+MW[314]+MW[414]+MW[514] ;

MATO[1] = %_iter_%_ALL_NL_MC.MTT, MO=601-614, dec = 14*3 ; Total Purpose Mode
Choice Trips
ENDRUN

```

# Appendix C Cube Voyager Scripts

```

;-----
; Summarize the Mode Choice Model Output to Juris. Level
;-----

DESCRIPT='Simulation - Year: %_year_ Alternative: %_alt_ Iteration: %_iter_ '

LOOP PURP=1,6 ; Outer Loop for Each Purpose (HBW,HBS,HBO,NHW,NHO, Total)
IF (PURP=1)
  pur = 'HBW'
  purfile = 'A_HBW.tbl'
  MCOUATAB='%_iter_%_HBW_NL_MC.MTT'
  PURPOSE = 'Internal HBW Trips'
ELSEIF (PURP=2)
  pur = 'HBS'
  purfile = 'B_HBS.tbl'
  MCOUATAB='%_iter_%_HBS_NL_MC.MTT'
  PURPOSE = 'Internal HBS Trips'
ELSEIF (PURP=3)
  pur = 'HBO'
  purfile = 'C_HBO.tbl'
  MCOUATAB='%_iter_%_HBO_NL_MC.MTT'
  PURPOSE = 'Internal HBO Trips'
ELSEIF (PURP=4)
  pur = 'NHW'
  purfile = 'D_NHW.tbl'
  MCOUATAB='%_iter_%_NHW_NL_MC.MTT'
  PURPOSE = 'Internal NHW Trips'
ELSEIF (PURP=5)
  pur = 'NHO'
  purfile = 'E_NHO.tbl'
  MCOUATAB='%_iter_%_NHO_NL_MC.MTT'
  PURPOSE = 'Internal NHO Trips'
ELSEIF (PURP=6)
  pur = 'ALL'
  purfile = 'F_ALL.tbl'
  MCOUATAB='%_iter_%_ALL_NL_MC.MTT'
  PURPOSE = 'Total Internal Trips'
ENDIF

;-----
; Summarize the Est./Obs Output Files to Juris. Level
;-----

COPY FILE=DJ.EQV
; -- Start of Jurisdiction-to-TAZ equivalency --
D 1=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; 0 DC Core
D 2=5,48,51,64,66-180,210-281,288-373,382-393 ; 0 DC Noncore
D 3=394-769 ; 1 Montgomery
D 4=771-776,778-1404 ; 2 Prince George
D 5=1471-1476, 1486-1489, 1495-1497 ; 3 ArlCore
D 6=1405-1470,1477-1485,1490-1494,1498-1545 ; 3 ArlNCore
D 7=1546-1610 ; 4 Alex
D 8=1611-2159 ; 5 FFX
D 9=2160-2441 ; 6 LDn
D 10=2442-2554,2556-2628,2630-2819 ; 7 PW
D 11=2820-2949 ; 9 Frd
D 12=3230-3265,3268-3287 ; 14 Car.
D 13=2950-3017 ; 10 How.
D 14=3018-3102,3104-3116 ; 11 AnnAr
D 15=3288-3334 ; 15 Calv
D 16=3335-3409 ; 16 STM
D 17=3117-3229 ; 12 Chs.
D 18=3604-3653 ; 21 Fau
D 19=3449-3477,3479-3481,3483-3494,3496-3541 ; 19 Stf.
D 20=3654-3662,3663-3675 ; 22/23 Clk,Jeff.

```

```

D 21=3435-3448,3542-3543,3545-3603 ; 18/20 Fbg,Spots
D 22=3410-3434 ; 17 KG.
D 23=3676-3722 ; Externals
; -- end of Jurisdiction-to-TAZ equivalency --
ENDCOPY

RUN PGM=MATRIX
PAGEHEIGHT= 32767
ZONES=3722
MATI[1]= @MCOUATAB@

MW[01] = MI.1.4 + MI.1.5 + MI.1.6 + MI.1.7 + MI.1.8 + ; 1/Transit
          MI.1.9 + MI.1.10 + MI.1.11 + MI.1.12 + MI.1.13 +
          MI.1.14
MW[02] = MI.1.1 + MI.1.2 + MI.1.3 ; 2/Auto_Psn
MW[03] = MI.1.1 + (MI.1.2/2.0) + MI.1.3/@HOV3_OCC@ ; 3/Auto_Drv

MW[04] = MW[1] + MW[2] ; 4/Person

MW[05] = MI.1.4 + MI.1.5 + MI.1.6 + MI.1.7 ; 5/TRN_WLK
MW[06] = MI.1.8 + MI.1.9 + MI.1.11 + MI.1.13 ; 6/TRN_PNR
MW[07] = MI.1.10 + MI.1.12 + MI.1.14 ; 7/TRN_KNR

MW[08] = MI.1.1 ; DR ALONE ; 8/SOV_Psn
MW[09] = MI.1.2 ; SR2 ; 9/HOV2_Psn
MW[10] = MI.1.3 ; SR3+ ; 10/HOV3_Psn

MW[11] = MI.1.4 ; WK-CR ; 11/WLK_CR
MW[12] = MI.1.5 ; WK-AB ; 12/WLK_AB
MW[13] = MI.1.6 ; WK-BM ; 13/WLK_BM
MW[14] = MI.1.7 ; WK-MR ; 14/WLK_MR

MW[15] = MI.1.8 ; PNR-CR ; 15/PNR_CR
MW[16] = MI.1.9 ; PNR-AB ; 16/PNR_AB
MW[17] = MI.1.10 ; KNR-AB ; 17/KNR_AB
MW[18] = MI.1.11 ; PNR-BM ; 18/PNR_BM
MW[19] = MI.1.12 ; KNR-BM ; 19/KNR_BM
MW[20] = MI.1.13 ; PNR-MR ; 20/PNR_MR
MW[21] = MI.1.14 ; KNR-MR ; 21/KNR_MR

MW[22] = MW[11] + MW[15] ; 22/cr
MW[23] = MW[12] + MW[16] + MW[17] ; 23/ab
MW[24] = MW[13] + MW[18] + MW[19] ; 24/bm
MW[25] = MW[14] + MW[20] + MW[21] ; 25/mr

MW[26] = MI.1.1/1.0 ; 26/SOV_vehs
MW[27] = MI.1.2/2.0 ; 27/HOV2_vehs
MW[28] = MI.1.3/@HOV3_OCC@ ; 28/HOV3_vehs

MW[30]= 0 ; dummy/placemaker table

;; ACCUMULATE MODAL TOTALS
Transit = Transit + ROWSUM(01)
Auto_Psn = Auto_Psn + ROWSUM(02)
Auto_Drv = Auto_Drv + ROWSUM(03)

Person = Person + ROWSUM(01) + ROWSUM(02)
SOV_Psn = SOV_Psn + ROWSUM(08)
HOV2_Psn = HOV2_Psn + ROWSUM(09)
HOV3_Psn = HOV3_Psn + ROWSUM(10)

```

# Appendix C Cube Voyager Scripts

```

SOV_Veh      = SOV_Veh  + ROWSUM(26)
HOV2_Veh    = HOV2_Veh + ROWSUM(27)
HOV3_Veh    = HOV3_Veh + ROWSUM(28)

Trn_WLK     = Trn_WLK  + ROWSUM(11) + ROWSUM(12) + ROWSUM(13) +
ROWSUM(14)
Trn_PNR     = Trn_PNR  + ROWSUM(15) + ROWSUM(16) + ROWSUM(18) +
ROWSUM(20)
Trn_KNR     = Trn_KNR  + ROWSUM(17) + ROWSUM(19) + ROWSUM(21)

CR          = CR       + ROWSUM(11) + ROWSUM(15)
AB          = AB       + ROWSUM(12) + ROWSUM(16) + ROWSUM(17)
BM          = BM       + ROWSUM(13) + ROWSUM(18) + ROWSUM(19)
MR          = MR       + ROWSUM(14) + ROWSUM(20) + ROWSUM(21)

WLK_CR     = WLK_CR   + ROWSUM(11)
WLK_AB     = WLK_AB   + ROWSUM(12)
WLK_BM     = WLK_BM   + ROWSUM(13)
WLK_MR     = WLK_MR   + ROWSUM(14)

PNR_CR     = PNR_CR   + ROWSUM(15)
PNR_AB     = PNR_AB   + ROWSUM(16)
PNR_BM     = PNR_BM   + ROWSUM(18)
PNR_MR     = PNR_MR   + ROWSUM(20)

KNR_AB     = KNR_AB   + ROWSUM(17)
KNR_BM     = KNR_BM   + ROWSUM(19)
KNR_MR     = KNR_MR   + ROWSUM(21)

IF (I=ZONES)
;;
;; compute regional rates
;;
Transit_Pct = Transit/Person * 100.00
Auto_Occ    = Auto_Psn/Auto_Drv

;; print global totals:
PRINT LIST=' Purpose: ','@pur@', ' Regional Totals Summary',
file= @purfile@
PRINT LIST=' '
PRINT FORM=12.0csv List= ' ',' Transit: ', Transit
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' Auto_Person: ', Auto_Psn
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' Total_Person: ', Person
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' '
,file= @purfile@;
PRINT FORM=12.2csv List= ' ',' Transit Pct.: ',
Transit_Pct ,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' SOV_Auto_Person: ', SOV_Psn
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' HOV2_Auto_Person: ', HOV2_Psn
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' HOV3+Auto_Person ', HOV3_Psn
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' Auto_Person: ', Auto_Psn
,file= @purfile@;

```

```

PRINT FORM=12.0csv List= ' ',' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' SOV_Auto_Driver: ', SOV_Veh
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' HOV2_Auto_Driver: ', HOV2_Veh
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' HOV3+Auto_Driver: ', HOV3_Veh
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Auto_Driver: ', Auto_Drv
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.2csv List= ' ',' ' Auto Occupancy: ', Auto_Occ
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Commuter_Rail: ', CR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' All_Bus: ', AB
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Bus&Metrorail: ', BM
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Metrorail_Only: ', MR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Transit: ', Transit
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Walk_Commuter_Rail: ', WLK_CR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Walk_All_Bus ', WLK_AB
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Walk_Bus_& Metrorail:', WLK_BM
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Walk_Metrorail_Only: ', WLK_MR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Total WLK Acc: ', Trn_WLK
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' PNR_Commuter_Rail: ', PNR_CR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' PNR_All_Bus ', PNR_AB
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' PNR_Bus_& Metrorail:', PNR_BM
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' PNR_Metrorail_Only: ', PNR_MR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' -----'
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' Total PNR Acc: ', Trn_PNR
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' '
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' KNR_ALL_Bus: ', KNR_AB
,file= @purfile@;
PRINT FORM=12.0csv List= ' ',' ' KNR_Bus_& Metrorail:', KNR_BM
,file= @purfile@;

```

## Appendix C Cube Voyager Scripts

```

        PRINT FORM=12.0csv List= '      ', '      KNR_Metrorail_Only: ', KNR_MR
,file= @purfile@;
        PRINT FORM=12.0csv List= '      ', '      ----- '
,file= @purfile@;
        PRINT FORM=12.0csv List= '      ', '      Total KNR Acc: ', Trn_KNR
,file= @purfile@;
        PRINT FORM=12.0csv List= '      ', '      '
,file= @purfile@;
        PRINT          LIST=' ===== End ', '@pur@', ' Purpose ===== '
,file= @purfile@;
        PRINT FORM=12.0csv List= '      ', '      ', '
,file= @purfile@;
        PRINT LIST='/et          '

ENDIF

;;
FILEO MATO[01] = TEMP.trn MO= 1,30
      MATO[02] = TEMP.apn MO= 2,30
      MATO[03] = TEMP.sov MO= 8,30
      MATO[04] = TEMP.hv2 MO= 9,30
      MATO[05] = TEMP.hv3 MO= 10,30
      MATO[06] = TEMP.adr MO= 3,30
      MATO[07] = TEMP.psn MO= 4,30
      MATO[08] = TEMP.cr MO=22,30
      MATO[09] = TEMP.ab MO=23,30
      MATO[10] = TEMP.bm MO=24,30
      MATO[11] = TEMP.mr MO=25,30
      MATO[12] = TEMP.trp MO=1,4
      MATO[13] = TEMP.occ MO=2,3

; renumber OUT.MAT according to DJ.EQV
RENUMBER FILE=DJ.EQV, MISSINGZI=M, MISSINGZO=W
ENDRUN

;
LOOP INDEX2=1,13 ; Inner Loop for Each Summary Type:
;
;
;
IF (INDEX2=1)          ; Parameters for each table:
      SQFNAME='temp.psn'          ;
      MODE ='Motorized Person'    ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=2)
      SQFNAME='temp.trn'          ; - name of squeezed modal trip table(s)
      MODE ='Transit              ' ; - mode label od trip table
      DCML=0                      ; - decimal specification
      TABTYPE=1                  ; - table type(1/2)-involves 1 or 2 trip tables
      SCALE=1                    ; - scale factor to be applied (if desired)
      OPER='+'                   ; - operation(if tabtype=2) Tab1(?)Tab2=Result
ELSEIF (INDEX2=3 )
      SQFNAME='temp.trp'          ;
      MODE ='Transit Percentage'  ;
      DCML=1                      ;
      TABTYPE=2                  ;
      SCALE=100                  ;
      OPER='/'                   ;
ELSEIF (INDEX2=4)
      SQFNAME='temp.apn'          ;
      MODE ='Auto Person          ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;

```

```

ELSEIF (INDEX2=5)
      SQFNAME='temp.sov'          ;
      MODE ='SOV Person          ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=6 )
      SQFNAME='temp.hv2'          ;
      MODE ='HOV2Occ Person      ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=7 )
      SQFNAME='temp.hv3'          ;
      MODE ='HOV3+Occ Person     ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=8 )
      SQFNAME='temp.adr'          ;
      MODE ='Auto Driver          ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=9 )
      SQFNAME='temp.occ'          ;
      MODE ='Auto Occupancy      ' ;
      DCML=2                      ;
      TABTYPE=2                  ;
      SCALE=1                    ;
      OPER='/'                   ;
ELSEIF (INDEX2=10)
      SQFNAME='temp.cr           ' ;
      MODE ='Commuter Rail       ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=11)
      SQFNAME='temp.ab           ' ;
      MODE ='All Bus             ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=12)
      SQFNAME='temp.bm'          ;
      MODE ='Bus & Metrorail     ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ELSEIF (INDEX2=13)
      SQFNAME='temp.mr'          ;
      MODE ='Metrorail Only      ' ;
      DCML=0                      ;
      TABTYPE=1                  ;
      SCALE=1                    ;
      OPER='+'                   ;
ENDIF

;
RUN PGM=MATRIX
PAGEHEIGHT= 32767
ZONES=23

```

## Appendix C Cube Voyager Scripts

```

FILEI MATI=@SQFNAME@
ARRAY CSUM=23,CSUM1=23,CSUM2=23
;-----
; -- Table Cell Value decalaration or computation (in MW[1])
;-----

FILLMW MW[1]=MI.1.1,2      ; read input tables in MW 2,3

IF (@TABTYPE@ = 2)
FILLMW MW[2]=MI.1.1,2      ; read input tables in MW 2,3
ENDIF

IF (@TABTYPE@=2)           ; Cell Value
  JLOOP                   ; computed for
  IF (MW[3][J]>0) MW[1]=MW[2]*@SCALE@@OPER@MW[3]; special summaries-
  ENDJLOOP                ; calculation in MW[1]
ENDIF

;-----
; --- ROW Marginal declaration or computation ---
;-----
RSUM = ROWSUM(1)          ; 'normal' table- row summary value
denom = ROWSUM(3)
IF (@TABTYPE@=2)
  if (denom>0) RSUM = @SCALE@*ROWSUM(2)@OPER@ROWSUM(3) ; non-'normal' table
ENDIF
; compute the row marginal(%)

;-----
; --- COLUMN/Total Marginal Accumulation ---
; --- The computation (if necessary) is done below ---
;-----

JLOOP                   ; COL/Total Accumulation
CSUM[J] = CSUM[J] + MW[1][J] ; for 'normal' table
TOTAL = TOTAL + MW[1]
ENDJLOOP

IF (@TABTYPE@=2)
JLOOP                   ; COL/Total Accumulation
CSUM1[J] = CSUM1[J] + MW[2][J] ; for non-'normal' Table
TOTAL1 = TOTAL1 + MW[2]
CSUM2[J] = CSUM2[J] + MW[3][J]
TOTAL2 = TOTAL2 + MW[3]
ENDJLOOP
ENDIF

IF (I=1)                ; print header

PRINT LIST='/bt      ', '@DESCRIPT@'
PRINT LIST='      ', 'Purpose: ', '@PURPOSE@', '   MODE: ', '@MODE@'
PRINT LIST='      '

PRINT LIST='          DESTINATION'
PRINT LIST=' ORIGIN |',
' 1', ' 2', ' 3', ' 4',
' 5', ' 6', ' 7', ' 8', ' 9',
' 10', ' 11', ' 12', ' 13', ' 14',
' 15', ' 16', ' 17', ' 18', ' 19',
' 20', ' 21', ' 22', ' 23', ' | TOTAL'

PRINT LIST='=====',
'=====',
'=====',
'=====',
'=====',

```

```

ENDIF

IF (I=1)
  CURDIST=STR(I,2,1)+' DC CR'+ '|' ; Make row header
ELSEIF (I=2)
  CURDIST=STR(I,2,1)+' DC NC'+ '|' ; Make row header
ELSEIF (I=3)
  CURDIST=STR(I,2,1)+' MTG '+ '|' ; Make row header
ELSEIF (I=4)
  CURDIST=STR(I,2,1)+' PG '+ '|' ; Make row header
ELSEIF (I=5)
  CURDIST=STR(I,2,1)+' ARLCR'+ '|' ; Make row header
ELSEIF (I=6)
  CURDIST=STR(I,2,1)+' ARNCR'+ '|' ; Make row header
ELSEIF (I=7)
  CURDIST=STR(I,2,1)+' ALX '+ '|' ; Make row header
ELSEIF (I=8)
  CURDIST=STR(I,2,1)+' FFX '+ '|' ; Make row header
ELSEIF (I=9)
  CURDIST=STR(I,2,1)+' LDN '+ '|' ; Make row header
ELSEIF (I=10)
  CURDIST=STR(I,2,1)+' PW '+ '|' ; Make row header
ELSEIF (I=11)
  CURDIST=STR(I,2,1)+' FRD '+ '|' ; Make row header
ELSEIF (I=12)
  CURDIST=STR(I,2,1)+' CAR '+ '|' ; Make row header
ELSEIF (I=13)
  CURDIST=STR(I,2,1)+' HOW '+ '|' ; Make row header
ELSEIF (I=14)
  CURDIST=STR(I,2,1)+' AAR '+ '|' ; Make row header
ELSEIF (I=15)
  CURDIST=STR(I,2,1)+' CAL '+ '|' ; Make row header
ELSEIF (I=16)
  CURDIST=STR(I,2,1)+' STM '+ '|' ; Make row header
ELSEIF (I=17)
  CURDIST=STR(I,2,1)+' CHS '+ '|' ; Make row header
ELSEIF (I=18)
  CURDIST=STR(I,2,1)+' FAU '+ '|' ; Make row header
ELSEIF (I=19)
  CURDIST=STR(I,2,1)+' STA '+ '|' ; Make row header
ELSEIF (I=20)
  CURDIST=STR(I,2,1)+' CL/JF'+ '|' ; Make row header
ELSEIF (I=21)
  CURDIST=STR(I,2,1)+' SP/FB'+ '|' ; Make row header
ELSEIF (I=22)
  CURDIST=STR(I,2,1)+' KGEO '+ '|' ; Make row header
ELSEIF (I=23)
  CURDIST=STR(I,2,1)+' EXTL '+ '|' ; Make row header
ELSE ; (I=24)
  CURDIST=STR(I,2,1)+' TOTAL'+ '|' ; Make row header
ENDIF

PRINT FORM=7.@DCML@ LIST=CURDIST, MW[1][1],MW[1][2],MW[1][3],MW[1][4],MW[1][5],
MW[1][6],MW[1][7],MW[1][8],MW[1][9],MW[1][10],
MW[1][11],MW[1][12],MW[1][13],MW[1][14],MW[1][15],
MW[1][16],MW[1][17],MW[1][18],MW[1][19],MW[1][20],
MW[1][21],MW[1][22],MW[1][23], ' | ', RSUM

IF (I==ZONES)
; Now at the end of Processed zone matrix
; Do final Column/Grand Total Computations
IF (@TABTYPE@=2)
  LOOP IDX = 1,ZONES
  IF (CSUM2[IDX] = 0)
    CSUM[IDX] = 0
  ELSE
    CSUM[IDX] = @SCALE@* CSUM1[IDX] @OPER@ CSUM2[IDX]
  ENDIF
ENDIF

```



## 22 MFARE1.S

```

=====
; MFARE1.S
; V2.3 Model
; Script Version of MFARE1 script
; Walk and Drive Access Metrorail Sta. to Sta. fares developed
; for AM Peak and off-peak periods
;
; Programmer: Milone
; Date: 1/11/07
; Metro station XY file name corrected (12/13/2006)
=====
STATSIZE = 150 ; No. of Metrorail Stations (Note: Max is
999)
MR_DST_FTR = 0.01 ; Factor to convert input skimmed Metrorail
distance units to whole miles
;-----
; Filenames:
MSTA_XYs = 'MFARE1.A1' ; Metrorail Sta XYs coords scaled
so computed units are in 1/100ths of miles
MSTA_Dst_Skims = 'RLDIST.SKM' ; Metrorail Sta/Sta Distance Skims
(Distance units: 1/100ths of miles)
MSTA_Discount = 'INPUTS\mfare1_Sta_Disc.ASC' ; Metrorail Sta fare discount array
in cents
MSTA_Tariff = 'INPUTS\tariff.txt' ; WMATA tariff policy
AM_Sta_Fares = 'AM_Metrorail_Fares.TXT' ; Output AM Statio-to-Station Fares
-text file
OP_Sta_Fares = 'OP_Metrorail_Fares.TXT' ; Output OP Statio-to-Station Fares
-text file
RUN PGM=MATRIX
ZONES=@STATSIZE@
read FILE=@MSTA_Tariff@
;
; Set up zone arrays for accumulating I/O variables
;
;
;-----
; Read Station Coordinate file =
;-----
LOOKUP Name=StaXYS,
LOOKUP[1] = 1,Result = 2, ; Xcrds
LOOKUP[2] = 1,Result = 3, ; YCrds
Interpolate = N, FAIL=0,0,0,list=Y,file=@MSTA_XYS@
;
;-----
; Read Station Fare Discount Lookup =
; - The station-specific discount values are in cents. =
; The discounts are subtracted from the final =
; computed fares to/from the station =
;-----
LOOKUP Name=StaDSC,
LOOKUP[1] = 1,Result = 2, ; AM Fare Discount in cents

```

```

LOOKUP[2] = 1,Result = 3, ; OP Fare Discount in cents
Interpolate = N, FAIL=0,0,0,list=Y,file=@MSTA_Discount@
;-----
; Over-the Rail Distance Skims =
;-----
FILEI MATI = RLDIST.SKM
MW[1]= MI.1.1 ; (Over-the-rail distance in 1/100s mi)
ROWSUM1 = ROWSUM(1)
;-----
; Now, loop through each station i/j, compute composite distance,
; and compute AM and Off peak fares. Use generalized cost
; calculation:
;
; FARE = (incremental cost + Rate*Distance) <-Per short Distance +
; (incremental cost + Rate*Distance) <-Per medium Distance +
; (incremental cost + Rate*Distance) <-Per long Distance
;-----
IF (rowsum1 > 0) ; exclude unused stations
JLOOP
IF (MW[1] !=0 || I=J) ; exclude station i/j's that are 'unused'
; Calculate airline distance (MW[2]) in 100s of miles
IxCrds = StaXYS(1,I)
JxCrds = StaXYS(1,J)
IyCrds = StaXYS(2,I)
JyCrds = StaXYS(2,J)
MW[2] = ((IxCrds-JxCrds)^2 + (IyCrds-JyCrds)^2)^0.5
; Calculate Composite (airline/over-the rail) distance MW[3] in whole miles
MW[3] = ((MW[1] + MW[2]) / 2.0) * @MR_Dst_Ftr@
; Calculate peak (MW[10]) and off-peak fares (MW[20]) based on comp distance
; Fares computed units in non-defaulted cents
;
; Peak Fare Calculation: -----
-----
PkDist1 = Pk_Fare_Dist1
PkDist2 = Pk_Fare_Dist1 + Pk_Fare_Dist2
IF (MW[3] <= PkDist1)
MW[10] = Pk_Fare_Incr1 + (Pk_Fare_Rate1 * MW[3])
ELSEIF (MW[3] > PkDist1 && MW[3] <= PkDist2)
MW[10] = Pk_Fare_Incr1 + (Pk_Fare_Rate1 * Pk_Fare_Dist1) +
Pk_Fare_Incr2 + (Pk_Fare_Rate2 * (MW[3] - PkDist1))
ELSEIF (MW[3] > PkDist2)
MW[10] = Pk_Fare_Incr1 + (Pk_Fare_Rate1 * Pk_Fare_Dist1) +
Pk_Fare_Incr2 + (Pk_Fare_Rate2 * Pk_Fare_Dist2) +
Pk_Fare_Incr3 + (Pk_Fare_Rate3 * (MW[3] - PkDist2))
ENDIF
; Round computed AM fare MW[10] to nearest nickle as in original program Final
Fare
; is 'FinAMFare'
FARE = MW[10]

```



## Appendix C Cube Voyager Scripts

```

temp1      = INT(Fare/10.0)
temp2      = temp1 * 10.0
DiffCheck  = Fare - temp2
IF         (DiffCheck < 2.5)
    FinAMFare = temp2
ELSEIF    (DiffCheck > 7.5)
    FinAMFare = temp2 + 10.0
ELSE
    FinAMFare = temp2 + 5.0
ENDIF

;   Impose Max Fare rule
    If (FinAMFare > Pk_Fare_Max)  FinAMFare = Pk_Fare_Max

;   Apply AM station discounts if used
    FinAMFare = FinAMFare - StaDSC(1,I) - StaDsc(1,J)

;   Compute IJ Index so station-to-station fares can be read in as a lookup
;   Index merges separate I/J numbers into one number (index for station 1 to
station 1 is '1001')
    IJindex = (I * 1000.0) + J

;   Write out the AM Fares:
    Print List = I(5),J(5), FinAMFare(6),IJindex(7),' ; ', MW[10](6),
MW[1](10.0),MW[2](10.0),MW[3](10.2),
    IxCrd(7), JxCrd(7), IyCrd(7), JyCrd(7),
    PkDist1(10.2),PkDist2(10.2),
    '<
I/J/AM_Fare_n5/AM_Fare/R_Dst100s/A_Dst100s/CmpDstMi/iXcrd/jXcrd/iYcrd/jYcrdI/Dist1/D
ist2',
    File=@AM_Sta_Fares@

;   END of Peak Fare Calculation -----
-----

;   Off-Peak Calculation: -----
-----

OpDist1 = Op_Fare_Dist1
OpDist2 = Op_Fare_Dist1 + Op_Fare_Dist2

IF      (MW[3] <= OpDist1)
    MW[20] = Op_Fare_Incr1 + (Op_Fare_Rate1 * MW[3])
ELSEIF  (MW[3] > OpDist1 && MW[3] <= OpDist2)
    MW[20] = Op_Fare_Incr1 + (Op_Fare_Rate1 * Op_Fare_Dist1) +
    Op_Fare_Incr2 + (Op_Fare_Rate2 * (MW[3] - OpDist1))
ELSEIF  (MW[3] > OpDist2)
    MW[20] = Op_Fare_Incr1 + (Op_Fare_Rate1 * Op_Fare_Dist1) +
    Op_Fare_Incr2 + (Op_Fare_Rate2 * Op_Fare_Dist2) +
    Op_Fare_Incr3 + (Op_Fare_Rate3 * (MW[3] - OpDist2))
ENDIF

;   Round computed Off-peak fare MW[20] to nearest nickle as in original program
Final Fare
;   is 'FinOPFare'
    FARE      = MW[20]
temp1      = INT(Fare/10.0)
temp2      = temp1 * 10.0
DiffCheck  = Fare - temp2
IF         (DiffCheck < 2.5)

```

```

    FinOPFare = temp2
ELSEIF    (DiffCheck > 7.5)
    FinOPFare = temp2 + 10.0
ELSE
    FinOPFare = temp2 + 5.0
ENDIF

;   Impose Max Fare rule
    If (FinOPFare > Op_Fare_Max)  FinOPFare = Op_Fare_Max

;   Apply Off-pk station discounts if used
    FinOPFare = FinOPFare - StaDSC(2,I) - StaDsc(2,J)

;   Compute IJ Index so station-to-station fares can be read in as a lookup
;   Index merges separate I/J numbers into one number (index for station 1 to
station 1 is '1001')
    IJindex = (I * 1000.0) + J

;   Write out the Off-Pk Fares:
    Print List = I(5),J(5),FinOPFare(6),IJindex(7),' ; ', MW[20](6),
MW[1](10.0),MW[2](10.0),MW[3](10.2),
    IxCrd(7), JxCrd(7), IyCrd(7), JyCrd(7),
    OpDist1(10.2),OpDist2(10.2),
    '<
I/J/OP_Fare/R_Dst100s/A_Dst100s/CmpDstMi/iXcrd/jXcrd/iYcrd/jYcrdI/Dist1/Dist2',
    File=@OP_Sta_Fares@

;   END of Off Peak Fare Calculation -----
-----

    ENDIF
ENDJLOOP
endif
ENDRUN

```

## 23 MFARE2.S

```

;=====
; MFARE2.S
; Version 2.3 Model
; TP+ Script Version of MFARE2 Program
; Walk and Drive Access Zonal Fares Developed for AM Peak and Off-Peak Periods
;
; Programmer: Milone
; Date: 12/11/10
;
;
; Update 2/21/07 to support nested logit work
; 01/03/08 JainM
; Update for including LRT in MR path. Use BUSFARAM.ASC and BUSFAROP.ASC in MR
paths.
; Condition the fares for Metrorail only path. Zero out fare for i/j with no transit
path.
;=====
; Set up tokens to either use or comment out commands for Cube Cluster (distributed
processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
    dp_token = ' '

```

## Appendix C Cube Voyager Scripts

```

else
  dp_token = ';'
endif
; useIdp = t (true) or f (false); this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=f

ZONESIZE      = 3722          ; No. of TAZs
LastIzn       = 3675          ; Last Internal TAZ No.
STATSIZE      = 150           ; Max No. of Metrorail Stations
BFZ_Size      = 21            ; No. of Bus Fare Zones
; -----
MSTA_Tariff   = 'INPUTS\tariff.txt'
TRN_Defl      = 'TRN_Deflator.txt'
; LOOP Through the Time Period/Access Mode combinations
;   - define I/P & O/P files:
;
LOOP PRDACC = 1,22

===== ; ; COMMUTER RAIL FARES
IF (PRDACC = 1) ; ----- AM Walk Access cycle:
-----
  USTOSFile      = '%_iter_%_AM_WK_CR.STA ' ; Input: Walk Acc. Station
  to Station Matrix (Brd Sta/T1, Ali Stat/T2)
  TRSkimFile     = '%_iter_%_AM_WK_CR.SKM ' ; Walk Acc. CR
  Transit Skims
  MR_FareFile    = 'AM_Metrorail_Fares.TXT ' ; Metrorail Fares in
  Current Year Cents
  BusFareMTX     = 'INPUTS\BUSFARAM.ASC' ; Bus Fare matrix
  21x21 (Bus fares zones '1' to '21')
  MF2ZonalDeck  = 'FARE_A2.ASC' ; Zonal A2 Deck
  (Bus fares zones referenced as '1' to '21')
  OutputMatrix   = '%_iter_%_AM_WK_CR.FAR ' ; Output: Total Fare Matrix
  OutputMatrix5  = '%_iter_%_AM_WK_CR.FR5 ' ; Fare Matrix (T1-5
  Total,bus onlr, rail, acc, egr fare file)
  OutputText     = '%_iter_%_AM_WK_CR.TXT ' ; Fare text file for
  checking fare components / selected ijs
  ELSEIF (PRDACC = 2) ; ----- AM Drive Access
  cycle: -----
    USTOSFile      = '%_iter_%_AM_DR_CR.STA ' ;
    TRSkimFile     = '%_iter_%_AM_DR_CR.SKM ' ;
    MR_FareFile    = 'AM_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFARAM.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_AM_DR_CR.FAR ' ;
    OutputMatrix5  = '%_iter_%_AM_DR_CR.FR5 ' ;
    OutputText     = '%_iter_%_AM_DR_CR.TXT ' ;
  ELSEIF (PRDACC = 3) ; ----- Off-Pk Walk Access
  cycle: -----
    USTOSFile      = '%_iter_%_OP_WK_CR.STA ' ;
    TRSkimFile     = '%_iter_%_OP_WK_CR.SKM ' ;
    MR_FareFile    = 'OP_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFAROP.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_OP_WK_CR.FAR ' ;
    OutputMatrix5  = '%_iter_%_OP_WK_CR.FR5 ' ;
    OutputText     = '%_iter_%_OP_WK_CR.TXT ' ;
  ELSEIF (PRDACC = 4) ; ----- Off-Pk Drive Access
  cycle: -----
    USTOSFile      = '%_iter_%_OP_DR_CR.STA ' ;
    TRSkimFile     = '%_iter_%_OP_DR_CR.SKM ' ;
    MR_FareFile    = 'OP_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFAROP.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_OP_DR_CR.FAR ' ;
    OutputMatrix5  = '%_iter_%_OP_DR_CR.FR5 ' ;
    OutputText     = '%_iter_%_OP_DR_CR.TXT ' ;

```

```

===== ; ; METRO RAIL ONLY FARES
ELSEIF (PRDACC = 5) ; ----- AM Walk Access cycle:
-----
  USTOSFile      = '%_iter_%_AM_WK_MR.STA ' ;
  TRSkimFile     = '%_iter_%_AM_WK_MR.SKM ' ;
  MR_FareFile    = 'AM_Metrorail_Fares.TXT ' ;
  BusFareMTX     = 'INPUTS\BUSFARAM.ASC' ;
  MF2ZonalDeck  = 'FARE_A2.ASC' ;
  OutputMatrix   = '%_iter_%_AM_WK_MR.FAR ' ;
  OutputMatrix5  = '%_iter_%_AM_WK_MR.FR5 ' ;
  OutputText     = '%_iter_%_AM_WK_MR.TXT ' ;
  ELSEIF (PRDACC = 6) ; ----- AM Drive Access
  cycle: -----
    USTOSFile      = '%_iter_%_AM_DR_MR.STA ' ;
    TRSkimFile     = '%_iter_%_AM_DR_MR.SKM ' ;
    MR_FareFile    = 'AM_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFARAM.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_AM_DR_MR.FAR ' ;
    OutputMatrix5  = '%_iter_%_AM_DR_MR.FR5 ' ;
    OutputText     = '%_iter_%_AM_DR_MR.TXT ' ;
  ELSEIF (PRDACC = 7) ; ----- AM KNR Access
  cycle: -----
    USTOSFile      = '%_iter_%_AM_KR_MR.STA ' ;
    TRSkimFile     = '%_iter_%_AM_KR_MR.SKM ' ;
    MR_FareFile    = 'AM_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFARAM.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_AM_KR_MR.FAR ' ;
    OutputMatrix5  = '%_iter_%_AM_KR_MR.FR5 ' ;
    OutputText     = '%_iter_%_AM_KR_MR.TXT ' ;
  ELSEIF (PRDACC = 8) ; ----- Off-Pk Walk Access
  cycle: -----
    USTOSFile      = '%_iter_%_OP_WK_MR.STA ' ;
    TRSkimFile     = '%_iter_%_OP_WK_MR.SKM ' ;
    MR_FareFile    = 'OP_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFAROP.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_OP_WK_MR.FAR ' ;
    OutputMatrix5  = '%_iter_%_OP_WK_MR.FR5 ' ;
    OutputText     = '%_iter_%_OP_WK_MR.TXT ' ;
  ELSEIF (PRDACC = 9) ; ----- Off-Pk Drive Access
  cycle: -----
    USTOSFile      = '%_iter_%_OP_DR_MR.STA ' ;
    TRSkimFile     = '%_iter_%_OP_DR_MR.SKM ' ;
    MR_FareFile    = 'OP_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFAROP.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_OP_DR_MR.FAR ' ;
    OutputMatrix5  = '%_iter_%_OP_DR_MR.FR5 ' ;
    OutputText     = '%_iter_%_OP_DR_MR.TXT ' ;
  ELSEIF (PRDACC = 10) ; ----- Off-Pk KNR Access
  cycle: -----
    USTOSFile      = '%_iter_%_OP_KR_MR.STA ' ;
    TRSkimFile     = '%_iter_%_OP_KR_MR.SKM ' ;
    MR_FareFile    = 'OP_Metrorail_Fares.TXT ' ;
    BusFareMTX     = 'INPUTS\BUSFAROP.ASC' ;
    MF2ZonalDeck  = 'FARE_A2.ASC' ;
    OutputMatrix   = '%_iter_%_OP_KR_MR.FAR ' ;
    OutputMatrix5  = '%_iter_%_OP_KR_MR.FR5 ' ;
    OutputText     = '%_iter_%_OP_KR_MR.TXT ' ;
; ; ALL BUS FARES
=====
ELSEIF (PRDACC = 11) ; ----- AM Walk Access cycle:
-----
  USTOSFile      = '%_iter_%_AM_WK_AB.STA ' ;

```

## Appendix C Cube Voyager Scripts

```

TRSkimFile      = '%_iter_%_AM_WK_AB.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_WK_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_AM_WK_AB.FR5 ' ;
OutputText      = '%_iter_%_AM_WK_AB.TXT ' ;
ELSEIF (PRDACC =12) ; ----- AM Drive Access
cycle: -----
USTOSFile       = '%_iter_%_AM_DR_AB.STA ' ;
TRSkimFile      = '%_iter_%_AM_DR_AB.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_DR_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_AM_DR_AB.FR5 ' ;
OutputText      = '%_iter_%_AM_DR_AB.TXT ' ;
ELSEIF (PRDACC =13) ; ----- AM KNR Access
cycle: -----
USTOSFile       = '%_iter_%_AM_KR_AB.STA ' ;
TRSkimFile      = '%_iter_%_AM_KR_AB.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_KR_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_AM_KR_AB.FR5 ' ;
OutputText      = '%_iter_%_AM_KR_AB.TXT ' ;
ELSEIF (PRDACC =14) ; ----- Off-Pk Walk Access
cycle: -----
USTOSFile       = '%_iter_%_OP_WK_AB.STA ' ;
TRSkimFile      = '%_iter_%_OP_WK_AB.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_WK_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_WK_AB.FR5 ' ;
OutputText      = '%_iter_%_OP_WK_AB.TXT ' ;
ELSEIF (PRDACC =15) ; ----- Off-Pk Drive Access
cycle: -----
USTOSFile       = '%_iter_%_OP_DR_AB.STA ' ;
TRSkimFile      = '%_iter_%_OP_DR_AB.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_DR_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_DR_AB.FR5 ' ;
OutputText      = '%_iter_%_OP_DR_AB.TXT ' ;
ELSEIF (PRDACC =16) ; ----- Off-Pk KNR Access
cycle: -----
USTOSFile       = '%_iter_%_OP_KR_AB.STA ' ;
TRSkimFile      = '%_iter_%_OP_KR_AB.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_KR_AB.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_KR_AB.FR5 ' ;
OutputText      = '%_iter_%_OP_KR_AB.TXT ' ;

=====
ELSEIF (PRDACC =17) ; ----- AM Walk Access cycle:
-----
USTOSFile       = '%_iter_%_AM_WK_BM.STA ' ;
TRSkimFile      = '%_iter_%_AM_WK_BM.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_WK_BM.FAR ' ;

```

```

OutputMatrix5   = '%_iter_%_AM_WK_BM.FR5 ' ;
OutputText      = '%_iter_%_AM_WK_BM.TXT ' ;
ELSEIF (PRDACC =18) ; ----- AM Drive Access
cycle: -----
USTOSFile       = '%_iter_%_AM_DR_BM.STA ' ;
TRSkimFile      = '%_iter_%_AM_DR_BM.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_DR_BM.FAR ' ;
OutputMatrix5   = '%_iter_%_AM_DR_BM.FR5 ' ;
OutputText      = '%_iter_%_AM_DR_BM.TXT ' ;
ELSEIF (PRDACC =19) ; ----- AM KNR Access
cycle: -----
USTOSFile       = '%_iter_%_AM_KR_BM.STA ' ;
TRSkimFile      = '%_iter_%_AM_KR_BM.SKM ' ;
MR_FareFile     = 'AM_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFARAM.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_AM_KR_BM.FAR ' ;
OutputMatrix5   = '%_iter_%_AM_KR_BM.FR5 ' ;
OutputText      = '%_iter_%_AM_KR_BM.TXT ' ;
ELSEIF (PRDACC =20) ; ----- Off-Pk Walk Access
cycle: -----
USTOSFile       = '%_iter_%_OP_WK_BM.STA ' ;
TRSkimFile      = '%_iter_%_OP_WK_BM.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_WK_BM.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_WK_BM.FR5 ' ;
OutputText      = '%_iter_%_OP_WK_BM.TXT ' ;
ELSEIF (PRDACC =21) ; ----- Off-Pk Drive Access
cycle: -----
USTOSFile       = '%_iter_%_OP_DR_BM.STA ' ;
TRSkimFile      = '%_iter_%_OP_DR_BM.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_DR_BM.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_DR_BM.FR5 ' ;
OutputText      = '%_iter_%_OP_DR_BM.TXT ' ;
ELSEIF (PRDACC =22) ; ----- Off-Pk KR Access
cycle: -----
USTOSFile       = '%_iter_%_OP_KR_BM.STA ' ;
TRSkimFile      = '%_iter_%_OP_KR_BM.SKM ' ;
MR_FareFile     = 'OP_Metrorail_Fares.TXT ' ;
BusFareMTX      = 'INPUTS\BUSFAROP.ASC'   ;
MF2ZonalDeck    = 'FARE_A2.ASC'          ;
OutputMatrix    = '%_iter_%_OP_KR_BM.FAR ' ;
OutputMatrix5   = '%_iter_%_OP_KR_BM.FR5 ' ;
OutputText      = '%_iter_%_OP_KR_BM.TXT ' ;

ENDIF

RUN PGM=MATRIX
@dp_token@DistributeIntrastep processId='mw cog', ProcessList=@subnode%
ZONES=@ZONESIZE@
;
read FILE=@MSTA_Tariff@
read FILE=@TRN_Defl@
;=====
; Read Station-to-Station Metrorail Fares as lookups =
; Fares read in based on IJ index =
; e.g., '1001' means 1 to 1 and '150150' means 150 to 150 =
; =
;=====

```

## Appendix C Cube Voyager Scripts

```

;
LOOKUP Name=STA_Fares,
LOOKUP[1] = 4,Result = 3, ; station to station fares
Interpolate = N, FAIL=0,0,0,list=N,file=@MR_FareFile@
;
;=====
; Read Bus Fare zone to Bus fare zone matrix =
; Fares are indexed to origin-end bus fare zone 'row';lookup =
; no. corresponds to a destin-end bus fare zone 'column' =
;=====
;
LOOKUP Name=BusFrMTX,
LOOKUP[01] = 1,Result = 2, ;
LOOKUP[02] = 1,Result = 3, ;
LOOKUP[03] = 1,Result = 4, ;
LOOKUP[04] = 1,Result = 5, ;
LOOKUP[05] = 1,Result = 6, ;
LOOKUP[06] = 1,Result = 7, ;
LOOKUP[07] = 1,Result = 8, ;
LOOKUP[08] = 1,Result = 9, ;
LOOKUP[09] = 1,Result =10, ;
LOOKUP[10] = 1,Result =11, ;
LOOKUP[11] = 1,Result =12, ;
LOOKUP[12] = 1,Result =13, ;
LOOKUP[13] = 1,Result =14, ;
LOOKUP[14] = 1,Result =15, ;
LOOKUP[15] = 1,Result =16, ;
LOOKUP[16] = 1,Result =17, ;
LOOKUP[17] = 1,Result =18, ;
LOOKUP[18] = 1,Result =19, ;
LOOKUP[19] = 1,Result =20, ;
LOOKUP[20] = 1,Result =21, ;
LOOKUP[21] = 1,Result =22, ;
Interpolate = N, FAIL=0,0,0,list=N,file=@BusFareMTX@

;; read Zone data file
LOOKUP Name=TAZLook,
LOOKUP[01] = 1,Result = 2, ; BusFare Zn 1 (1-21)
LOOKUP[02] = 1,Result = 3, ; BusFare Zn 2 (1-21)
LOOKUP[03] = 1,Result = 4, ; Orig Walk Pct in 10ths of pct
('1000'=100%)
LOOKUP[04] = 1,Result = 5, ; Dest Walk Pct in 10ths of pct
('1000'=100%)
LOOKUP[05] = 1,Result = 6, ; BusFare Zn 1 associated w/ Metro
station (1-21)
LOOKUP[06] = 1,Result = 7, ; BusFare Zn 2 associated w/ Metro
station (1-21)
LOOKUP[07] = 1,Result = 8, ; Jurcode: 0/DC, 1/MD, 2/VA Area1, 3/VA
Area2
LOOKUP[08] = 1,Result = 9, ; Origin-end Bus Fare Override value
(in current yr cents)
LOOKUP[09] = 1,Result =10, ; Destin-end Bus Fare Override value
(in current yr cents)
Interpolate = N, FAIL=0,0,0,list=N,file=@MF2ZonalDeck@

;
; Establish Discount Array
;
ARRAY RB_Disc = 4

RB_Disc[1] = DC_RailBus_Disc
RB_Disc[2] = MD_RailBus_Disc
RB_Disc[3] = VA1_RailBusDisc
RB_Disc[4] = VA2_RailBusDisc

IF (TAZLook(7,I) > 3 || TAZLook(7,I) < 0)
LIST = 'Jurisdiction Code NOT within convention values; I Quit'

```

```

ABORT
ENDIF

IF (TAZLook(3,I) < 0 || TAZLook(3,I) > 1000.0 )
LIST = 'Orig. Walk Pcts NOT within tolerances(0.0 to 1000.0) ; I
Quit'

ABORT
ENDIF

IF (TAZLook(4,I) < 0 || TAZLook(4,I) > 1000.0 )
LIST = 'Destin. Walk Pcts NOT within tolerances(0.0 to 1000.0) ; I
Quit'

ABORT
ENDIF

IF (TAZLook(1,I) > @BFZ_SIZE@ || TAZLook(2,I) > @BFZ_SIZE@ ||
TAZLook(5,I) > @BFZ_SIZE@ || TAZLook(6,I) > @BFZ_SIZE@)
LIST = 'Zonal / Metrorail Bus Fare Zn No. equivalence exceeds:
','@BFZ_SIZE@',' : I Quit'
ABORT
ENDIF

;=====
; Read in the USTOS files here & Declare output matrix =
;=====

MATI[01] = @USTOSFile@
MW[11] = MI.1.1 ; On-Station
MW[12] = MI.1.2 ; Off-Station

MATI[02] = @TRSkimFile@
MW[13] = MI.2.1 ;---- ivt-local bus (0.01 min)
MW[14] = MI.2.2 ;---- ivt-exp bus (0.01 min)
MW[15] = MI.2.3 ;---- ivt-metrorail (0.01 min)
MW[16] = MI.2.4 ;---- ivt-commuter rail (0.01 min)
MW[17] = MI.2.5 ;---- ivt-light rail (0.01 min)
MW[18] = MI.2.6 ;---- ivt-new mode (0.01 min)

MATO[1]= @OutputMatrix@,MO=21,FORMAT=MINUTP; total deflated fare/t1
MATO[2]= @OutputMatrix5@,MO=41-45 ; total deflated fare/t1,
; busonly(undef1) /t2,
; rail(undef) /t3,
; acc(undef) /t4,
; egr(undef) /t5

;=====
; Now begin i/j level fare calculation process =
;=====

JLOOP

MW[19] = MW[13]+MW[14]+MW[15]+MW[16]+MW[17]+MW[18] ; total transit in-vehicle
time
MW[20] = MW[13]+MW[14]+MW[16]+MW[17]+MW[18] ; Non-Metrorail in-vehicle
time

IF (I > @LastIZN@ || J > @LastIZN@) Continue ; Skip current
i/j if either is external
; Start afresh all fare related variables at the current i/j

BusFare = 0.0

```

## Appendix C Cube Voyager Scripts

```

RailFare      = 0.0
RailAccFare   = 0.0
RailEgrFare   = 0.0
TotalFare     = 0.0
TotalFareDef  = 0.0
IBFZ1        = 0.0
IBFZ2        = 0.0
JBFZ1        = 0.0
JBFZ2        = 0.0
Acc_NoWlk_Prop = 0.0
Egr_NoWlk_Prop = 0.0
ISTA         = 0.0
JSTA         = 0.0
IJIDX        = 0.0
RailFare     = 0.0
RIBFZ1      = 0.0
RIBFZ2      = 0.0
RJBFZ1      = 0.0
RJBFZ2      = 0.0
_AccFare1   = 0.0
_AccFare2   = 0.0
_AccFare12  = 0.0
_EgrFare1   = 0.0
_EgrFare2   = 0.0
_EgrFare12  = 0.0
RailAccFare = 0.0
RailEgrFare = 0.0
Acc_Discount = 0.0
Egr_Discount = 0.0
I_FareOvr   = 0.0
J_FareOvr   = 0.0

; Make sure station numbers are appropriate:
IF (MW[11] > @STATSIZE@ || MW[12] > @STATSIZE@ )
  LIST = 'USTOS Station number(s) are out of range; I Quit'
  ABORT
ENDIF

;
; Define Rail-to-bus fare discount. The discount will be applied
; at the access end and egress - end on a 50/50 basis (per MFARE2)
;
AccRBDx = TAZLook(7,I) + 1 ; convert JurCode 0-3 to Rail/Bus
discount array index 1-4
EgrRBDx = TAZLook(7,J) + 1 ;
Acc_Discount = RB_Disc[AccRBDx] * 0.50
Egr_Discount = RB_Disc[EgrRBDx] * 0.50

; Lookup Bus Fares
;
IBFZ1 = TAZLOOK(1,I)
IBFZ2 = TAZLOOK(2,I)
IF (IBFZ2 = 0) IBFZ2 = IBFZ1

JBFZ1 = TAZLOOK(1,J)
JBFZ2 = TAZLOOK(2,J)
IF (JBFZ2 = 0) JBFZ2 = JBFZ1

;
; Define Zonal Non-walk area percentages at
; Access end and egress end:
Acc_NoWlk_Prop = 1.0 - (TAZLOOK(3,I)/1000.0) ; Zonal non-walk proportion to
station (Access-end)
Egr_NoWlk_Prop = 1.0 - (TAZLOOK(4,J)/1000.0) ; Zonal non-walk proportion to
station (Egress-end)

```

```

;-----
; If no transit path exists for i/j then zero-out fares
;-----
IF (MW[19][j] = 0 )

  TotalFare      = 0.0
  TotalFareDef   = 0.0
  BusFare        = 0.0
  RailFare       = 0.0
  _AccFare12     = 0.0
  _EgrFare12     = 0.0

  MW[21][j]     = TotalFareDef

  MW[41][j]     = TotalFareDef
  MW[42][j]     = BusFare
  MW[43][j]     = RailFare
  MW[44][j]     = _AccFare12
  MW[45][j]     = _EgrFare12

ELSEIF (MW[11][j] = 0 && MW[12][j] = 0) ;
;-----
; ..ElseIf no rail stations used, compute Bus fare (BUSFARE)
;-----
  BusFare = (BusFrMTX(JBFZ1,IBFZ1) +
             BusFrMTX(JBFZ2,IBFZ1) +
             BusFrMTX(JBFZ1,IBFZ2) +
             BusFrMTX(JBFZ2,IBFZ2)) * 0.250

  TotalFare = BusFare + RailFare + RailAccFare + RailEgrFare ;
  undeinflated transit fare, Bus-Only paths
  TotalFareDef = Round(TotalFare * DeflationPTR)
  MW[21][j] = TotalFareDef

  MW[41][j] = TotalFareDef
  MW[42][j] = BusFare
  MW[43][j] = RailFare
  MW[44][j] = _AccFare12
  MW[45][j] = _EgrFare12

ELSE
;-----
; ... Else compute rail related fares
; if USTOS stations exist for current I/J
;-----
  ISTA = MW[11][J] ; Origin Metrorail
  Station No.
  JSTA = MW[12][J] ; Destin Metrorail
  Station No.
  IJIDX = ISTA*1000.0 + JSTA ; Sta I/J index,
('0001001' means from sta# 1 to sta#1)
  RailFare = Sta_Fares(1,IJIDX) ; Fare from current Sta.I
  to Sta.J

;
; Define Station-related Bus Fare Zones
; Access-End:

RIBFZ1 = TAZLOOK(5,ISTA)
RIBFZ2 = TAZLOOK(6,ISTA)
IF (RIBFZ2 = 0) RIBFZ2 = RIBFZ1

; Egress-End:

RJBFBZ1 = TAZLOOK(5,JSTA)

```

## Appendix C Cube Voyager Scripts

```

RJBZF2 = TAZLOOK(6,JSTA)
IF (RJBZF2 = 0) RJBZF2 = RJBZF1

  _AccFare1 = MIN(BusFrMTX(RIBFZ1,IBFZ1),BusFrMTX(RIBFZ2,IBFZ1))
  _AccFare2 = MIN(BusFrMTX(RIBFZ1,IBFZ2),BusFrMTX(RIBFZ2,IBFZ2))
  _AccFare12= ((_AccFare1 + _AccFare2) * 0.50) - Acc_Discount

  _EgrFare1 = MIN(BusFrMTX(JBFZ1,RJBZF1),BusFrMTX(JBFZ1,RJBZF2))
  _EgrFare2 = MIN(BusFrMTX(JBFZ2,RJBZF1),BusFrMTX(JBFZ2,RJBZF2))
  _EgrFare12= ((_EgrFare1 + _EgrFare2) * 0.50) - Egr_Discount

  RailAccFare = _AccFare12 * Acc_NoWlk_Prop
  RailEgrFare = _EgrFare12 * Egr_NoWlk_Prop

; If Only Metrorail is used then TotalFare equals RailFare

  IF (MW[20][j] = 0)

    TotalFare = RailFare ; undeflated
transit fare, Metrorail Only-Related paths

    BusFare = 0.0
    _AccFare12 = 0.0
    _EgrFare12 = 0.0
    RailAccFare = 0.0
    RailEgrFare = 0.0

  ELSE

    TotalFare = BusFare + RailFare + RailAccFare + RailEgrFare ; undeflated
transit fare, Metrorail-Related paths

  ENDIF

; If an I/J override value exists use it instead of the total computed fare
value

  IF ( TAZLOOK(8,I) > 0.0 )
    I_FareOvr = TAZLOOK(8,I)
    TotalFare = I_FareOvr
  ENDIF
  IF ( TAZLOOK(9,J) > 0.0 )
    J_FareOvr = TAZLOOK(9,J)
    TotalFare = J_FareOvr
  ENDIF

; Apply Deflator to Total fare to write out constant dollars

  TotalFareDef = Round(TotalFare * DeflationFTR)
  MW[21][j] = TotalFareDef

  MW[41][j] = TotalFareDef
  MW[42][j] = BusFare
  MW[43][j] = RailFare
  MW[44][j] = _AccFare12
  MW[45][j] = _EgrFare12

ENDIF

; write out the results of sample IJs here:
IF (i = 8, 64, 345, 362, 464, 578, 829, 927, 1043, 1231, 1236, 1337,
1537, 1554, 1619, 1698, 1716, 1842, 1942, 1967 &
j = 8, 64, 345, 362, 1231, 1236, 1337, 1537)

  print Form=7.1 list= i(6),j(6),TotalFareDef(6), TotalFare, BusFare,
RailFare, _AccFare12,_EgrFare12,

```

```

I_FareOvr,J_FareOvr,
' <<-
I/J/DefFare/UnDefFare/BusFare/RailFare/AccFare/EgrFare/IOvrFare/JOvrFare/',
file=@Outputtext@
ENDIF

ENDJLOOP

ENDRUN

ENDLOOP

```

## 24 Misc\_Time-of-Day.S

```

; =====
; Misc_Time-of-Day.s
; MWCOCG Version 2.3 Model - 3722 TAZ System
;
; Distribute Truck and Miscellaneous (non-modeled) trips among
; among three time periods:
; - AM peak 6:00 AM - 8:59 AM (3 Hrs)
; - Midday 9:00 AM - 2:59 PM (6 Hrs)
; - PM peak 3:00 PM - 6:59 PM (4 Hrs)
; - Night All remaining hrs. (11 Hrs)
;
; Note: The miscellaneous purpose 'School Auto Dr.' is no longer used in V2.3
;
; =====
; Environment Variable:
; _iter_ (Iteration indicator = 'pp','il'-'i6'
;
; Parameters:
;
; ZONESIZE = 3722 ; No. of TAZs //
; LastIZN = 3675 ; Last Internal TAZ no. //
; FExt = LastIZN + 1 ; First External TAZ no. //
; ////////////////////////////////////////////////////////////////////
; Input/Output filenames:
; ////////////////////////////////////////////////////////////////////
; COM/TRK Calibration Adjustment Tables //
TKDELTA = '..\support\tkdelta_3722.trp' ; MTK/HTK delta //
CVDELTA = '..\support\cvdelta_3722.trp' ; COM delta ;
//
; I/P Truck & Exogenous trip Tables:
XXCVTRK = 'inputs\XXCVT.vtt' ; Com/Mtk/Htk XX Trips (t1-3) //
XXAUTDR = 'inputs\xxaut.vtt' ; Auto Dr XX Trips (t1) //
;
TAXIADR = 'inputs\taxi.adr' ; TAXI Auto Dr Trips //
VISIADR = 'inputs\visi.adr' ; Visitor A.Dr Trips //
SchIADR = 'inputs\schl.adr' ; School A.Dr Trips //
;
COMTDOUT = '%_iter_%_commer.ptt' ; Comm Vehs t1-Intl, t2-Extl
MTKTDOUT = '%_iter_%_mtruck.ptt' ; Med Trks t1-Intl, t2-Extl
HTKTDOUT = '%_iter_%_htruck.ptt' ; Hvy Trks t1-Intl, t2-Extl /
;
APXADR = 'inputs\airpax.adr' ; Air Passenger Auto Dr. //
;
; O/P Truck and Exogenous Tabs by time of day //
MISCAM = '%_iter_%_am_misc.tt' ; AM Non-Modeled Trips //
MISCMD = '%_iter_%_md_misc.tt' ; Midday Non-Modeled Trips //

```



## Appendix C Cube Voyager Scripts

```

mw[25] = mi.3.4      ; mi.3.htkii
mw[26] = mi.3.5      ; mi.3.htkext
mw[27] = mi.3.5.t    ; mi.3.htkext.t
mw[28] = mi.3.6      ; mi.3.htkxx

; Add in the deltas. First, for I/I and I/X.
if (i = 1-@LastIZN@)
  jloop
  mw[31] = max(mw[1] + mw[21],0) ;mtk ii
  mw[32] = max(mw[2] + mw[22],0) ;mtk ix

  mw[35] = max(mw[4] + mw[25],0) ;htk ii
  mw[36] = max(mw[5] + mw[26],0) ;htk ix
endjloop
endif

if (i > @LastIZN@)

; Now for X/I.
mw[33] = max(mw[3] + mw[23],0), include = 1-@LastIZN@ ; xi mtk
mw[37] = max(mw[6] + mw[27],0), include = 1-@LastIZN@ ; xi htk

; Now for X/X.
mw[34] = max(mw[7] + mw[24],0), include = @FExt@-@ZONESIZE@ ; xx mtk
mw[38] = max(mw[8] + mw[28],0), include = @FExt@-@ZONESIZE@ ; xx htk
endif

; Sum I/I and External here
; MTK
mw[71] = 0.250 * (mw[31] + 0.7 * mw[33] + 0.3 * mw[32]) ; AM
mw[72] = 0.450 * (mw[31] + 0.5 * mw[33] + 0.5 * mw[32]) ; MD
mw[73] = 0.200 * (mw[31] + 0.3 * mw[33] + 0.7 * mw[32]) ; PM
mw[74] = 0.100 * (mw[31] + 0.5 * mw[33] + 0.5 * mw[32]) ; OP

; HTK
mw[75] = 0.200 * (mw[35] + 0.7 * mw[37] + 0.3 * mw[36]) ; AM
mw[76] = 0.500 * (mw[35] + 0.5 * mw[37] + 0.5 * mw[36]) ; MD
mw[77] = 0.100 * (mw[35] + 0.3 * mw[37] + 0.7 * mw[36]) ; PM
mw[78] = 0.200 * (mw[35] + 0.5 * mw[37] + 0.5 * mw[36]) ; OP

; Keep X/X separate
; MTK
mw[79] = 0.250 * mw[34]
mw[80] = 0.450 * mw[34]
mw[81] = 0.200 * mw[34]
mw[82] = 0.100 * mw[34]

; HTK
mw[83] = 0.200 * mw[38]
mw[84] = 0.500 * mw[38]
mw[85] = 0.100 * mw[38]
mw[86] = 0.200 * mw[38]
endrun
;=====
; end of com veh, med, hvy truck time of day processing
;=====
RUN PGM=MATRIX ; Read in Daily Miscellaneous Trips
MATI[1]=@XXAUTDR@ ; Thru Auto Driver Trips
MATI[2]=@TAXIADR@ ; Taxi Auto Driver Trips
MATI[3]=@VISIADR@ ; Visitor/Tourist Auto Driver Trips
MATI[4]=@SchIADR@ ; School Auto Driver Trips

MATI[5]=@APXADR@ ; Air Passenger auto driver Trips

; Read in COM/TRK trips, already split by time period above.
MATI[6]=tempcom.trp
MATI[7]=temptrk.trp

```

```

; Put Misc Trips in Work Mats 2-8 (it simplifies the
; numbering of the other tables, below).
MW[2] = MI.1.1
MW[3] = MI.2.1
MW[4] = MI.3.1
MW[5] = MI.4.1

MW[8] = MI.5.1

; Put COM/TRK trips by TOD in their proper work matrices. We're just
; passing them through from the steps above.

MW[110] = MI.7.9 ; AM X/X MTK
MW[111] = MI.7.13 ; AM X/X HTK
MW[112] = MI.7.9 + MI.7.13; AM X/X TRK
MW[116] = MI.7.1 ; AM I/I + EXT MTK
MW[117] = MI.7.5 ; AM I/I + EXT HTK
MW[119] = MI.6.1 ; AM I/I + EXT COM

MW[140] = MI.7.10 ; MD X/X MTK
MW[141] = MI.7.14 ; MD X/X HTK
MW[142] = MI.7.10 + MI.7.14; MD X/X TRK
MW[146] = MI.7.2 ; MD I/I + EXT MTK
MW[147] = MI.7.6 ; MD I/I + EXT HTK
MW[149] = MI.6.2 ; MD I/I + EXT COM

MW[120] = MI.7.11 ; PM X/X MTK
MW[121] = MI.7.15 ; PM X/X HTK
MW[122] = MI.7.11 + MI.7.15; PM X/X TRK
MW[126] = MI.7.3 ; PM I/I + EXT MTK
MW[127] = MI.7.7 ; PM I/I + EXT HTK
MW[129] = MI.6.3 ; PM I/I + EXT COM

MW[130] = MI.7.12 ; OP X/X MTK
MW[131] = MI.7.16 ; OP X/X HTK
MW[132] = MI.7.12 + MI.7.16; OP X/X TRK
MW[136] = MI.7.4 ; OP I/I + EXT MTK
MW[137] = MI.7.8 ; OP I/I + EXT HTK
MW[139] = MI.6.4 ; OP I/I + EXT COM

; Apply TOD Factors
; put AM trips in work mats 10-19
; put MD trips in work mats 40-49
; put PM trips in work mats 20-29
; put Off-Peak trips in work mats 30-39
;
JLOOP
; AM Peak Period Trips -----
MW[12] = 0.18700 * MW[2] + MI.6.5[J] ; AM Thru Auto Driver + COM
MW[13] = 0.18700 * MW[3] ; AM Taxi Auto Driver
MW[14] = 0.18700 * MW[4] ; AM Visitor Auto Driver
MW[15] = 0.18700 * MW[5] ; AM School Auto Driver

MW[18] = 0.2310 * MW[8] ; AM Air Pax Auto Driver

; Midday Period Trips -----
MW[42] = 0.32630 * MW[2] + MI.6.6[J] ; MD Thru Auto Driver + COM
MW[43] = 0.32630 * MW[3] ; MD Taxi Auto Driver
MW[44] = 0.32630 * MW[4] ; MD Visitor Auto Driver
MW[45] = 0.32630 * MW[5] ; MD School Auto Driver

MW[48] = 0.3657 * MW[8] ; MD Air Pax Auto Driver

; PM Peak Period Trips -----
MW[22] = 0.32890 * MW[2] + MI.6.7[J] ; PM Thru Auto Driver + COM
MW[23] = 0.32890 * MW[3] ; PM Taxi Auto Driver
MW[24] = 0.32890 * MW[4] ; PM Visitor Auto Driver
MW[25] = 0.32890 * MW[5] ; PM School Auto Driver

```



# Appendix C Cube Voyager Scripts

```

MW[28] = 0.2538      * MW[8] ; PM Air Pax  Auto Driver

; Off-Peak Period Trips -----
MW[32] = 0.15780    * MW[2] + MI.6.8[J] ; OP Thru   Auto Driver + COM
MW[33] = 0.15780    * MW[3] ; OP Taxi    Auto Driver
MW[34] = 0.15780    * MW[4] ; OP Visitor  Auto Driver
MW[35] = 0.15780    * MW[5] ; OP School  Auto Driver

MW[38] = 0.1495     * MW[8] ; OP Air Pax  Auto Driver

ENDJLOOP

; LET'S SUMMARIZE NEATLY
jloop
DAYXXMTK = DAYXXMTK + MW[110] + MW[120] + MW[130] + MW[140] ; ACCUMULATE
TOTAL DAILY Medium THRU TRUCKS
DAYXXHTK = DAYXXHTK + MW[111] + MW[121] + MW[131] + MW[141] ; ACCUMULATE
TOTAL DAILY Heavy THRU TRUCKS
DAYXXAD = DAYXXAD + MW[2] + MI.6.5[J] + MI.6.6[J] + MI.6.7[J] + MI.6.8[J];
ACCUMULATE TOTAL DAILY THRU AUTO DRV + COM
DAYTXAD = DAYTXAD + MW[3] ; ACCUMULATE
TOTAL DAILY TAXI ADR TRIPS
DAYVSAD = DAYVSAD + MW[4] ; ACCUMULATE
TOTAL DAILY VISITOR ADR TRIPS
DAYSCAD = DAYSCAD + MW[5] ; ACCUMULATE
TOTAL DAILY School ADR TRIPS

DAYMTRK = DAYMTRK + MW[116] + MW[126] + MW[136] + MW[146]; ACCUMULATE TOTAL
DAILY MED. TRUCK TRIPS
DAYHTRK = DAYHTRK + MW[117] + MW[127] + MW[137] + MW[147]; ACCUMULATE TOTAL
DAILY HVY. TRUCK TRIPS
DAYAPAX = DAYAPAX + MW[8] ; ACCUMULATE TOTAL
DAILY AIR PAX ADR TRIPS
DAYCOM = DAYCOM + MW[119] + MW[129] + MW[139] + MW[149]; ACCUMULATE TOTAL
DAILY COMMERCIAL TRIPS
;---
AMXXMTK = AMXXMTK + MW[110] ; ACCUMULATE TOTAL AM XX Medium TRUCKS
AMXXHTK = AMXXHTK + MW[111] ; ACCUMULATE TOTAL AM XX Heavy TRUCKS
AMXXAD = AMXXAD + MW[12] ; ACCUMULATE TOTAL AM XX ADR + XX COM TRIPS
AMTXAD = AMTXAD + MW[13] ; ACCUMULATE TOTAL AM TAXI ADR TRIPS
AMVSAD = AMVSAD + MW[14] ; ACCUMULATE TOTAL AM VISIT ADR TRIPS
AMScAD = AMScAD + MW[15] ; ACCUMULATE TOTAL AM SchoolADR TRIPS

AMMTRK = AMMTRK + MW[116] ; ACCUMULATE TOTAL AM MED TRUCK TRIPS
AMHTRK = AMHTRK + MW[117] ; ACCUMULATE TOTAL AM HVY TRUCK TRIPS
AMAPAX = AMAPAX + MW[18] ; ACCUMULATE TOTAL AM AIR PAX ADR TRIPS
AMCOM = AMCOM + MW[119] ; ACCUMULATE TOTAL AM COMMERCIAL TRIPS
;---
MDXXMTK = MDXXMTK + MW[140] ; ACCUMULATE TOTAL MD XX Medium TRUCKS
MDXXHTK = MDXXHTK + MW[141] ; ACCUMULATE TOTAL MD XX Heavy TRUCKS
MDXXAD = MDXXAD + MW[42] ; ACCUMULATE TOTAL MD XX ADR + XX COM TRIPS
MDTXAD = MDTXAD + MW[43] ; ACCUMULATE TOTAL MD TAXI ADR TRIPS
MDVSAD = MDVSAD + MW[44] ; ACCUMULATE TOTAL MD VISIT ADR TRIPS
MDScAD = MDScAD + MW[45] ; ACCUMULATE TOTAL MD SchoolADR TRIPS

MDMTRK = MDMTRK + MW[146] ; ACCUMULATE TOTAL MD MED TRUCK TRIPS
MDHTRK = MDHTRK + MW[147] ; ACCUMULATE TOTAL MD HVY TRUCK TRIPS
MDAPAX = MDAPAX + MW[48] ; ACCUMULATE TOTAL MD AIRPAX ADR TRIPS
MDCOM = MDCOM + MW[149] ; ACCUMULATE TOTAL MD COMMERCIAL TRIPS
;---
PMXXMTK = PMXXMTK + MW[120] ; ACCUMULATE TOTAL PM XX Medium TRUCKS
PMXXHTK = PMXXHTK + MW[121] ; ACCUMULATE TOTAL PM XX Heavy TRUCKS
PMXXAD = PMXXAD + MW[22] ; ACCUMULATE TOTAL PM XX ADR + XX COM TRIPS
PMTXAD = PMTXAD + MW[23] ; ACCUMULATE TOTAL PM TAXI ADR TRIPS
PMVSAD = PMVSAD + MW[24] ; ACCUMULATE TOTAL PM VISIT ADR TRIPS
PMScAD = PMScAD + MW[25] ; ACCUMULATE TOTAL PM SchoolADR TRIPS

```

```

PMMTRK = PMMTRK + MW[126] ; ACCUMULATE TOTAL PM MED TRUCK TRIPS
PMHTRK = PMHTRK + MW[127] ; ACCUMULATE TOTAL PM HVY TRUCK TRIPS
PMAPAX = PMAPAX + MW[28] ; ACCUMULATE TOTAL PM AIR PAX ADR TRIPS
PMCOM = PMCOM + MW[129] ; ACCUMULATE TOTAL PM COMMERCIAL TRIPS
;---
OPXXMTK = OPXXMTK + MW[130] ; ACCUMULATE TOTAL OP XX Medium TRUCKS
OPXXHTK = OPXXHTK + MW[131] ; ACCUMULATE TOTAL OP XX Heavy TRUCKS
OPXXAD = OPXXAD + MW[32] ; ACCUMULATE TOTAL OP XX ADR + XX COM TRIPS
OPTXAD = OPTXAD + MW[33] ; ACCUMULATE TOTAL OP TAXI ADR TRIPS
OPVSAD = OPVSAD + MW[34] ; ACCUMULATE TOTAL OP VISIT ADR TRIPS
OPScAD = OPScAD + MW[35] ; ACCUMULATE TOTAL OP SchoolADR TRIPS

OPMTRK = OPMTRK + MW[136] ; ACCUMULATE TOTAL OP MED TRUCK TRIPS
OPHTRK = OPHTRK + MW[137] ; ACCUMULATE TOTAL OP HVY TRUCK TRIPS
OPAPAX = OPAPAX + MW[38] ; ACCUMULATE TOTAL OP AIR PAX ADR TRIPS
OPCOM = OPCOM + MW[139] ; ACCUMULATE TOTAL OP COMMERCIAL TRIPS
;---
; total input misc trips
ipmisc = ipmisc + MW[02] + MW[03] + MW[04] + MW[05] + MW[08] +
MW[110] + MW[111] + MW[116] + MW[117] + MW[119] +
MW[120] + MW[121] + MW[126] + MW[127] + MW[129] +
MW[130] + MW[131] + MW[136] + MW[137] + MW[139] +
MW[140] + MW[141] + MW[146] + MW[147] + MW[149] +
MI.6.5[J] + MI.6.6[J] + MI.6.7[J] + MI.6.8[J]

; total output misc trips
opmisc = opmisc +
MW[110]+MW[111] +MW[12]+MW[13]+MW[14]+MW[15]+MW[116]+MW[117] +MW[18]+MW[119]+
MW[120]+MW[121] +MW[22]+MW[23]+MW[24]+MW[25]+MW[126]+MW[127] +MW[28]+MW[129]+
MW[130]+MW[131] +MW[32]+MW[33]+MW[34]+MW[35]+MW[136]+MW[137] +MW[38]+MW[139]+
MW[140]+MW[141] +MW[42]+MW[43]+MW[44]+MW[45]+MW[146]+MW[147] +MW[48]+MW[149]

ENDJLOOP

IF (I=ZONES) ; LIST OUT THE TOTALS IF AT THE END OF THE I-LOOP
; get regional I/O differences
diff = opmisc-ipmisc ;

LIST = '/bt '
LIST = ' MISCELLANEOUS/TRUCK TIME-OF-DAY TOTALS ','\n',
list = ' '

list = 'Input Misc/Truck Total: ',ipmisc(10.0c)
list = 'Output Misc/Truck Total: ',opmisc(10.0c)
list = 'Diff. (Output-Input): ',diff(10.0)
list = ' '

LIST = 'DAILY XX MedTrk: ' ,dayxxmtk(9.0c),' AM, MD, PM, Off-Pk totals:
',AMXXmTk(9.0c),' ',MDXXmTk(9.0c),' ',PMXXmTk(9.0c),' ',OPXXmTk(9.0c)
LIST = 'DAILY XX HvyTrk: ' ,dayxxhtk(9.0c),' AM, MD, PM, Off-Pk totals:
',AMXXhTk(9.0c),' ',MDXXhTk(9.0c),' ',PMXXhTk(9.0c),' ',OPXXhTk(9.0c)
LIST = 'DAILY XX ADR/CV: ' ,dayxxad(9.0c),' AM, MD, PM, Off-Pk totals:
',AMXXAD(9.0c),' ',MDXXAD(9.0c),' ',PMXXAD(9.0c),' ',OPXXAD(9.0c)
LIST = 'DAILY TAXI ADRS: ' ,dayTxAd(9.0c),' AM, MD, PM, Off-Pk totals:
',AMTXAD(9.0c),' ',MDTXAD(9.0c),' ',PMTXAD(9.0c),' ',OPTXAD(9.0c)
LIST = 'DAILY VISI ADRS: ' ,dayVSAD(9.0c),' AM, MD, PM, Off-Pk totals:
',AMVSAD(9.0c),' ',MDVSAD(9.0c),' ',PMVSAD(9.0c),' ',OPVSAD(9.0c)
LIST = 'DAILY Schl ADRS: ' ,dayScAd(9.0c),' AM, MD, PM, Off-Pk totals:
',AMScAD(9.0c),' ',MDScAD(9.0c),' ',PMScAD(9.0c),' ',OPScAD(9.0c)
LIST = 'DAILY COM VEHS: ' ,dayCOM(9.0c),' AM, MD, PM, Off-Pk totals:
',AMCOM(9.0c),' ',MDCOM(9.0c),' ',PMCOM(9.0c),' ',OPCOM(9.0c)
LIST = 'DAILY MED TRKS: ' ,dayMTRK(9.0c),' AM, MD, PM, Off-Pk totals:
',AMMTRK(9.0c),' ',MDMTRK(9.0c),' ',PMMTRK(9.0c),' ',OPMTRK(9.0c)
LIST = 'DAILY HVY TRKS: ' ,dayHTRK(9.0c),' AM, MD, PM, Off-Pk totals:
',AMHTRK(9.0c),' ',MDHTRK(9.0c),' ',PMHTRK(9.0c),' ',OPHTRK(9.0c)
LIST = 'DAILY APX ADRS: ' ,dayAPAX(9.0c),' AM, MD, PM, Off-Pk totals:
',AMAPAX(9.0c),' ',MDAPAX(9.0c),' ',PMAPAX(9.0c),' ',OPAPAX(9.0c)

```

## Appendix C Cube Voyager Scripts

```
LIST = '/et
endif
; Write out the Miscellaneous Trips in time period-specific files
MATO[1] = @MISCAM@, MO=112,12,13,14,15,116,117,18,119, ; AM MISC Trips

name=AM_XXTrk,AM_XXAdr,AM_TxAdr,AM_VtAdr,AM_ScAdr,AM_MedTk,AM_HvyTk,AM_APAAdr,AM_ComV
e

MATO[2] = @MISCMD@, MO=142,42,43,44,45,146,147,48,149, ; MD MISC Trips

name=MD_XXTrk,MD_XXAdr,MD_TxAdr,MD_VtAdr,MD_ScAdr,MD_MedTk,MD_HvyTk,MD_APAAdr,MD_ComV
e

MATO[3] = @MISCPM@, MO=122,22,23,24,25,126,127,28,129, ; PM MISC Trips

name=PM_XXTrk,PM_XXAdr,PM_TxAdr,PM_VtAdr,PM_ScAdr,PM_MedTk,PM_HvyTk,PM_APAAdr,PM_ComV
e

MATO[4] = @MISCOP@, MO=132,32,33,34,35,136,137,38,139, ; OP MISC Trips

name=OP_XXTrk,OP_XXAdr,OP_TxAdr,OP_VtAdr,OP_ScAdr,OP_MedTk,OP_HvyTk,OP_APAAdr,OP_ComV
e
ENDRUN
;
*del tempcom.trp
*del temptrk.trp
```

## 25 modnet.s

```
;; 4/16/11 HWYNET modules changed to 'NETWORK'
;; 4/16/11 zones increased to 7999; 'MDLIMIT = 0' and 'NTLIMIT=0' added
pageheight=32767 ; Preclude header breaks

;; write out list of highway nodes with a-nodes, b-nodes, distance, TAZ, and Ftype
;; for the walkacc program
RUN PGM=NETWORK
NETI = '%_iter_%_hwy.net'

IF (Ftype != 0)
print list= a(8), b(8), distance(8.2), ftype(8),TAZ(8), File= WalkAcc_links.txt
ENDIF
ENDRUN

RUN PGM=MATRIX
ZONES=1
FILEI RECI = WalkAcc_links.txt,
a= 1, b= 2, distance= 3, ftype= 4,TAZ= 5

n=n+1

RECO[1] ="WalkAcc Links.dbf",
Fields = a(8), b(8), distance(8.2), ftype(8),TAZ(8)

WRITE RECO=1
endrun
;

;; write out network with added station centroid connectors
RUN PGM=NETWORK
NETI = '%_iter_%_hwy.net'
```

```
NETO = '%_iter_%_HWYMOD.NET'

PARAMETERS ZONES=7999

IF (A=3723-7999 || B=3723-7999)
AMLIMIT = 0
PMLIMIT = 0
OPLIMIT = 0
MDLIMIT = 0
NTLIMIT = 0
ENDIF
ENDRUN
```

## 26 MWCOG\_Prepare\_Inputs\_to\_Walksh ed\_Process\_PT.s

```
;
;
; This Voyager script creates shapefile inputs for Walkshed Process
; Outputs are:
; 1) ALLStops_PK.shp (.dbf, .shx)
; 2) ALLStops_OP.shp (.dbf, .shx)
; 3) MetroandLRT_AllDay.shp (.dbf, .shx)
;
; 10/17/2013, Author: KCP with support from Citilabs
;-----
; Reference: V2.3.52_Users_Guide_v2_w_appA.pdf Page 113
; Mode 3= Metrorail
; Mode 4 = Commuter rail
; Mode 5 = light rail
; Mode 10= BRT, Streetcar

;
; Step 1) Create network with all nodes
;
RUN PGM=HWYNET

FILEI NODEI[1] = "..\Node.dbf"
FILEI NODEI[2] = "..\Met_Node.tb",
VAR=N,9-14,
VAR=X,18-27,
VAR=Y,31-40

FILEI NODEI[3] = "..\Com_Node.tb",
VAR=N,9-14,
VAR=X,18-27,
VAR=Y,31-40

FILEI NODEI[4] = "..\LRT_Node.tb",
VAR=N,9-14,
VAR=X,18-27,
VAR=Y,31-40

FILEI NODEI[5] = "..\NEW_Node.tb",
VAR=N,9-14,
VAR=X,18-27,
VAR=Y,31-40

FILEI LINKI[1] = "..\Link.dbf"
FILEI LINKI[2] = "..\Met_Link.tb",
VAR=A,12-17,
VAR=B,19-24, REV=2

FILEI LINKI[3] = "..\Com_Link.tb",
VAR=A,12-17,
```

## Appendix C Cube Voyager Scripts

```

        FILEI LINKI[4] = "..\LRT_Link.tb",          VAR=B,19-24, REV=2
                                                VAR=A,12-17,
                                                VAR=B,19-24, REV=2
        FILEI LINKI[5] = "..\NEW_Link.tb",        VAR=A,12-17,
                                                VAR=B,19-24, REV=2

        ZONES      = 3722
        NETO       = "input\temp_HwyNetWithTrnNodes.NET"

ENDRUN

;
; Create a Dummy Factors File for PT
;

*DEL /Q input\Dummy_FactorFile.txt

*ECHO ;Global Settings
>> input\Dummy_FactorFile.txt
*ECHO BESTPATHONLY=T
>> input\Dummy_FactorFile.txt
*ECHO MAXFERS=5
>> input\Dummy_FactorFile.txt
*ECHO SERVICEMODEL=FREQUENCY
>> input\Dummy_FactorFile.txt
*ECHO RECOSTMAX=250.0
>> input\Dummy_FactorFile.txt
*ECHO FREQBYMODE=T
>> input\Dummy_FactorFile.txt
*ECHO.
>> input\Dummy_FactorFile.txt
*ECHO ;Global Settings
>> input\Dummy_FactorFile.txt
*ECHO DELMODE = 1
>> input\Dummy_FactorFile.txt
*ECHO.
>> input\Dummy_FactorFile.txt
*ECHO ;Boarding and Transfer Penalties
>> input\Dummy_FactorFile.txt
*ECHO IWAITCURVE=1, NODES=1100-99999
>> input\Dummy_FactorFile.txt
*ECHO XWAITCURVE=1, NODES=1100-99999
>> input\Dummy_FactorFile.txt
*ECHO WAITFACTOR=2.5, n=1-99999
>> input\Dummy_FactorFile.txt
*ECHO XFERFACTOR=2.5, from=1-10, to=1-10
>> input\Dummy_FactorFile.txt
*ECHO.
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=9999, from=12, to=16,
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=9999, from=16, to=12,
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=9999, from=16, to=16
>> input\Dummy_FactorFile.txt
*ECHO.
>> input\Dummy_FactorFile.txt
*ECHO BRDPEN=0,0,0,0,0,0 ;
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=9999, from=1, to=3-6
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=9999, from=3-6, to=1
>> input\Dummy_FactorFile.txt
*ECHO XFERPEN=2.0, from=7, to=3
>> input\Dummy_FactorFile.txt

```

```

;
; Create a Dummy System File for PT
;

*DEL /Q input\Dummy_SystemFile.txt

*ECHO ;;<<PT>><<SYSTEM>>;
>> input\Dummy_SystemFile.txt
*ECHO.
>> input\Dummy_SystemFile.txt
*ECHO ; Wait curve definitions
>> input\Dummy_SystemFile.txt
*ECHO WAITCRVDEF NUMBER=1 LONGNAME="Initial and Transfer Wait" NAME=InitXferWait ,
>> input\Dummy_SystemFile.txt
*ECHO CURVE=0-0,1-0.25,4-1,60-15,120-30
>> input\Dummy_SystemFile.txt

;
; Copy All PT Line Files into One
; This is done so as to have the option to have an empty line file
;
*COPY ..\MODE*AM.Lin input\Combined_MODEsAM.Lin

;
; Step 2a) Read TRNBUILD line files and export links (ALLStops_PK)
;
RUN PGM=PUBLIC TRANSPORT

        FILEI NETI      = "input\temp_HwyNetWithTrnNodes.NET"
        FILEO LINKO    = "input\temp_LinkALLPK.DBF", ONOFFS=Y

        FILEI FACTORI[1] = "input\Dummy_FactorFile.txt"
        FILEI SYSTEMI   = "input\Dummy_SystemFile.txt"

;Globals
PARAMETERS USERCLASSES=1,FARE=N, HDWAYPERIOD=1,NOROUTEERRS=99999999,
NOROUTEMSGS=999999999,TRANTIME=100, SKIPBADLINES=Y

        FILEI LINEI      = "input\Combined_MODEsAM.Lin"

; FILEI LINEI[1]      = "..\MODE1AM.Lin"
; FILEI LINEI[2]      = "..\MODE2AM.Lin"
; FILEI LINEI[3]      = "..\MODE3AM.Lin"
; FILEI LINEI[4]      = "..\MODE4AM.Lin"
; FILEI LINEI[5]      = "..\MODE5AM.Lin"
; FILEI LINEI[6]      = "..\MODE6AM.Lin"
; FILEI LINEI[7]      = "..\MODE7AM.Lin"
; FILEI LINEI[8]      = "..\MODE8AM.Lin"
; FILEI LINEI[9]      = "..\MODE9AM.Lin"
; FILEI LINEI[10]     = "..\MODE10AM.Lin"

        PHASE=DATAPREP
        ENDPHASE

        PHASE=SKIMIJ
        ENDPHASE

ENDRUN

;
; Copy All PT Line Files into One
; This is done so as to have the option to have an empty line file
;
*COPY ..\MODE*OP.Lin input\Combined_MODEsOP.Lin

;
; Step 2b) Read TRNBUILD line files and export links (ALLStops_OP)
;

```

## Appendix C Cube Voyager Scripts

```

RUN PGM=PUBLIC TRANSPORT

FILEI NETI      = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO     = "input\temp_LinkALLOP.DBF", ONOFFS=Y

FILEI FACTORI[1] = "input\Dummy_FactorFile.txt"
FILEI SYSTEMI   = "input\Dummy_SystemFile.txt"

;Globals
PARAMETERS USERCLASSES=1,FARE=N, HDWAYPERIOD=1,NOROUTEERRS=99999999,
NOROUTEMSGS=99999999,TRANTIME=100, SKIPBADLINES=Y

FILEI LINEI     = "input\Combined_MODEsOP.Lin"

; FILEI LINEI[1]   = "..\MODE1OP.Lin"
; FILEI LINEI[2]   = "..\MODE2OP.Lin"
; FILEI LINEI[3]   = "..\MODE3OP.Lin"
; FILEI LINEI[4]   = "..\MODE4OP.Lin"
; FILEI LINEI[5]   = "..\MODE5OP.Lin"
; FILEI LINEI[6]   = "..\MODE6OP.Lin"
; FILEI LINEI[7]   = "..\MODE7OP.Lin"
; FILEI LINEI[8]   = "..\MODE8OP.Lin"
; FILEI LINEI[9]   = "..\MODE9OP.Lin"
; FILEI LINEI[10]  = "..\MODE10OP.Lin"

PHASE=DATAPREP
ENDPHASE

PHASE=SKIMIJ
ENDPROCESS

ENDRUN

;
; Copy Metro & LRT PT Line Files into One
; This is done so as to have the option to have an empty line file
;
*COPY ..\MODE3AM.Lin + ..\MODE5AM.Lin input\Combined_MetroLRTAM.Lin

;
; Step 2c) Read TRNBUILD line files and export links (MetroandLRT_AllDay)
;
RUN PGM=PUBLIC TRANSPORT

FILEI NETI      = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO     = "input\temp_LinkMetroLRT_PK.DBF", ONOFFS=Y

FILEI FACTORI[1] = "input\Dummy_FactorFile.txt"
FILEI SYSTEMI   = "input\Dummy_SystemFile.txt"

;Globals
PARAMETERS USERCLASSES=1,FARE=N, HDWAYPERIOD=1,NOROUTEERRS=99999999,
NOROUTEMSGS=99999999,TRANTIME=100, SKIPBADLINES=Y

FILEI LINEI     = "input\Combined_MetroLRTAM.Lin"

; FILEI LINEI[3]   = "..\MODE3OP.Lin"
; FILEI LINEI[5]   = "..\MODE5OP.Lin"

PHASE=DATAPREP
ENDPHASE

PHASE=SKIMIJ
ENDPROCESS

ENDRUN

;

```

```

; Copy Metro & LRT PT Line Files into One
; This is done so as to have the option to have an empty line file
;
*COPY ..\MODE3OP.Lin + ..\MODE5OP.Lin input\Combined_MetroLRTOP.Lin

RUN PGM=PUBLIC TRANSPORT

FILEI NETI      = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO     = "input\temp_LinkMetroLRT_OP.DBF", ONOFFS=Y

FILEI FACTORI[1] = "input\Dummy_FactorFile.txt"
FILEI SYSTEMI   = "input\Dummy_SystemFile.txt"

;Globals
PARAMETERS USERCLASSES=1,FARE=N, HDWAYPERIOD=1,NOROUTEERRS=99999999,
NOROUTEMSGS=99999999,TRANTIME=100, SKIPBADLINES=Y

FILEI LINEI     = "input\Combined_MetroLRTOP.Lin"

; FILEI LINEI[3]   = "..\MODE3OP.Lin"
; FILEI LINEI[5]   = "..\MODE5OP.Lin"

PHASE=DATAPREP
ENDPHASE

PHASE=SKIMIJ
ENDPROCESS

ENDRUN

;
; Step 3a) Export nodes (ALLStops_PK)
;
RUN PGM=MATRIX

FILEI RECI      = "input\temp_LinkALLPK.DBF" ;LINKO
file from TRNBUILD
FILEO RECO[1]   = "input\temp_NodeALLPK.DBF" FIELDS=N,STOP

STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOPA=1) WRITE RECO=1 ;Select
Mode
N=RI.B
IF (RI.MODE>0 & RI.STOPB=1) WRITE RECO=1

ENDRUN

;
; Step 3b) Export nodes (ALLStops_OP)
;
RUN PGM=MATRIX

FILEI RECI      = "input\temp_LinkALLOP.DBF" ;LINKO
file from TRNBUILD
FILEO RECO[1]   = "input\temp_NodeALLOP.DBF" FIELDS=N,STOP

STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOPA=1) WRITE RECO=1 ;Select
Mode
N=RI.B
IF (RI.MODE>0 & RI.STOPB=1) WRITE RECO=1

ENDRUN

;

```

## Appendix C Cube Voyager Scripts

```

; Step 3c) Export nodes (MetroandLRT_AllDay)
;
RUN PGM=MATRIX

      FILEI RECI      = "input\temp_LinkMetroLRT_PK.DBF"          ;LINKO
file from TRNBUILD
      FILEO RECO[1]   = "input\temp_NodeMetroLRT_PK.DBF"  FIELDS=N,STOP

      STOP=1
      N=RI.A
      IF (RI.MODE>0 & RI.STOPA=1) WRITE RECO=1 ;Select Mode
      N=RI.B
      IF (RI.MODE>0 & RI.STOPB=1) WRITE RECO=1 ;Select Mode

ENDRUN
RUN PGM=MATRIX

      FILEI RECI      = "input\temp_LinkMetroLRT_OP.DBF"          ;LINKO
file from TRNBUILD
      FILEO RECO[1]   = "input\temp_NodeMetroLRT_OP.DBF"  FIELDS=N,STOP

      STOP=1
      N=RI.A
      IF (RI.MODE>0 & RI.STOPA=1) WRITE RECO=1 ;Select Mode
      N=RI.B
      IF (RI.MODE>0 & RI.STOPB=1) WRITE RECO=1 ;Select Mode

ENDRUN

;
; Step 4a) Use nodes and network to export stop-shapefile (ALLStops_PK)
;
RUN PGM=NETWORK

      FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET"      ;Input
highway network
      FILEI NODEI[2] = "input\temp_NodeALLPK.DBF"  COMBINE=T      ;Stop
node file from previous MATRIX step
      FILEO NODEO    = "input\ALLStops_PK.shp"  FORMAT=SHP        ;stop
nodes shape file.

      MERGE RECORD=T

      PHASE=NODEMERGE
      IF (ni.2.stop=0) delete
      ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\ALL_Stops_PK.prj

;
; Step 4b) Use nodes and network to export stop-shapefile (ALLStops_OP)
;
RUN PGM=NETWORK

      FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET"      ;Input
highway network
      FILEI NODEI[2] = "input\temp_NodeALLOP.DBF"  COMBINE=T      ;Stop
node file from previous MATRIX step
      FILEO NODEO    = "input\ALLStops_OP.shp"  FORMAT=SHP        ;stop
nodes shape file.

      MERGE RECORD=T

```

```

      PHASE=NODEMERGE
      IF (ni.2.stop=0) delete
      ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\ALLStops_OP.prj

;
; Step 4c) Use nodes and network to export stop-shapefile (MetroandLRT_AllDay)
;
RUN PGM=NETWORK

      FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET"      ;Input
highway network
      FILEI NODEI[2] = "input\temp_NodeMetroLRT_PK.DBF"  COMBINE=T      ;Stop
node file from previous MATRIX step
      FILEI NODEI[3] = "input\temp_NodeMetroLRT_OP.DBF"  COMBINE=T      ;Stop
node file from previous MATRIX step
      FILEO NODEO    = "input\MetroandLRT_AllDay.shp"  FORMAT=SHP        ;stop
nodes shape file.

      MERGE RECORD=T

      PHASE=NODEMERGE
      IF (ni.2.stop=0) delete
      IF (ni.3.stop=0) delete
      ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\MetroandLRT_AllDay.prj

;
; This Voyager script creates shapefile inputs for Walkshed Process
; Outputs are:
; 1) ALLStops_PK.shp      (.dbf, .shx)
; 2) ALLStops_OP.shp     (.dbf, .shx)
; 3) MetroandLRT_AllDay.shp (.dbf, .shx)
;
; 10/17/2013, Author: KCP with support from Citilabs
;-----
; Reference: V2.3.52_Users_Guide_v2_w_appA.pdf Page 113
; Mode 3= Metrorail
; Mode 4 = Commuter rail
; Mode 5 = light rail
; Mode 10= BRT, Streetcar

```

## Appendix C Cube Voyager Scripts

```

;
; Step 1) Create network with all nodes
;
RUN PGM=HWYNET

FILEI NODEI[1] = "..\Node.dbf"
FILEI NODEI[2] = "..\Met_Node.tb",
                VAR=N,9-14,
                VAR=X,18-27,
                VAR=Y,31-40

FILEI NODEI[3] = "..\Com_Node.tb",
                VAR=N,9-14,
                VAR=X,18-27,
                VAR=Y,31-40

FILEI NODEI[4] = "..\LRT_Node.tb",
                VAR=N,9-14,
                VAR=X,18-27,
                VAR=Y,31-40

FILEI NODEI[5] = "..\NEW_Node.tb",
                VAR=N,9-14,
                VAR=X,18-27,
                VAR=Y,31-40

FILEI LINKI[1] = "..\Link.dbf"
FILEI LINKI[2] = "..\Met_Link.tb",
                VAR=A,12-17,
                VAR=B,19-24, REV=2

FILEI LINKI[3] = "..\Com_Link.tb",
                VAR=A,12-17,
                VAR=B,19-24, REV=2

FILEI LINKI[4] = "..\LRT_Link.tb",
                VAR=A,12-17,
                VAR=B,19-24, REV=2

FILEI LINKI[5] = "..\NEW_Link.tb",
                VAR=A,12-17,
                VAR=B,19-24, REV=2

ZONES          = 3722
NETO           = "input\temp_HwyNetWithTrnNodes.NET"

ENDRUN

;
; Step 2a) Read TRNBUILD line files and export links (ALLStops_PK)
;
RUN PGM=TRNBUILD

FILEI NETI     = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO    = "input\temp_LinkALLPK.DBF"

parameters buildpaths=false
phase1 hwytime=DISTANCE ;give a
link variable for path cost. This has no effect on the stop node shape file output
support modes=200 dist=100 N=1-2
;generate a dummy support link. Use a mode which is not used in your transit line
file

READ FILE     = "..\MODE1AM.TB"
READ FILE     = "..\MODE2AM.TB"
READ FILE     = "..\MODE3AM.TB"
READ FILE     = "..\MODE4AM.TB"
READ FILE     = "..\MODE5AM.TB"
READ FILE     = "..\MODE6AM.TB"
READ FILE     = "..\MODE7AM.TB"
READ FILE     = "..\MODE8AM.TB"
READ FILE     = "..\MODE9AM.TB"
READ FILE     = "..\MODE10AM.TB"

```

```

ENDRUN

;
; Step 2b) Read TRNBUILD line files and export links (ALLStops_OP)
;
RUN PGM=TRNBUILD

FILEI NETI     = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO    = "input\temp_LinkALLOP.DBF"

parameters buildpaths=false
phase1 hwytime=DISTANCE ;give a
link variable for path cost. This has no effect on the stop node shape file output
support modes=200 dist=100 N=1-2
;generate a dummy support link. Use a mode which is not used in your transit line
file

READ FILE     = "..\MODE10P.TB"
READ FILE     = "..\MODE20P.TB"
READ FILE     = "..\MODE30P.TB"
READ FILE     = "..\MODE40P.TB"
READ FILE     = "..\MODE50P.TB"
READ FILE     = "..\MODE60P.TB"
READ FILE     = "..\MODE70P.TB"
READ FILE     = "..\MODE80P.TB"
READ FILE     = "..\MODE90P.TB"
READ FILE     = "..\MODE100P.TB"

ENDRUN

;
; Step 2c) Read TRNBUILD line files and export links (MetroandLRT_AllDay)
;
RUN PGM=TRNBUILD

FILEI NETI     = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO    = "input\temp_LinkMetroLRT_PK.DBF"

parameters buildpaths=false
phase1 hwytime=DISTANCE ;give a
link variable for path cost. This has no effect on the stop node shape file output
support modes=200 dist=100 N=1-2
;generate a dummy support link. Use a mode which is not used in your transit line
file

READ FILE     = "..\MODE3AM.TB"
READ FILE     = "..\MODE5AM.TB"

ENDRUN

RUN PGM=TRNBUILD

FILEI NETI     = "input\temp_HwyNetWithTrnNodes.NET"
FILEO LINKO    = "input\temp_LinkMetroLRT_OP.DBF"

parameters buildpaths=false
phase1 hwytime=DISTANCE ;give a
link variable for path cost. This has no effect on the stop node shape file output
support modes=200 dist=100 N=1-2
;generate a dummy support link. Use a mode which is not used in your transit line
file

READ FILE     = "..\MODE30P.TB"
READ FILE     = "..\MODE50P.TB"

ENDRUN

;
; Step 3a) Export nodes (ALLStops_PK)

```

## Appendix C Cube Voyager Scripts

```

;
RUN PGM=MATRIX

file from FILEI RECI = "input\temp_LinkALLPK.DBF" ;LINKO
TRNBUILD
FILEO RECO[1] = "input\temp_NodeALLPK.DBF" FIELDS=N,STOP

Mode
STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOP_A=1) WRITE RECO=1 ;Select
N=RI.B
IF (RI.MODE>0 & RI.STOP_B=1) WRITE RECO=1

ENDRUN

;
; Step 3b) Export nodes (ALLStops_OP)
;
RUN PGM=MATRIX

file from FILEI RECI = "input\temp_LinkALLOP.DBF" ;LINKO
TRNBUILD
FILEO RECO[1] = "input\temp_NodeALLOP.DBF" FIELDS=N,STOP

Mode
STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOP_A=1) WRITE RECO=1 ;Select
N=RI.B
IF (RI.MODE>0 & RI.STOP_B=1) WRITE RECO=1

ENDRUN

;
; Step 3c) Export nodes (MetroandLRT_AllDay)
;
RUN PGM=MATRIX

file from FILEI RECI = "input\temp_LinkMetroLRT_PK.DBF" ;LINKO
TRNBUILD
FILEO RECO[1] = "input\temp_NodeMetroLRT_PK.DBF" FIELDS=N,STOP

Mode
STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOP_A=1) WRITE RECO=1 ;Select Mode
N=RI.B
IF (RI.MODE>0 & RI.STOP_B=1) WRITE RECO=1 ;Select Mode

ENDRUN
RUN PGM=MATRIX

file from FILEI RECI = "input\temp_LinkMetroLRT_OP.DBF" ;LINKO
TRNBUILD
FILEO RECO[1] = "input\temp_NodeMetroLRT_OP.DBF" FIELDS=N,STOP

Mode
STOP=1
N=RI.A
IF (RI.MODE>0 & RI.STOP_A=1) WRITE RECO=1 ;Select Mode
N=RI.B
IF (RI.MODE>0 & RI.STOP_B=1) WRITE RECO=1 ;Select Mode

ENDRUN

;
; Step 4a) Use nodes and network to export stop-shapefile (ALLStops_PK)
;
RUN PGM=NETWORK

```

```

FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET" ;Input
highway network
FILEI NODEI[2] = "input\temp_NodeALLPK.DBF" COMBINE=T ;Stop
node file from previous MATRIX step
FILEO NODEO = "input\ALLStops_PK.shp" FORMAT=SHP ;stop
nodes shape file.

MERGE RECORD=T

PHASE=NODEMERGE
IF (ni.2.stop=0) delete
ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\ALL_Stops_PK.prj

;
; Step 4b) Use nodes and network to export stop-shapefile (ALLStops_OP)
;
RUN PGM=NETWORK

FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET" ;Input
highway network
FILEI NODEI[2] = "input\temp_NodeALLOP.DBF" COMBINE=T ;Stop
node file from previous MATRIX step
FILEO NODEO = "input\ALLStops_OP.shp" FORMAT=SHP ;stop
nodes shape file.

MERGE RECORD=T

PHASE=NODEMERGE
IF (ni.2.stop=0) delete
ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\ALLStops_OP.prj

;
; Step 4c) Use nodes and network to export stop-shapefile (MetroandLRT_AllDay)
;
RUN PGM=NETWORK

FILEI LINKI[1] = "input\temp_HwyNetWithTrnNodes.NET" ;Input
highway network
FILEI NODEI[2] = "input\temp_NodeMetroLRT_PK.DBF" COMBINE=T ;Stop
node file from previous MATRIX step
FILEI NODEI[3] = "input\temp_NodeMetroLRT_OP.DBF" COMBINE=T ;Stop
node file from previous MATRIX step
FILEO NODEO = "input\MetroandLRT_AllDay.shp" FORMAT=SHP ;stop
nodes shape file.

MERGE RECORD=T

PHASE=NODEMERGE
IF (ni.2.stop=0) delete
IF (ni.3.stop=0) delete

```

## Appendix C Cube Voyager Scripts

```

ENDPHASE

ENDRUN

;
; Cube does not create a projection (.prj) file, so copy from template
;
*COPY ..\..\Scripts\Maryland1900Ft_ShapefileProjection_TEMPLATE.prj
input\MetroandLRT_AllDay.prj

```

## 28 Parker.s

```

*del voya*.prn
; Parker.s - PNR to Station Link development
; Dimensions:
;
;;Input Files:
Sta_File = 'inputs\Station.dbf ' ; Std. Station file
;
; Output Files:

met_am_pnr = 'met_am_pnr.tb' ;unit:21x
com_am_pnr = 'com_am_pnr.tb' ; 22
bus_am_pnr = 'bus_am_pnr.tb' ; 23
lrt_am_pnr = 'lrt_am_pnr.tb' ; 24
new_am_pnr = 'new_am_pnr.tb' ; 25
met_op_pnr = 'met_op_pnr.tb' ; 31x
com_op_pnr = 'com_op_pnr.tb' ; 32
bus_op_pnr = 'bus_op_pnr.tb' ; 33
lrt_op_pnr = 'lrt_op_pnr.tb' ; 34
new_op_pnr = 'new_op_pnr.tb' ; 35
;
; Params:

VOTperHr = 10.00 ; Assumed Value of time in $/hr
VOTperMin = VOTperHr/60.0 ; Derived Value of time in $/min
MinPerDoll = 1.0/VOTperMin ; Derived Value of minutes per dollar (see comment
below for units)

RUN PGM=MATRIX

ZONES=1
FILEI DBI[1] = "@Sta_File@"

FILEO PRINTO[1] = "@met_am_pnr@"
FILEO PRINTO[2] = "@com_am_pnr@"
FILEO PRINTO[3] = "@bus_am_pnr@"
FILEO PRINTO[4] = "@lrt_am_pnr@"
FILEO PRINTO[5] = "@new_am_pnr@"
FILEO PRINTO[6] = "@met_op_pnr@"
FILEO PRINTO[7] = "@com_op_pnr@"
FILEO PRINTO[8] = "@bus_op_pnr@"
FILEO PRINTO[9] = "@lrt_op_pnr@"
FILEO PRINTO[10] = "@new_op_pnr@"

; Read in Station File
LOOP K = 1,dbi.1.NUMRECORDS
x = DBIReadRecord(1,k)

MM = di.1.MM ; Mode code ('M','C','B','L','N')
STAPARK = di.1.STAPARK ; Station Parking lot flag ('Y' or blank)
STAUSE = di.1.STAUSE ; Station Active flag ('Y' or blank)

```

```

series) STAT = di.1.STAT ; Station node (9000 - 11999)
series) STAZ = di.1.STAZ ; Nearest TAZ centroid ( 1 - 3722)
series) STAC = di.1.STAC ; Station centroid ( 5000 - 8000)
series) STAP = di.1.STAP ; Station PNR node (12000 - 14999)
series) STAN1 = di.1.STAN1 ; Bus Node connector
STAPCAP = di.1.STAPCAP ; Parking lot capacity
STAPKCOST = di.1.STAPKCOST ; AM Pk daily parking cost (per-day cost
for AM, cents)
STAOPCOST = di.1.STAOPCOST ; Offpk parking cost (per-hour cost
for OP, cents; assumes 1 hour duration)
STAPKSHAD = di.1.STAPKSHAD ; AM Shadow parking cost (per-day cost
for AM, hundredths of minutes)
STAOPSHAD = di.1.STAOPSHAD ; Offpk Shadow parking cost (per-hour cost
for OP, hundredths of minutes; assumes 1 hour duration)

STACnt = dbi.1.NUMRECORDS

_parkam = INT(max(1.0,(STAPKCOST/2.0))) ; One-way AM period parking cost
_parkop = INT(max(1.0,(STAOPCOST/2.0))) ; One-way Off Pk period parking cost
; Note: computed as truncated integer for
consistency w/ Parker.for

_Walkk = 100.0 ; Base KNR walk connector time in
hundredths of min ('100.0' = 1 min)
_Walkpk = 200.0 ; Base AM PNR walk connector time in
hundredths of min ('100.0' = 1 min)
_Walkop = 200.0 ; Base OP PNR walk connector time in
hundredths of min ('100.0' = 1 min)

IF (STAPCAP > 500) _Walkpk = 250.0 ; Peak times are longer for stations with
larger lots
IF (STAPCAP > 1000) _Walkpk = 350.0 ;
IF (STAPCAP > 1500) _Walkpk = 400.0 ;
IF (STAPCAP > 2000) _Walkpk = 500.0 ;

;; write out Metrorail PNR-to-Station links (for AM & Offpeak periods)
IF (MM = 'M' && STAPARK = 'Y' && STAUSE = 'Y')
; units for _time are hundredths of minutes
; units for @MinPerDoll@ are hundredths of minutes per cent, which equals minutes
per dollar
_time = _walkpk + STAPKSHAD + (@MinPerDoll@ * _parkam)
_xtime = _time/100.0

Print printo =1 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
'DIST= ',_parkam(5),' TIME= ', _Xtime(8.2)

_time = _walkop + STAOPSHAD + (@MinPerDoll@ * _parkop)
_xtime = _time/100.0

Print printo =6 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
'DIST= ',_parkop(5),' TIME= ', _Xtime(8.2)

ENDIF

;; write out CommRail PNR-to-Station links (for AM & Offpeak periods)
IF (MM = 'C' && STAPARK = 'Y' && STAUSE = 'Y')

_time = _walkpk + STAPKSHAD + (@MinPerDoll@ * _parkam)
_xtime = _time/100.0

Print printo =2 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',

```



## Appendix C Cube Voyager Scripts

```

                ' DIST= ',_parkam(5),' TIME= ', _Xtime(8.2)
        _time = _walkop + STAOPSHAD + (@MinPerDoll@ * _parkop)
        _xtime = _time/100.0
        Print printo =7 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkop(5),' TIME= ', _Xtime(8.2)
        ENDIF

;; write out BUS PNR-to-Bus Stop Node links (for AM & Offpeak periods)
IF (MM = 'B' && STAPARK = 'Y' && STAUSE = 'Y')
        _time = _walkpk + STAPKSHAD + (@MinPerDoll@ * _parkam)
        _xtime = _time/100.0
        Print printo =3 list = 'SUPPLINK N=',STAP(5),'-',STAN1(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkam(5),' TIME= ', _Xtime(8.2)
        _time = _walkop + STAOPSHAD + (@MinPerDoll@ * _parkop)
        _xtime = _time/100.0
        Print printo =8 list = 'SUPPLINK N=',STAP(5),'-',STAN1(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkop(5),' TIME= ', _Xtime(8.2)
        ENDIF

;; write out Light Rail PNR-to-Station links (for AM & Offpeak periods)
IF (MM = 'L' && STAPARK = 'Y' && STAUSE = 'Y')
        _time = _walkpk + STAPKSHAD + (@MinPerDoll@ * _parkam)
        _xtime = _time/100.0
        Print printo =4 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkam(5),' TIME= ', _Xtime(8.2)
        _time = _walkop + STAOPSHAD + (@MinPerDoll@ * _parkop)
        _xtime = _time/100.0
        Print printo =9 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkop(5),' TIME= ', _Xtime(8.2)
        ENDIF

;; write out BRT/New PNR-to-Station links (for AM & Offpeak periods)
IF (MM = 'N' && STAPARK = 'Y' && STAUSE = 'Y')
        _time = _walkpk + STAPKSHAD + (@MinPerDoll@ * _parkam)
        _xtime = _time/100.0
        Print printo =5 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkam(5),' TIME= ', _Xtime(8.2)
        _time = _walkop + STAOPSHAD + (@MinPerDoll@ * _parkop)
        _xtime = _time/100.0
        Print printo =10 list = 'SUPPLINK N=',STAP(5),'-',STAT(5),' ONEWAY=Y
MODE=15',
                ' DIST= ',_parkop(5),' TIME= ', _Xtime(8.2)
        ENDIF

```

ENDLOOP

## 29 pathTrace.s

```

; pathTrace.s
; This file will get inserted into Transit Skims Steps to perform path traces
; for select i/j's (origins and destinations)
;
; 2010-10-08 MSM
;
; 3722 Juris
; TAZ Code Location Orig Dest
; -----
; 37 0 Downtown DC (Farragut West) x x
; 283 0 Union Station, DC x x
; 492 1 Gaithersburg, near Mont Co Airpark, MD x
; 520 1 Shady Grove, MD x
; 589 1 North Silver Spring, MD x
; 623 1 Silver Spring, MD x x
; 662 1 Bethesda, MD x x
; 717 1 Rockville, MD x x
; 906 2 Greenbelt, MD x
; 982 2 College Park, Univ. of Maryland x
; 1003 2 New Carrollton, MD x
; 1342 2 Andrews Air Force Base, MD x
; 1472 3 Rosslyn, Arlington, VA x
; 1496 3 Pentagon, Arlington, VA x x
; 1501 3 Crystal City, Arlington, VA x x
; 1599 4 Old Town Alexandria, VA x x
; 1679 5 South of Dulles Airport, VA x
; 1768 5 Reston, VA x
; 1823 5 Vienna, VA x
; 1843 5 Tysons Corner, VA x x
; 2032 5 Franconia-Springfield, VA x
; 2112 5 Fort Belvoir, VA x
; 2139 5 Rolling Road VRE Station, VA x
; 2250 6 Loudoun Co. near Brunswick MARC sta. x
; 2270 6 Leesburg, VA x
; 2632 7 Manassas City, Prince William Co, VA x
; 2751 7 Dale City, Prince William Co, VA x
; 2807 7 Quantico VRE, VA x
; 2928 9 City of Frederick, Fred. Co, MD x
; 3004 10 Jessup MARC Station, Howard Co, MD x
; 3007 10 North Laurel, Howard Co, MD x
; 3197 12 La Plata, Charles Co, MD x
; 3580 20 Spotsylvania Co, VA x

; ***** Comment out this section when running the model; Keep this for building
only select paths
;***** select i =
;***** 37, 492, 520, 589, 623, 662, 717, 906, 982, 1003, 1496, 1501,
;***** 1599, 1768, 1823, 1843, 2032, 2139, 2250, 2270, 2632, 2751, 2807,
;***** 2928, 3004, 3007, 3197, 3580,
;***** j =
;***** 37, 283, 623, 662, 717, 1342, 1472, 1496, 1501, 1599, 1679, 1843,
2112
; ***** End of section to be commented out when running travel model

select trace = (i =
37, 492, 520, 589, 623, 662, 717, 906, 982, 1003, 1496, 1501,
1599, 1768, 1823, 1843, 2032, 2139, 2250, 2270, 2632, 2751, 2807,
2928, 3004, 3007, 3197, 3580 &&
j =

```

## Appendix C Cube Voyager Scripts

37, 283, 623, 662, 717, 1342, 1472, 1496, 1501, 1599, 1679, 1843, 2112)

### 30 PP\_Auto\_Drivers.s

```

;-----
; PP_Auto_Drivers.S   Creating auto driver trips by occupant level (1,2,3+)
;                    from the pump prime trip distribution output
;                    using pre-existing NL model modal targets by market area
;                    (This process substitutes for a mode choice model run)
;                    in the initial 4-step iteration
; The 5 output matrix files will be:
;
;     1  <iter>_HBW_adr.mat
;     2  <iter>_HBS_adr.mat
;     3  <iter>_HBO_adr.mat
;     4  <iter>_NHW_adr.mat
;     5  <iter>_NHO_adr.mat
; .. each file with 3 tabs: 1occ,2occ,3+occ auto drivers
; Milone:- 1/5/11
;-----
; First, establish Input/Output filenames:
LOOP PURP=1,5 ; We'll Loop 5 times, for each purpose
;-----
; write out zonal person trip table that reflects
; Auto Person trips, based on HTS Auto drivers (nonHBW trip factored by 1.75)
; and transit trips adjusted to match the targets
;-----
; global auto occs from HTS and estimated occupancies by occ. group
; Purp avg_occ 1-occShr 2-occShr 3+occShr
; HBW 1.06 0.943806 0.054412 0.001782
; HBS 1.45 0.674235 0.239570 0.086195
; HBO 1.63 0.559809 0.307970 0.132221
; NHW 1.11 0.893861 0.104172 0.001967
; NHO 1.50 0.642450 0.258570 0.098980
;-----
;
IF (PURP=1) ; HBW Loop
MCFILE = 'INPUTS\HBW_NL_MC.MTT' ;AECOM Mode Choice file (Input)
TDFILE = '%_iter%_hbw.ptt' ;Trip distribution output
(Input)
MCL23OCC = '%_iter%_HBW_adr.mat' ;HBW Auto Drv trips- 1,2,3+ Occ.
(Output)
PURPOSE = 'HBW' ;
Avg3P_Occ = 3.50 ; Avg Auto Occupancy for autos w/ 3+ person
CarOcc = 1.06 ; Avg External Auto Occ.
adroccshr1 = 0.943806 ; assumed share of adrs that are 1 occ
adroccshr2 = 0.054412 ; 2 occ
adroccshr3 = 0.001782 ; 3+ occ
TDFile = '6' ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=2) ; HBS Loop
MCFILE = 'INPUTS\HBS_NL_MC.MTT' ;AECOM Mode Choice file (Input)
TDFILE = '%_iter%_hbs.ptt' ;Trip distribution output
(Input)
MCL23OCC = '%_iter%_HBS_adr.mat' ;HBS Auto Drv trips- 1,2,3+ Occ.
(Output)
PURPOSE = 'HBS' ;
Avg3P_Occ = 3.50 ; Avg Auto Occupancy for autos w/ 3+ person
CarOcc = 1.45 ; Avg External Auto Occ.
adroccshr1 = 0.674235 ; assumed share of adrs that are 1 occ
adroccshr2 = 0.239570 ; 2 occ

```

```

adroccshr3 = 0.086195 ; 3+ occ
TDFile = '6' ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=3) ; HBO Loop
MCFILE = 'INPUTS\HBO_NL_MC.MTT' ;AECOM Mode Choice file (Input)
TDFILE = '%_iter%_hbo.ptt' ;Trip distribution output
(Input)
MCL23OCC = '%_iter%_HBO_adr.mat' ;HBO Auto Drv trips- 1,2,3+ Occ.
(Output)
PURPOSE = 'HBO' ;
Avg3P_Occ = 3.50 ; Avg Auto Occupancy for autos w/ 3+ person
CarOcc = 1.63 ; Avg External Auto Occ.
adroccshr1 = 0.559809 ; assumed share of adrs that are 1 occ
adroccshr2 = 0.307970 ; 2 occ
adroccshr3 = 0.132221 ; 3+ occ
TDFile = '6' ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=4) ; NHW Loop
MCFILE = 'INPUTS\NHW_NL_MC.MTT' ;AECOM Mode Choice file (Input)
TDFILE = '%_iter%_nhw.ptt' ;Trip distribution output
(Input)
MCL23OCC = '%_iter%_NHW_adr.mat' ;NHW Auto Drv trips- 1,2,3+ Occ.
(Output)
PURPOSE = 'NHW' ;
Avg3P_Occ = 3.50 ; Avg Auto Occupancy for autos w/ 3+ person
CarOcc = 1.11 ; Avg External Auto Occ.
adroccshr1 = 0.893861 ; assumed share of adrs that are 1 occ
adroccshr2 = 0.104172 ; 2 occ
adroccshr3 = 0.001967 ; 3+ occ
TDFile = '3' ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=5) ; NHO Loop
MCFILE = 'INPUTS\NHO_NL_MC.MTT' ;AECOM Mode Choice file (Input)
TDFILE = '%_iter%_nho.ptt' ;Trip distribution output
(Input)
MCL23OCC = '%_iter%_NHO_adr.mat' ;NHO Auto Drv trips- 1,2,3+ Occ.
(Output)
PURPOSE = 'NHO' ;
Avg3P_Occ = 3.50 ; Avg Auto Occupancy for autos w/ 3+ person
CarOcc = 1.50 ; Avg External Auto Occ.
adroccshr1 = 0.642450 ; assumed share of adrs that are 1 occ
adroccshr2 = 0.258570 ; 2 occ
adroccshr3 = 0.098980 ; 3+ occ
TDFile = '3' ; Total Psn Trip tab no. in Trip Dist. output
file

ENDIF
;
;//////////////////////////////////////
; Step 1:
; - First read trip distribution person trips (from which auto drivers are to be
estimated) and
; - read a pre-existing nested logit mode choice model output.
; - Summarize both to the 20 market segments (seg. 21 refers to external areas)
; - computed auto person shares for each market area based on the NL output file
; - apply market level 'seed' auto person shares to the trip dist. person trips
; - write out the computed 'target' auto person trips at the market level.
; (these will be used in step 2 to apportion zone level trip dist person trips
among auto psn/drv by occ level)
RUN PGM=MATRIX
PAGEHEIGHT= 32767
array NLmkt_trips=5,21 ; array to summarize NL seed trips by mode
(1,2,3+occ apsn transit, psn) and market area 1-21 (21 is external)
array TDmkt_trips=5,21 ; array to summarize computed TD est. trips by
mode, based on seed shares

```

## Appendix C Cube Voyager Scripts

```

array Tdmtk_share=5,21 ; array to summarize computed TD est. shares by
mode, based on seed shares

MATI[1]=@TDFILE@ ; TRIP DISTRIBUTION OUTPUT FILE
MATI[2]=@MCFILE@ ; NL MODE CHOICE MODEL OUTPUT FILE (INTL TRIPS)

;; read in Trip Dist. person trips and NL model output seed trips, by mode

MW[101] = MI.1.@TDTAB@ ; put TOTAL PP
motorized person trips in mtx 101

MW[201] = MI.2.1 + MI.2.2 + MI.2.3 + MI.2.4 + MI.2.5 +
MI.2.6 + MI.2.7 + MI.2.8 + MI.2.9 + MI.2.10 +
MI.2.11 + MI.2.12 + MI.2.13 + MI.2.14 ; put 'seed' NL
MC psn trips by mode in mats 201-214 (I-I only)

MW[211] = MI.2.1 ;seed locc auto psn
MW[212] = MI.2.2 ;seed 2occ auto psn
MW[213] = MI.2.3 ;seed 3+occ auto psn
MW[214] = MW[201] - (MW[211] + MW[212] + MW[213]) ; seed transit

;; now summarize TD psn trips and seed trips by mode (transit, adr psn by 1,2,3+
occ)

LOOKUP Name=TAZ_NLMkt,
LOOKUP[1] = 1,Result = 2, ; Market no 1 to 7
Interpolate = N, FAIL=0,0,0,list=n,
file= ..\support\TAZ3722_to_7Mrkts.txt

jloop
IM = TAZ_NLMkt(1,I)
JM = TAZ_NLMkt(1,J)

Mkt = 21 ; default/external area
;; define zonal market index no. 1 through 7-- put value in matrix 99
if ((IM= 1 || IM= 3) && (JM= 1)) mkt= 1
if ((IM= 1 || IM= 3) && (JM= 2)) mkt= 2
if ((IM= 1 || IM= 3) && (JM= 3 || JM= 4 || JM= 5)) mkt= 3
if ((IM= 1 || IM= 3) && (JM= 6 || JM= 7)) mkt= 4

if ((IM= 4) && (JM= 1)) mkt= 5
if ((IM= 4) && (JM= 2)) mkt= 6
if ((IM= 4) && (JM= 3 || JM= 4 || JM= 5)) mkt= 7
if ((IM= 4) && (JM= 6 || JM= 7)) mkt= 8

if ((IM= 2 || IM= 5) && (JM= 1)) mkt= 9
if ((IM= 2 || IM= 5) && (JM= 2)) mkt=10
if ((IM= 2 || IM= 5) && (JM= 3 || JM= 4 || JM= 5)) mkt=11
if ((IM= 2 || IM= 5) && (JM= 6 || JM= 7)) mkt=12

if ((IM= 6) && (JM= 1)) mkt=13
if ((IM= 6) && (JM= 2)) mkt=14
if ((IM= 6) && (JM= 3 || JM= 4 || JM= 5)) mkt=15
if ((IM= 6) && (JM= 6 || JM= 7)) mkt=16

if ((IM= 7) && (JM= 1)) mkt=17
if ((IM= 7) && (JM= 2)) mkt=18
if ((IM= 7) && (JM= 3 || JM= 4 || JM= 5)) mkt=19
if ((IM= 7) && (JM= 6 || JM= 7)) mkt=20

MW[99] = mkt

;; summarize seed trips by mode, mkt

IF (Mkt > 0)

NLMkt_trips[1][mkt] = NLMkt_trips[1][mkt] + MW[211] ; NL seed 1-occ apsn

```

```

NLMkt_trips[2][mkt] = NLMkt_trips[2][mkt] + MW[212] ; NL seed 2-occ apsn
NLMkt_trips[3][mkt] = NLMkt_trips[3][mkt] + MW[213] ; NL seed 3+occ apsn
NLMkt_trips[4][mkt] = NLMkt_trips[4][mkt] + MW[214] ; NL seed transit
NLMkt_trips[5][mkt] = NLMkt_trips[5][mkt] + MW[211] + MW[212] + MW[213] +
MW[214] ; NL seed person

Tdmtk_trips[5][mkt] = Tdmtk_trips[5][mkt] + MW[101] ; Trip Dist Psn
trips
ENDIF
endjloop

IF (I=zones) ;; if at the end of program, write out dbf file with market shares

;; estimate TD trips based on NL model shares
Loop Mkt= 1,21
IF (NLMkt_Trips[5][mkt] > 0)
Tdmtk_trips[1][mkt] = Tdmtk_trips[5][mkt] * NLMkt_trips[1][mkt] /
NLMkt_trips[5][mkt]; est 1 occpsn Trip Dist Psn trips
Tdmtk_trips[2][mkt] = Tdmtk_trips[5][mkt] * NLMkt_trips[2][mkt] /
NLMkt_trips[5][mkt]; est 2 occpsn Trip Dist Psn trips
Tdmtk_trips[3][mkt] = Tdmtk_trips[5][mkt] * NLMkt_trips[3][mkt] /
NLMkt_trips[5][mkt]; est 3+occpsn Trip Dist Psn trips
Tdmtk_trips[4][mkt] = Tdmtk_trips[5][mkt] * NLMkt_trips[4][mkt] /
NLMkt_trips[5][mkt]; est Transit Trip Dist Psn trips
ELSE
Tdmtk_trips[1][mkt] = Tdmtk_trips[5][mkt] * @adrocchr1@
Tdmtk_trips[2][mkt] = Tdmtk_trips[5][mkt] * @adrocchr2@
Tdmtk_trips[3][mkt] = Tdmtk_trips[5][mkt] * @adrocchr3@
ENDIF
ENDLOOP

;; compute TD auto driver shares
loop Mkt= 1,21
IF ( Tdmtk_trips[5][mkt] > 0)
Tdmtk_share[1][mkt] = Tdmtk_trips[1][mkt] /Tdmtk_trips[5][mkt]
Tdmtk_share[2][mkt] = Tdmtk_trips[2][mkt] /Tdmtk_trips[5][mkt]
Tdmtk_share[3][mkt] = Tdmtk_trips[3][mkt] /Tdmtk_trips[5][mkt]
Tdmtk_share[4][mkt] = Tdmtk_trips[4][mkt] /Tdmtk_trips[5][mkt]
ENDIF

FILEO reco[1] = TD_Shares@Purpose@.dbf, fields= mkt(5),
TDPsn1(12.2), TDPsn2(12.2), TDPsn3(12.2), TDtrn(12.2),
TDpsn(12.2),
TDPsn1Shr(12.6), TDPsn2Shr(12.6),TDPsn3Shr(12.6),TDtrnShr(12.6)

ro.mkt = mkt
ro.TDPsn1 = Tdmtk_trips[1][mkt] ; auto psn 1 occ trips
ro.TDPsn2 = Tdmtk_trips[2][mkt] ; auto psn 2 occ trips
ro.TDPsn3 = Tdmtk_trips[3][mkt] ; auto psn 3+occ trips
ro.TDTrn = Tdmtk_trips[4][mkt] ; transit trips
ro.TDPsn = Tdmtk_trips[5][mkt] ; person trips

ro.TDPsn1shr = Tdmtk_share[1][mkt] ; auto psn 1 occ trips share
ro.TDPsn2shr = Tdmtk_share[2][mkt] ; auto psn 2 occ trips share
ro.TDPsn3shr = Tdmtk_share[3][mkt] ; auto psn 3+occ trips share
ro.TDTrnshr = Tdmtk_share[4][mkt] ; transit trip share

WRITE RECO=1
ENDLOOP
endif

FILEO MATO[1] = Market1_21.Mtx, MO=99
ENDRUN
;

;//////////////////////////////////////
; Step 2:

```

## Appendix C Cube Voyager Scripts

```

; - read the computed 'target' auto person trips (developed above) at the market
level.
; - compute auto person shares from these targets at market level
; - apply shares to TD person trips, compute auto person/driver trips by occ. level
(1,2,3+)
;; - write out the PP auto driver trips
;; Note: There may be a small loss in the conservation of auto driver trips in
applying shares to trips at zone level
; (particularly for the higher auto occ. levels). This is acceptable for
the pump prime iteration
RUN PGM=MATRIX
  ZONES=3722
  MATI[1] = @TDFILE@ ; TRIP DISTRIBUTION OUTPUT FILE
  MATI[2] = Market1_21.Mtx ; zone file containing mkt index no (21 =
extl)
  FILEI DBI[1] ="TD_Shares@Purpose@.dbf" ; mkt level shares and target trips by
mode, computed above

  MW[101] = MI.1.@TDtab@ ; put TOTAL PP
  motorized person trips in mtx 101
  MW[201] = MI.2.1 ; put zonal mkt
  index in mtx 201

  array TDmkt_share = 9,21 ; array to summarize computed TD est. shares by
mode, based on seed shares
; and target INPUT trips from above

  array OTDmkt_trips= 8,21 ; array to summarize OUTPUT zone level TD est.
trips by mode, based on mkt level seed shares
; 8 modes:1/
Apsn1occ,2/Apsn2occ,3/Apsn3+occ,4/TRn,5/ADr1occ,6/ADr2occ,7/ADr3+occ,8/Psn
;
;; read share file into array
IF (I=1)
  LOOP K = 1,dbi.1.NUMRECORDS
    x = DBIReadRecord(1,k)
    mkt = di.1.mkt
    TDmkt_share[1][mkt] = di.1.TDPsn1shr
    TDmkt_share[2][mkt] = di.1.TDPsn2shr
    TDmkt_share[3][mkt] = di.1.TDPsn3shr
    TDmkt_share[4][mkt] = di.1.TDTrnshr
    TDmkt_share[5][mkt] = di.1.TDPsn1
    TDmkt_share[6][mkt] = di.1.TDPsn2
    TDmkt_share[7][mkt] = di.1.TDPsn3
    TDmkt_share[8][mkt] = di.1.TDtrn
    TDmkt_share[9][mkt] = di.1.TDpsn

    ;; echo print
    ;; print form=12.6 list = mkt(5),
    ;; TDmkt_share[1][mkt],
    ;; TDmkt_share[2][mkt],
    ;; TDmkt_share[3][mkt],
    ;; TDmkt_share[4][mkt],
    file=Share_@purpose@_Chk.txt
  ENDDOOP
ENDIF

;; Apply mkt level shares to zonal person trips
Jloop

IF (mw[201] > 0) ; ;
  mkt= mw[201] ; Est:
  mw[301] = MW[101] * TDmkt_share[1][mkt] ; zonal 1-occ auto
persons

```

```

persons mw[302] = MW[101] * TDmkt_share[2][mkt] ; zonal 2-occ auto
person mw[303] = MW[101] * TDmkt_share[3][mkt] ; zonal 3-occ auto
person mw[304] = MW[101] * TDmkt_share[4][mkt] ; zonal TRANSIT
mw[305] = MW[101] * TDmkt_share[1][mkt] / 1.0 ; zonal 1-occ auto
drivers mw[306] = MW[101] * TDmkt_share[2][mkt] / 2.0 ; zonal 2-occ auto
drivers mw[307] = MW[101] * TDmkt_share[3][mkt] / @Avg3P_Occ@ ; zonal 3-occ auto
drivers

; ; otherwise
; ; apply external default pct's
ELSE
  mw[301] = MW[101] * @adroccshr1@ ; zonal 1-occ auto
persons
  mw[302] = MW[101] * @adroccshr2@ ; zonal 2-occ auto
persons
  mw[303] = MW[101] * @adroccshr3@ ; zonal 3-occ auto
persons

  mw[305] = MW[101] * @adroccshr1@ / 1.0 ; zonal 1-occ auto drivers
  mw[306] = MW[101] * @adroccshr2@ / 2.0 ; zonal 2-occ auto drivers
  mw[307] = MW[101] * @adroccshr3@ / @Avg3P_Occ@ ; zonal 3-occ auto drivers
ENDIF

; ; Accumulate computed trips by mode

  OTDmkt_trips[1][mkt] = OTDmkt_trips[1][mkt] + MW[301]
; TD est. 1-occ psn
  OTDmkt_trips[2][mkt] = OTDmkt_trips[2][mkt] + MW[302]
; TD est. 2-occ psn
  OTDmkt_trips[3][mkt] = OTDmkt_trips[3][mkt] + MW[303]
; TD est. 3+occ psn
  OTDmkt_trips[4][mkt] = OTDmkt_trips[4][mkt] + MW[304]
; TD est. transit
  OTDmkt_trips[5][mkt] = OTDmkt_trips[5][mkt] + MW[305]
; TD est. 1-occ adr
  OTDmkt_trips[6][mkt] = OTDmkt_trips[6][mkt] + MW[306]
; TD est. 2-occ adr
  OTDmkt_trips[7][mkt] = OTDmkt_trips[7][mkt] + MW[307]
; TD est. 3+occ adr
  OTDmkt_trips[8][mkt] = OTDmkt_trips[8][mkt] + MW[301] + MW[302] + MW[303] +
MW[304] ; TD est. Person

  ENDJLOOP

  FILEO MATO[1] = @MC123OCC@,mo=305,306,307 ; ; output auto driver matrix - 3tabs
(1,2,3+occ adrs)

; ; At the end of processing, write out the OUTPUT trips by mode along with INPUT
trips by mode for checking
IF (I=zones)
  loop Mkt= 1,21
    FILEO reco[1] = PP_Auto_Drivers_@Purpose@.dbf, fields= mkt(5),
    OTDpsn1(12.2),OTDpsn2(12.2), OTDpsn3(12.2),OTDTrn(12.2),
    OTDadr1(12.2),OTDadr2(12.2), OTDadr3(12.2),OTDpsn(12.2),
    ITDpsn1(12.2), ITDpsn2(12.2), ITDpsn3(12.2),ITDtrn(12.2),
    ITDpsn(12.2)

    ro.mkt = mkt
    ro.OTDpsn1 = OTDmkt_trips[1][mkt] ; OUTPUT auto drv 1 occ
trips
    ro.OTDpsn2 = OTDmkt_trips[2][mkt] ; OUTPUT auto drv 2 occ
trips

```



## Appendix C Cube Voyager Scripts

```

;
; Programmer: Milone
; Date: 4/08/08
;
;=====
;
;=====
; Set Parameters:
;=====
ZONESIZE = 3722 ; No. of TAZs
LastIZN = 3675 ; Last Internal TAZ no.
PCostRng = 51 ; No. of ranges in the parking cost Model
TTimeRng = 5 ; No. of ranges in the terminal time
Model

Rept = 'Prepare_MC_Zfile.txt' ; Summary Reports
OFile = 'ZONEV2.A2F ' ; Output ZFile for the NL Mode Choice Model

;=====
; Set Input Files:
;=====
ZNFIL LU = 'inputs\zone.dbf' ; Input Zonal Land Use File
ZNFIL MCMrkt = 'inputs\areadef3722.prn' ; Input Zonal TAZ-Mode choice district
equiv.

;=====
;//////////////////////////////////////
; Begin TP+ Matrix Routine :
;=====
\//////////////////////////////////////
;=====

RUN PGM=MATRIX
ZONES=@ZONESIZE@
ARRAY MetroShortA=101, ; Arrays for counting TAZs in pct walk bins of 0-100
MetroLongA=101,
AMShortA=101,
AMLongA=101,
OPShortA=101,
OPLongA=101,

MetroShortj=24, ; Arrays for counting TAZs in juris bins
MetroLongj=24,
AMShortj=24,
AMLongj=24,
OPShortj=24,
OPLongj=24,
Total_Area=24

;=====
; Read Zonal Area Type Lookup file =
;=====
;
;
FileI LOOKUPI[1] = "@atypfile@"
LOOKUP LOOKUPI=1, NAME=ZNAT,
LOOKUP[1] = TAZ, RESULT=AType, ;
LOOKUP[2] = TAZ, RESULT=EMPDEN, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

```

```

;=====
; End of LookUps Now read the input files
;=====
;=====

;; First initialize all current values to zero:

HBWParkCost = 0
HBSParkCost = 0
HBOParkCost = 0
NHParkCost = 0
HB_TermTime = 0
NHB_TermTime = 0
MetroShort = 0
MetroLong = 0
AMShort = 0
AMLong = 0
OPShort = 0
OPLong = 0
EMP = 0
jur = 0
area = 0
AMMELONG = 0

; read Zonal land use files into Z-File
ZDATI[1] = @ZNFIL LU@;Z
LANDAREA ;EMP TOTEMP INDEMP RETEMP OFFEMP OTHEMP JURCODE
;jur
;Area

; Current Zonal Totals:
EMP = zi.1.TOTEMP[I]
jur = zi.1.jurcode[I] + 1 ; convert 0-23 jur
codes to 1 to 24 for indexing
Area = zi.1.LandArea[I]
IF (Area > 0)
EMPDENSITY = ROUND(EMP/AREA)
ELSE
EMPDENSITY = 0
ENDIF

; Accumulate Regional Totals:
TOTEMP = TOTEMP + zi.1.TOTEMP[I]
TOTArea = TOTArea + zi.1.LandArea[I]

; Zonal MC TAZ -District Equiv. File
ZDATI[2] = @ZNFIL MCMrkt@, Z = #1,
MCDistrict = #2
MCDistrict = zi.2.MCDistrict[I]

; Zonal Transit Walk Shares
ZDATI[3] = @ZNFIL TrPcts@ , Z = #1, ; TAZ
MetroShort = #2, ; % of TAZ that is w/in short walk
distance (0.5mi) to Metrorail
MetroLong = #3, ; % of TAZ that is w/in long walk
distance (1.0mi) to Metrorail
AMShort = #4, ; % of TAZ that is w/in short walk
distance (0.5mi) to AM Transit
AMLong = #5, ; % of TAZ that is w/in long walk
distance (1.0mi) to AM Transit
OPShort = #6, ; % of TAZ that is w/in short walk
distance (0.5mi) to OP Transit
OPLong = #7 ; % of TAZ that is w/in long walk
distance (1.0mi) to OP Transit

```

## Appendix C Cube Voyager Scripts

```

;; Convert walk shares to percents (i.e., 1.00 will be 100)
MetroShort = Round(zi.3.MetroShort[I] * 100.0)
MetroLong  = Round(zi.3.MetroLong[I]  * 100.0)
AMShort    = Round(zi.3.AMShort[I]    * 100.0)
AMLong     = Round(zi.3.AMLong[I]     * 100.0)
OPShort    = Round(zi.3.OPShort[I]    * 100.0)
OPLong     = Round(zi.3.OPLong[I]     * 100.0)

AMMELON = 0.0 ; AM Long-mutually exclusive of AM Short
area
AMMELONG = 0.0
IF (AMSHORT = 100.0 )
AMMELONG = AMLONG - AMShort IF (AMSHORT > 0.0 && AMSHORT < 100.0 && AMLONG > 0)
AMMELONG = AMLONG IF (AMSHORT = 0.0 && AMLONG > 0.0)

;; Do some QC checks on the Percent walk data
IF (MetroShort < 0 || MetroShort > 100)
List = ' MetroShort value: ', MetroShort, ' out of expected range at
TAZ:',I
Abort
ENDIF
IF (MetroLong < 0 || MetroLong > 100)
List = ' MetroLong value: ', MetroLong, ' out of expected range at
TAZ:',I
Abort
ENDIF
IF (AMShort < 0 || AMShort > 100)
List = ' AMShort value: ', AMShort, ' out of expected range at
TAZ:',I
Abort
ENDIF
IF (AMLong < 0 || AMLong > 100)
List = ' AMLong value: ', AMLong, ' out of expected range at
TAZ:',I
Abort
ENDIF
IF (OPShort < 0 || OPShort > 100)
List = ' OPShort value: ', OPShort, ' out of expected range at
TAZ:',I
Abort
ENDIF
IF (OPLong < 0 || OPLong > 100)
List = ' OPLong value: ', OPLong, ' out of expected range at
TAZ:',I
Abort
ENDIF

; Accumulate the count of TAZs in pct walk bins (0 to 100) for reporting
IF (Area > 0)
LOOP Idx = 1, 101 ;; indexes 1-101 refer to values 0 to 100
IF (MetroShort = (idx-1)) MetroShortA[idx] = MetroShortA[idx] + 1
IF (MetroLong = (idx-1)) MetroLongA[idx] = MetroLongA[idx] + 1

```

```

IF (AMShort = (idx-1)) AMShortA[idx] = AMShortA[idx] + 1
IF (AMLong = (idx-1)) AMLongA[idx] = AMLongA[idx] + 1
IF (OPShort = (idx-1)) OPShortA[idx] = OPShortA[idx] + 1
IF (OPLong = (idx-1)) OPLongA[idx] = OPLongA[idx] + 1
ENDLOOP
ActiveTAZCnt = ActiveTAZCnt + 1
ENDIF
; Accumulate the Area of each walk shed for reporting
MetroShortArea = MetroShortArea + (MetroShort/100.00 * Area)
MetroLongArea = MetroLongArea + (MetroLong /100.00 * Area)
AMShortArea = AMShortArea + (AMShort /100.00 * Area)
AMLongArea = AMLongArea + (AMLong /100.00 * Area)
OPShortArea = OPShortArea + (OPShort /100.00 * Area)
OPLongArea = OPLongArea + (OPLong /100.00 * Area)

; Accumulate the area of TAZs in juris. bins for reporting
IF (Area > 0)
LOOP Idx = 1, 24 ;; indexes 1-101 refer to values 0 to 100
IF (jur = idx ) MetroShortj[idx] = MetroShortj[idx] + (MetroShort/100.00 *
Area)
IF (jur = idx ) MetroLongj[idx] = MetroLongj[idx] + (MetroLong /100.00 *
Area)
IF (jur = idx ) AMShortj[idx] = AMShortj[idx] + (AMShort /100.00 *
Area)
IF (jur = idx ) AMLongj[idx] = AMLongj[idx] + (AMLong /100.00 *
Area)
IF (jur = idx ) OPShortj[idx] = OPShortj[idx] + (OPShort /100.00 *
Area)
IF (jur = idx ) OPLongj[idx] = OPLongj[idx] + (OPLong /100.00 *
Area)
IF (jur = idx ) Total_Area[idx] = Total_Area[idx] + Area
ENDLOOP
ENDIF

;;-----
;; Define hwy terminal times based on Area Type
;;-----
_AType = ZNAT(1,I) ; Area Type
_FEmpDen = ZNAT(2,I) ; Floating 1-mi zonal Employment density
if (_Atype = 1 ) Termtm= 5.0
if (_Atype = 2 ) Termtm= 4.0
if (_Atype = 3 ) Termtm= 3.0
if (_Atype = 4 ) Termtm= 2.0
if (_Atype = 5 ) Termtm= 1.0
if (_Atype = 6 ) Termtm= 1.0

if (I > @LastIZN@) Termtm = 0.0

HB_TermTime = TermTm
NHB_TermTime = TermTm

;;-----
;; Define hwy Parking costs based on Area Type --ALL IN 2007 CENTS
;;-----
;; HBW 8-Hour Parking Cost
IF (_Atype >0 && _Atype <= 3)
HBWParkCost = MAX( (217.24 * (Ln(_FEmpDen)) - 1553.3), 0.0 )
ELSE
HBWParkCost = 0.0
ENDIF

;; non-HBW 1-Hour Parking Cost
IF (_Atype = 1)
HrNonWkPkCost = 200.0

```





## Appendix C Cube Voyager Scripts

```

Print form=10, file=@Rept@, list = value,
MetroShortA[idx],
MetroLongA[idx],
AMShortA[idx],
AMLongA[idx],
OPShortA[idx],
OPLongA[idx]
ENDLOOP
Print form=10, file=@Rept@ list = '\n', '\n',
-----'\n',
Total', ActiveTAZCnt,
ActiveTAZCnt,ActiveTAZCnt,ActiveTAZCnt,
ActiveTAZCnt, ActiveTAZCnt
-----
;;-----
ENDIF
ENDRUN

;;opy TPPL*.prn Prepare_MC_ZFile.RPT

```

## 32 Prepare\_Ext\_Auto\_Ends.s

```

*del voya*.prn
;
;=====
; Prepare_Ext_Auto_Ends.s
=
; This process prepares Auto-related external Ps, As for the External Trip
Distribution Process =
; The zonal level internal Ps & As are scaled (or balanced) to match external As &
Ps, respectively =
;=====
ZONESIZE = 3722 ; No. of TAZs
Purps = 5 ; No. of purposes
LastIZn = 3675 ; Last Internal TAZ no.
Scaled_IntPsAs = '%_iter_%_Ext_Trip_Gen_PsAs.dbf' ;; OUTPUT external zonal Ps,As
file, HBW,HBS,HBO,NHW,NHO purposes

RUN PGM=MATRIX
ZONES=1

Fileo printo[1] = '%_iter_%_Ext_Trip_Gen_PsAs.txt' ;; report file

Array ZProdA = 5,3722 ; input zonal productions array /Unscaled
Array ZAttrA = 5,3722 ; input zonal attractions array /Unscaled

Array S_ZProdA = 5,3722 ; output zonal productions / intls scaled to extl
attr. totals
Array S_ZAttrA = 5,3722 ; output zonal attractions / intls scaled to extl
prod. totals

Array TotProda=5, IntProda=5, ExtProda=5, TotscaleP=5, TotscaleA=5
Array TotAttrA=5, IntAttrA=5, ExtAttrA=5, Pscale=5,Ascale=5, IntScaleP=5,
IntScaleA=5

;; INPUT Zonal trip productions
FILEI DBI[1] = "%_iter_%_Trip_Gen_productions_Comp.dbf"
;; variables in file:

```

```

;;TAZ HBW_MTR_PS HBW_NMT_PS HBW_ALL_PS HBWMTRP_I1 HBWMTRP_I2
HBWMTRP_I3 HBWMTRP_I4
;; HBS_MTR_PS HBS_NMT_PS HBS_ALL_PS HBSMTRP_I1 HBSMTRP_I2
HBSMTRP_I3 HBSMTRP_I4
;; HBO_MTR_PS HBO_NMT_PS HBO_ALL_PS HBOMTRP_I1 HBOMTRP_I2
HBOMTRP_I3 HBOMTRP_I4
;; NHW_MTR_PS NHW_NMT_PS NHW_ALL_PS NHO_MTR_PS NHO_NMT_PS
NHO_ALL_PS

```

```

;;INPUT Zonal final/scaled trip attractions
FILEI DBI[2] = "%_iter_%_Trip_Gen_Attractions_Comp.dbf"
;; variables in file:
;;TAZ HBW_MTR_AS HBW_NMT_AS HBW_ALL_AS HBWMTRA_I1 HBWMTRA_I2
HBWMTRA_I3 HBWMTRA_I4
;; HBS_MTR_AS HBS_NMT_AS HBS_ALL_AS HBSMTRA_I1 HBSMTRA_I2
HBSMTRA_I3 HBSMTRA_I4
;; HBO_MTR_AS HBO_NMT_AS HBO_ALL_AS HBOMTRA_I1 HBOMTRA_I2
HBOMTRA_I3 HBOMTRA_I4
;; NHW_MTR_AS NHW_NMT_AS NHW_ALL_AS NHO_MTR_AS NHO_NMT_AS
NHO_ALL_AS

```

;; Read productions into zonal array and accumulate, totals, internals, and externals by purpose

```

LOOP K = 1,dbi.1.NUMRECORDS
x = DBIReadRecord(1,k)
ZProda[1][di.1.TAZ] = di.1.HBW_Mtr_Ps
ZProda[2][di.1.TAZ] = di.1.HBS_Mtr_Ps
ZProda[3][di.1.TAZ] = di.1.HBO_Mtr_Ps
ZProda[4][di.1.TAZ] = di.1.NHW_Mtr_Ps
ZProda[5][di.1.TAZ] = di.1.NHO_Mtr_Ps

```

```

;; Accumulate total, internal and external P's by purpose
TotProda[1] = TotProda[1] + ZProda[1][di.1.TAZ]
TotProda[2] = TotProda[2] + ZProda[2][di.1.TAZ]
TotProda[3] = TotProda[3] + ZProda[3][di.1.TAZ]
TotProda[4] = TotProda[4] + ZProda[4][di.1.TAZ]
TotProda[5] = TotProda[5] + ZProda[5][di.1.TAZ]
TotProdaSum = TotProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ] + ZProda[4][di.1.TAZ] +
ZProda[5][di.1.TAZ]

```

```

IF (K <= @LastIZn@)
IntProda[1] = IntProda[1] + ZProda[1][di.1.TAZ]
IntProda[2] = IntProda[2] + ZProda[2][di.1.TAZ]
IntProda[3] = IntProda[3] + ZProda[3][di.1.TAZ]
IntProda[4] = IntProda[4] + ZProda[4][di.1.TAZ]
IntProda[5] = IntProda[5] + ZProda[5][di.1.TAZ]
IntProdaSum = IntProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ] + ZProda[4][di.1.TAZ] +
ZProda[5][di.1.TAZ]
ELSE
ExtProda[1] = ExtProda[1] + ZProda[1][di.1.TAZ]
ExtProda[2] = ExtProda[2] + ZProda[2][di.1.TAZ]
ExtProda[3] = ExtProda[3] + ZProda[3][di.1.TAZ]
ExtProda[4] = ExtProda[4] + ZProda[4][di.1.TAZ]
ExtProda[5] = ExtProda[5] + ZProda[5][di.1.TAZ]
ExtProdaSum = ExtProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ] + ZProda[4][di.1.TAZ] +
ZProda[5][di.1.TAZ]
ENDIF
ENDLOOP

```

;; Read attractions into zonal array and accumulate, totals, internals, and externals by purpose

```

LOOP K = 1,dbi.2.NUMRECORDS
x = DBIReadRecord(2,k)

```

## Appendix C Cube Voyager Scripts

```

ZAttr[1][di.2.TAZ] = di.2.HBW_Mtr_As
ZAttr[2][di.2.TAZ] = di.2.HBS_Mtr_As
ZAttr[3][di.2.TAZ] = di.2.HBO_Mtr_As
ZAttr[4][di.2.TAZ] = di.2.NHW_Mtr_As
ZAttr[5][di.2.TAZ] = di.2.NHO_Mtr_As

;; Accumulate total, internal and external P's by purpose
TotAttr[1] = TotAttr[1] + ZAttr[1][di.2.TAZ]
TotAttr[2] = TotAttr[2] + ZAttr[2][di.2.TAZ]
TotAttr[3] = TotAttr[3] + ZAttr[3][di.2.TAZ]
TotAttr[4] = TotAttr[4] + ZAttr[4][di.2.TAZ]
TotAttr[5] = TotAttr[5] + ZAttr[5][di.2.TAZ]
TotAttrSum = TotAttrSum + ZAttr[1][di.2.TAZ] +
ZAttr[2][di.2.TAZ] + ZAttr[3][di.2.TAZ] + ZAttr[4][di.2.TAZ] +
ZAttr[5][di.2.TAZ]

IF (K <= @LastIzn@)
  IntAttr[1] = IntAttr[1] + ZAttr[1][di.2.TAZ]
  IntAttr[2] = IntAttr[2] + ZAttr[2][di.2.TAZ]
  IntAttr[3] = IntAttr[3] + ZAttr[3][di.2.TAZ]
  IntAttr[4] = IntAttr[4] + ZAttr[4][di.2.TAZ]
  IntAttr[5] = IntAttr[5] + ZAttr[5][di.2.TAZ]
  IntAttrSum = IntAttrSum + ZAttr[1][di.2.TAZ] +
ZAttr[2][di.2.TAZ] + ZAttr[3][di.2.TAZ] + ZAttr[4][di.2.TAZ] +
ZAttr[5][di.2.TAZ]
ELSE
  ExtAttr[1] = ExtAttr[1] + ZAttr[1][di.2.TAZ]
  ExtAttr[2] = ExtAttr[2] + ZAttr[2][di.2.TAZ]
  ExtAttr[3] = ExtAttr[3] + ZAttr[3][di.2.TAZ]
  ExtAttr[4] = ExtAttr[4] + ZAttr[4][di.2.TAZ]
  ExtAttr[5] = ExtAttr[5] + ZAttr[5][di.2.TAZ]
  ExtAttrSum = ExtAttrSum + ZAttr[1][di.2.TAZ] +
ZAttr[2][di.2.TAZ] + ZAttr[3][di.2.TAZ] + ZAttr[4][di.2.TAZ] +
ZAttr[5][di.2.TAZ]
ENDIF
ENDLOOP

;; compute scaling factors by purpose

Loop pp= 1, @Purps@

  If (IntProda[pp] != 0) Pscale[pp] = ExtAttr[pp]/IntProda[pp]
  If (IntAttr[pp] != 0) Ascale[pp] = ExtProda[pp]/IntAttr[pp]

ENDLOOP

;;print input P/A results by intl, external groups
print printo=1 List = ' Listing of INPUT P/A Totals by Purpose and
computed scaling factors '
print printo= 1 form=12.2 list = ' '

print printo =1 list = ' Purpose>>> ' , '
HBW HBS HBO NHW NHO ALL'
print printo= 1 list = ' '
print printo= 1 form=16.2csv list = ' Total Internal Ps by purpose: ',
IntProda[1], IntProda[2], IntProda[3], IntProda[4], IntProda[5], IntProdaSum
print printo= 1 form=16.2csv list = ' Total External Ps by purpose: ',
ExtProda[1], ExtProda[2], ExtProda[3], ExtProda[4], ExtProda[5], ExtProdaSum
print printo= 1 form=16.2csv list = ' Total Intl&Extl Ps by purpose: ',
TotProda[1], TotProda[2], TotProda[3], TotProda[4], TotProda[5], TotProdaSum
print printo= 1 list = ' '
print printo= 1 form=16.2csv list = ' Total Internal As by purpose: ',
IntAttr[1], IntAttr[2], IntAttr[3], IntAttr[4], IntAttr[5], IntAttrSum
print printo= 1 form=16.2csv list = ' Total External As by purpose: ',
ExtAttr[1], ExtAttr[2], ExtAttr[3], ExtAttr[4], ExtAttr[5], ExtAttrSum
print printo= 1 form=16.2csv list = ' Total Intl&Extl As by purpose: ',
TotAttr[1], TotAttr[2], TotAttr[3], TotAttr[4], TotAttr[5], TotAttrSum
print printo= 1 list = ' '

```

```

print printo= 1 form=16.6csv list = 'Prod_scale fts ExtAs/IntlPs: ',
Pscale[1], Pscale[2], Pscale[3], Pscale[4], Pscale[5]
print printo= 1 form=16.6csv list = 'Attr_scale fts ExtPs/ExtlPs: ',
Ascale[1], Ascale[2], Ascale[3], Ascale[4], Ascale[5]
print printo= 1 list = ' '
print printo= 1 list = ' '
print printo= 1 list = ' '

;;set up out file

;; DEFINE OUTPUT FILE & VARIABLES
FILEO RECO[1] = "@Scaled_IntPsAs@",
fields = TAZ(5),
SHBW_MtrPs(15.2), SHBS_MtrPs(15.2), SHBO_MtrPs(15.2),
SNHW_MtrPs(15.2), SNHO_MtrPs(15.2),
SHBW_MtrAs(15.2), SHBS_MtrAs(15.2), SHBO_MtrAs(15.2),
SNHW_MtrAs(15.2), SNHO_MtrAs(15.2),
NHIIAs(15.2), NHOIIAs(15.2)

;;
;; Now loop through each internal TAZ and
;; 1) scale INT Attractions to EXT productions
;; 2) scale INT Productions to EXT attractions
;; 3) write out scaled/INT Ps As and unscaled EXT P's, As

Loop zz= 1, @ZONESIZE@

  Loop pp= 1, @Purps@

    IF (zz <= @LastIzn@) ;;if TAZ is internal, then scale and accumulate
      S_ZProda[pp][zz] = ZProda[pp][zz] * Pscale[pp]
      S_ZAttr[pp][zz] = ZAttr[pp][zz] * Ascale[pp]

;; accumulate scaled internal Ps, As by purpose and for total
IntScaleP[pp] = IntScaleP[pp] + S_ZProda[pp][zz]
IntScaleA[pp] = IntScaleA[pp] + S_ZAttr[pp][zz]

IntScalePSum = IntScalePSum + S_ZProda[pp][zz]
IntScaleASum = IntScaleASum + S_ZAttr[pp][zz]

ELSE ;; Else TAZ is external, final scaled P/S equals
input P,A
  S_ZProda[pp][zz] = ZProda[pp][zz]
  S_ZAttr[pp][zz] = ZAttr[pp][zz]

ENDIF ;;
;; Accum. total of scaled intl and untouched extls for reporting, by
purpose and for total
TotScaleP[pp] = TotScaleP[pp] + S_ZProda[pp][zz]
TotScaleA[pp] = TotScaleA[pp] + S_ZAttr[pp][zz]

TotScalePSum = TotScalePSum + S_ZProda[pp][zz]
TotScaleASum = TotScaleASum + S_ZAttr[pp][zz]

ENDLOOP

;; Write out the unscaled and scaled Ps,As by purpose
;; The scaled internal productions will equal the sum of external attractions
;; The scaled internal attractions will equal the sum of external productions
;; The external Ps, As will remain unchanged
ro.TAZ = zz
ro.SHBW_MtrPs = S_ZProda[1][zz]
ro.SHBS_MtrPs = S_ZProda[2][zz]
ro.SHBO_MtrPs = S_ZProda[3][zz]

```

## Appendix C Cube Voyager Scripts

```

ro.SNHW_MtrPs = S_ZAttrA[4][zz]
ro.SNHO_MtrPs = S_ZAttrA[5][zz]

ro.SHBW_MtrAs = S_ZAttrA[1][zz]
ro.SHBS_MtrAs = S_ZAttrA[2][zz]
ro.SHBO_MtrAs = S_ZAttrA[3][zz]
ro.SNHW_MtrAs = S_ZAttrA[4][zz]
ro.SNHO_MtrAs = S_ZAttrA[5][zz]

IF (ZZ <= @LastIZn@)
  ro.NHWIIAs = ZAttrA[4][zz]
  ro.NHOIIAs = ZAttrA[5][zz]
ELSE
  ro.NHWIIAs = 0.0
  ro.NHOIIAs = 0.0
ENDIF

WRITE RECO=1

ENDLOOP

print printo=1      List = ' Listing of OUTPUT P/A Totals by purpose to be
used in the External Trip Distribution Process '
;;print input P/A results by intl, external groups

  print printo= 1      list = '
  print printo =1      list = ' Purpose>>>
HBW      HBS      HBO      NHW      NHO      ALL'
  print printo= 1      list = '
  print printo= 1 form=16.2csv list = ' Internal Ps, scaled to Extl As:',
IntScaleP[1], IntScaleP[2],IntScaleP[3], IntScaleP[4], IntScaleP[5], IntScalePSum
  print printo= 1 form=16.2csv list = ' External Ps by purpose:
ExtProda[1], ExtProda[2], ExtProda[3], ExtProda[4], ExtProda[5], ExtProdaSum
  print printo= 1 form=16.2csv list = ' Total Ps by purpose:
TotScaleP[1], TotScaleP[2],TotScaleP[3], TotScaleP[4], TotScaleP[5], TotScalePSum
  print printo= 1      list = '
  print printo= 1 form=16.2csv list = ' Internal As, scaled to Extl Ps:',
IntScaleA[1], IntScaleA[2],IntScaleA[3], IntScaleA[4], IntScaleA[5], IntScaleASum
  print printo= 1 form=16.2csv list = ' Total External As by purpose:
ExtAttrA[1], ExtAttrA[2], ExtAttrA[3], ExtAttrA[4], ExtAttrA[5], ExtAttrASum
  print printo= 1 form=16.2csv list = ' Total Intl&Extl As by purpose:
TotScaleA[1], TotScaleA[2],TotScaleA[3], TotScaleA[4], TotScaleA[5], TotScaleASum
  print printo= 1      list = '

ENDRUN
*copy voya*.prn mod2.rpt

```

### 33 Prepare\_Ext\_ComTruck\_Ends.s

```

*del voya*.prn
;
;=====
; Prepare_Ext_ComTrk_Ends.s
=
; This process prepares CV and Truck-related external Ps, As for the External Trip
Distribution Process =
; The zonal level internal Ps & As are scaled (or balanced) to match external As &
Ps, respectively =
;=====

```

```

ZONESIZE      = 3722                ; No. of TAZs
Purps         = 3                    ; No. of purposes
LastIZn       = 3675                ; Last Internal TAZ no.
Scaled_IntPsAs = '%_iter_%_Ext_CVTruck_Gen_PsAs.dbf' ; OUTPUT external zonal Ps,As
file, HBW,HBS,HBO,NHW,NHO purposes

RUN PGM=MATRIX
ZONES=1

Fileo printo[1] = '%_iter_%_Ext_CVTruck_Gen_PsAs.txt' ;; report file

Array ZProda   = 5,3722            ; input zonal productions array /Unscaled
Array ZAttrA   = 5,3722            ; input zonal attractions array /Unscaled

Array S_ZProda = 5,3722            ; output zonal productions / intls scaled to extl
attr. totals
Array S_ZAttrA = 5,3722            ; output zonal attractions / intls scaled to extl
prod. totals

Array TotProda=5, IntProda=5, ExtProda=5, TotscaleP=5, TotscaleA=5
Array TotAttrA=5, IntAttrA=5, ExtAttrA=5, Pscale=5,Ascale=5, IntScaleP=5,
IntScaleA=5

;; INPUT Zonal trip productions

;;INPUT Zonal comm, med truck, heavy truck trip ends
FILEI DBI[1] = "%_iter_%_ComVeh_Truck_Ends.dbf"
;; variables in file:
; TAZ      COMM_VEH      MED_TRUCK      HVY_TRUCK

;; Read productions into zonal array and accumulate, totals, internals, and
externals by purpose
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  ZProda[1][di.1.TAZ] = di.1.Comm_Veh
  ZProda[2][di.1.TAZ] = di.1.Med_Truck
  ZProda[3][di.1.TAZ] = di.1.Hvy_Truck

;; Accumulate total, internal and external P's by purpose
  TotProda[1] = TotProda[1] + ZProda[1][di.1.TAZ]
  TotProda[2] = TotProda[2] + ZProda[2][di.1.TAZ]
  TotProda[3] = TotProda[3] + ZProda[3][di.1.TAZ]
  TotProdaSum = TotProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ]

  IF (K <= @LastIZn@)
    IntProda[1] = IntProda[1] + ZProda[1][di.1.TAZ]
    IntProda[2] = IntProda[2] + ZProda[2][di.1.TAZ]
    IntProda[3] = IntProda[3] + ZProda[3][di.1.TAZ]
    IntProdaSum = IntProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ]
  ELSE
    ExtProda[1] = ExtProda[1] + ZProda[1][di.1.TAZ]
    ExtProda[2] = ExtProda[2] + ZProda[2][di.1.TAZ]
    ExtProda[3] = ExtProda[3] + ZProda[3][di.1.TAZ]
    ExtProdaSum = ExtProdaSum + ZProda[1][di.1.TAZ] +
ZProda[2][di.1.TAZ] + ZProda[3][di.1.TAZ]
  ENDIF
ENDIF

ENDLOOP

;; Read attractions into zonal array and accumulate, totals, internals, and
externals by purpose
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  ZAttrA[1][di.1.TAZ] = di.1.Comm_Veh
  ZAttrA[2][di.1.TAZ] = di.1.Med_Truck

```

## Appendix C Cube Voyager Scripts

```

ZAttr[3][di.1.TAZ] = di.1.Hvy_Truck
;; Accumulate total, internal and external P's by purpose
TotAttr[1] = TotAttr[1] + ZAttr[1][di.1.TAZ]
TotAttr[2] = TotAttr[2] + ZAttr[2][di.1.TAZ]
TotAttr[3] = TotAttr[3] + ZAttr[3][di.1.TAZ]
TotAttrSum = TotAttrSum + ZAttr[1][di.1.TAZ] +
ZAttr[2][di.1.TAZ] + ZAttr[3][di.1.TAZ]
IF (K <= @LastIzn@)
  IntAttr[1] = IntAttr[1] + ZAttr[1][di.1.TAZ]
  IntAttr[2] = IntAttr[2] + ZAttr[2][di.1.TAZ]
  IntAttr[3] = IntAttr[3] + ZAttr[3][di.1.TAZ]
  IntAttrSum = IntAttrSum + ZAttr[1][di.1.TAZ] +
ZAttr[2][di.1.TAZ] + ZAttr[3][di.1.TAZ]
ELSE
  ExtAttr[1] = ExtAttr[1] + ZAttr[1][di.1.TAZ]
  ExtAttr[2] = ExtAttr[2] + ZAttr[2][di.1.TAZ]
  ExtAttr[3] = ExtAttr[3] + ZAttr[3][di.1.TAZ]
  ExtAttrSum = ExtAttrSum + ZAttr[1][di.1.TAZ] +
ZAttr[2][di.1.TAZ] + ZAttr[3][di.1.TAZ]
ENDIF
ENDLOOP
;; compute scaling factors by purpose
Loop pp= 1, @Purps@
  If (IntProda[pp]!= 0) Pscale[pp] = ExtAttr[pp]/IntProda[pp]
  If (IntAttr[pp]!= 0) Ascale[pp] = ExtProda[pp]/IntAttr[pp]
ENDLOOP
;;print input P/A results by intl, external groups
print printo=1 List = ' Listing of INPUT Commercial Veh. and Truck
P/A Totals by Purpose and computed scaling factors '
print printo= 1 form=12.2 list = '
print printo =1 list = ' Purpose>>> ',
Com_Veh MedTrk HvyTrk ALL'
Print printo= 1 list = '
print printo= 1 form=16.2csv list = ' Total Internal Ps by purpose: ',
IntProda[1], IntProda[2], IntProda[3], IntProdaSum
print printo= 1 form=16.2csv list = ' Total External Ps by purpose: ',
ExtProda[1], ExtProda[2], ExtProda[3], ExtProdaSum
print printo= 1 form=16.2csv list = ' Total Intl&Extl Ps by purpose: ',
TotProda[1], TotProda[2], TotProda[3], TotProdaSum
print printo= 1 list = '
print printo= 1 form=16.2csv list = ' Total Internal As by purpose: ',
IntAttr[1], IntAttr[2], IntAttr[3], IntAttrSum
print printo= 1 form=16.2csv list = ' Total External As by purpose: ',
ExtAttr[1], ExtAttr[2], ExtAttr[3], ExtAttrSum
print printo= 1 form=16.2csv list = ' Total Intl&Extl As by purpose: ',
TotAttr[1], TotAttr[2], TotAttr[3], TotAttrSum
print printo= 1 list = '
print printo= 1 form=16.6csv list = 'Prod_scale fts ExtAs/IntlPs: ',
Pscale[1], Pscale[2], Pscale[3]
print printo= 1 form=16.6csv list = 'Attr_scale fts ExtPs/ExtlPs: ',
Ascale[1], Ascale[2], Ascale[3]
print printo= 1 list = '
print printo= 1 list = '
print printo= 1 list = '
;;set up out file
;; DEFINE OUTPUT FILE & VARIABLES

```

```

FILEO RECO[1] = "@Scaled_IntPsAs@",
fields = TAZ(5),
SCom_VehPs(15.2), SMed_TrkPs(15.2), SHvy_TrkPs(15.2),
SCom_VehAs(15.2), SMed_TrkAs(15.2), SHvy_TrkAs(15.2)
;;
;; Now loop through each internal TAZ and
;; 1) scale INT Attractions to EXT productions
;; 2) scale INT Productions to EXT attractions
;; 3) write out scaled/INT Ps As and unscaled EXT P's, As
Loop zz= 1, @ZONESIZE@
  Loop pp= 1, @Purps@
    IF (zz <= @LastIzn@) ;;if TAZ is internal, then scale and accumulate
      S_ZProda[pp][zz] = ZProda[pp][zz] * Pscale[pp]
      S_ZAttr[pp][zz] = ZAttr[pp][zz] * Ascale[pp]
;; accumulate scaled internal Ps, As by purpose and for total
IntScaleP[pp] = IntScaleP[pp] + S_ZProda[pp][zz]
IntScaleA[pp] = IntScaleA[pp] + S_ZAttr[pp][zz]
IntScalePSum = IntScalePSum + S_ZProda[pp][zz]
IntScaleASum = IntScaleASum + S_ZAttr[pp][zz]
ELSE ;; Else TAZ is external, final scaled P/S equals
input P,A
  S_ZProda[pp][zz] = ZProda[pp][zz]
  S_ZAttr[pp][zz] = ZAttr[pp][zz]
ENDIF ;;
;; Accum. total of scaled intl and untouched extls for reporting, by
purpose and for total
TotScaleP[pp] = TotScaleP[pp] + S_ZProda[pp][zz]
TotScaleA[pp] = TotScaleA[pp] + S_ZAttr[pp][zz]
TotScalePSum = TotScalePSum + S_ZProda[pp][zz]
TotScaleASum = TotScaleASum + S_ZAttr[pp][zz]
ENDLOOP
;; Write out the unscaled and scaled Ps,As by purpose
;; The scaled internal productions will equal the sum of external attractions
;; The scaled internal attractions will equal the sum of external productions
;; The external Ps, As will remain unchanged
ro.TAZ = zz
ro.SCom_VehPs = S_ZProda[1][zz]
ro.SMed_TrkPs = S_ZProda[2][zz]
ro.SHvy_TrkPs = S_ZProda[3][zz]
ro.SCom_VehAs = S_ZAttr[1][zz]
ro.SMed_TrkAs = S_ZAttr[2][zz]
ro.SHvy_TrkAs = S_ZAttr[3][zz]
WRITE RECO=1
ENDLOOP
print printo=1 List = ' Listing of OUTPUT Commercial Veh. and Truck P/A
Totals by purpose to be used in the External Trip Distribution Process '
;;print input P/A results by intl, external groups
print printo= 1 list = '

```

## Appendix C Cube Voyager Scripts

```

print printo=1 list = ' Purpose>>> ',
ComVeh MedTrk HvyTrk ALL'
print printo=1 list = '
print printo=1 form=16.2csv list = ' Internal Ps, scaled to Extl As:',
IntScaleP[1], IntScaleP[2],IntScaleP[3], IntScalePSum
print printo=1 form=16.2csv list = ' External Ps by purpose: ',
ExtProda[1], ExtProda[2], ExtProda[3], ExtProdaSum
print printo=1 form=16.2csv list = ' Total Ps by purpose: ',
TotScaleP[1], TotScaleP[2],TotScaleP[3], TotScalePSum
print printo=1 list = '
print printo=1 form=16.2csv list = ' Internal As, scaled to Extl Ps:',
IntScaleA[1], IntScaleA[2],IntScaleA[3], IntScaleASum
print printo=1 form=16.2csv list = ' Total External As by purpose: ',
ExtAttrA[1], ExtAttrA[2], ExtAttrA[3], ExtAttrASum
print printo=1 form=16.2csv list = ' Total Intl&Extl As by purpose: ',
TotScaleA[1], TotScaleA[2],TotScaleA[3], TotScaleASum
print printo=1 list = '

ENDRUN
*copy voya*.prn mod2.rpt

```

## 34 Prepare\_Internal\_Ends.s

```

*del voya*.prn
;
=====
; Prepare_Internal_Ends.s
=
; This process prepares Internal auto & truck-related Ps and As.
=
; The zonal level internal Ps & As are scaled (or balanced) to match external As &
Ps, respecti vely
;=====
ZONESIZE = 3722 ; No. of TAZs
Purps = 8 ; No. of purposes
LastIzn = 3675 ; Last Internal TAZ no.

Inp_ExtHBW = '%_iter_%_HBWext.ptt' ; INPUT External HBW trips
Inp_ExtHBS = '%_iter_%_HBSext.ptt' ; INPUT External HBS trips
Inp_ExtHBO = '%_iter_%_HBOext.ptt' ; INPUT External HBO trips
Inp_ExtNHW = '%_iter_%_NHWext.ptt' ; INPUT External NHW trips
Inp_ExtNHO = '%_iter_%_NHOext.ptt' ; INPUT External NHO trips
Inp_ExtCOM = '%_iter_%_COMext.vtt' ; INPUT External COM trips
Inp_ExtMTK = '%_iter_%_MTKext.vtt' ; INPUT External MTK trips
Inp_ExtHTK = '%_iter_%_HTKext.vtt' ; INPUT External HTK trips

Final_IntPsAs = '%_iter_%_Final_Int_Motor_PsAs.dbf' ; OUTPUT Internal zonal Ps,As
file, HBW,HBS,HBO,NHW,NHO, comm, mtk, htk purposes

;; -----
;; First extract zonal Ps, As, from the external trip tables
;; The external Ps (X-I trips) will be subsequently subtracted from Total As and
;; The external As (I-X trips) will be subsequently subtracted from Total Ps and
;;-----

RUN PGM=MATRIX
zones=@zonesize@

MATI[1] = @Inp_ExtHBW@ ; Input external person/vehicle trip tabs by purpose

```

```

MATI[2] = @Inp_ExtHBS@ ;
MATI[3] = @Inp_ExtHBO@ ;
MATI[4] = @Inp_ExtNHW@ ;
MATI[5] = @Inp_ExtNHO@ ;
MATI[6] = @Inp_ExtCOM@ ;
MATI[7] = @Inp_ExtMTK@ ;
MATI[8] = @Inp_ExtHTK@ ;

MW[100] = 1.0
MW[200] = 1.0

IF (I <= @LastIzn@) MW[200] = 0.0 ; screen matrix for ExtPs (X-I movements)
IF (I > @LastIzn@) MW[100] = 0.0 ; screen matrix for Ext As (I-X movements)

MW[11] = mi.1.1 * MW[100] MW[12] = mi.1.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[21] = mi.2.1 * MW[100] MW[22] = mi.2.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[31] = mi.3.1 * MW[100] MW[32] = mi.3.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[41] = mi.4.1 * MW[100] MW[42] = mi.4.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[51] = mi.5.1 * MW[100] MW[52] = mi.5.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[61] = mi.6.1 * MW[100] MW[62] = mi.6.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[71] = mi.7.1 * MW[100] MW[72] = mi.7.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table
MW[81] = mi.8.1 * MW[100] MW[82] = mi.8.1 * MW[200] ; HBW X-I trip table & HBW X-I
trip table

mato[1]=ext.tem, mo = 11,12,21,22,31,32,41,42,51,52,61,62,71,72,81,82

ENDRUN

RUN PGM=MATRIX
zones=@zonesize@
MATI[1] = ext.tem

MW[11] = mi.1.1 MW[12] = mi.1.2.T ; <-- HBW matrix, HBW matrix X-posed
MW[21] = mi.1.3 MW[22] = mi.1.4.T ; <-- HBS matrix, HBS matrix X-posed
MW[31] = mi.1.5 MW[32] = mi.1.6.T ; <-- HBO matrix, HBO matrix X-posed
MW[41] = mi.1.7 MW[42] = mi.1.8.T ; <-- NHW matrix, NHW matrix X-posed
MW[51] = mi.1.9 MW[52] = mi.1.10.T ; <-- NHO matrix, NHO matrix X-posed
MW[61] = mi.1.11 MW[62] = mi.1.12.T ; <-- COM matrix, COM matrix X-posed
MW[71] = mi.1.13 MW[72] = mi.1.14.T ; <-- MTK matrix, MTK matrix X-posed
MW[81] = mi.1.15 MW[82] = mi.1.16.T ; <-- HTK matrix, HTK matrix X-posed

FILEO RECO[1] = "ExternalPsAs.dbf",
FIELDS = TAZ, HBWXP, HBWXAs, HBSXP, HBSXAs, HBOXP, HBOXAs,
NHWXP, NHWXAs, NHOXP, NHOXAs, COMXP, COMXAs,
MTKXP, MTKXAs, HTKXP, HTKXAs

TAZ=i

HBWXP = ROWSUM(12) ;HBW X-I movements
HBSXP = ROWSUM(22) ;HBS
HBOXP = ROWSUM(32) ;HBO
NHWXP = ROWSUM(42) ;NHW
NHOXP = ROWSUM(52) ;NHO
COMXP = ROWSUM(62) ;COM
MTKXP = ROWSUM(72) ;MTK
HTKXP = ROWSUM(82) ;HTK

HBWXAs = ROWSUM(11) ;HBW I-X movements
HBSXAs = ROWSUM(21) ;HBS
HBOXAs = ROWSUM(31) ;HBO

```

## Appendix C Cube Voyager Scripts

```

NHWXAs = ROWSUM(41) ;NHW
NHOXAs = ROWSUM(51) ;NHO
COMXAs = ROWSUM(61) ;COM
MTKXAs = ROWSUM(71) ;MTK
HTKXAs = ROWSUM(81) ;HTK

WRITE RECO = 1

ENDRUN

;; -----
;; Now Read TOTAL Ps and As by Purpose, from the Trip_Generation.s and
Truck_Com_Trip_Generation.s scripts,
;; and subtract the external trip ends from above to arrive at zonal internal P's
and A's
;; to be used in the internal Trip Distribution process
;;-----

RUN PGM=MATRIX
ZONES=1

Fileo printo[1] = '%_iter_%_Prepare_Internal_Ends.txt' ;; report file

Array TmProdA = 8,3722 ; input TOTAL zonal motorized productions array (8
purposes)
Array TMattra = 8,3722 ; input TOTAL zonal motorized attractions array
Array TNMProdA = 8,3722 ; input TOTAL zonal non-motorized productions array (8
purposes)
Array TNMAttrA = 8,3722 ; input TOTAL zonal non-motorized attractions array
Array TMNMProdA = 8,3722 ; input TOTAL zonal motor&non-motr productions array (8
purposes)
Array TMNMAttrA = 8,3722 ; input TOTAL zonal motor&non-motr attractions array
Array TProdIncA = 3,3722,4 ; input TOTAL zonal motorized productions array (3
stratified purposes, 4 Inc groups)
Array TAttrIncA = 3,3722,4 ; input TOTAL zonal motorized attractions array (3
stratified purposes, 4 Inc groups)

Array XProdA = 8,3722 ; input EXTERNAL zonal productions array (8 purposes)
Array XAttrA = 8,3722 ; input EXTERNAL zonal attractions array

Array IMProdA = 8,3722 ; output INTERNAL motorized zonal productions array
(8 purposes)
Array IMAttrA = 8,3722 ; output INTERNAL motorized zonal attractions array
Array INMProdA = 8,3722 ; output INTERNAL non-motorized zonal productions array
(8 purposes)
Array INMAttrA = 8,3722 ; output INTERNAL non-motorized zonal attractions array
Array IMNMProdA = 8,3722 ; output INTERNAL motor&non-motr zonal productions array
(8 purposes)
Array IMNMAttrA = 8,3722 ; output INTERNAL motor&non-motr zonal attractions array
Array IProdIncA = 3,3722,4 ; output INTERNAL zonal motorized productions array
(3 stratified purposes, 4 Inc groups)
Array IAttrIncA = 3,3722,4 ; output INTERNAL zonal motorized attractions array
(3 stratified purposes, 4 Inc groups)

Array ScIMAttrA = 8,3722 ; output Scaled INTERNAL motorized zonal
attractions array
Array ScINMAttrA = 8,3722 ; output Scaled INTERNAL non-motorized zonal
attractions array
Array ScIMNMAttrA = 8,3722 ; output Scaled INTERNAL motor&non-Motr zonal
attractions array

;; Sum of total, external, internal Motorized Ps, As by purpose:
Array SUMTOTMP = 8, SUMTOTNMP =8, SUMTOTMNMP = 8

```

```

Array SumTotMA = 8, SumExtMP = 8, SumExtMA = 8, SumIntMP = 8, SumIntMA = 8,
AttScalFtr = 8, SumScIntMA=8

;; Sum of total, Non-Motorized Ps, As by purpose:
Array SUMINTNMP = 8, SUMINTNMA = 8, SumScIntNMA=8, SumScIntMNMA=8

;; Sum of total, Non-Motorized Ps, As by purpose:
Array SUMINTMNMP = 8, SUMINTMNMA = 8
Array SUMTOTNMA = 8, SUMTOTMNMA = 8
;;=====
;;=====

;; INPUT TOTAL Zonal trip productions
FILEI DBI[1] = "%_iter_%_Trip_Gen_productions_Comp.dbf"
;;variables in file:
;;TAZ HBW_MTR_PS HBW_NMT_PS HBW_ALL_PS HBWMTRP_I1 HBWMTRP_I2
HBWMTRP_I3 HBWMTRP_I4
;; HBS_MTR_PS HBS_NMT_PS HBS_ALL_PS HBSMTRP_I1 HBSMTRP_I2
HBSMTRP_I3 HBSMTRP_I4
;; HBO_MTR_PS HBO_NMT_PS HBO_ALL_PS HBOMTRP_I1 HBOMTRP_I2
HBOMTRP_I3 HBOMTRP_I4
;; NHW_MTR_PS NHW_NMT_PS NHW_ALL_PS NHO_MTR_PS NHO_NMT_PS
NHO_ALL_PS

;;INPUT TOTAL Zonal trip attractions
FILEI DBI[2] = "%_iter_%_Trip_Gen_Attractions_Comp.dbf"
;;variables in file:
;;TAZ HBW_MTR_AS HBW_NMT_AS HBW_ALL_AS HBWMTRA_I1 HBWMTRA_I2
HBWMTRA_I3 HBWMTRA_I4
;; HBS_MTR_AS HBS_NMT_AS HBS_ALL_AS HBSMTRA_I1 HBSMTRA_I2
HBSMTRA_I3 HBSMTRA_I4
;; HBO_MTR_AS HBO_NMT_AS HBO_ALL_AS HBOMTRA_I1 HBOMTRA_I2
HBOMTRA_I3 HBOMTRA_I4
;; NHW_MTR_AS NHW_NMT_AS NHW_ALL_AS NHO_MTR_AS NHO_NMT_AS
NHO_ALL_AS

;;INPUT Zonal comm, med truck, heavy truck trip ends
FILEI DBI[3] = "%_iter_%_ComVeh_Truck_Ends.dbf"
;; variables in file:
; TAZ COMM_VEH MED_TRUCK HVY_TRUCK

;;INPUT EXTERNAL Ps, As
FILEI DBI[4] = "ExternalPsAs.dbf"
;; FIELDS = TAZ, HBWXPs, HBWXAs, HBSXPs, HBSXAs, HBOXPs, HBOXAs,
;; NHWXPs, NHWXAs, NHOXPs, NHOXAs, COMXPs, COMXAs,
;; MTKXPs, MTKXAs, HTKXPs, HTKXAs

;;OUTPUT Internal Motorized P/A file:
FILEO RECO[1] = "@Final_IntPsAs@",
FIELDS = TAZ, HBWMIP, HBWNMIP, HBWMNMIP, HBWMNMIP,
HBWMIP1,HBWMIP2, HBWMIP3, HBWMIP4,
HBWMIA, HBWNMIA, HBWMNMIA,
HBWMIA1,HBWMIA2, HBWMIA3, HBWMIA4,
HBSMIP, HBSNMIP, HBSNMNIP,
HBSMIP1,HBSMIP2, HBSMIP3, HBSMIP4,
HBSMIA, HBSNMIA, HBSNMNIA,
HBSMIA1,HBSMIA2, HBSMIA3, HBSMIA4,
HBOMIP, HBONMIP, HBONNMIP,
HBOMIP1,HBOMIP2, HBOMIP3, HBOMIP4,
HBOMIA, HBONMIA, HBONNMIA,
HBOMIA1,HBOMIA2, HBOMIA3, HBOMIA4,
NHWMIP, NHWNMIP, NHWMNMIP,
NHWMIA, NHWNMIA, NHWMNMIA,

```

## Appendix C Cube Voyager Scripts

```

NHOMIP, NHONMIP, NHOMNMIP,
NHOMIA, NHONMIA, NHOMNMIA,
COMIP, COMIA,
MTKIP, MTKIA,
HTKIP, HTKIA

;; Read Total Internal productions into zonal array and accumulate internals by
purpose
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  IF (K <= @LastIZn@)
    TMsProda[1][di.1.TAZ] = di.1.HBW_Mtr_Ps
    TMsProda[2][di.1.TAZ] = di.1.HBS_Mtr_Ps
    TMsProda[3][di.1.TAZ] = di.1.HBO_Mtr_Ps
    TMsProda[4][di.1.TAZ] = di.1.NHW_Mtr_Ps
    TMsProda[5][di.1.TAZ] = di.1.NHO_Mtr_Ps

    TNMProda[1][di.1.TAZ] = di.1.HBW_NMt_Ps
    TNMProda[2][di.1.TAZ] = di.1.HBS_NMt_Ps
    TNMProda[3][di.1.TAZ] = di.1.HBO_NMt_Ps
    TNMProda[4][di.1.TAZ] = di.1.NHW_NMt_Ps
    TNMProda[5][di.1.TAZ] = di.1.NHO_NMt_Ps

    TMNMPProda[1][di.1.TAZ] = di.1.HBW_Mtr_Ps + di.1.HBW_ALL_Ps
    TMNMPProda[2][di.1.TAZ] = di.1.HBS_Mtr_Ps + di.1.HBS_ALL_Ps
    TMNMPProda[3][di.1.TAZ] = di.1.HBO_Mtr_Ps + di.1.HBO_ALL_Ps
    TMNMPProda[4][di.1.TAZ] = di.1.NHW_Mtr_Ps + di.1.NHW_ALL_Ps
    TMNMPProda[5][di.1.TAZ] = di.1.NHO_Mtr_Ps + di.1.NHO_ALL_Ps

    TProdInca[1][di.1.TAZ][1] = di.1.HBWMTRP_I1
    TProdInca[1][di.1.TAZ][2] = di.1.HBWMTRP_I2
    TProdInca[1][di.1.TAZ][3] = di.1.HBWMTRP_I3
    TProdInca[1][di.1.TAZ][4] = di.1.HBWMTRP_I4

    TProdInca[2][di.1.TAZ][1] = di.1.HBSMTRP_I1
    TProdInca[2][di.1.TAZ][2] = di.1.HBSMTRP_I2
    TProdInca[2][di.1.TAZ][3] = di.1.HBSMTRP_I3
    TProdInca[2][di.1.TAZ][4] = di.1.HBSMTRP_I4

    TProdInca[3][di.1.TAZ][1] = di.1.HBOMTRP_I1
    TProdInca[3][di.1.TAZ][2] = di.1.HBOMTRP_I2
    TProdInca[3][di.1.TAZ][3] = di.1.HBOMTRP_I3
    TProdInca[3][di.1.TAZ][4] = di.1.HBOMTRP_I4

    ;; Accumulate total P's by purpose
    SumTotMP[1] = SumTotMP[1] + TMsProda[1][di.1.TAZ]
    SumTotMP[2] = SumTotMP[2] + TMsProda[2][di.1.TAZ]
    SumTotMP[3] = SumTotMP[3] + TMsProda[3][di.1.TAZ]
    SumTotMP[4] = SumTotMP[4] + TMsProda[4][di.1.TAZ]
    SumTotMP[5] = SumTotMP[5] + TMsProda[5][di.1.TAZ]

    SumTotNMP[1] = SumTotNMP[1] + TNMProda[1][di.1.TAZ]
    SumTotNMP[2] = SumTotNMP[2] + TNMProda[2][di.1.TAZ]
    SumTotNMP[3] = SumTotNMP[3] + TNMProda[3][di.1.TAZ]
    SumTotNMP[4] = SumTotNMP[4] + TNMProda[4][di.1.TAZ]
    SumTotNMP[5] = SumTotNMP[5] + TNMProda[5][di.1.TAZ]

    SumTotMNMP[1] = SumTotMNMP[1] + TMsProda[1][di.1.TAZ] +
    TNMProda[1][di.1.TAZ]
    SumTotMNMP[2] = SumTotMNMP[2] + TMsProda[2][di.1.TAZ] +
    TNMProda[2][di.1.TAZ]
    SumTotMNMP[3] = SumTotMNMP[3] + TMsProda[3][di.1.TAZ] +
    TNMProda[3][di.1.TAZ]
    SumTotMNMP[4] = SumTotMNMP[4] + TMsProda[4][di.1.TAZ] +
    TNMProda[4][di.1.TAZ]

```

```

SumTotMNMP[5] = SumTotMNMP[5] + TMsProda[5][di.1.TAZ] +
    TNMProda[5][di.1.TAZ]
  ENDIF
ENDLOOP

;; Read Total Internal attractions into zonal array and accumulate internals by
purpose
LOOP K = 1,dbi.2.NUMRECORDS
  x = DBIReadRecord(2,k)
  IF (K <= @LastIZn@)
    TMAAttr[1][di.2.TAZ] = di.2.HBW_Mtr_As
    TMAAttr[2][di.2.TAZ] = di.2.HBS_Mtr_As
    TMAAttr[3][di.2.TAZ] = di.2.HBO_Mtr_As
    TMAAttr[4][di.2.TAZ] = di.2.NHW_Mtr_As
    TMAAttr[5][di.2.TAZ] = di.2.NHO_Mtr_As

    TNMAAttr[1][di.2.TAZ] = di.2.HBW_NMt_As
    TNMAAttr[2][di.2.TAZ] = di.2.HBS_NMt_As
    TNMAAttr[3][di.2.TAZ] = di.2.HBO_NMt_As
    TNMAAttr[4][di.2.TAZ] = di.2.NHW_NMt_As
    TNMAAttr[5][di.2.TAZ] = di.2.NHO_NMt_As

    TMNMAAttr[1][di.2.TAZ] = di.2.HBW_Mtr_As + di.2.HBW_NMt_As
    TMNMAAttr[2][di.2.TAZ] = di.2.HBS_Mtr_As + di.2.HBS_NMt_As
    TMNMAAttr[3][di.2.TAZ] = di.2.HBO_Mtr_As + di.2.HBO_NMt_As
    TMNMAAttr[4][di.2.TAZ] = di.2.NHW_Mtr_As + di.2.NHW_NMt_As
    TMNMAAttr[5][di.2.TAZ] = di.2.NHO_Mtr_As + di.2.NHO_NMt_As

    TAttrInca[1][di.2.TAZ][1] = di.2.HBWMTRA_I1
    TAttrInca[1][di.2.TAZ][2] = di.2.HBWMTRA_I2
    TAttrInca[1][di.2.TAZ][3] = di.2.HBWMTRA_I3
    TAttrInca[1][di.2.TAZ][4] = di.2.HBWMTRA_I4

    TAttrInca[2][di.2.TAZ][1] = di.2.HBSMTRA_I1
    TAttrInca[2][di.2.TAZ][2] = di.2.HBSMTRA_I2
    TAttrInca[2][di.2.TAZ][3] = di.2.HBSMTRA_I3
    TAttrInca[2][di.2.TAZ][4] = di.2.HBSMTRA_I4

    TAttrInca[3][di.2.TAZ][1] = di.2.HBOMTRA_I1
    TAttrInca[3][di.2.TAZ][2] = di.2.HBOMTRA_I2
    TAttrInca[3][di.2.TAZ][3] = di.2.HBOMTRA_I3
    TAttrInca[3][di.2.TAZ][4] = di.2.HBOMTRA_I4

    ;; Accumulate total P's by purpose
    SumTotMA[1] = SumTotMA[1] + TMAAttr[1][di.2.TAZ]
    SumTotMA[2] = SumTotMA[2] + TMAAttr[2][di.2.TAZ]
    SumTotMA[3] = SumTotMA[3] + TMAAttr[3][di.2.TAZ]
    SumTotMA[4] = SumTotMA[4] + TMAAttr[4][di.2.TAZ]
    SumTotMA[5] = SumTotMA[5] + TMAAttr[5][di.2.TAZ]

    SumTotNMA[1] = SumTotNMA[1] + TNMAAttr[1][di.2.TAZ]
    SumTotNMA[2] = SumTotNMA[2] + TNMAAttr[2][di.2.TAZ]
    SumTotNMA[3] = SumTotNMA[3] + TNMAAttr[3][di.2.TAZ]
    SumTotNMA[4] = SumTotNMA[4] + TNMAAttr[4][di.2.TAZ]
    SumTotNMA[5] = SumTotNMA[5] + TNMAAttr[5][di.2.TAZ]

    SumTotMNMA[1] = SumTotMNMA[1] + TMNMAAttr[1][di.2.TAZ]
    SumTotMNMA[2] = SumTotMNMA[2] + TMNMAAttr[2][di.2.TAZ]
    SumTotMNMA[3] = SumTotMNMA[3] + TMNMAAttr[3][di.2.TAZ]
    SumTotMNMA[4] = SumTotMNMA[4] + TMNMAAttr[4][di.2.TAZ]
    SumTotMNMA[5] = SumTotMNMA[5] + TMNMAAttr[5][di.2.TAZ]

  ENDIF
ENDLOOP

;; Read internal commercial, truck Ps/As zonal array and accumulate, totals,
internals, and externals by purpose

```

## Appendix C Cube Voyager Scripts

```

LOOP K = 1,dbi.3.NUMRECORDS
  x = DBIReadRecord(3,k)
  IF (K <= @LastIZn@)
    TMProda[6][di.3.TAZ] = di.3.Comm_Veh
    TMProda[7][di.3.TAZ] = di.3.Med_Truck
    TMProda[8][di.3.TAZ] = di.3.Hvy_Truck

    TMattra[6][di.3.TAZ] = di.3.Comm_Veh
    TMattra[7][di.3.TAZ] = di.3.Med_Truck
    TMattra[8][di.3.TAZ] = di.3.Hvy_Truck

    TMNMP[6][di.3.TAZ] = di.3.Comm_Veh
    TMNMP[7][di.3.TAZ] = di.3.Med_Truck
    TMNMP[8][di.3.TAZ] = di.3.Hvy_Truck

    TMNMA[6][di.3.TAZ] = di.3.Comm_Veh
    TMNMA[7][di.3.TAZ] = di.3.Med_Truck
    TMNMA[8][di.3.TAZ] = di.3.Hvy_Truck

    ;; Accumulate total P's by purpose
    SumTotMP[6] = SumTotMP[6] + TMProda[6][di.3.TAZ]
    SumTotMP[7] = SumTotMP[7] + TMProda[7][di.3.TAZ]
    SumTotMP[8] = SumTotMP[8] + TMProda[8][di.3.TAZ]

    SumTotMA[6] = SumTotMA[6] + TMattra[6][di.3.TAZ]
    SumTotMA[7] = SumTotMA[7] + TMattra[7][di.3.TAZ]
    SumTotMA[8] = SumTotMA[8] + TMattra[8][di.3.TAZ]

    ;; Accumulate total P's by purpose
    SumTotMNMP[6] = SumTotMP[6] + TMProda[6][di.3.TAZ]
    SumTotMNMP[7] = SumTotMP[7] + TMProda[7][di.3.TAZ]
    SumTotMNMP[8] = SumTotMP[8] + TMProda[8][di.3.TAZ]

    SumTotMNMA[6] = SumTotMA[6] + TMattra[6][di.3.TAZ]
    SumTotMNMA[7] = SumTotMA[7] + TMattra[7][di.3.TAZ]
    SumTotMNMA[8] = SumTotMA[8] + TMattra[8][di.3.TAZ]
  ENDIF
ENDLOOP

;; Read ALL External Ps and As zonal array and accumulate totals
LOOP K = 1,dbi.4.NUMRECORDS
  x = DBIReadRecord(4,k)
  XProda[1][di.4.TAZ] = di.4.HBWxps
  XProda[2][di.4.TAZ] = di.4.HBSXps
  XProda[3][di.4.TAZ] = di.4.HBOXps
  XProda[4][di.4.TAZ] = di.4.NHWXps
  XProda[5][di.4.TAZ] = di.4.NHOXps
  XProda[6][di.4.TAZ] = di.4.COMXps
  XProda[7][di.4.TAZ] = di.4.MTKXps
  XProda[8][di.4.TAZ] = di.4.HTKXps

  XAttr[1][di.4.TAZ] = di.4.HBWxas
  XAttr[2][di.4.TAZ] = di.4.HBSXas
  XAttr[3][di.4.TAZ] = di.4.HBOXas
  XAttr[4][di.4.TAZ] = di.4.NHWXas
  XAttr[5][di.4.TAZ] = di.4.NHOXas
  XAttr[6][di.4.TAZ] = di.4.COMXas
  XAttr[7][di.4.TAZ] = di.4.MTKXas
  XAttr[8][di.4.TAZ] = di.4.HTKXas

  ;; Accumulate total, internal and external P's by purpose
  SumExtMP[1] = SumExtMP[1] + XProda[1][di.4.TAZ]
  SumExtMP[2] = SumExtMP[2] + XProda[2][di.4.TAZ]
  SumExtMP[3] = SumExtMP[3] + XProda[3][di.4.TAZ]
  SumExtMP[4] = SumExtMP[4] + XProda[4][di.4.TAZ]

```

```

SumExtMP[5] = SumExtMP[5] + XProda[5][di.4.TAZ]
SumExtMP[6] = SumExtMP[6] + XProda[6][di.4.TAZ]
SumExtMP[7] = SumExtMP[7] + XProda[7][di.4.TAZ]
SumExtMP[8] = SumExtMP[8] + XProda[8][di.4.TAZ]

SumExtMA[1] = SumExtMA[1] + XAttr[1][di.4.TAZ]
SumExtMA[2] = SumExtMA[2] + XAttr[2][di.4.TAZ]
SumExtMA[3] = SumExtMA[3] + XAttr[3][di.4.TAZ]
SumExtMA[4] = SumExtMA[4] + XAttr[4][di.4.TAZ]
SumExtMA[5] = SumExtMA[5] + XAttr[5][di.4.TAZ]
SumExtMA[6] = SumExtMA[6] + XAttr[6][di.4.TAZ]
SumExtMA[7] = SumExtMA[7] + XAttr[7][di.4.TAZ]
SumExtMA[8] = SumExtMA[8] + XAttr[8][di.4.TAZ]

ENDLOOP

;;
;; compute INTERNAL Trip Ps,A by subtracting EXTERNAL Ends from TOTAL MOTORIZED&NON-
MOTR.ENDS, scale Ps by income group accordingly
;;
  LOOP ZZ = 1,@LastIZn@
    LOOP PP = 1, @Purps@

      IMProda[PP][ZZ] = MAX( 0, (TMProda[PP][ZZ] - XAttr[PP][ZZ]))
      IMAttr[PP][ZZ] = MAX( 0, (TMattra[PP][ZZ] - XProda[PP][ZZ]))

      ;; scale motorized trips by income level to match new Internal
      motorized total
      IF (PP<4)
        if (IMProda[PP][zz] = 0)

          IncScale = 0.0

        Else

          IncScale = IMProda[PP][ZZ]/TMProda[PP][ZZ]

        ENDIF

      IMProdInca[pp][zz][1] = TProdInca[pp][zz][1] * IncScale
      IMProdInca[pp][zz][2] = TProdInca[pp][zz][2] * IncScale
      IMProdInca[pp][zz][3] = TProdInca[pp][zz][3] * IncScale
      IMProdInca[pp][zz][4] = TProdInca[pp][zz][4] * IncScale

    ENDIF

    ;; Accumulate new motorized and non-motorized Final Internal Ps,
    unscaled) As
    SumIntMP[PP] = SumIntMP[PP] + IMProda[PP][ZZ]
    SumIntMA[PP] = SumIntMA[PP] + IMAttr[PP][ZZ]

    SumIntNMP[PP] = SumIntNMP[PP] + TMNMP[PP][ZZ]
    SumIntNMA[PP] = SumIntNMA[PP] + TMNMA[PP][ZZ]

    SumIntMNMP[PP] = SumIntMNMP[PP] + IMProda[PP][ZZ] +
    TMNMP[PP][ZZ]
    SumIntMNMA[PP] = SumIntMNMA[PP] + IMAttr[PP][ZZ] +
    TMNMA[PP][ZZ]
  ENDLOOP
ENDLOOP

;;
;; compute scaling factors for INTERNAL Attractions
;; - This scaling will be based on INTERNAL Motorized Ps and nonMotorized Trips
;; - Attractions are based on computed attractions from Trip Generation

  LOOP PP= 1,8

```



## Appendix C Cube Voyager Scripts

```

        IF ( SumTotMNMA[PP] > 0.0 ) AttScalFtr[PP] = ((SUMintMNMP[PP] +
SumExtMP[PP]) - SumExtMA[PP]) / SumTotMNMA[PP]
        print form= 10.0 list = 'Purpose: ', PP, ' SumTotMNMP[PP]:
',SumTotMNMP[PP], file = debug.txt
        print form= 10.0 list = ' ', ' SUMintMNMA[PP]:
',SumintMNMA[PP], file = debug.txt
        print form= 10.0 list = ' ', ' SumExtMP[PP]:
',SumExtMP[PP] , file = debug.txt
        print form= 10.0 list = ' ', ' SumExtMA[PP]:
',SumExtMA[PP] , file = debug.txt
        print form= 10.4 list = ' ', ' AttScalFtr[PP]
',AttScalFtr[PP], file = debug.txt

ENDLOOP

;;
;; Apply Motor & Non-Motor INTERNAL scaling factors to INTERNAL Attractions
;;
    LOOP ZZ = 1, @LastIzn@
      LOOP PP = 1, @Purps@

        ScIMAttrra[PP][ZZ] = TMAttrra[PP][ZZ] * AttScalFtr[PP]
        ScINMAttrra[PP][ZZ] = TNMAttrra[PP][ZZ] * AttScalFtr[PP]

        ScIMNMAAttrra[PP][ZZ] = ScIMAttrra[PP][ZZ] + ScINMAttrra[PP][ZZ]

        IF (PP<4)
          if (ScIMAttrra[PP][ZZ] = 0)

            IncScale = 0.0

          Else

            IncScale = ScIMAttrra[PP][ZZ]/TMAttrra[PP][ZZ]

          ENDIF

          IAttrInca[pp][zz][1] = TAttrInca[pp][zz][1] * IncScale
          IAttrInca[pp][zz][2] = TAttrInca[pp][zz][2] * IncScale
          IAttrInca[pp][zz][3] = TAttrInca[pp][zz][3] * IncScale
          IAttrInca[pp][zz][4] = TAttrInca[pp][zz][4] * IncScale

        ENDIF

        ; Accumulate Internal Ps, As:

        SumScIntMA[PP] = SumScIntMA[PP] + ScIMAttrra[PP][ZZ]
        SumScIntNMA[PP] = SumScIntNMA[PP] + ScINMAttrra[PP][ZZ]
        SumScIntMNMA[PP] = SumScIntMNMA[PP] + ScIMNMAAttrra[PP][ZZ]

      ENDLOOP
    ENDLOOP

;; Write out zonal INTERNAL Ps,As by purpose

LOOP ZZ = 1,@ZONESIZE@

  RO.TAZ = ZZ
  RO.HEWMIP = IMProda[1][ZZ]
  RO.HBWNMIP = TNMProda[1][ZZ]
  RO.HEWMMIP = IMProda[1][ZZ] + TNMProda[1][ZZ]
  RO.HEWMIPI = IProdInca[1][ZZ][1]

```

```

  RO.HEWMIP2 = IProdInca[1][ZZ][2]
  RO.HEWMIP3 = IProdInca[1][ZZ][3]
  RO.HEWMIP4 = IProdInca[1][ZZ][4]

  RO.HEWMIA = ScIMAttrra[1][ZZ]
  RO.HBWNMIA = ScINMAttrra[1][ZZ]
  RO.HBWNMIA = ScIMNMAAttrra[1][ZZ]
  RO.HEWMIA1 = IAttrInca[1][ZZ][1]
  RO.HEWMIA2 = IAttrInca[1][ZZ][2]
  RO.HEWMIA3 = IAttrInca[1][ZZ][3]
  RO.HEWMIA4 = IAttrInca[1][zz][4]

  RO.HBSMIP = IMProda[2][ZZ]
  RO.HBSNMIP = TNMProda[2][ZZ]
  RO.HBSNMNIP = IMProda[2][ZZ] + TNMProda[2][ZZ]
  RO.HBSMIP1 = IProdInca[2][ZZ][1]
  RO.HBSMIP2 = IProdInca[2][ZZ][2]
  RO.HBSMIP3 = IProdInca[2][ZZ][3]
  RO.HBSMIP4 = IProdInca[2][ZZ][4]

  RO.HBSMIA = ScIMAttrra[2][ZZ]
  RO.HBSNMIA = ScINMAttrra[2][ZZ]
  RO.HBSNMNIA = ScIMNMAAttrra[2][ZZ]
  RO.HBSMIA1 = IAttrInca[2][ZZ][1]
  RO.HBSMIA2 = IAttrInca[2][ZZ][2]
  RO.HBSMIA3 = IAttrInca[2][ZZ][3]
  RO.HBSMIA4 = IAttrInca[2][zz][4]

  RO.HBOMIP = IMProda[3][ZZ]
  RO.HBONMIP = TNMProda[3][ZZ]
  RO.HBONNMIP = IMProda[3][ZZ] + TNMProda[3][ZZ]
  RO.HBOMIP1 = IProdInca[3][ZZ][1]
  RO.HBOMIP2 = IProdInca[3][ZZ][2]
  RO.HBOMIP3 = IProdInca[3][ZZ][3]
  RO.HBOMIP4 = IProdInca[3][ZZ][4]

  RO.HBOMIA = ScIMAttrra[3][ZZ]
  RO.HBONMIA = ScINMAttrra[3][ZZ]
  RO.HBONNMIA = ScIMNMAAttrra[3][ZZ]
  RO.HBOMIA1 = IAttrInca[3][ZZ][1]
  RO.HBOMIA2 = IAttrInca[3][ZZ][2]
  RO.HBOMIA3 = IAttrInca[3][ZZ][3]
  RO.HBOMIA4 = IAttrInca[3][zz][4]

  RO.NHWMIP = IMProda[4][ZZ]
  RO.NHWNMIP = TNMProda[4][ZZ]
  RO.NHWNMIP = IMProda[4][ZZ] + TNMProda[4][ZZ]
  RO.NHWMIA = ScIMAttrra[4][ZZ]
  RO.NHWNMIA = ScINMAttrra[4][ZZ]
  RO.NHWNMIA = ScIMNMAAttrra[4][ZZ]

  RO.NHOMIP = IMProda[5][ZZ]
  RO.NHONMIP = TNMProda[5][ZZ]
  RO.NHONNMIP = IMProda[5][ZZ] + TNMProda[5][ZZ]
  RO.NHOMIA = ScIMAttrra[5][ZZ]
  RO.NHONMIA = ScINMAttrra[5][ZZ]
  RO.NHONMIA = ScIMNMAAttrra[5][ZZ]

  RO.COMIP = IMProda[6][ZZ]
  RO.COMIA = ScIMAttrra[6][ZZ]
  RO.MTKIP = IMProda[7][ZZ]
  RO.MTKIA = ScIMAttrra[7][ZZ]
  RO.HTKIP = IMProda[8][ZZ]
  RO.HTKIA = ScIMAttrra[8][ZZ]

```

```
WRITE RECO = 1
```

## Appendix C Cube Voyager Scripts

```

ENDLOOP

;; -----
;;print input P/A results by intl, external groups

print printo=1          List = ' Listing of OUTPUT P/A Totals by purpose to be
                        used in the INTERNAL Trip Distribution Process '

      print printo=1          list = '
      print printo =1        list = '          ' PRODUCTIONS SUMMARY '
      print printo =1        list = '          ' External Mtr   Internal Mtr

Internal Mtr   Internal NonMtr   Internal Total
print printo =1          list = ' Purpose ' As (I/P)          Ps (I/P)
Ps (O/P)       Ps (I/P&O/P)      Ps (O/P)
      print printo=1          list = '
      print printo= 1 form=16.2csv list = ' HBW   ' , SumExtMA[1], SumTotMP[1],
SumIntMP[1], SumIntNMP[1], SumIntMNMP[1]
      print printo= 1 form=16.2csv list = ' HBS   ' , SumExtMA[2], SumTotMP[2],
SumIntMP[2], SumIntNMP[2], SumIntMNMP[2]
      print printo= 1 form=16.2csv list = ' HBO   ' , SumExtMA[3], SumTotMP[3],
SumIntMP[3], SumIntNMP[3], SumIntMNMP[3]
      print printo= 1 form=16.2csv list = ' NHW   ' , SumExtMA[4], SumTotMP[4],
SumIntMP[4], SumIntNMP[4], SumIntMNMP[4]
      print printo= 1 form=16.2csv list = ' NHO   ' , SumExtMA[5], SumTotMP[5],
SumIntMP[5], SumIntNMP[5], SumIntMNMP[5]
      print printo= 1 form=16.2csv list = ' COM   ' , SumExtMA[6], SumTotMP[6],
SumIntMP[6], SumIntNMP[6], SumIntMNMP[6]
      print printo= 1 form=16.2csv list = ' MTK   ' , SumExtMA[7], SumTotMP[7],
SumIntMP[7], SumIntNMP[7], SumIntMNMP[7]
      print printo= 1 form=16.2csv list = ' HTK   ' , SumExtMA[8], SumTotMP[8],
SumIntMP[8], SumIntNMP[8], SumIntMNMP[8]
      print printo= 1          list = '
      print printo =1        list = '          ' ATTRACTIONS SUMMARY

      print printo =1          list = '          ' External Mtr   Internal Mtr
Intl NonMtr   Intl All
print printo =1          list = ' Purpose ' Ps (I/P)          As (I/P)
As (I/P)       As (I/P)
      print printo= 1          list = '
      print printo= 1 form=16.2csv list = ' HBW   ' , SumExtMP[1], SumTotMA[1],
SumTotNMA[1],SumTotMNMA[1]
      print printo= 1 form=16.2csv list = ' HBS   ' , SumExtMP[2], SumTotMA[2],
SumTotNMA[2],SumTotMNMA[2]
      print printo= 1 form=16.2csv list = ' HBO   ' , SumExtMP[3], SumTotMA[3],
SumTotNMA[3],SumTotMNMA[3]
      print printo= 1 form=16.2csv list = ' NHW   ' , SumExtMP[4], SumTotMA[4],
SumTotNMA[4],SumTotMNMA[4]
      print printo= 1 form=16.2csv list = ' NHO   ' , SumExtMP[5], SumTotMA[5],
SumTotNMA[5],SumTotMNMA[5]
      print printo= 1 form=16.2csv list = ' COM   ' , SumExtMP[6], SumTotMA[6],
SumTotNMA[6],SumTotMNMA[6]
      print printo= 1 form=16.2csv list = ' MTK   ' , SumExtMP[7], SumTotMA[7],
SumTotNMA[7],SumTotMNMA[7]
      print printo= 1 form=16.2csv list = ' HTK   ' , SumExtMP[8], SumTotMA[8],
SumTotNMA[8],SumTotMNMA[8]
      print printo= 1          list = '
      print printo =1        list = '          ' Scaled Intl   Scaled Intl
Scaled Intl   Scaling
      print printo =1          list = ' Purpose ' Mtr As          Non-Mtr As
ALL As Factor
      print printo= 1          list = '
      print printo= 1 form=16.2csv list = ' HBW   ' , SumScIntMA[1], SumScIntNMA[1],
SumScIntMNMA[1], AttScalFtr[1]
      print printo= 1 form=16.2csv list = ' HBS   ' , SumScIntMA[2], SumScIntNMA[2],
SumScIntMNMA[2], AttScalFtr[2]

```

```

      print printo= 1 form=16.2csv list = ' HBO   ' , SumScIntMA[3], SumScIntNMA[3],
SumScIntMNMA[3], AttScalFtr[3]
      print printo= 1 form=16.2csv list = ' NHW   ' , SumScIntMA[4], SumScIntNMA[4],
SumScIntMNMA[4], AttScalFtr[4]
      print printo= 1 form=16.2csv list = ' NHO   ' , SumScIntMA[5], SumScIntNMA[5],
SumScIntMNMA[5], AttScalFtr[5]
      print printo= 1 form=16.2csv list = ' COM   ' , SumScIntMA[6], SumScIntNMA[6],
SumScIntMNMA[6], AttScalFtr[6]
      print printo= 1 form=16.2csv list = ' MTK   ' , SumScIntMA[7], SumScIntNMA[7],
SumScIntMNMA[7], AttScalFtr[7]
      print printo= 1 form=16.2csv list = ' HTK   ' , SumScIntMA[8], SumScIntNMA[8],
SumScIntMNMA[8], AttScalFtr[8]
ENDRUN
*copy voya*.prn mod2.rpt

```

## 35 Prepare\_Trip\_Tables\_for\_Assignment.s

```

; Prepare_Trip_tables_for_Assignment.s
;4/3/12 three decimals maintained in output files ("dec=6*3" added to MATO
statement)
;-----
; Step 1 - Modeled & Non-Modeled Trip Table Consolidation
; for the Version 2.3 Highway Assignment
;
; - 4 Trip files built for AM, Midday, PM, Off-Peak Time Periods
; - Each file has 6 Trip tables:
; 1) 1-occ adrs
; 2) 2-occ adrs
; 3) 3+occ adrs
; 4) Commercial Vehicle
; 5) Trucks (Medium and Heavy)
; 6) Airport Pax Adrs
;-----
; I/P Auto Dr. Pct. tables:
ADRAM = '%_iter_%_am_adr.mat'
ADRMD = '%_iter_%_md_adr.mat'
ADRPM = '%_iter_%_pm_adr.mat'
ADRNT = '%_iter_%_nt_adr.mat'
;
; I/P MISC Auto Dr. Tables:
MISCAM = '%_iter_%_am_misc.tt'
MISCMD = '%_iter_%_md_misc.tt'
MISCPM = '%_iter_%_pm_misc.tt'
MISCNT = '%_iter_%_nt_misc.tt'
;
; O/P Vehicle Trips:
AM_VT = '%_iter_%_AM.VTT'
MD_VT = '%_iter_%_MD.VTT'
PM_VT = '%_iter_%_PM.VTT'
NT_VT = '%_iter_%_NT.VTT'
;
; avg xx auto occ. is 1.72 basis for: //
XXAD1OCC = 0.5021 ; ASSUMED SHARE OF THRU ADRS Which are 1-OCC vehs.
XXAD2OCC = 0.3426 ; ASSUMED SHARE OF THRU ADRS Which are 2-OCC vehs.
XXAD3OCC = 0.1553 ; ASSUMED SHARE OF THRU ADRS Which are 3+OCC vehs.
;//////////////////////////////////////
RUN PGM=MATRIX
;; Input files:
;; Auto Driver trips by time period
;; each file contains 3 tables (1-occ, 2-occ., and 3+occ auto driver trips)
MATI[1]=@ADRAM@ ;; AM Modeled Auto Drivers
MATI[2]=@ADRMD@

```

## Appendix C Cube Voyager Scripts

```

MATI[3]=@ADRFPM@
MATI[4]=@ADRNT@

;; Miscellaneous Trips by time period
;;
;; Each file contains 8 tables -
;; 1/xx truck,2/xx autodr,3/taxi adr,4/visitor-tourist adr,
;; 5/med.truck, 6/hvy truck, 7/air passenger adr, 8/comm veh.
MATI[5]=@MISCAM@
MATI[6]=@MISCMD@
MATI[7]=@MISCPM@
MATI[8]=@MISCNT@

;AM Modeled Auto Drivers:
MW[101]= MI.1.1 ; 1-Occ adrs
MW[102]= MI.1.2 ; 2-Occ adrs
MW[103]= MI.1.3 ; 3+Occ adrs

;MD Modeled Auto Drivers:
MW[201]= MI.2.1 ; 1-Occ adrs
MW[202]= MI.2.2 ; 2-Occ adrs
MW[203]= MI.2.3 ; 3+Occ adrs

;PM Modeled Auto Drivers:
MW[301]= MI.3.1 ; 1-Occ adrs
MW[302]= MI.3.2 ; 2-Occ adrs
MW[303]= MI.3.3 ; 3+Occ adrs

;OP Modeled Auto Drivers:
MW[401]= MI.4.1 ; 1-Occ adrs
MW[402]= MI.4.2 ; 2-Occ adrs
MW[403]= MI.4.3 ; 3+Occ adrs

;
; AM Peak Period MISC Trips
MW[111] = MI.5.1 ; Thru Truck
MW[112] = MI.5.2*@XXAD1OCC@ ; Thru Auto Driver-1 OCC
MW[113] = MI.5.2*@XXAD2OCC@ ; Thru Auto Driver-2 OCC
MW[114] = MI.5.2*@XXAD3OCC@ ; Thru Auto Driver-3+OCC
MW[115] = MI.5.3 ; Taxi Auto Driver
MW[116] = MI.5.4 ; Visitor Auto Driver
MW[117] = MI.5.6 ; I-I,I-E,E-I Medium Truck
MW[118] = MI.5.7 ; I-I,I-E,E-I Heavy Truck
MW[119] = MI.5.8 ; Air Pax Auto Driver
MW[120] = MI.5.9 ; I-I,I-E,E-I Comm. Veh
MW[121] = MI.5.5 ; School Auto Driver

;
; MD Peak Period MISC Trips
MW[211] = MI.6.1 ; Thru Truck
MW[212] = MI.6.2*@XXAD1OCC@ ; Thru Auto Driver-1 OCC
MW[213] = MI.6.2*@XXAD2OCC@ ; Thru Auto Driver-2 OCC
MW[214] = MI.6.2*@XXAD3OCC@ ; Thru Auto Driver-3+OCC
MW[215] = MI.6.3 ; Taxi Auto Driver
MW[216] = MI.6.4 ; Visitor Auto Driver
MW[217] = MI.6.6 ; I-I,I-E,E-I Medium Truck
MW[218] = MI.6.7 ; I-I,I-E,E-I Heavy Truck
MW[219] = MI.6.8 ; Air Pax Auto Driver
MW[220] = MI.6.9 ; I-I,I-E,E-I Comm. Veh
MW[221] = MI.6.5 ; School Auto Driver

;
; PM Peak Period MISC Trips
MW[311] = MI.7.1 ; Thru Truck
MW[312] = MI.7.2*@XXAD1OCC@ ; Thru Auto Driver-1 OCC
MW[313] = MI.7.2*@XXAD2OCC@ ; Thru Auto Driver-2 OCC
MW[314] = MI.7.2*@XXAD3OCC@ ; Thru Auto Driver-3+OCC
MW[315] = MI.7.3 ; Taxi Auto Driver
MW[316] = MI.7.4 ; Visitor Auto Driver
MW[317] = MI.7.6 ; I-I,I-E,E-I Medium Truck
MW[318] = MI.7.7 ; I-I,I-E,E-I Heavy Truck
MW[319] = MI.7.8 ; Air Pax Auto Driver

```

```

MW[320] = MI.7.9 ; I-I,I-E,E-I Comm. Veh
MW[321] = MI.7.5 ; School Auto Driver

```

```

;
; OP Peak Period MISC Trips
MW[411] = MI.8.1 ; Thru Truck
MW[412] = MI.8.2*@XXAD1OCC@ ; Thru Auto Driver-1 OCC
MW[413] = MI.8.2*@XXAD2OCC@ ; Thru Auto Driver-2 OCC
MW[414] = MI.8.2*@XXAD3OCC@ ; Thru Auto Driver-3+OCC
MW[415] = MI.8.3 ; Taxi Auto Driver
MW[416] = MI.8.4 ; Visitor Auto Driver
MW[417] = MI.8.6 ; I-I,I-E,E-I Medium Truck
MW[418] = MI.8.7 ; I-I,I-E,E-I Heavy Truck
MW[419] = MI.8.8 ; Air Pax Auto Driver
MW[420] = MI.8.9 ; I-I,I-E,E-I Comm. Veh
MW[421] = MI.8.5 ; School Auto Driver

```

; Add up vehicle tables into the appropriate TOD Categories

```

; AM
MW[151] = MW[101] + MW[112] + MW[121] ; SOV Vehicle Trips
MW[152] = MW[102] + MW[113] + MW[115] + MW[116] ; HOV2 Vehicle Trips
MW[153] = MW[103] + MW[114] ; HOV3+ Vehicle Trips
MW[154] = MW[120] ; Comm. Vehs
MW[155] = MW[111] + MW[117] + MW[118] ; Med/Hvy Truck Trips
MW[156] = MW[119] ; Airport Pax Adr Trips

```

```

; MD
MW[251] = MW[201] + MW[212] + MW[221] ; SOV Vehicle Trips
MW[252] = MW[202] + MW[213] + MW[215] + MW[216] ; HOV2 Vehicle Trips
MW[253] = MW[203] + MW[214] ; HOV3+ Vehicle Trips
MW[254] = MW[220] ; Comm. Vehs
MW[255] = MW[211] + MW[217] + MW[218] ; Med/Hvy Truck Trips
MW[256] = MW[219] ; Airport Pax Adr Trips

```

```

; PM
MW[351] = MW[301] + MW[312] + MW[321] ; SOV Vehicle Trips
MW[352] = MW[302] + MW[313] + MW[315] + MW[316] ; HOV2 Vehicle Trips
MW[353] = MW[303] + MW[314] ; HOV3+ Vehicle Trips
MW[354] = MW[320] ; Comm. Vehs
MW[355] = MW[311] + MW[317] + MW[318] ; Med/Hvy Truck Trips
MW[356] = MW[319] ; Airport Pax Adr Trips

```

```

; OP
MW[451] = MW[401] + MW[412] + MW[421] ; SOV Vehicle Trips
MW[452] = MW[402] + MW[413] + MW[415] + MW[416] ; HOV2 Vehicle Trips
MW[453] = MW[403] + MW[414] ; HOV3+ Vehicle Trips
MW[454] = MW[420] ; Comm. Vehs
MW[455] = MW[411] + MW[417] + MW[418] ; Med/Hvy Truck Trips
MW[456] = MW[419] ; Airport Pax Adr Trips

```

```

;
; Now let's accumulate totals for neat regional summaries
jloop
vehs = vehs + (MW[151]+MW[152]+MW[153]+MW[154]+MW[155]+MW[156]) + ;
(MW[251]+MW[252]+MW[253]+MW[254]+MW[255]+MW[256]) + ;
(MW[351]+MW[352]+MW[353]+MW[354]+MW[355]+MW[356]) + ;
(MW[451]+MW[452]+MW[453]+MW[454]+MW[455]+MW[456]) ; daily

```

```

comveh = comveh + mw[120] + mw[220] + mw[320] + mw[420] ; daily CVs

```

```

;AM group
amvehs = amvehs + (MW[151]+MW[152]+MW[153]+MW[154]+MW[155]+MW[156]) ; all am
vehs

```

## Appendix C Cube Voyager Scripts

```

am1occ = am1occ + MW[151] ; am modeled 1-occveh's
am2occ = am2occ + MW[152] ; am modeled 2-occveh's
am3occ = am3occ + MW[153] ; am modeled 3+occveh's
amtrks = amtrks + MW[155] ; am trucks
amapax = amapax + MW[156] ; am airpax adrs
am1occad = am1occad + MW[101] ; am locc adr
am2occad = am2occad + MW[102] ; am 2occ adr
am3occad = am3occad + MW[103] ; am 3+occ adr
amadr = amadr + MW[101] + MW[102] + MW[103] ; am total adr(modeled)
amxxtrk = amxxtrk + MW[111] ; am Thru Truck
amxxad1 = amxxad1 + MW[112] ; am Thru locc Adr
amxxad2 = amxxad2 + MW[113] ; am Thru 2occ Adr
amxxad3 = amxxad3 + MW[114] ; am Thru 3+occAdr
amxxadr = amxxadr + MW[112]+MW[113]+MW[114] ; am total xx adr
amtaxi = amtaxi + MW[115] ; am Taxi ADR
amvisi = amvisi + MW[116] ; am visitor ADR
amschl = amschl + MW[121] ; am School ADR
amtrk = amtrk + MW[117] ; am int,ext MedTk
amhtrk = amhtrk + MW[118] ; am int,ext HvyTk
amairpax = amairpax + MW[119] ; am air pax auto dr
amcomveh = amcomveh + MW[120] ; am int,ext,ComVeh

;MD group
mdvchs = mdvchs + (MW[251]+MW[252]+MW[253]+MW[254]+MW[255]+MW[256]) ; all md
veh
md1occ = md1occ + MW[251] ; md modeled 1-occveh's
md2occ = md2occ + MW[252] ; md modeled 2-occveh's
md3occ = md3occ + MW[253] ; md modeled 3+occveh's
mdtrks = mdtrks + MW[255] ; md trucks
mdapax = mdapax + MW[256] ; md airpax adrs
md1occad = md1occad + MW[201] ; md locc adr
md2occad = md2occad + MW[202] ; md 2occ adr
md3occad = md3occad + MW[203] ; md 3+occ adr
mdadr = mdadr + MW[201] + MW[202] + MW[203] ; md total adr(modeled)
mdxxtrk = mdxxtrk + MW[211] ; md Thru Truck
mdxxad1 = mdxxad1 + MW[212] ; md Thru locc Adr
mdxxad2 = mdxxad2 + MW[213] ; md Thru 2occ Adr
mdxxad3 = mdxxad3 + MW[214] ; md Thru 3+occAdr
mdxxadr = mdxxadr + MW[212] + MW[213] + MW[214] ; md total xx adr
mdtaxi = mdtaxi + MW[215] ; md Taxi ADR
mdvisi = mdvisi + MW[216] ; md visitor ADR
mdSchl = mdSchl + MW[221] ; md School ADR
mdmtrk = mdmtrk + MW[217] ; md int,ext MedTk
mdhtrk = mdhtrk + MW[218] ; md int,ext HvyTk
mdairpax = mdairpax + MW[219] ; md air pax auto dr
mdcomveh = mdcomveh + MW[220] ; md int,ext,ComVeh

;PM group
pmvchs = pmvchs + (MW[351]+MW[352]+MW[353]+MW[354]+MW[355]+MW[356]) ; all pm
veh
pm1occ = pm1occ + MW[351] ; pm modeled 1-occveh's
pm2occ = pm2occ + MW[352] ; pm modeled 2-occveh's
pm3occ = pm3occ + MW[353] ; pm modeled 3+occveh's
pmtrks = pmtrks + MW[355] ; pm trucks
pmapax = pmapax + MW[356] ; pm airpax adrs
pm1occad = pm1occad + MW[301] ; pm locc adr
pm2occad = pm2occad + MW[302] ; pm 2occ adr
pm3occad = pm3occad + MW[303] ; pm 3+occ adr
pmadr = pmadr + MW[301] + MW[302] + MW[303] ; pm total adr(modeled)
pmxxtrk = pmxxtrk + MW[311] ; pm Thru Truck
pmxxad1 = pmxxad1 + MW[312] ; pm Thru locc Adr
pmxxad2 = pmxxad2 + MW[313] ; pm Thru 2occ Adr
pmxxad3 = pmxxad3 + MW[314] ; pm Thru 3+occAdr
pmxxadr = pmxxadr + MW[312] + MW[313] + MW[314] ; pm total xx adr
pmtaxi = pmtaxi + MW[315] ; pm Taxi ADR
pmvisi = pmvisi + MW[316] ; pm visitor ADR

```

```

pmschl = pmschl + MW[321] ; pm school ADR
pmmtrk = pmmtrk + MW[317] ; pm int,ext MedTk
pmhtrk = pmhtrk + MW[318] ; pm int,ext HvyTk
pmairpax = pmairpax + MW[319] ; pm air pax auto dr
pmcomveh = pmcomveh + MW[320] ; pm int,ext,ComVeh

;OP group
opvchs = opvchs + (MW[451]+MW[452]+MW[453]+MW[454]+MW[455]+MW[456]) ; all
op/nt vehs
op1occ = op1occ + MW[451] ; op/nt modeled 1-occveh's
op2occ = op2occ + MW[452] ; op/nt modeled 2-occveh's
op3occ = op3occ + MW[453] ; op/nt modeled 3+occveh's
optrks = optrks + MW[455] ; op/nt trucks
opapax = opapax + MW[456] ; op/nt airpax adrs
op1occad = op1occad + MW[401] ; op/nt locc adr
op2occad = op2occad + MW[402] ; op/nt 2occ adr
op3occad = op3occad + MW[403] ; op/nt 3+occ adr
opadr = opadr + MW[401] + MW[402] + MW[403] ; op/nt total adr(modeled)
opxxtrk = opxxtrk + MW[411] ; op/nt Thru Truck
opxxad1 = opxxad1 + MW[412] ; op/nt Thru locc Adr
opxxad2 = opxxad2 + MW[413] ; op/nt Thru 2occ Adr
opxxad3 = opxxad3 + MW[414] ; op/nt Thru 3+occAdr
opxxadr = opxxadr + MW[412] + MW[413] + MW[414] ; op/nt total xx adr
optaxi = optaxi + MW[415] ; op/nt Taxi ADR
opvisi = opvisi + MW[416] ; op/nt visitor ADR
opschl = opschl + MW[421] ; op/nt school ADR
opmtrk = opmtrk + MW[417] ; op/nt int,ext MedTk
ophtrk = ophtrk + MW[418] ; op/nt int,ext HvyTk
opairpax = opairpax + MW[419] ; op/nt air pax auto dr
opcomveh = opcomveh + MW[420] ; op/nt int,ext,ComVeh

; Sum up output trip table totals
; AM
AMSOVs = AMSOVs + MW[151]
AMHOV2s = AMHOV2s + MW[152]
AMHOV3s = AMHOV3s + MW[153]
AMComVehs = AMComVehs + MW[154]
AMTrucks = AMTrucks + MW[155]
AMAirPaxs = AMAirPaxs + MW[156]

; MD
MDSOVs = MDSOVs + MW[251]
MDHOV2s = MDHOV2s + MW[252]
MDHOV3s = MDHOV3s + MW[253]
MDCOMVehs = MDCOMVehs + MW[254]
MDTrucks = MDTrucks + MW[255]
MDAirPaxs = MDAirPaxs + MW[256]

; PM
PMSOVs = PMSOVs + MW[351]
PMHOV2s = PMHOV2s + MW[352]
PMHOV3s = PMHOV3s + MW[353]
PMComVehs = PMComVehs + MW[354]
PMTrucks = PMTrucks + MW[355]
PMAirPaxs = PMAirPaxs + MW[356]

; OP
OPSOVs = OPSOVs + MW[451]
OPHOV2s = OPHOV2s + MW[452]
OPHOV3s = OPHOV3s + MW[453]
OPComVehs = OPComVehs + MW[454]
OPTrucks = OPTrucks + MW[455]
OPAirPaxs = OPAirPaxs + MW[456]
endjloop

if (i=zones) ; print out results
Print list = '/bt '

```



## Appendix C Cube Voyager Scripts

```
Print form= 12.0csv list = ' pm school Adr ',pmschl
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' pm int,ext MedTk ',pmmtrk
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' pm int,ext HvyTk ',pmhtrk
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' pm air pax auto dr ',pmairpax
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' pm int,ext,ComVeh ',pmcomveh
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' all pm vehs ',pmvehs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
;OP group
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = 'Off-Peak Totals: '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt modeled 1-occvehs ',op1occ
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt modeled 2-occvehs ',op2occ
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt modeled 3+occvehs ',op3occ
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt trucks ',optrks
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt locc adr ',op1occad
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt 2occ adr ',op2occad
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt 3+occ adr ',op3occad
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt total adr(modeled) ',opadr
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt Thru Truck ',opxxtrk
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt Thru locc Adr ',opxxadr1
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt Thru 2occ Adr ',opxxadr2
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt Thru 3+occAdr ',opxxadr3
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt total xx adr ',opxxadr
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt Taxi Adr ',optaxi
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt visitOr Adr ',opvisi
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt school Adr ',opschl
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt int,ext MedTk ',opmtrk
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt int,ext HvyTk ',ophtrk
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt air pax auto dr ',opairpax
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' nt int,ext,ComVeh ',opcomveh
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' all nt vehs ',opvehs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' SUM OF ALL VEHICLES: ',vehs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = '%_iter_% Trip Table Output Totals: '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
; AM
Print form= 12.0csv list = ' AMSOVs ',AMSOVs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' AMHOV2s ',AMHOV2s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' AMHOV3s ',AMHOV3s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' AMComVehs
',AMComVehs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' AMTrucks ',AMTrucks
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' AMAirPaxs
',AMAirPaxs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
; MD
Print form= 12.0csv list = ' MDSOVs ',MDSOVs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' MDHOV2s ',MDHOV2s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' MDHOV3s ',MDHOV3s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' MDComVehs
',MDComVehs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' MDTrucks ',MDTrucks
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' MDAirPaxs
',MDAirPaxs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
; PM
Print form= 12.0csv list = ' PMSOVs ',PMSOVs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' PMHOV2s ',PMHOV2s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' PMHOV3s ',PMHOV3s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' PMComVehs
',PMComVehs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' PMTrucks ',PMTrucks
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' PMAirPaxs
',PMAirPaxs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

```
; OP
Print form= 12.0csv list = ' NTSOVs ',OPSOVs
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' NTHOV2s ',OPHOV2s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' NTHOV3s ',OPHOV3s
,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' NTComVehs
',OPComVehs,file=%_iter_% Prepare_Trip_Tables_For_Assignment.txt
```

## Appendix C Cube Voyager Scripts

```

Print form= 12.0csv list = ' NTTrucks          ',OPTrucks
,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' NTAirPaxs
',OPAirPaxs,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print form= 12.0csv list = ' SUM OF ALL VEHICLES: ',vehs
,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print list = ' '
,file=%_iter_%_Prepare_Trip_Tables_For_Assignment.txt
Print list = '/et
endif

; Write out the auto driver tables by time period
MATO[1] = @AM_VT@, MO=151-156, ; AM Veh Trips
1,2,3+occ, comveh, trucks,Air Pax Vehs
name=AM_SOVs,AM_HV2s,AM_HV3s,AM_COMs,AM_TRKs,AM_APVs, dec=6*3

MATO[2] = @MD_VT@, MO=251-256, ; MD Veh Trips
1,2,3+occ, comveh, trucks,Air Pax Vehs
name=MD_SOVs,MD_HV2s,MD_HV3s,MD_COMs,MD_TRKs,MD_APVs, dec=6*3

MATO[3] = @PM_VT@, MO=351-356, ; PM Veh Trips
1,2,3+occ, comveh, trucks,Air Pax Vehs
name=PM_SOVs,PM_HV2s,PM_HV3s,PM_COMs,PM_TRKs,PM_APVs, dec=6*3

MATO[4] = @NT_VT@, MO=451-456, ; NT Veh Trips
1,2,3+occ, comveh, trucks,Air Pax Vehs
name=NT_SOVs,NT_HV2s,NT_HV3s,NT_COMs,NT_TRKs,NT_APVs, dec=6*3

ENDRUN

```

## 36 Refine\_Station\_File.s

```

*del voya*.prn
;; Refine_Station_File.s - program to read standard V2.3 station file with odd
formats, column widths
;;
and writes the SAME data with a neater appearance

Station_Input = 'Station_2040_FromMary_4_13_11.DBF' ; input file of this
script
Station_Output = 'Station_2040_Final_4_13_13.DBF' ; output file of this
script

RUN PGM=MATRIX
ZONES=1

FILEI DBI[1] = "@Station_Input@"

RECO[1] = "@Station_Output@", form=10.0,
Fields = SEQNO, MM(C8), NCT, STAPARK, STAUSE, SNAME(c27), STAC, STAZ,
STAT, STAP,
STAN1, STAN2, STAN3, STAN4, STAPCAP, STAX, STAY,
STAPKCost(13), STAOPCost(13), STAPKShad(13), STAOPShad(13), FirstYr,
Sta_CenD

cnt = 0

```

```

; All Station Nodes:
LOOP L= 1,dbi.1.NUMRECORDS
x=DBIReadRecord(1,L)
cnt=cnt+1
ro.segno =di.1.segno
RO.MM =di.1.MM
RO.NCT =di.1.NCT
RO.stapark =di.1.STAPARK
RO.stause =di.1.stause
RO.Sname =di.1.Sname
RO.STAC =di.1.STAC
RO.STAZ =di.1.STAZ
RO.STAT =di.1.STAT
RO.STAP =di.1.STAP
RO.STAN1 =di.1.STAN1
RO.STAN2 =di.1.STAN2
RO.STAN3 =di.1.STAN3
RO.STAN4 =di.1.STAN4
RO.STAPCAP =di.1.STAPCAP
RO.STAX =di.1.STAX
RO.STAY =di.1.STAY
RO.STAPKCost =di.1.STAPKCost
RO.STAOPCost =di.1.STAOPCost
RO.STAPKShad =di.1.STAPKShad
RO.STAOPShad =di.1.STAOPShad
RO.FirstYr =di.1.FirstYr
RO.STA_CenD =di.1.Sta_CenD

write reco=1

ENDDLOOP

ENDRUN

```

## 37 Remove\_PP\_Speed.s

```

;; Remove initial 'lookup' speeds on highway links defined during hwy network
building
;; The speeds will be replaced by restrained speeds created in the 'pump prime'
assignment
;;
*copy zonehwy.net zonehwy.tem
*del zonehwy.net
RUN PGM=NETWORK
NETI = ZONEHWY.tem
NETO = zonehwy.net, exclude= PPAMSPD, PPPMSPD, PPMDSPD, PPNTSPD, PPOPSPD
ENDRUN

```

## 38 Set\_CPI.S

```

;-----
; SET_CPI.S Version 2.3 Model
; Used to define Transit and Highway Deflators consistently
;
;-----
CPI_File = 'INPUTS\CPI_File.TXT' ; Input parameters from the \INPUTS
subdir.

```

## Appendix C Cube Voyager Scripts

```

ModeledYear      = '%_year_%'              ; Simulation Year (Defined in
runall_ModelSteps_<year>.bat file)
;; Two one-line files are produced:
;;          TRN_Deflator.txt   (Transit Deflation Factor)
;;          HWY_Deflator.txt   (Highway Deflation Factor)
;-----
CPI_Rept        = 'MFARE2_CPI.TXT'          ; Output Reporting file
pageheight=32767 ; Preclude header breaks
;
RUN PGM=MATRIX
ZONES=1
READ file=@CPI_File@

IF (Defl_Override != 0 )                   ; if explicit deflation factor is provided by
user
    DEFLATIONFTR = Defl_Override           ; then use it, otherwise compute it using the
most recent CPI table
    Print List="Deflation Factor is based on Override (Defl_Override) in the
CPI_File.txt file: ', DEFLATIONFTR(8.5), file=@CPI_rept@
        ELSE                               ;
;
;;
;; Now establish the Deflation factor depending on the modeled year
;; and available historic US BLS data
;;
    _BseCPI      = CPI_Table(1,BaseCPIYear)
    _CurCPI     = CPI_Table(1,CurrCPIYear)
    _CurCPIdefl = CPI_Table(1,BaseCPIYear) / CPI_Table(1,CurrCPIYear)

    IF (@ModeledYear@ < BaseCPIYear)       ; Deflation ftr can't be
developed if yr < 2007
        LIST = 'Modeled Year is earlier than Base Year in CPI Lookup; I Quit'
        ABORT

    ELSEIF (@ModeledYear@ = BaseCPIYear)    ; If Modeled year is Base CPI
year
        _BseGrowthRate = CPI_Table(2,@ModeledYear@) ; then use the defaltion
rate from table
        _AltGrowthRate = CPI_Table(2,@ModeledYear@)

        _FutBseCPI = CPI_Table(1,@ModeledYear@)
        _FutAltCPI = CPI_Table(1,@ModeledYear@)

        DEFLATIONFTR = CPI_Table(3,@ModeledYear@)

    ELSEIF (@ModeledYear@ > BaseCPIYear && @ModeledYear@ <= CurrCPIYear) ; If
Modeled year is Base CPI year
        _BseGrowthRate = CPI_Table(2,@ModeledYear@) ;
then use the defaltion rate from table
        _AltGrowthRate = CPI_Table(2,@ModeledYear@) * INFLATIONFTR

        _FutBseCPI = CPI_Table(1,BaseCPIYear) * ((1.0 +
_BseGrowthRate)^(@ModeledYear@ - BaseCPIYear))
        _FutAltCPI = CPI_Table(1,BaseCPIYear) * ((1.0 +
_AltGrowthRate)^(@ModeledYear@ - BaseCPIYear))

        DEFLATIONFTR = (_FutAltCPI / _FutBseCPI) * CPI_Table(3,@ModeledYear@)

    ELSE
        _BseGrowthRate = ( CPI_Table(1,CurrCPIYear) / CPI_Table(1,BaseCPIYear)) ^
(1.0 / (CurrCPIYear - BaseCPIYear)) - 1.0
        _AltGrowthRate = _BseGrowthRate * INFLATIONFTR

        _FutBseCPI = CPI_Table(1,BaseCPIYear) * ((1.0 +
_BseGrowthRate)^(@ModeledYear@ - BaseCPIYear))

```

```

    _FutAltCPI = CPI_Table(1,BaseCPIYear) * ((1.0 +
_AltGrowthRate)^(@ModeledYear@ - BaseCPIYear))

    DEFLATIONFTR = (_FutAltCPI / _FutBseCPI) * CPI_Table(3,CurrCPIYear)
ENDIF

; print out small text file containing deflation factor derivation:
Print List="Modeled Year: ',
@ModeledYear@(8.0), '\n',
        file=@CPI_rept@
Print List="Base Year & CPI: ',
BaseCPIYear(8.0), _BseCPI(8.1), '\n',
        file=@CPI_rept@
Print List="Current Year & CPI & deflator (Base CPI/Curr CPI): ',
CurrCPIYear(8.0), _CurCPI(8.1), _CurCPIdefl(8.5), '\n', file=@CPI_rept@
Print List="Inflation Factor Assumption (1.00 = direct CPI): ',
INFLATIONFTR(8.5), '\n',
        file=@CPI_rept@
Print List="Modeled Year Growth rate & CPI w/ Full CPI: (A)',
_BseGrowthRate(8.5), _FutBseCPI(8.1), '(forecasts years only)', '\n', file=@CPI_rept@
Print List="Modeled Year growth rate & CPI w/ Infla. Factor: (B)',
_AltGrowthRate(8.5), _FutAltCPI(8.1), '(forecasts years only)', '\n', file=@CPI_rept@
Print List="Deflation Factor ((B)/(A)) * Current Deflator: ',
DEFLATIONFTR(8.5),
        file=@CPI_rept@

ENDIF

Print List = 'DEFLATIONFTR = ', DEFLATIONFTR(8.5), ' ; ; Transit Deflation
Factor ', File = TRN_Deflator.txt
Print List = 'DEFLATIONFTR = ', DEFLATIONFTR(8.5), ' ; ; Highway Deflation
Factor ', File = HWY_Deflator.txt

```

```

ENDRUN

```

```

;=====
; End of CPI/Deflation section
;=====

```

## 39 Set\_Factors.s

```

;-----
; SET_FACTORS.S Version 2.3 Model-3722 TAZ system
; Updated 4-14-15 RJM: added step to create Station name file from
; scenario specific station file
;-----
; MWCOCG Version 2.3 Model
; Set up K-factor files used in Trip Distribution
;
; Zonal K-factor Files created by this script

HBSK = 'hbs_k.mat' ; zonal K-factor matrix
HBSK = 'hbs_k.mat' ; zonal K-factor matrix
HBOK = 'hbo_k.mat' ; zonal K-factor matrix
NHWK = 'nhw_k.mat' ; zonal K-factor matrix
NHOK = 'nho_k.mat' ; zonal K-factor matrix

;-----
;
; //////////////////////////////////////////////////
; //////////////////////////////////////////////////
; ////////////////////////////////////////////////// 5) Begin K-Factor building, by trip purpose. \
; ////////////////////////////////////////////////// K-Factors values below are scaled by 1000. \
; ////////////////////////////////////////////////// (i.e., a value of 1000 below means K-Ftr of 1) MW[100] = 1
; //////////////////////////////////////////////////

```



## Appendix C Cube Voyager Scripts

```

; /////////////// The will be applied across income strata in trip  \\
; /////////////// distribution.                                     \\
; /////////////// ////////////////////////////////////////////////////

;; 2/27/13 All bridge-related K-factors are REMOVED

RUN PGM=MATRIX
ZONES=3722
; Now Begin the K-Factor Establishment
; Initialize K-factor matrices for each purpose:

MW[1] = 1000.0 ; HBW      K-factor matrix
MW[2] = 1000.0 ; HBS      K-factor matrix
MW[3] = 1000.0 ; HBO      K-factor matrix
MW[4] = 1000.0 ; NHW      K-factor matrix
MW[5] = 1000.0 ; NHO      K-factor matrix

;;; /* ***** Bridge penalty section ***** */
;;; ;-----
;;; ;Define K-Factor production areas in mtx 100
;;; ;1/DC&Mtg&PG, 2/Suburban VA, 3/OuterMD, 4/OuterVA, 5/Ext1
;;;
;;; IF (I= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ) MW[100] = 1 ; 0
DC Core
;;; IF (I= 5,48,51,64,66-180,210-281,288-373,382-393 ) MW[100] = 1 ; 0
DC Noncore
;;; IF (I= 394-769 ) MW[100] = 1 ; 1
Montgomery
;;; IF (I= 771-776,778-1404 ) MW[100] = 1 ; 2
Prince George
;;;
;;; IF (I=1471-1476, 1486-1489, 1495-1497 ) MW[100] = 2 ; 3
ArlCore
;;; IF (I=1405-1470,1477-1485,1490-1494,1498-1545 ) MW[100] = 2 ; 3
ArlNCore
;;; IF (I=1546-1610 ) MW[100] = 2 ; 4
Alex
;;; IF (I=1611-2159 ) MW[100] = 2 ; 5
FFx
;;; IF (I=2160-2441 ) MW[100] = 2 ; 6
LDn
;;; IF (I=2442-2554,2556-2628,2630-2819 ) MW[100] = 2 ; 7
PW
;;;
;;; IF (I=2820-2949 ) MW[100] = 3 ; 9
Frd
;;; IF (I=3230-3265,3268-3287 ) MW[100] = 3 ; 14
Car.
;;; IF (I=2950-3017 ) MW[100] = 3 ; 10
How.
;;; IF (I=3018-3102,3104-3116 ) MW[100] = 3 ; 11
AnnAr
;;; IF (I=3288-3334 ) MW[100] = 3 ; 15
Calv
;;; IF (I=3335-3409 ) MW[100] = 3 ; 16
StM
;;; IF (I=3117-3229 ) MW[100] = 3 ; 12
Chs.
;;;
;;; IF (I=3604-3653 ) MW[100] = 4 ; 21
Fau
;;; IF (I=3449-3477,3479-3481,3483-3494,3496-3541 ) MW[100] = 4 ; 19
Stf.
;;; IF (I=3654-3662,3663-3675 ) MW[100] = 4 ; 22/23
Clk,Jeff.
;;; IF (I=3435-3448,3542-3543,3545-3603 ) MW[100] = 4 ; 18/20
Fbg,Spots

```

```

;;; IF (I=3410-3434 ) MW[100] = 4 ; 17
KG.
;;;
;;; IF (I=3676-3722 ) MW[100] = 5 ;
Externals
;;; ;-----
;;;
;;; ;-----
;;; ;Define K-Factor attraction areas in mtx 200
;;; ;1/DC&Mtg&PG, 2/Suburban VA, 3/OuterMD, 4/OuterVA, 5/Ext1
;;; JLOOP
;;; IF (J= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ) MW[200] = 1 ; 0
DC Core
;;; IF (J= 5,48,51,64,66-180,210-281,288-373,382-393 ) MW[200] = 1 ; 0
DC Noncore
;;; IF (J= 394-769 ) MW[200] = 1 ; 1
Montgomery
;;; IF (J= 771-776,778-1404 ) MW[200] = 1 ; 2
Prince George
;;;
;;; IF (J=1471-1476, 1486-1489, 1495-1497 ) MW[200] = 2 ; 3
ArlCore
;;; IF (J=1405-1470,1477-1485,1490-1494,1498-1545 ) MW[200] = 2 ; 3
ArlNCore
;;; IF (J=1546-1610 ) MW[200] = 2 ; 4
Alex
;;; IF (J=1611-2159 ) MW[200] = 2 ; 5
FFx
;;; IF (J=2160-2441 ) MW[200] = 2 ; 6
LDn
;;; IF (J=2442-2554,2556-2628,2630-2819 ) MW[200] = 2 ; 7
PW
;;;
;;; IF (J=2820-2949 ) MW[200] = 3 ; 9
Frd
;;; IF (J=3230-3265,3268-3287 ) MW[200] = 3 ; 14
Car.
;;; IF (J=2950-3017 ) MW[200] = 3 ; 10
How.
;;; IF (J=3018-3102,3104-3116 ) MW[200] = 3 ; 11
AnnAr
;;; IF (J=3288-3334 ) MW[200] = 3 ; 15
Calv
;;; IF (J=3335-3409 ) MW[200] = 3 ; 16
StM
;;; IF (J=3117-3229 ) MW[200] = 3 ; 12
Chs.
;;;
;;; IF (J=3604-3653 ) MW[200] = 4 ; 21
Fau
;;; IF (J=3449-3477,3479-3481,3483-3494,3496-3541 ) MW[200] = 4 ; 19
Stf.
;;; IF (J=3654-3662,3663-3675 ) MW[200] = 4 ; 22/23
Clk,Jeff.
;;; IF (J=3435-3448,3542-3543,3545-3603 ) MW[200] = 4 ; 18/20
Fbg,Spots
;;; IF (J=3410-3434 ) MW[200] = 4 ; 17
KG.
;;;
;;; IF (J=3676-3722 ) MW[200] = 5 ;
Externals
;;;
;;; ; Establish K factors for each purpose:
;;; ;;; HBWK HBSK HBOK
;;; ;;;
;;; ;;; -----

```

## Appendix C Cube Voyager Scripts

```

;;; IF (MW[100] = 1 && MW[200] = 1) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; DC/SubMD to DC/SubMD
;;; IF (MW[100] = 1 && MW[200] = 2) mw[1] = 800 mw[2] = 250 mw[3] = 300
mw[4] = 600 mw[5] = 300 ; DC/SubMD to SubVA
;;; IF (MW[100] = 1 && MW[200] = 3) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; DC/SubMD to OuterMD
;;; IF (MW[100] = 1 && MW[200] = 4) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; DC/SubMD to OuterVA
;;;
;;; IF (MW[100] = 2 && MW[200] = 1) mw[1] = 900 mw[2] = 250 mw[3] = 700
mw[4] = 600 mw[5] = 300 ; SubVA to DC/SubMD
;;; IF (MW[100] = 2 && MW[200] = 2) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; SubVA to SubVA
;;; IF (MW[100] = 2 && MW[200] = 3) mw[1] = 500 mw[2] = 500 mw[3] = 300
mw[4] = 500 mw[5] = 500 ; SubVA to OuterMD
;;; IF (MW[100] = 2 && MW[200] = 4) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; SubVA to OuterVA
;;;
;;; IF (MW[100] = 3 && MW[200] = 1) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterMD to DC/SubMD
;;; IF (MW[100] = 3 && MW[200] = 2) mw[1] = 700 mw[2] = 1000 mw[3] = 1000
mw[4] = 500 mw[5] = 400 ; OuterMD to SubVA
;;; IF (MW[100] = 3 && MW[200] = 3) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterMD to OuterMD
;;; IF (MW[100] = 3 && MW[200] = 4) mw[1] = 500 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterMD to OuterVA
;;;
;;; IF (MW[100] = 4 && MW[200] = 1) mw[1] = 700 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterVA to DC/SubMD
;;; IF (MW[100] = 4 && MW[200] = 2) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterVA to SubVA
;;; IF (MW[100] = 4 && MW[200] = 3) mw[1] = 300 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterVA to OuterMD
;;; IF (MW[100] = 4 && MW[200] = 4) mw[1] = 1000 mw[2] = 1000 mw[3] = 1000
mw[4] = 1000 mw[5] = 1000 ; OuterVA to OuterVA
;;;
;;; ENDJLOOP
;;;
;;; /* ***** End Bridge penalty section ***** */
;-----
;Define K-Factor production areas
;
; HBW: Same as those used in the Ver. 2.2 model, but dropped
; * pw-dc core
; * frd-frd
;
if (i = 5,48,51,64,66-180,210-281,288-373,382-393)
mw[1] = 2000, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; DC non-
core to DC core
elseif (i = 394-769)
mw[1] = 2000, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; Mont to
DC core
elseif (i = 394- 769)
mw[1] = 2500, include= 394- 769 ; Mont to
Mont
elseif (i = 771-776,778-1404)
mw[1] = 1500, include=771-776,778-1404 ; PG to PG
elseif (i = 471-1476, 1486-1489, 1495-1497)
mw[1] = 2500, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; Arl cr
to DC cr
elseif (i = 1405-1470,1477-1485,1490-1494,1498-1545)
mw[1] = 1700, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; Arl
non-cr to DC cr
elseif (i = 1546-1610)
mw[1] = 2000, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; Alx to
DC cr
elseif (i = 1611-2159)

```

```

mw[1] = 1500, include= 1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; Ffx to
DC cr
elseif (i = 1611-2159)
mw[1] = 1000, include= 5,48,51,64,66-180,210-281,288-373,382-393 ; Ffx to DC
non-cr
elseif (i = 1611-2159)
mw[1] = 1200, include= 1611-2159 ; Ffx to Ffx
elseif (i = 2442-2554,2556-2628,2630-2819)
mw[1] = 2000, include= 1611-2159 ; PW to Ffx
endif

; HBS = 1.5 for most intra-jurisdiction movements
; Exceptions: DC; Mont; PG; Ffx; Staf all = 2.0

if (i = 5,48,51,64,66-180,210-281,288-373,382-393)
mw[2] = 2500, include=5,48,51,64,66-180,210-281,288-373,382-393 ; DC non-
core to DC non-core
elseif (i = 394- 769)
mw[2] = 2000, include= 394- 769 ; Mont to Mont
elseif (i = 771-776,778-1404)
mw[2] = 2500, include=771-776,778-1404 ; PG to PG
elseif (i = 1405-1470,1477-1485,1490-1494,1498-1545)
mw[2] = 2000, include= 1405-1470,1477-1485,1490-1494,1498-1545 ; Arl non-
core to Arl non-core
elseif (i = 1546-1610)
mw[2] = 2000, include= 1546-1610 ; Alx to Alx
elseif (i = 1611-2159)
mw[2] = 2500, include= 1611-2159 ; Ffx to Ffx
elseif (i = 2160-2441)
mw[2] = 1500, include= 2160-2441 ; Ldn to Ldn
elseif (i = 2442-2554,2556-2628,2630-2819)
mw[2] = 1750, include= 2442-2554,2556-2628,2630-2819 ; PW to PW
elseif (i = 2820-2949)
mw[2] = 1500, include= 2820-2949 ; Frd to Frd
elseif (i = 3230-3265,3268-3287)
mw[2] = 1500, include= 3230-3265,3268-3287 ; Car to Car
elseif (i = 2950-3017)
mw[2] = 1500, include= 2950-3017 ; How to How
elseif (i = 3018-3102,3104-3116)
mw[2] = 1500, include= 3018-3102,3104-3116 ; Ann to Ann
elseif (i = 3288-3334)
mw[2] = 1500, include= 3288-3334 ; Calv to Calv
elseif (i = 3335-3409)
mw[2] = 1500, include= 3335-3409 ; StM to StM
elseif (i = 3117-3229)
mw[2] = 1500, include= 3117-3229 ; Chs to Chs
elseif (i = 3604-3653)
mw[2] = 1500, include= 3604-3653 ; Fau to Fau
elseif (i = 3449-3477,3479-3481,3483-3494,3496-3541)
mw[2] = 1500, include= 3449-3477,3479-3481,3483-3494,3496-3541 ; Staf to
Staf
elseif (i = 3654-3662)
mw[2] = 1500, include= 3654-3662 ; Clrk to Clrk
elseif (i = 3663-3675)
mw[2] = 1500, include= 3663-3675 ; Jef to Jef
elseif (i = 3435-3448)
mw[2] = 1500, include= 3435-3448 ; Frbrg to Frbrg
elseif (i = 3542-3543,3545-3603)
mw[2] = 1500, include= 3542-3543,3545-3603 ; Spots to Spots
elseif (i = 3410-3434)
mw[2] = 1500, include= 3410-3434 ; KingG to KingG
endif

; HBO = 1.5 for some intra-jurisdiction movements
; = 2.0 for other intra-jurisdiction movements

if (i = 5,48,51,64,66-180,210-281,288-373,382-393)

```

## Appendix C Cube Voyager Scripts

```

mw[3] = 2200, include= 5,48,51,64,66-180,210-281,288-373,382-393 ; DC
non-core to DC non-core
elseif (i = 394- 769)
mw[3] = 2200, include= 394- 769 ; Mont to Mont
elseif (i = 771-776,778-1404)
mw[3] = 2500, include=771-776,778-1404 ; PG to PG
elseif (i = 1405-1470,1477-1485,1490-1494,1498-1545)
mw[3] = 2200, include= 1405-1470,1477-1485,1490-1494,1498-1545 ; Arl non-
core to Arl non-core
elseif (i = 1546-1610)
mw[3] = 2200, include= 1546-1610 ; Alx to Alx
elseif (i = 1611-2159)
mw[3] = 2500, include= 1611-2159 ; Ffx to Ffx
elseif (i = 2160-2441)
mw[3] = 2200, include= 2160-2441 ; Ldn to Ldn
elseif (i = 2442-2554,2556-2628,2630-2819)
mw[3] = 2200, include= 2442-2554,2556-2628,2630-2819 ; PW to PW
elseif (i = 2820-2949)
mw[3] = 2200, include= 2820-2949 ; Frd to Frd
elseif (i = 3230-3265,3268-3287)
mw[3] = 2200, include= 3230-3265,3268-3287 ; Car to Car
elseif (i = 2950-3017)
mw[3] = 2200, include= 2950-3017 ; How to How
elseif (i = 3018-3102,3104-3116)
mw[3] = 2200, include= 3018-3102,3104-3116 ; Ann to Ann
elseif (i = 3288-3334)
mw[3] = 1500, include= 3288-3334 ; Calv to Calv
elseif (i = 3335-3409)
mw[3] = 1500, include= 3335-3409 ; StM to StM
elseif (i = 3117-3229)
mw[3] = 1500, include= 3117-3229 ; Chs to Chs
elseif (i = 3604-3653)
mw[3] = 1500, include= 3604-3653 ; Fau to Fau
elseif (i = 3449-3477,3479-3481,3483-3494,3496-3541)
mw[3] = 1000, include= 3449-3477,3479-3481,3483-3494,3496-3541 ; Staf to
Staf
elseif (i = 3654-3662)
mw[3] = 1500, include= 3654-3662 ; Clrk to Clrk
elseif (i = 3663-3675)
mw[3] = 1500, include= 3663-3675 ; Jef to Jef
elseif (i = 3435-3448)
mw[3] = 1000, include= 3435-3448 ; Frbrg to Frbrg
elseif (i = 3542-3543,3545-3603)
mw[3] = 1000, include= 3542-3543,3545-3603 ; Spots to Spots
elseif (i = 3410-3434)
mw[3] = 1500, include= 3410-3434 ; KingG to KingG
endif

; NHW = 1.5 for most intra-jurisdiction movements
; = 2.0 for some intra-jurisdiction movements

if (i = 5,48,51,64,66-180,210-281,288-373,382-393)
mw[4] = 1500, include= 5,48,51,64,66-180,210-281,288-373,382-393 ; DC
non-core to DC non-core
elseif (i = 394- 769)
mw[4] = 2200, include= 394- 769 ; Mont to Mont
elseif (i = 771-776,778-1404)
mw[4] = 1500, include=771-776,778-1404 ; PG to PG
elseif (i = 1405-1470,1477-1485,1490-1494,1498-1545)
mw[4] = 1700, include= 1405-1470,1477-1485,1490-1494,1498-1545 ; Arl non-
core to Arl non-core
elseif (i = 1546-1610)
mw[4] = 1700, include= 1546-1610 ; Alx to Alx
elseif (i = 1611-2159)
mw[4] = 2000, include= 1611-2159 ; Ffx to Ffx
elseif (i = 2160-2441)
mw[4] = 1700, include= 2160-2441 ; Ldn to Ldn

```

```

elseif (i = 2442-2554,2556-2628,2630-2819)
mw[4] = 1500, include= 2442-2554,2556-2628,2630-2819 ; PW to PW
elseif (i = 2820-2949)
mw[4] = 1500, include= 2820-2949 ; Frd to Frd
elseif (i = 3230-3265,3268-3287)
mw[4] = 1500, include= 3230-3265,3268-3287 ; Car to Car
elseif (i = 2950-3017)
mw[4] = 1700, include= 2950-3017 ; How to How
elseif (i = 3018-3102,3104-3116)
mw[4] = 1500, include= 3018-3102,3104-3116 ; Ann to Ann
elseif (i = 3288-3334)
mw[4] = 1500, include= 3288-3334 ; Calv to Calv
elseif (i = 3335-3409)
mw[4] = 1500, include= 3335-3409 ; StM to StM
elseif (i = 3117-3229)
mw[4] = 1500, include= 3117-3229 ; Chs to Chs
elseif (i = 3604-3653)
mw[4] = 1500, include= 3604-3653 ; Fau to Fau
elseif (i = 3449-3477,3479-3481,3483-3494,3496-3541)
mw[4] = 1500, include= 3449-3477,3479-3481,3483-3494,3496-3541 ; Staf to
Staf
elseif (i = 3654-3662)
mw[4] = 1500, include= 3654-3662 ; Clrk to Clrk
elseif (i = 3663-3675)
mw[4] = 1500, include= 3663-3675 ; Jef to Jef
elseif (i = 3435-3448)
mw[4] = 1500, include= 3435-3448 ; Frbrg to Frbrg
elseif (i = 3542-3543,3545-3603)
mw[4] = 1500, include= 3542-3543,3545-3603 ; Spots to Spots
elseif (i = 3410-3434)
mw[4] = 1500, include= 3410-3434 ; KingG to KingG
endif

; NHO = 1.5 for most intra-jurisdiction movements
; = 2.0 for some intra-jurisdiction movements

if (i = 5,48,51,64,66-180,210-281,288-373,382-393)
mw[5] = 2500, include= 5,48,51,64,66-180,210-281,288-373,382-393 ; DC
non-core to DC non-core
elseif (i = 394- 769)
mw[5] = 1500, include= 394- 769 ; Mont to Mont
elseif (i = 771-776,778-1404)
mw[5] = 1700, include=771-776,778-1404 ; PG to PG
elseif (i = 1405-1470,1477-1485,1490-1494,1498-1545)
mw[5] = 1700, include= 1405-1470,1477-1485,1490-1494,1498-1545 ; Arl non-
core to Arl non-core
elseif (i = 1546-1610)
mw[5] = 1700, include= 1546-1610 ; Alx to Alx
elseif (i = 1611-2159)
mw[5] = 2100, include= 1611-2159 ; Ffx to Ffx
elseif (i = 2160-2441)
mw[5] = 1500, include= 2160-2441 ; Ldn to Ldn
elseif (i = 2442-2554,2556-2628,2630-2819)
mw[5] = 1500, include= 2442-2554,2556-2628,2630-2819 ; PW to PW
elseif (i = 2820-2949)
mw[5] = 1500, include= 2820-2949 ; Frd to Frd
elseif (i = 3230-3265,3268-3287)
mw[5] = 1500, include= 3230-3265,3268-3287 ; Car to Car
elseif (i = 2950-3017)
mw[5] = 1700, include= 2950-3017 ; How to How
elseif (i = 3018-3102,3104-3116)
mw[5] = 1700, include= 3018-3102,3104-3116 ; Ann to Ann
elseif (i = 3288-3334)
mw[5] = 1700, include= 3288-3334 ; Calv to Calv
elseif (i = 3335-3409)
mw[5] = 1700, include= 3335-3409 ; StM to StM
elseif (i = 3117-3229)

```

## Appendix C Cube Voyager Scripts

```

mw[5] = 1700, include= 3117-3229      ; Chs to Chs
elseif (i = 3604-3653)
mw[5] = 1700, include= 3604-3653      ; Fau to Fau
elseif (i = 3449-3477,3479-3481,3483-3494,3496-3541)
mw[5] = 1700, include= 3449-3477,3479-3481,3483-3494,3496-3541      ; Staf to
Staf
elseif (i = 3654-3662)
mw[5] = 1300, include= 3654-3662      ; Clrk to Clrk
elseif (i = 3663-3675)
mw[5] = 1300, include= 3663-3675      ; Jef to Jef
elseif (i = 3435-3448)
mw[5] = 1700, include= 3435-3448      ; Frbrg to Frbrg
elseif (i = 3542-3543,3545-3603)
mw[5] = 1700, include= 3542-3543,3545-3603      ; Spots to Spots
elseif (i = 3410-3434)
mw[5] = 1500, include= 3410-3434      ; KingG to KingG
endif

MATO[1] =@HBWK@ ,MO=1
MATO[2] =@HBSK@ ,MO=2
MATO[3] =@HBOK@ ,MO=3
MATO[4] =@NHWK@ ,MO=4
MATO[5] =@NHOK@ ,MO=5

; ////////////////////////////////////////////////////
; // End of K-Factor Specifications for All Purposes //
; ////////////////////////////////////////////////////

endrun

; ////////////////////////////////////////////////////
; // Added step to create station name file for the Linesum //
; // program directly from scenario specific station file //
; ////////////////////////////////////////////////////

; Program step to write a station names file (used by the LINESUM program) directly
from
; the "Station.dbf" file. The O/P file will be sent to the inputs subdirectory; make
sure
; that this is reflected in the LINESUM_MR_Access.ctl file

; Input file:
Sta_File = 'inputs\Station.dbf' ; Std. Station file

; Output File:
Sta_Names = 'inputs\station_names.dbf' ;

RUN PGM=MATRIX
Zones=1
FILEI DBI[1] ="@Sta_File@"
FILEO RECO[1] ="@Sta_Names@", Fields= ID(12.0),Station(c35)
;
LOOP K = 1,dbi.1.NUMRECORDS
x = DBIReadRecord(1,k)
idx = dbi.1.recno ;
STAT = di.1.STAT ; Station no.
STAN1 = di.1.STAN1 ; First Bus Node no.
MM = di.1.MM ; Mode Code (M,C,L,B,N)
SNAME = di.1.SNAME ; Mode Code (M,C,L,B,N)
;
; Index = STAT unless mode = "B"- in that case the Station will equal
STAN1

```

```

RO.ID = STAT
IF (MM = 'B') RO.ID = STAN1
RO.STATION = SNAME
WRITE RECO = 1
ENDLOOP
ENDRUN

```

## 40 Time-of-Day.s

```

; =====
; Time-of-Day.s
; MWCOG Version 2.3 Model
; 4/3/12 - three decimals maintained for the MATO files ("dec = 3*3" added to mato
statements)
;
;
; Distribute Modeled Pump Prime Auto Driver Trips, i.e,
; 4 Purposes (HBW,HBS,HBO,NHW), 3 Modes (1,2,3+Occ Adrs)
; among three time periods:
;
; - AM peak (6:00AM - 9:00 AM) 3 Hrs.
; - Midday (9:00AM - 3:00 PM) 6 Hrs.
; - PM peak (3:00PM - 7:00 PM) 4 Hrs.
; - Off-peak (All Other hrs ) 11 Hrs.
;
; file named: 'todcomp_2008HTS_AdjOP.dbf' is used.
; It contains trip percentages for each time period
; by purpose, mode, and direction.
;
; Environment Variable:
; _iter_ (Iteration indicator = 'pp','il'-'i6'
; =====
;
; ////////////////////////////////////////////////////
; //
; Input/Output filenames: //
; //
; TODFtrs = '..\support\todcomp_2008HTS.dbf' ; Time of Day Factor File //
; //
; I/P PP Auto Driver Trip Tables: // ; I/P PP Auto
Driver Trip Tables: //
HBWADR = '%_iter_%_HBW_adr.mat' ; HBW 1,2,3+ Occ Adr Trips (t1-3) //
HBSADR = '%_iter_%_HBS_adr.mat' ; HBS 1,2,3+ Occ Adr Trips (t1-3) //
HBOADR = '%_iter_%_HBO_adr.mat' ; HBO 1,2,3+ Occ Adr Trips (t1-3) //
NHWADR = '%_iter_%_NHW_adr.mat' ; NHW 1,2,3+ Occ Adr Trips (t1-3) //
NHOADR = '%_iter_%_NHO_adr.mat' ; NHO 1,2,3+ Occ Adr Trips (t1-3) //
; //
; O/P Auto Dr. Pct. tables: //
ADRAM = '%_iter_%_am_adr.mat' ; AM Modeled Total Auto Drivers //
ADRFM = '%_iter_%_pm_adr.mat' ; PM Modeled Total Auto Drivers //
ADRMD = '%_iter_%_md_adr.mat' ; Midday Modeled Total Auto Drivers //
ADRNT = '%_iter_%_nt_adr.mat' ; Night Modeled Total Auto Drivers //
;
; ; define TOD ARRAY parameters
Pur = 5 ; 1/HBW, 2/HBS, 3/HBO, 4/NHW, 5/NHO
Mod = 4 ; 1/Adr, 2/DrAlone 3/CarPoolPsn 4/Transit
Dir = 2 ; 1/H>NH, 2/NH>H
Per = 4 ; 1/AM, 2/MD, 3/PM, 4/NT

```

## Appendix C Cube Voyager Scripts

```

RUN PGM=MATRIX
pageheight=32767          ; Preclude header breaks
MATI[1]=@HBWADR@         ; HBW 1,2,3+-Occ. Auto Drv. Trips(T1-3)
MATI[2]=@HBSADR@         ; HBS 1,2,3+-Occ. Auto Drv. Trips(T1-3)
MATI[3]=@HBOADR@         ; HBO 1,2,3+-Occ. Auto Drv. Trips(T1-3)
MATI[4]=@NHWADR@        ; NHW 1,2,3+-Occ. Auto Drv. Trips(T1-3)
MATI[5]=@NHOADR@        ; NHO 1,2,3+-Occ. Auto Drv. Trips(T1-3)

; These are in P/A format and represent the Home-to-NonHome direction

      FILLMW MW[111] = MI.1.1, MI.1.2, MI.1.3 ;Work 1,2,3+ Occ Adrs P/A
t111-t113
      FILLMW MW[121] = MI.2.1, MI.2.2, MI.2.3 ;Shop 1,2,3+ Occ Adrs P/A
t121-t123
      FILLMW MW[131] = MI.3.1, MI.3.2, MI.3.3 ;Othr 1,2,3+ Occ Adrs P/A
t131-t133
      FILLMW MW[141] = MI.4.1, MI.4.2, MI.4.3 ;NHW 1,2,3+ Occ Adrs P/A
t141-t143
      FILLMW MW[151] = MI.5.1, MI.5.2, MI.5.3 ;NHO 1,2,3+ Occ Adrs P/A
t151-t153

; Put Transpose of the above
; HBW, HBS, HBO, NHW, NHO trip tables
;
MW[211]=MI.1.1.T, MW[212]=MI.1.2.T, MW[213]=MI.1.3.T; HBW 1,2,3+ Occ Adrs A/P
t211-213
MW[221]=MI.2.1.T, MW[222]=MI.2.2.T, MW[223]=MI.2.3.T; HBS 1,2,3+ Occ Adrs A/P
t221-223
MW[231]=MI.3.1.T, MW[232]=MI.3.2.T, MW[233]=MI.3.3.T; HBO 1,2,3+ Occ Adrs A/P
t231-233
MW[241]=MI.4.1.T, MW[242]=MI.4.2.T, MW[243]=MI.4.3.T; NHW 1,2,3+ Occ Adrs A/P
t241-243
MW[251]=MI.5.1.T, MW[252]=MI.5.2.T, MW[253]=MI.5.3.T; NHO 1,2,3+ Occ Adrs A/P
t251-253

;
; Now read TOD factors file
;
Array TODFtrs  =@Pur@,@Mod@,@Dir@,@Per@

;=====
;=====
;=====
; Read in Time of Day factor file and populate TOD factor array

FILEI DBI[1]  ="@TODFtrs@"
LOOP K = 1,dbi.1.NUMRECORDS      ;;PURP  MODE  DIR  AM  MD  PM
OP

      x = DBIReadRecord(1,k)
      count = dbi.1.recno
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][1] = di.1.AM
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][2] = di.1.MD
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][3] = di.1.PM
      TODFtrs[di.1.Purp][di.1.Mode][di.1.DIR][4] = di.1.OP
ENDLOOP
;=====
;=====
;=====

JLOOP

;;          Trips          p m d p          Trips          p m d p

```

```

;;          in          u o i e          in          u o i e
;;          H-NH Dir          r d r r          H-NH Dir          r d r r
;;          |          | | | |          |          | | | |
;
      mw[501] = (MW[111] * (TODFtrs[1][2][1][1]/100.00) + MW[211]*
(TODFtrs[1][1][2][2][1]/100.00) / 2.0 ; HBW / DA *****
      mw[502] = (MW[112] * (TODFtrs[1][3][1][1]/100.00) + MW[212]*
(TODFtrs[1][3][2][2][1]/100.00) / 2.0 ; HBW / 2-occ carpool * *
      mw[503] = (MW[113] * (TODFtrs[1][3][1][1]/100.00) + MW[213]*
(TODFtrs[1][3][2][2][1]/100.00) / 2.0 ; HBW / 3+occ carpool * A *

;
      * M *
      mw[504] = (MW[121] * (TODFtrs[2][2][1][1]/100.00) + MW[221]*
(TODFtrs[2][2][2][2][1]/100.00) / 2.0 ; HBS / DA * *
      mw[505] = (MW[122] * (TODFtrs[2][3][1][1]/100.00) + MW[222]*
(TODFtrs[2][3][2][2][1]/100.00) / 2.0 ; HBS / 2-occ carpool * P *
      mw[506] = (MW[123] * (TODFtrs[2][3][1][1]/100.00) + MW[223]*
(TODFtrs[2][3][2][2][1]/100.00) / 2.0 ; HBS / 3+occ carpool * E *

;
      * A *
      mw[507] = (MW[131] * (TODFtrs[3][2][1][1]/100.00) + MW[231]*
(TODFtrs[3][2][2][2][1]/100.00) / 2.0 ; HBO / DA * K *
      mw[508] = (MW[132] * (TODFtrs[3][3][1][1]/100.00) + MW[232]*
(TODFtrs[3][3][2][2][1]/100.00) / 2.0 ; HBO / 2-occ carpool * *
      mw[509] = (MW[133] * (TODFtrs[3][3][1][1]/100.00) + MW[233]*
(TODFtrs[3][3][2][2][1]/100.00) / 2.0 ; HBO / 3+occ carpool * P *

;
      * E *
      mw[510] = (MW[141] * (TODFtrs[4][2][1][1]/100.00) + MW[241]*
(TODFtrs[4][2][2][2][1]/100.00) / 2.0 ; NHW / DA * R *
      mw[511] = (MW[142] * (TODFtrs[4][3][1][1]/100.00) + MW[242]*
(TODFtrs[4][3][2][2][1]/100.00) / 2.0 ; NHW / 2-occ carpool * I *
      mw[512] = (MW[143] * (TODFtrs[4][3][1][1]/100.00) + MW[243]*
(TODFtrs[4][3][2][2][1]/100.00) / 2.0 ; NHW / 3+occ carpool * O *

;
      * D *
      mw[513] = (MW[151] * (TODFtrs[5][2][1][1]/100.00) + MW[251]*
(TODFtrs[5][2][2][2][1]/100.00) / 2.0 ; NHO / DA * *
      mw[514] = (MW[152] * (TODFtrs[5][3][1][1]/100.00) + MW[252]*
(TODFtrs[5][3][2][2][1]/100.00) / 2.0 ; NHO / 2-occ carpool * *
      mw[515] = (MW[153] * (TODFtrs[5][3][1][1]/100.00) + MW[253]*
(TODFtrs[5][3][2][2][1]/100.00) / 2.0 ; NHO / 3+occ carpool *****

;;

      mw[516] = (MW[111] * (TODFtrs[1][2][1][2]/100.00) + MW[211]*
(TODFtrs[1][2][2][2][1]/100.00) / 2.0 ; HBW / DA *****
      mw[517] = (MW[112] * (TODFtrs[1][3][1][2]/100.00) + MW[212]*
(TODFtrs[1][3][2][2][1]/100.00) / 2.0 ; HBW / 2-occ carpool * *
      mw[518] = (MW[113] * (TODFtrs[1][3][1][2]/100.00) + MW[213]*
(TODFtrs[1][3][2][2][1]/100.00) / 2.0 ; HBW / 3+occ carpool * M *

;
      * I *
      mw[519] = (MW[121] * (TODFtrs[2][2][1][2]/100.00) + MW[221]*
(TODFtrs[2][2][2][2][1]/100.00) / 2.0 ; HBS / DA * D *
      mw[520] = (MW[122] * (TODFtrs[2][3][1][2]/100.00) + MW[222]*
(TODFtrs[2][3][2][2][1]/100.00) / 2.0 ; HBS / 2-occ carpool * D *
      mw[521] = (MW[123] * (TODFtrs[2][3][1][2]/100.00) + MW[223]*
(TODFtrs[2][3][2][2][1]/100.00) / 2.0 ; HBS / 3+occ carpool * A *

;
      * Y *
      mw[522] = (MW[131] * (TODFtrs[3][2][1][2]/100.00) + MW[231]*
(TODFtrs[3][2][2][2][1]/100.00) / 2.0 ; HBO / DA * *
      mw[523] = (MW[132] * (TODFtrs[3][3][1][2]/100.00) + MW[232]*
(TODFtrs[3][3][2][2][1]/100.00) / 2.0 ; HBO / 2-occ carpool * *
      mw[524] = (MW[133] * (TODFtrs[3][3][1][2]/100.00) + MW[233]*
(TODFtrs[3][3][2][2][1]/100.00) / 2.0 ; HBO / 3+occ carpool * P *

```

## Appendix C Cube Voyager Scripts

```

;
; * E *
mw[525] = (MW[141] * (TODFtrs[4][2][1][2]/100.00) + MW[241]*
(TODFtrs[4][2][2][2]/100.00)) / 2.0 ; NHW / DA * R *
mw[526] = (MW[142] * (TODFtrs[4][3][1][2]/100.00) + MW[242]*
(TODFtrs[4][3][2][2]/100.00)) / 2.0 ; NHW / 2-occ carpool * I *
mw[527] = (MW[143] * (TODFtrs[4][3][1][2]/100.00) + MW[243]*
(TODFtrs[4][3][2][2]/100.00)) / 2.0 ; NHW / 3+occ carpool * O *

;
; * D *
mw[528] = (MW[151] * (TODFtrs[5][2][1][2]/100.00) + MW[251]*
(TODFtrs[5][2][2][2]/100.00)) / 2.0 ; NHO / DA * *
mw[529] = (MW[152] * (TODFtrs[5][3][1][2]/100.00) + MW[252]*
(TODFtrs[5][3][2][2]/100.00)) / 2.0 ; NHO / 2-occ carpool * *
mw[530] = (MW[153] * (TODFtrs[5][3][1][2]/100.00) + MW[253]*
(TODFtrs[5][3][2][2]/100.00)) / 2.0 ; NHO / 3+occ carpool *****

;;

mw[531] = (MW[111] * (TODFtrs[1][2][1][3]/100.00) + MW[211]*
(TODFtrs[1][2][2][3]/100.00)) / 2.0 ; HBW / DA *****
mw[532] = (MW[112] * (TODFtrs[1][3][1][3]/100.00) + MW[212]*
(TODFtrs[1][3][2][3]/100.00)) / 2.0 ; HBW / 2-occ carpool * *
mw[533] = (MW[113] * (TODFtrs[1][3][1][3]/100.00) + MW[213]*
(TODFtrs[1][3][2][3]/100.00)) / 2.0 ; HBW / 3+occ carpool * P *

;
; * M *
mw[534] = (MW[121] * (TODFtrs[2][2][1][3]/100.00) + MW[221]*
(TODFtrs[2][2][2][3]/100.00)) / 2.0 ; HBS / DA * *
mw[535] = (MW[122] * (TODFtrs[2][3][1][3]/100.00) + MW[222]*
(TODFtrs[2][3][2][3]/100.00)) / 2.0 ; HBS / 2-occ carpool * P *
mw[536] = (MW[123] * (TODFtrs[2][3][1][3]/100.00) + MW[223]*
(TODFtrs[2][3][2][3]/100.00)) / 2.0 ; HBS / 3+occ carpool * E *

;
; * A *
mw[537] = (MW[131] * (TODFtrs[3][2][1][3]/100.00) + MW[231]*
(TODFtrs[3][2][2][3]/100.00)) / 2.0 ; HBO / DA * K *
mw[538] = (MW[132] * (TODFtrs[3][3][1][3]/100.00) + MW[232]*
(TODFtrs[3][3][2][3]/100.00)) / 2.0 ; HBO / 2-occ carpool * *
mw[539] = (MW[133] * (TODFtrs[3][3][1][3]/100.00) + MW[233]*
(TODFtrs[3][3][2][3]/100.00)) / 2.0 ; HBO / 3+occ carpool * P *

;
; * E *
mw[540] = (MW[141] * (TODFtrs[4][2][1][3]/100.00) + MW[241]*
(TODFtrs[4][2][2][3]/100.00)) / 2.0 ; NHW / DA * R *
mw[541] = (MW[142] * (TODFtrs[4][3][1][3]/100.00) + MW[242]*
(TODFtrs[4][3][2][3]/100.00)) / 2.0 ; NHW / 2-occ carpool * I *
mw[542] = (MW[143] * (TODFtrs[4][3][1][3]/100.00) + MW[243]*
(TODFtrs[4][3][2][3]/100.00)) / 2.0 ; NHW / 3+occ carpool * O *

;
; * D *
mw[543] = (MW[151] * (TODFtrs[5][2][1][3]/100.00) + MW[251]*
(TODFtrs[5][2][2][3]/100.00)) / 2.0 ; NHO / DA * *
mw[544] = (MW[152] * (TODFtrs[5][3][1][3]/100.00) + MW[252]*
(TODFtrs[5][3][2][3]/100.00)) / 2.0 ; NHO / 2-occ carpool * *
mw[545] = (MW[153] * (TODFtrs[5][3][1][3]/100.00) + MW[253]*
(TODFtrs[5][3][2][3]/100.00)) / 2.0 ; NHO / 3+occ carpool *****

;;

mw[546] = (MW[111] * (TODFtrs[1][2][1][4]/100.00) + MW[211]*
(TODFtrs[1][2][2][4]/100.00)) / 2.0 ; HBW / DA *****
mw[547] = (MW[112] * (TODFtrs[1][3][1][4]/100.00) + MW[212]*
(TODFtrs[1][3][2][4]/100.00)) / 2.0 ; HBW / 2-occ carpool * O *
mw[548] = (MW[113] * (TODFtrs[1][3][1][4]/100.00) + MW[213]*
(TODFtrs[1][3][2][4]/100.00)) / 2.0 ; HBW / 3+occ carpool * F *

;
; * F *

```

```

mw[549] = (MW[121] * (TODFtrs[2][2][1][4]/100.00) + MW[221]*
(TODFtrs[2][2][2][4]/100.00)) / 2.0 ; HBS / DA * *
mw[550] = (MW[122] * (TODFtrs[2][3][1][4]/100.00) + MW[222]*
(TODFtrs[2][3][2][4]/100.00)) / 2.0 ; HBS / 2-occ carpool * P *
mw[551] = (MW[123] * (TODFtrs[2][3][1][4]/100.00) + MW[223]*
(TODFtrs[2][3][2][4]/100.00)) / 2.0 ; HBS / 3+occ carpool * E *

;
; * A *
mw[552] = (MW[131] * (TODFtrs[3][2][1][4]/100.00) + MW[231]*
(TODFtrs[3][2][2][4]/100.00)) / 2.0 ; HBO / DA * K *
mw[553] = (MW[132] * (TODFtrs[3][3][1][4]/100.00) + MW[232]*
(TODFtrs[3][3][2][4]/100.00)) / 2.0 ; HBO / 2-occ carpool * *
mw[554] = (MW[133] * (TODFtrs[3][3][1][4]/100.00) + MW[233]*
(TODFtrs[3][3][2][4]/100.00)) / 2.0 ; HBO / 3+occ carpool * P *

;
; * E *
mw[555] = (MW[141] * (TODFtrs[4][2][1][4]/100.00) + MW[241]*
(TODFtrs[4][2][2][4]/100.00)) / 2.0 ; NHW / DA * R *
mw[556] = (MW[142] * (TODFtrs[4][3][1][4]/100.00) + MW[242]*
(TODFtrs[4][3][2][4]/100.00)) / 2.0 ; NHW / 2-occ carpool * I *
mw[557] = (MW[143] * (TODFtrs[4][3][1][4]/100.00) + MW[243]*
(TODFtrs[4][3][2][4]/100.00)) / 2.0 ; NHW / 3+occ carpool * O *

;
; * D *
mw[558] = (MW[151] * (TODFtrs[5][2][1][4]/100.00) + MW[251]*
(TODFtrs[5][2][2][4]/100.00)) / 2.0 ; NHO / DA * *
mw[559] = (MW[152] * (TODFtrs[5][3][1][4]/100.00) + MW[252]*
(TODFtrs[5][3][2][4]/100.00)) / 2.0 ; NHO / 2-occ carpool * *
mw[560] = (MW[153] * (TODFtrs[5][3][1][4]/100.00) + MW[253]*
(TODFtrs[5][3][2][4]/100.00)) / 2.0 ; NHO / 3+occ carpool *****

;
;-----
; Summarize by purpose for checking - 601/hbw, 602/hbs, 603/hbo, 604/nhw, 605/nho
; Total HBW:
MW[601] = MW[501]+MW[502]+MW[503] + MW[516]+MW[517]+MW[518] +
MW[531]+MW[532]+MW[533] + MW[546]+MW[547]+MW[548]
; Total HBS:
MW[602] = MW[504]+MW[505]+MW[506] + MW[519]+MW[520]+MW[521] +
MW[534]+MW[535]+MW[536] + MW[549]+MW[550]+MW[551]
; Total HBO:
MW[603] = MW[507]+MW[508]+MW[509] + MW[522]+MW[523]+MW[524] +
MW[537]+MW[538]+MW[539] + MW[552]+MW[553]+MW[554]
; Total NHW:
MW[604] = MW[510]+MW[511]+MW[512] + MW[525]+MW[526]+MW[527] +
MW[540]+MW[541]+MW[542] + MW[555]+MW[556]+MW[557]
; Total NHO:
MW[605] = MW[513]+MW[514]+MW[515] + MW[528]+MW[529]+MW[530] +
MW[543]+MW[544]+MW[545] + MW[558]+MW[559]+MW[560]

;-----
; Summarize by Time period, Occ Group for Assignment 611-622
;
MW[611] = MW[501]+MW[504]+MW[507]+MW[510]+MW[513] ; AM 1-Occ adrs
MW[612] = MW[502]+MW[505]+MW[508]+MW[511]+MW[514] ; AM 2-Occ adrs
MW[613] = MW[503]+MW[506]+MW[509]+MW[512]+MW[515] ; AM 3+Occ adrs
;
MW[614] = MW[516]+MW[519]+MW[522]+MW[525]+MW[528] ; MD 1-Occ adrs
MW[615] = MW[517]+MW[520]+MW[523]+MW[526]+MW[529] ; MD 2-Occ adrs
MW[616] = MW[518]+MW[521]+MW[524]+MW[527]+MW[530] ; MD 3+Occ adrs
;
MW[617] = MW[531]+MW[534]+MW[537]+MW[540]+MW[543] ; PM 1-Occ adrs
MW[618] = MW[532]+MW[535]+MW[538]+MW[541]+MW[544] ; PM 2-Occ adrs
MW[619] = MW[533]+MW[536]+MW[539]+MW[542]+MW[545] ; PM 3+Occ adrs
;
MW[620] = MW[546]+MW[549]+MW[552]+MW[555]+MW[558] ; OP 1-Occ adrs
MW[621] = MW[547]+MW[550]+MW[553]+MW[556]+MW[559] ; OP 2-Occ adrs
MW[622] = MW[548]+MW[551]+MW[554]+MW[557]+MW[560] ; OP 3+Occ adrs

```

## Appendix C Cube Voyager Scripts

```

; Now summarize regional totals to summarize neatly

;;AM;;
; am hbw, hbs, hbo, nhb by occupant totals:
amhbw1=amhbw1+MW[501], amhbw2=amhbw2+MW[502], amhbw3=amhbw3+MW[503]
amhbs1=amhbs1+MW[504], amhbs2=amhbs2+MW[505], amhbs3=amhbs3+MW[506]
amhbo1=amhbo1+MW[507], amhbo2=amhbo2+MW[508], amhbo3=amhbo3+MW[509]
amnhw1=amnhw1+MW[510], amnhw2=amnhw2+MW[511], amnhw3=amnhw3+MW[512]
amnho1=amnho1+MW[513], amnho2=amnho2+MW[514], amnho3=amnho3+MW[515]
; am hbw, hbs, hbo, nhb totals:
amhbw =amhbw + MW[501] + MW[502] + MW[503]
amhbs =amhbs + MW[504] + MW[505] + MW[506]
amhbo =amhbo + MW[507] + MW[508] + MW[509]
amnhw =amnhw + MW[510] + MW[511] + MW[512]
amnho =amnho + MW[513] + MW[514] + MW[515]
; am occupant level totals:
am1 =am1 +MW[611],am2 =am2 +MW[612],am3 =am3 +MW[613]
; am totals:
am =am +MW[611] +MW[612] +MW[613]

;;MD;;
; md hbw, hbs, hbo, nhb by occupant totals:
mdhbw1=mdhbw1+MW[516], mdhbw2=mdhbw2+MW[517], mdhbw3=mdhbw3+MW[518]
mdhbs1=mdhbs1+MW[519], mdhbs2=mdhbs2+MW[520], mdhbs3=mdhbs3+MW[521]
mdhbo1=mdhbo1+MW[522], mdhbo2=mdhbo2+MW[523], mdhbo3=mdhbo3+MW[524]
mdnhw1=mdnhw1+MW[525], mdnhw2=mdnhw2+MW[526], mdnhw3=mdnhw3+MW[527]
mdnho1=mdnho1+MW[528], mdnho2=mdnho2+MW[529], mdnho3=mdnho3+MW[530]
; md hbw, hbs, hbo, nhb totals:
mdhbw =mdhbw + MW[516] + MW[517] + MW[518]
mdhbs =mdhbs + MW[519] + MW[520] + MW[521]
mdhbo =mdhbo + MW[522] + MW[523] + MW[524]
mdnhw =mdnhw + MW[525] + MW[526] + MW[527]
mdnho =mdnho + MW[528] + MW[529] + MW[530]
; md occupant level totals:
md1 =md1 +MW[614],md2 =md2 +MW[615],md3 =md3 +MW[616]
; md totals:
md =md +MW[614] +MW[615] +MW[616]

;;PM;;
; pm hbw, hbs, hbo, nhb by occupant totals:
pmhbw1=pmhbw1+MW[531], pmhbw2=pmhbw2+MW[532], pmhbw3=pmhbw3+MW[533]
pmhbs1=pmhbs1+MW[534], pmhbs2=pmhbs2+MW[535], pmhbs3=pmhbs3+MW[536]
pmhbo1=pmhbo1+MW[537], pmhbo2=pmhbo2+MW[538], pmhbo3=pmhbo3+MW[539]
pmnhw1=pmnhw1+MW[540], pmnhw2=pmnhw2+MW[541], pmnhw3=pmnhw3+MW[542]
pmnho1=pmnho1+MW[543], pmnho2=pmnho2+MW[544], pmnho3=pmnho3+MW[545]
; pm hbw, hbs, hbo, nhb totals:
pmhbw =pmhbw + MW[531] + MW[532] + MW[533]
pmhbs =pmhbs + MW[534] + MW[535] + MW[536]
pmhbo =pmhbo + MW[537] + MW[538] + MW[539]
pmnhw =pmnhw + MW[540] + MW[541] + MW[542]
pmnho =pmnho + MW[543] + MW[544] + MW[545]
; pm occupant level totals:
pm1 =pm1 +MW[617],pm2 =pm2 +MW[618],pm3 =pm3 +MW[619]
; pm totals:
pm =pm +MW[617] +MW[618] +MW[619]

;;OP;;
; op hbw, hbs, hbo, nhb by occupant totals:
ophbw1=ophbw1+MW[546], ophbw2=ophbw2+MW[547], ophbw3=ophbw3+MW[548]
ophbs1=ophbs1+MW[549], ophbs2=ophbs2+MW[550], ophbs3=ophbs3+MW[551]
ophbo1=ophbo1+MW[552], ophbo2=ophbo2+MW[553], ophbo3=ophbo3+MW[554]
opnhw1=opnhw1+MW[555], opnhw2=opnhw2+MW[556], opnhw3=opnhw3+MW[557]
opnho1=opnho1+MW[558], opnho2=opnho2+MW[559], opnho3=opnho3+MW[560]
; op hbw, hbs, hbo, nhb totals:
ophbw =ophbw + MW[546] + MW[547] + MW[548]
ophbs =ophbs + MW[549] + MW[550] + MW[551]

```

```

ophbo =ophbo + MW[552] + MW[553] + MW[554]
opnhw =opnhw + MW[555] + MW[556] + MW[557]
opnho =opnho + MW[558] + MW[559] + MW[560]
; op occupant level totals:
op1 =op1 +MW[620],op2 =op2 +MW[621],op3 =op3 +MW[622]
; op totals:
op =op +MW[620] +MW[621] +MW[622]

;=====
;=====
; total output trips by purpose--output total:
ohbw=ohbw+MW[601], ohbs=ohbs+MW[602], ohbo=ohbo+MW[603], onhw=onhw+MW[604],
onho=onho+MW[605]

; total grand Total of output auto driver trips:
adr = adr + MW[601] + MW[602] + MW[603] + MW[604] + MW[605]

; total input trips by purpose
ihbw=ihbw + MW[111] + MW[112] + MW[113]
ihbs=ihbs + MW[121] + MW[122] + MW[123]
ihbo=ihbo + MW[131] + MW[132] + MW[133]
inhw=inhw + MW[141] + MW[142] + MW[143]
inho=inho + MW[151] + MW[152] + MW[153]

ENDJLOOP

; now write out the totals neatly:
if (i=zones)
; get differences by purpose (output - Input)
dfhbw = ohbw - ihbw;
dfhbs = ohbs - ihbs;
dfhbo = ohbo - ihbo;
dfnhw = onhw - inhw;
dfnho = onho - inho;

LIST = '/bt '
LIST = ' Modeled Pump Prime Time-of-Day Results','\n'
list = 'AM Period: 1-Occ. 2-Occ. 3+Occ. Total'
list = 'HBW ',amhbw1(8.0),amhbw2(8.0),amhbw3(8.0),' ',amhbw(8.0)
list = 'HBS ',amhbs1(8.0),amhbs2(8.0),amhbs3(8.0),' ',amhbs(8.0)
list = 'HBO ',amhbo1(8.0),amhbo2(8.0),amhbo3(8.0),' ',amhbo(8.0)
list = 'NHW ',amnhw1(8.0),amnhw2(8.0),amnhw3(8.0),' ',amnhw(8.0)
list = 'NHO ',amnho1(8.0),amnho2(8.0),amnho3(8.0),' ',amnho(8.0)
list = '-----'
list = 'Subtotal: ',am1(8.0),am2(8.0),am3(8.0),' ',am(8.0)
list = ' '
list = ' '
list = 'Midday: 1-Occ. 2-Occ. 3+Occ. Total'
list = 'HBW ',mdhbw1(8.0),mdhbw2(8.0),mdhbw3(8.0),' ',mdhbw(8.0)
list = 'HBS ',mdhbs1(8.0),mdhbs2(8.0),mdhbs3(8.0),' ',mdhbs(8.0)
list = 'HBO ',mdhbo1(8.0),mdhbo2(8.0),mdhbo3(8.0),' ',mdhbo(8.0)
list = 'NHW ',mdnhw1(8.0),mdnhw2(8.0),mdnhw3(8.0),' ',mdnhw(8.0)
list = 'NHO ',mdnho1(8.0),mdnho2(8.0),mdnho3(8.0),' ',mdnho(8.0)
list = '-----'
list = 'Subtotal: ',md1(8.0),md2(8.0),md3(8.0),' ',md(8.0)
list = ' '
list = ' '
list = 'PM Period: 1-Occ. 2-Occ. 3+Occ. Total'
list = 'HBW ',pmhbw1(8.0),pmhbw2(8.0),pmhbw3(8.0),' ',pmhbw(8.0)
list = 'HBS ',pmhbs1(8.0),pmhbs2(8.0),pmhbs3(8.0),' ',pmhbs(8.0)
list = 'HBO ',pmhbo1(8.0),pmhbo2(8.0),pmhbo3(8.0),' ',pmhbo(8.0)
list = 'NHW ',pmnhw1(8.0),pmnhw2(8.0),pmnhw3(8.0),' ',pmnhw(8.0)
list = 'NHO ',pmnho1(8.0),pmnho2(8.0),pmnho3(8.0),' ',pmnho(8.0)
list = '-----'
list = 'Subtotal: ',pm1(8.0),pm2(8.0),pm3(8.0),' ',pm(8.0)
list = ' '
list = ' '

```

## Appendix C Cube Voyager Scripts

```

list = 'Night:      1-Occ.  2-Occ.  3+Occ.      Total'
list = 'HBW      ', ophbw1(8.0), ophbw2(8.0), ophbw3(8.0), ' ', ophbw(8.0)
list = 'HBS      ', ophbs1(8.0), ophbs2(8.0), ophbs3(8.0), ' ', ophbs(8.0)
list = 'HBO      ', ophbo1(8.0), ophbo2(8.0), ophbo3(8.0), ' ', ophbo(8.0)
list = 'NHW      ', opnhw1(8.0), opnhw2(8.0), opnhw3(8.0), ' ', opnhw(8.0)
list = 'NHO      ', opnho1(8.0), opnho2(8.0), opnho3(8.0), ' ', opnho(8.0)
list = '-----'
list = 'Subtotal: ', op1(8.0), op2(8.0), op3(8.0), ' ', op(8.0)
list = ' '
list = ' '
list = 'Input / Output Totals by Purpose:'
list = '          Diff.'
list = '          Input      Output      (O-I)'
list = 'HBW      ', ihbw(8.0), ' ', ohbw(8.0), ' ', dfhbw(8.0)
list = 'HBS      ', ihbs(8.0), ' ', ohbs(8.0), ' ', dfhbs(8.0)
list = 'HBO      ', ihbo(8.0), ' ', ohbo(8.0), ' ', dfhbo(8.0)
list = 'NHW      ', inhbw(8.0), ' ', onhw(8.0), ' ', dfnhw(8.0)
list = 'NHO      ', inho(8.0), ' ', onho(8.0), ' ', dfnho(8.0)
list = ' '
list = 'Total Auto Drv: ', adr(8.0)

list = '/et
endif

am1 =am1 +MW[611],am2 =am2 +MW[612],am3 =am3 +MW[613]
md1 =md1 +MW[614],md2 =md2 +MW[615],md3 =md3 +MW[616]
pm1 =pm1 +MW[617],pm2 =pm2 +MW[618],pm3 =pm3 +MW[619]
op1 =op1 +MW[620],op2 =op2 +MW[621],op3 =op3 +MW[622]

;; Write out the auto driver files for each time period, 3 tables in each file (1-
,2-, 3+occ)

MATO[1] = @ADRAM@, MO=611-613, ; AM peak period Auto Drv Trips 1,2,3+occ tabs 1-3
name = AM_ADRs_1,AM_ADRs_2,AM_ADRs_3, dec=3*3

MATO[2] = @ADRMD@, MO=614-616, ; Midday period Auto Drv Trips 1,2,3+occ tabs 1-3
name = MD_ADRs_1,MD_ADRs_2,MD_ADRs_3, dec=3*3

MATO[3] = @ADRPM@, MO=617-619, ; PM peak period Auto Drv Trips 1,2,3+occ tabs 1-3
name = PM_ADRs_1,PM_ADRs_2,PM_ADRs_3, dec=3*3

MATO[4] = @ADRNT@, MO=620-622, ; Night period Auto Drv Trips 1,2,3+occ tabs 1-3
name = NT_ADRs_1,NT_ADRs_2,NT_ADRs_3, dec=3*3

ENDRUN
;

```

## 41 Transit\_Accessibility.s

```

;-----
; Transit_Accessibility.s
;
; Develop transit accessibility files needed in the demographic modeling
; - the AM transit accessibility to jobs w/in 35, 40, 45, 50 min
; - Metrorail related accessibility only (BM & MR only).
;-----
;

Loop Pr_ = 1,2
  IF (PR_ =1) per = 'AM'
  IF (PR_ =2) per = 'OP'

```

```

Loop Ac_ =1,2
  IF (Ac_ =1) Acc = 'WK'
  IF (Ac_ =2) Acc = 'DR'

Loop Pth =1,2
  IF (Pth =1) Path = 'BM'
  IF (Pth =2) Path = 'MR'
  ; IF (Pth =3) Path = 'AB'
  ; IF (Pth =4) Path = 'CR'

pageheight=32767 ; Preclude header breaks
ZONESIZE = 3722
RUN PGM=MATRIX
MATI[1] =%_iter_%_@per@_@Acc@_@Path@.ttt

ZDATI[1] =INPUTS\ZONE.dbf

_ACCESS = 0
_TAZ = i

MW[100] = Mi.1.1

JLOOP

IF (MW[100] =0.0) MW[100] =1000000

IF (MW[100] =1000000)
  NotConnected = NotConnected + 1
ELSE
  Connected = Connected + 1
ENDIF

IF (MW[100] < 1000000 )
  _ACCESS = _ACCESS + MW[100]
ENDIF

ENDJLOOP

IF ( _ACCESS > 0 )
MW[100][I] = 1
ENDIF

_EMP35 = 0
_EMP40 = 0
_EMP45 = 0
_EMP50 = 0
_EMPTOT = 0

JLOOP
IF (MW[100] = 1-35)
  _EMP35 = _EMP35 + ZI.1.TOTEMP[J] ; jobs w/35 Min
ENDIF

IF (MW[100] = 1-40)
  _EMP40 = _EMP40 + ZI.1.TOTEMP[J] ; jobs w/40 Min
ENDIF

IF (MW[100] = 1-45)
  _EMP45 = _EMP45 + ZI.1.TOTEMP[J] ; jobs w/45 Min
ENDIF

IF (MW[100] = 1-50)
  _EMP50 = _EMP50 + ZI.1.TOTEMP[J] ; jobs w/50 Min
ENDIF

_EMPTOT = _EMPTOT + ZI.1.TOTEMP[J] ; total regional jobs

```



## Appendix C Cube Voyager Scripts

```

ENDJLOOP

;;; ;; Print Accessibility to jobs file

FILEO RECO[1] = "%_iter_%_per@_@Acc@_@Path@_JobAcc.dbf",
              Fields = TAZ(5), Emp35(10), Emp40(10), Emp45(10), Emp50(10),
EMPTOT(10)

              ro.TAZ      = _TAZ
              ro.emp35    = _emp35
              ro.emp40    = _emp40
              ro.emp45    = _emp45
              ro.emp50    = _emp50
              ro.emptot   = _emptot

              WRITE RECO=1 ;

;; Print out text file containing best path stats
IF (I= @ZONESIZE@)
  PRINT FILE=%_iter_%_per@_@Acc@_@Path@_JOBACC.txt, FORM=12csv, LIST=
  'Accessibility_Report: ',
  ', '%_iter%',
  ', '@per',
  ', '@Acc',
  ', '@Path',
  IJs: ' ', Connected,
  IJs: ' ', NotConnected
ENDIF

ENDRUN
ENDLOOP
ENDLOOP
ENDLOOP

```

## 42 Transit\_Assignment\_AB.s

```

-----
;Transit_Assignment_AB.s
;TPB Version 2.3 travel model on the 3,722-TAZ area system
;
;   - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
;   - iteration (_iter_) global variables used
;Assign Transit Trips by Time Period and Access Mode
; Input Files:
;   Cube Voyager Highway Network = ZONEHWY.NET
;   Transit Line Files           = MODE[1-10][AM|OP].TB
;   Transit Network Data        = MET*.TB, COM*.TB, BUS*.TB
;   Walk and Drive Access       = WALKACC.TB, *_PNR_pp.TB
;   Walk Sidewalk Network       = SIDEWALK.ASC
;   Transit Trip Tables         = '%_iter_%_AMMS.TRP', '%_iter_%_OPMS.TRP'
; Output Files:
;   Transit Assignment Link and Node Files
;
; Step 1: AM Peak Walk Assignment

```

```

; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: WKABAMnode.dbf; WKABAMlink.dbf
; Step 2: AM Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: DRABAMnode.dbf; DRABAMlink.dbf
; Step 3: AM Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: KRABAMnode.dbf; KRABAMlink.dbf
; Step 4: Off Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: WKABOPnode.dbf; WKABOPlink.dbf
; Step 5: Off Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: DRABOPnode.dbf; DRABOPlink.dbf
; Step 6: Off Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: KRABOPnode.dbf; KRABOPlink.dbf
;-----

```

```

; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

```

```
pageheight=32767 ; Preclude header breaks
```

```
;-----
; Loop through each period and access mode
;-----

```

```
LOOP PERIOD = 1, 2
```

```
IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

```

```
MATIN='%_iter_%_AMMS.TRP'
AM=' '
OP=';'

```

```
ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

```

```
MATIN='%_iter_%_OPMS.TRP'
AM=';'
OP=' '

```

```
ENDIF
```

```
;---- start the access mode loop ----
```

```
LOOP ACCESS = 1,3
```

```
IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  KR_MODEL = ';'
  TABIN = 'MI.1.2'
ELSEIF (ACCESS = 2)
  ACCESS_MODE = 'DR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ' '
  KR_MODEL = ';'
  TABIN = 'MI.1.6'

```

## Appendix C Cube Voyager Scripts

```

ELSE
  ACCESS_MODE = 'KR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ';'
  KR_MODEL = ''
  TABIN = 'MI.1.7'
ENDIF

;-----
; Step 1, 2, 3, 4, 5 & 6 Assign All Bus Transit Trips
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATI = @MATIN@
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

;--- set default zone access and line parameters ---

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11
@KR_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERRUNTIME = Y

;---- rules for combining multiple line and headways ----
COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----
MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----
IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----
XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50
XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, Y

;---- Parameters ----
LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with perceived time > 240 min.
FREPERIOD = 1 ;--- Use the First Headway value
USERRUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25
;-----
; write out support links for later viewing in VIPER

```

## Appendix C Cube Voyager Scripts

```
fileo supporto = supl_AB_@access_mode@_@time_period@.asc modes=11-16 oneway=t
fixed=y

;---- Rail Stations & Links (modes 3 & 4) ----

;READ FILE = met_node.tb ;---- Metrorail stations
;READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
;READ FILE = lrt_node.tb ;---- LRT stations
;READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model10 Stations
READ FILE = new_link.tb ;---- Model10 links
;---- Park and Ride Lots (mode 15) ----

@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model10 PNR lots (nodes)

@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@.tb ;---- Commuter Rail-PNR connectors
(links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

;READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
;READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model10

;@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors (links)
@KR_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
;READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
```

```
;READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

; output files
@WALK_MODEL@@AM@FILEO NODEO = %_iter_%_WKABAMnode.dbf ; output node file
@WALK_MODEL@@OP@FILEO NODEO = %_iter_%_WKABOPnode.dbf ; output node file
@DRIVE_MODEL@@AM@FILEO NODEO = %_iter_%_DRABAMnode.dbf ; output node file
@DRIVE_MODEL@@OP@FILEO NODEO = %_iter_%_DRABOPnode.dbf ; output node file
@KR_MODEL@@AM@FILEO NODEO = %_iter_%_KRABAMnode.dbf ; output node file
@KR_MODEL@@OP@FILEO NODEO = %_iter_%_KRABOPnode.dbf ; output node file

@WALK_MODEL@@AM@FILEO LINKO = %_iter_%_WKABAMlink.dbf ; output link file
@WALK_MODEL@@OP@FILEO LINKO = %_iter_%_WKABOPlink.dbf ; output link file
@DRIVE_MODEL@@AM@FILEO LINKO = %_iter_%_DRABAMlink.dbf ; output link file
@DRIVE_MODEL@@OP@FILEO LINKO = %_iter_%_DRABOPlink.dbf ; output link file
@KR_MODEL@@AM@FILEO LINKO = %_iter_%_KRABAMlink.dbf ; output link file
@KR_MODEL@@OP@FILEO LINKO = %_iter_%_KRABOPlink.dbf ; output link file

TRIPS MATRIX=@TABIN@, ASSIGN=Y, VOLUMES=Y, BOARDS=Y, EXITS=Y
REPORT LINKVOL=Y,LINEVOL=Y

ENDRUN

ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----
```

## 43 Transit\_Assignment\_BM.s

```
-----
;Transit_Assignment_BM.s
;TPB Version 2.3 travel model on the 3,722-TAZ area system
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Assign Transit Trips by Time Period and Access Mode
; Input Files:
; Cube Voyager Highway Network = ZONEHWY.NET
; Transit Line Files = MODE[1-10][AM|OP].TB
; Transit Network Data = MET*.TB, COM*.TB, BUS*.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Transit Trip Tables = '%_iter_%_AMMS.TRP', '%_iter_%_OPMS.TRP'
; Output Files:
; Transit Assignment Link and Node Files
;
; Step 1: AM Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: WKBAMnode.dbf; WKBAMlink.dbf
; Step 2: AM Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: DRBAMnode.dbf; DRBAMlink.dbf
; Step 3: AM Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: KRBAMnode.dbf; KRBAMlink.dbf
; Step 4: Off Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: WKBOPnode.dbf; WKBOPlink.dbf
; Step 5: Off Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: DRBOPnode.dbf; DRBOPlink.dbf
; Step 6: Off Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
```

## Appendix C Cube Voyager Scripts

```

; Output Files: KRBMOPlnode.dbf; KRBMOPlink.dbf
;-----
; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

pageheight=32767 ; Preclude header breaks

;-----
; Loop through each period and access mode
;-----

LOOP PERIOD = 1, 2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

  MATIN='%_iter_%_OPMS.TRP'
  AM=';'
  OP=' '
ENDIF

;---- start the access mode loop ----

LOOP ACCESS = 1,3

  IF (ACCESS = 1)
    ACCESS_MODE = 'WK'
    WALK_MODEL = ' '
    DRIVE_MODEL = ';'
    KR_MODEL = ';'
    TABIN = 'MI.1.3'
  ELSEIF (ACCESS = 2)
    ACCESS_MODE = 'DR'
    WALK_MODEL = ';'
    DRIVE_MODEL = ' '
    KR_MODEL = ';'
    TABIN = 'MI.1.8'
  ELSE
    ACCESS_MODE = 'KR'
    WALK_MODEL = ' ';
    DRIVE_MODEL = ' ';
    KR_MODEL = ' '
    TABIN = 'MI.1.9'
  ENDIF

;-----
; Step 1, 2, 3 , 4, 5 & 6 Assign Bus/MR Transit Trips
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATI = @MATIN@
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

;-- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

```

```

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11
@KR_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERRUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50
XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50

```

## Appendix C Cube Voyager Scripts

```

XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n
NOX[12] = n, n, n, n, n, n, n, n, n, Y, Y, n, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREQPERIOD = 1 ;--- Use the First Headway value
USERRUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_BM_access_mode@_time_period@.asc modes=11-16 oneway=
fixed=y
;

;---- Rail Stations & Links (modes 3 & 4) ----

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model10 Stations
READ FILE = new_link.tb ;---- Model10 links

;---- Park and Ride Lots (mode 15) ----

@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model10 PNR lots (nodes)

```

```

@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@.tb ;---- Commuter Rail-PNR connectors
(links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model10

@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors (links)
@KR_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODEL1@TIME_PERIOD@.TB ;---- M1- metrobus local
READ FILE = MODEL2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODEL3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODEL4@TIME_PERIOD@.TB ;---- M4- commuter rail
READ FILE = MODEL5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODEL6@TIME_PERIOD@.TB ;---- M6- other local bus
READ FILE = MODEL7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODEL8@TIME_PERIOD@.TB ;---- M8- other local bus
READ FILE = MODEL9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODEL10@TIME_PERIOD@.TB ;---- M10- other bus (future)

; output files
@WALK_MODEL@@AM@FILEO NODEO = %_iter_% WKBMAMnode.dbf ; output node file
@WALK_MODEL@@OP@FILEO NODEO = %_iter_% WKBMOPnode.dbf ; output node file
@DRIVE_MODEL@@AM@FILEO NODEO = %_iter_% DRBMAMnode.dbf ; output node file
@DRIVE_MODEL@@OP@FILEO NODEO = %_iter_% DRBMOPnode.dbf ; output node file
@KR_MODEL@@AM@FILEO NODEO = %_iter_% KRBAMnode.dbf ; output node file
@KR_MODEL@@OP@FILEO NODEO = %_iter_% KRBMOPnode.dbf ; output node file ; Added
"O" to filename

@WALK_MODEL@@AM@FILEO LINKO = %_iter_% WKBMAMlink.dbf ; output link file
@WALK_MODEL@@OP@FILEO LINKO = %_iter_% WKBMOPlink.dbf ; output link file
@DRIVE_MODEL@@AM@FILEO LINKO = %_iter_% DRBMAMlink.dbf ; output link file
@DRIVE_MODEL@@OP@FILEO LINKO = %_iter_% DRBMOPlink.dbf ; output link file
@KR_MODEL@@AM@FILEO LINKO = %_iter_% KRBAMlink.dbf ; output link file
@KR_MODEL@@OP@FILEO LINKO = %_iter_% KRBMOPlink.dbf ; output link file

```

## Appendix C Cube Voyager Scripts

```
TRIPS MATRIX=@TABIN@, ASSIGN=Y, VOLUMES=Y, BOARDS=Y, EXITS=Y
REPORT LINKVOL=Y,LINEVOL=Y
```

```
ENDRUN
```

```
ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----
```

## 44 Transit\_Assignment\_CR.s

```
-----
;Transit_Assignment_CR.s
;TPB Version 2.3 travel model on the 3,722-TAZ area system
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration ( _iter_ ) global variables used
;Assign Transit Trips by Time Period and Access Mode
; Input Files:
; Cube Voyager Highway Network = ZONEHWY.NET
; Transit Line Files           = MODE[1-10][AM|OP].TB
; Transit Network Data         = MET_*.TB, COM_*.TB, BUS_*.TB
; Walk and Drive Access        = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network        = SIDEWALK.ASC
; Transit Trip Tables          = '%_iter_%_AMMS.TRP', '%_iter_%_OPMS.TRP'
; Output Files:
; Transit Assignment Link and Node Files
;
; Step 1: AM Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: WKCRAmnode.dbf; WKCRAmlink.dbf
; Step 2: AM Peak Drive, K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: DRCRAmnode.dbf; DRCRAmlink.dbf
; Step 3: Off Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: WKCROpnode.dbf; WKCROPlink.dbf
; Step 4: Off Peak Drive, K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: DRCROpnode.dbf; DRCROPlink.dbf
-----

; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

pageheight=32767 ; Preclude header breaks

-----
; Loop through each period and access mode
-----

LOOP PERIOD = 1, 2

IF (PERIOD = 1)
  TIME PERIOD = 'AM'
  COMBINE = 5.0

  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
ELSE
  TIME PERIOD = 'OP'
  COMBINE = 10.0

  MATIN='%_iter_%_OPMS.TRP'
  AM=';'

```

```
OP=' '
ENDIF

;---- start the access mode loop ----

LOOP ACCESS = 1, 2

IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  TABIN = 'MI.1.1'
ELSE
  ACCESS_MODE = 'DR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ' '
  TABIN = 'MI.1.5'
ENDIF

-----
; Step 1, 2, 3 & 4 Assign CR Transit Trips
-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATI = @MATIN@
maxnode = 60000

HWYTIME = @TIME_PERIOD@HRTIME

;--- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
```

## Appendix C Cube Voyager Scripts

```

XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50
XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, Y, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREQPERIOD = 1 ;--- Use the First Headway value
USERUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.

```

```

;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_CR_@access_mode@_@time_period@.asc modes=11-16 oneway=t
fixed=y
;

;---- Rail Stations & Links (modes 3 & 4) ----

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
READ FILE = com_node.tb ;---- Commuter Rail stations
READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model10 Stations
READ FILE = new_link.tb ;---- Model10 links
;---- Park and Ride Lots (mode 15) ----

;@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model10 PNR lots (nodes)

;@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@_pnr.tb ;---- Commuter Rail-PNR
connectors (links)
;@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

;@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
;@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail

```

## Appendix C Cube Voyager Scripts

```

READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

; output files
@WALK_MODEL@AM@FILEO NODEO = %_iter_% WKCGRAMnode.dbf ; output node file
@WALK_MODEL@OP@FILEO NODEO = %_iter_% WKCROPNODE.dbf ; output node file
@DRIVE_MODEL@AM@FILEO NODEO = %_iter_% DRCRAMnode.dbf ; output node file
@DRIVE_MODEL@OP@FILEO NODEO = %_iter_% DRCROPNODE.dbf ; output node file

@WALK_MODEL@AM@FILEO LINKO = %_iter_% WKCGRAMlink.dbf ; output link file
@WALK_MODEL@OP@FILEO LINKO = %_iter_% WKCROPLINK.dbf ; output link file
@DRIVE_MODEL@AM@FILEO LINKO = %_iter_% DRCRAMlink.dbf ; output link file
@DRIVE_MODEL@OP@FILEO LINKO = %_iter_% DRCROPLINK.dbf ; output link file

TRIPS MATRIX=@TABIN@, ASSIGN=Y, VOLUMES=Y, BOARDS=Y, EXITS=Y
REPORT LINKVOL=Y,LINEVOL=Y

ENDRUN

ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----

```

## 45 Transit\_Assignment\_MR.s

```

;-----
;Transit_Assignment_MR.s
;TPB Version 2.3 travel model on the 3,722-TAZ area system
;
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Assign Transit Trips by Time Period and Access Mode
; Input Files:
; Cube Voyager Highway Network = ZONEHWY.NET
; Transit Line Files = MODE[1-10][AM|OP].TBe
; Transit Network Data = MET*.TB, COM*.TB, BUS*.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Transit Trip Tables = '%_iter_%_AMMS.TRP', '%_iter_%_OPMS.TRP'
; Output Files:
; Transit Assignment Link and Node Files
;
; Step 1: AM Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: WKMRAMnode.dbf; WKMRAMlink.dbf
; Step 2: AM Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: DRMRAMnode.dbf; DRMRAMlink.dbf
; Step 3: AM Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
; Output Files: KRMRAMnode.dbf; KRMRAMlink.dbf
; Step 4: Off Peak Walk Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: WKMROPnode.dbf; WKMROPLink.dbf
; Step 5: Off Peak Drive Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: DRMROPnode.dbf; DRMROPLink.dbf
; Step 6: Off Peak K/R Assignment
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
; Output Files: KRMROPnode.dbf; KRMROPLink.dbf
;-----

```

```

; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

pageheight=32767 ; Preclude header breaks

;-----
; Loop through each period and access mode
;-----

LOOP PERIOD=1,2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

  MATIN='%_iter_%_OPMS.TRP'
  AM=';'
  OP=' '
ENDIF

;---- start the access mode loop ----

LOOP ACCESS=1,3

IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  KR_MODEL = ';'
  TABIN = 'MI.1.4'
ELSEIF (ACCESS = 2)
  ACCESS_MODE = 'DR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ' '
  KR_MODEL = ';'
  TABIN = 'MI.1.10'
ELSE
  ACCESS_MODE = 'KR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ';'
  KR_MODEL = ' '
  TABIN = 'MI.1.11'
ENDIF

;-----
; Step 1, 2, 3, 4, 5 & 6 Assign MR Transit Trips
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATI = @MATIN@
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

;--- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11

```



## Appendix C Cube Voyager Scripts

```

@KRW_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERRUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3] = 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12] = 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15] = 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16] = 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1] = 16*2.50
XPENFAC[2] = 16*2.50
XPENFAC[3] = 16*2.50
XPENFAC[4] = 16*2.50
XPENFAC[5] = 16*2.50
XPENFAC[6] = 16*2.50
XPENFAC[7] = 16*2.50
XPENFAC[8] = 16*2.50
XPENFAC[9] = 16*2.50
XPENFAC[10] = 16*2.50
XPENFAC[11] = 16*2.50
XPENFAC[12] = 16*2.50

XPENFAC[13] = 16*2.50
XPENFAC[14] = 16*2.50
XPENFAC[15] = 16*2.50
XPENFAC[16] = 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, n, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, n, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with perceived time > 240 min.
FREPERIOD = 1 ;--- Use the First Headway value
USERRUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_MR_@access_mode@_@time_period@.asc modes=11-16 oneway=t
fixed=y
;

;---- Rail Stations & Links (modes 3 & 4) ----

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
;READ FILE = new_node.tb ;---- Mode10 Stations
;READ FILE = new_link.tb ;---- Mode10 links
;---- Park and Ride Lots (mode 15) ----

;@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Mode10 PNR lots (nodes)

;@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors
(links)

```

## Appendix C Cube Voyager Scripts

```
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@.tb ;---- Commuter Rail-PNR connectors
(links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
;@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
;READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
;KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
;@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model10

@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

;READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
;READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
;READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
;READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

; output files
@WALK_MODEL@@@AM@FILEO NODEO = %_iter_%_WKMRAMnode.dbf ; output node file
@WALK_MODEL@@@OP@FILEO NODEO = %_iter_%_WKMRONode.dbf ; output node file
@DRIVE_MODEL@@@AM@FILEO NODEO = %_iter_%_DRMRAMnode.dbf ; output node file
@DRIVE_MODEL@@@OP@FILEO NODEO = %_iter_%_DRMRONode.dbf ; output node file
@KR_MODEL@@@AM@FILEO NODEO = %_iter_%_KMRAMnode.dbf ; output node file
@KR_MODEL@@@OP@FILEO NODEO = %_iter_%_KMRONode.dbf ; output node file

@WALK_MODEL@@@AM@FILEO LINKO = %_iter_%_WKMRAMlink.dbf ; output link file
@WALK_MODEL@@@OP@FILEO LINKO = %_iter_%_WKMRONlink.dbf ; output link file
@DRIVE_MODEL@@@AM@FILEO LINKO = %_iter_%_DRMRAMlink.dbf ; output link file
@DRIVE_MODEL@@@OP@FILEO LINKO = %_iter_%_DRMROlink.dbf ; output link file
@KR_MODEL@@@AM@FILEO LINKO = %_iter_%_KMRAMlink.dbf ; output link file
@KR_MODEL@@@OP@FILEO LINKO = %_iter_%_KMRONlink.dbf ; output link file

TRIPS MATRIX=@TABIN@, ASSIGN=Y, VOLUMES=Y, BOARDS=Y, EXITS=Y
REPORT LINKVOL=Y,LINEVOL=Y
```

ENDRUN

```
ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----
```

## 46 Transit\_Skims\_AB.s

```
-----
;Transit_Skims_AB.s
;MWCOCG Version 2.2 Model
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Build Transit Skims by Time Period and Access Mode
; Input Files:
; TP+ Highway Network = ZONEHWY.NET
; Transit Line Files = MODE?_pp.TB
; Transit Network Data = MET_*.TB, COM_*.TB, BUS_*.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Output Files:
; Walk and Drive Access Skims = pp_aa_mo.SKM
; Walk and Drive Station Data = pp_aa_mo.STA
;
; Step 1: AM Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 2: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_AM_WK_AB.SKM, %_iter_%_AM_WK_AB.STA,
; Step 3: AM Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 4: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_AM_DR_AB.SKM, %_iter_%_AM_DR_AB.STA,
; Step 5: AM Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 6: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_AM_KR_AB.SKM, %_iter_%_AM_KR_AB.STA,
; Step 7: Off Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 8: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_OP_WK_AB.SKM, %_iter_%_OP_WK_AB.STA,
; Step 9: Off Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 10: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_OP_DR_AB.SKM, %_iter_%_OP_DR_AB.STA,
; Step 11: Off Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.ab.skm
; Step 12: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.ab.skm
; Output Files: %_iter_%_OP_KR_AB.SKM, %_iter_%_OP_KR_AB.STA,
;
;-----
; Added Mode 15 access links for KNR to Bus path, JainM 09.19.06
; @KR_MODEL@ READ FILE = bus@TIME_PERIOD@pnr.tb ;---- Bus-PNR connectors (links)

; rm 4/7/08 ;
```

## Appendix C Cube Voyager Scripts

```

; Added table #19 (Total Transit time in min.) to output transit.temp.ab.skm file
; create total transit time skims named:
;   %_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_@AB.ttt

; 2010-10-22 Previously, only bus PNR links were built to bus PNR & bus KNR paths.
;           Now, we have created bus KNR access links from TAZ to bus stop node,
;           instead of TAZ to PNR node (rjm/msm)

;-----
; useIdp = t (true) or f (false); this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=%useMdp%
;
;-----
;           Loop through each period and access mode
;-----
pageheight=32767 ; Preclude header breaks
;
; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

LOOP PERIOD = 1, 2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

ENDIF

;---- start the access mode loop ----

LOOP ACCESS = 1,3

IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  KR_MODEL = ';'
ELSEIF (ACCESS = 2)
  ACCESS_MODE = 'DR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ' '
  KR_MODEL = ';'
ELSE
  ACCESS_MODE = 'KR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ';'
  KR_MODEL = ' '
ENDIF

;-----
; Step 1, 3, 5, 7, 9 & 11 Build Transit Path
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATO = transit.temp.ab.skm
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

```

```

;--- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11
@KR_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50

```

## Appendix C Cube Voyager Scripts

```

XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, n, Y, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREQPERIOD = 1 ;--- Use the First Headway value
USERUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25
;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_@time_period@_@access_mode@_ab.asc modes=11-16 oneway=t
fixed=y
fileo nodeo = supn_@time_period@_@access_mode@_ab.dbf
fileo linko = trnl_@time_period@_@access_mode@_ab.dbf ; Can be used to create
transit shapefile

;---- specify output skims ----

MATRICES NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT,
XWAIT, WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D,
PNR_IMP, PNR_CST, I_STA, J_STA,
MW[1] = TIME(1,6,8), ;---- ivt-local bus (0.01 min)
MW[2] = TIME(2,7,9), ;---- ivt-exp bus (0.01 min)
MW[3] = TIME(3), ;---- ivt-metrorail (0.01 min)
MW[4] = TIME(4), ;---- ivt-commuter rail(0.01 min)
MW[5] = TIME(5), ;---- ivt-new rail mode(0.01 min)
MW[6] = TIME(10), ;---- ivt-new bus mode (0.01 min)
MW[7] = IWAIT, ;---- ini.wait time (0.01 min)
MW[8] = XWAIT(1,2,3,4,5,6,7,8,9,10), ;---- xfr wait time (0.01 min)

```

```

MW[9] = TIME(14,16), ;---- walk acc time (0.01 min)
MW[10] = TIME(12,13), ;---- other walk time (0.01 min)
MW[11] = XPEN, ;---- added xfer time (0.01 min)
MW[12] = BOARDS, ;---- boardings (1+)
MW[13] = TIME(11), ;---- drv acc time (0.01 min)
MW[14] = DIST(11), ;---- drv acc distance (0.01 mile)
MW[15] = TIME(15), ;---- pnr impedance (0.01 min)
MW[16] = DIST(15), ;---- pnr cost (cents)
MW[17] = NODE0(3) - 8000.0, ;---- metro board sta (1-150)
MW[18] = NODEL(3) - 8000.0 ;---- metro alight sta (1-150)

;---- Rail Stations & Links (modes 3 & 4) ----

;READ FILE = met_node.tb ;---- Metrorail stations
;READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
;READ FILE = lrt_node.tb ;---- LRT stations
;READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model10 Stations
READ FILE = new_link.tb ;---- Model10 links
;---- Park and Ride Lots (mode 15) ----

@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model10 PNR lots (nodes)

@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@_pnr.tb ;---- Commuter Rail-PNR
connectors (links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

;READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
;READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model10

;@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors (links)
@KR_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

```

## Appendix C Cube Voyager Scripts

```

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
;READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
;READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

/* Transit path traces for select i/j pairs */
read file = ..\scripts\pathTrace.s

ENDRUN
;-----
;Step 2, 4, 6 & 8 Condition & Split Skims into Multiple Files
;-----
RUN PGM=MATRIX
; If we keep IDP here, we will need 16 cores, so we have commented it out
;@dtp_token@DistributeIntrastep processId='mwcog', ProcessList=%subnode%
MATI[1]=transit.temp.ab.skm
MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.SKM, MO = 1-16,
NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_LRT, IVT_BRT, IWAIT, XWAIT,
WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D, PNR_IMP, PNR_CST

MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.STA, MO = 17-18,
NAME = I_STA, J_STA

MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.ttt, MO = 100,
NAME = sumtrntm

MW[1] = MI.1.1 ;---- ivt-local bus (0.01 min)
MW[2] = MI.1.2 ;---- ivt-exp bus (0.01 min)
MW[3] = MI.1.3 ;---- ivt-metrorail (0.01 min)
MW[4] = MI.1.4 ;---- ivt-commuter rail(0.01 min)
MW[5] = MI.1.5 ;---- ivt-new rail mode(0.01 min)
MW[6] = MI.1.6 ;---- ivt-new bus mode (0.01 min)
MW[7] = MI.1.7 ;---- ini.wait time (0.01 min)
MW[8] = MI.1.8 ;---- xfr wait time (0.01 min)
MW[9] = MI.1.9 ;---- walk acc time (0.01 min)
MW[10] = MI.1.10 ;---- other walk time (0.01 min)
MW[11] = MI.1.11 ;---- added xfer time (0.01 min)
MW[12] = MI.1.12 ;---- transfers (0+)
MW[13] = MI.1.13 ;---- drv acc time (0.01 min)
MW[14] = MI.1.14 ;---- drv acc distance (0.01 mile)
MW[15] = MI.1.15 ;---- pnr time (0.01 min)
MW[16] = MI.1.16 ;---- pnr cost (cents)

MW[17] = MI.1.17 ;---- metro board sta (1-150)
MW[18] = MI.1.18 ;---- metro alight sta (1-150)

;4
;
;

JLOOP
MW[12] = MW[12] - 1
IF (MW[16] = 1 ) MW[16] = 0
MW[15] = MW[15] - MW[16] * 6.0
IF (MW[17] < 0 || MW[17] > 150 ) MW[17] = 0
IF (MW[18] < 0 || MW[18] > 150 ) MW[18] = 0
ENDJLOOP

MW[100] = (MW[1] + MW[2] + MW[3] + MW[4] + MW[5] +
MW[6] + MW[7] + MW[8] + MW[9] + MW[10] +

```

```

MW[11] + MW[13]) * 0.01 ;; Total Real Transit Time in Whole Minutes
(not incl. PNR 'impedance')

```

```

ENDRUN
ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----

```

## 47 Transit\_Skims\_BM.s

```

;-----
;Transit Skims BM.s
;MWCOCG Version 2.2 Model
;
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Build Transit Skims by Time Period and Access Mode
; Input Files:
; TP+ Highway Network = ZONEHWY.NET
; Transit Line Files = MODE?_pp.TB
; Transit Network Data = MET_*.TB, COM_*.TB, BUS_*.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Output Files:
; Walk and Drive Access Skims = pp_aa_mo.SKM
; Walk and Drive Station Data = pp_aa_mo.STA
;
; Step 1: AM Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 2: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_AM_WK_BM.SKM, %_iter_%_AM_WK_BM.STA,
; Step 3: AM Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 4: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_AM_DR_BM.SKM, %_iter_%_AM_DR_BM.STA,
; Step 5: AM Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 6: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_AM_KR_BM.SKM, %_iter_%_AM_KR_BM.STA,
; Step 7: Off Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 8: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_OP_WK_BM.SKM, %_iter_%_OP_WK_BM.STA,
; Step 9: Off Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 10: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_OP_DR_BM.SKM, %_iter_%_OP_DR_BM.STA
; Step 11: Off Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.bm.skm
; Step 12: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.bm.skm
; Output Files: %_iter_%_OP_KR_BM.SKM, %_iter_%_OP_KR_BM.STA,
;

```

## Appendix C Cube Voyager Scripts

```

;-----
; Added Mode 15 access links for KNR to Bus path, JainM 09.19.06
; @KR_MODEL@ READ FILE = bus@TIME_PERIOD@pnr.tb ;---- Bus-PNR connectors (links)

; rm 4/7/08
; Added table #19 (Total Transit time in min.) to output transit.temp.bm.skm file
; create total transit time skims named:
; %_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.ttt

; 2010-10-22 Previously, only bus PNR links were built to bus PNR & bus KNR paths.
; Now, we have created bus KNR access links from TAZ to bus stop node,
; instead of TAZ to PNR node (rjm/msm)

;-----
; useIdp = t (true) or f (false); this is set in the wrapper batch file
; distribute intrastep=%useIdp% multistep=%useMdp%
;
; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

;-----
; Loop through each period and access mode
;-----
pageheight=32767 ; Preclude header breaks
LOOP PERIOD = 1, 2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

ENDIF

;---- start the access mode loop ----
LOOP ACCESS = 1,3

IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  KR_MODEL = ';'
ELSEIF (ACCESS = 2)
  ACCESS_MODE = 'DR'
  WALK_MODEL = ' '
  DRIVE_MODEL = ' '
  KR_MODEL = ';'
ELSE
  ACCESS_MODE = 'KR'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
  KR_MODEL = ' '
ENDIF

;-----
; Step 1, 3, 5, 7, 9 & 11 Build Transit Path
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATO = transit.temp.bm.skm
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

```

```

;--- set default zone access and line parameters ----
ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11
@KR_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50

```

## Appendix C Cube Voyager Scripts

```

XPENFAC[6]= 16*2.50
XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, n, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, n, n, n, n, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREPPERIOD = 1 ;--- Use the First Headway value
USERUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_@time_period@_@access_mode@_bm.asc modes=11-16 oneway=t
fixed=y
fileo nodeo = supn_@time_period@_@access_mode@_bm.dbf
fileo linko = trnl_@time_period@_@access_mode@_bm.dbf ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT,
XWAIT, WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D,
PNR_IMP, PNR_CST, I_STA, J_STA,

MW[1] = TIME(1,6,8), ;---- ivt-local bus (0.01 min)
MW[2] = TIME(2,7,9), ;---- ivt-exp bus (0.01 min)
MW[3] = TIME(3), ;---- ivt-metrorail (0.01 min)
MW[4] = TIME(4), ;---- ivt-commuter rail(0.01 min)
MW[5] = TIME(5), ;---- ivt-new rail mode(0.01 min)
MW[6] = TIME(10), ;---- ivt-new bus mode (0.01 min)
MW[7] = IWAIT, ;---- ini.wait time (0.01 min)

```

```

MW[8] = XWAIT(1,2,3,4,5,6,7,8,9,10), ;---- xfr wait time (0.01 min)
MW[9] = TIME(14,16), ;---- walk acc time (0.01 min)
MW[10] = TIME(12,13), ;---- other walk time (0.01 min)
MW[11] = XPEN, ;---- added xfer time (0.01 min)
MW[12] = BOARDS, ;---- boardings (1+)
MW[13] = TIME(11), ;---- drv acc time (0.01 min)
MW[14] = DIST(11), ;---- drv acc distance (0.01 mile)
MW[15] = TIME(15), ;---- pnr impedance (0.01 min)
MW[16] = DIST(15), ;---- pnr cost (cents)
MW[17] = NODE0(3) - 8000.0, ;---- metro board sta (1-150)
MW[18] = NODEL(3) - 8000.0 ;---- metro alight sta (1-150)

```

;---- Rail Stations & Links (modes 3 & 4) ----

```

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model10 Stations
READ FILE = new_link.tb ;---- Model10 links

```

;---- Park and Ride Lots (mode 15) ----

```

@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model10 PNR lots (nodes)

```

```

@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@_pnr.tb ;---- Commuter Rail-PNR
connectors (links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors
(links)

```

;---- Access Links (modes 11, 12 and 16) ----

```

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model10 bus-LRT links&xfer car

```

READ FILE = walkacc.asc ;--- walk to local transit

```

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model10

```

```

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model10

```

```

@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model10-PNR connectors (links)
@KR_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors (links)

```

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

## Appendix C Cube Voyager Scripts

```

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

/* Transit path traces for select i/j pairs */
read file = ..\scripts\pathTrace.s

ENDRUN
;-----
;Step 2, 4, 6 & 8 Condition & Split Skims into Multiple Files
;-----
RUN PGM=MATRIX
; If we keep IDP here, we will need 16 cores, so we have commented it out
;@dp_token@distributedintrastep processId='mwcoq', ProcessList=%subnode%
MATI[1]=transit.temp.bm.skm
MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.SKM, MO = 1-16,
NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT, XWAIT,
WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D, PNR_IMP, PNR_CST

MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.STA, MO = 17-18,
NAME = I_STA, J_STA

MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.ttt, MO = 100,
NAME = sumtrntm

MW[1] = MI.1.1 ;---- ivt-local bus (0.01 min)
MW[2] = MI.1.2 ;---- ivt-exp bus (0.01 min)
MW[3] = MI.1.3 ;---- ivt-metrorail (0.01 min)
MW[4] = MI.1.4 ;---- ivt-commuter rail(0.01 min)
MW[5] = MI.1.5 ;---- ivt-new rail mode(0.01 min)
MW[6] = MI.1.6 ;---- ivt-new bus mode (0.01 min)
MW[7] = MI.1.7 ;---- ini.wait time (0.01 min)
MW[8] = MI.1.8 ;---- xfr wait time (0.01 min)
MW[9] = MI.1.9 ;---- walk acc time (0.01 min)
MW[10] = MI.1.10 ;---- other walk time (0.01 min)
MW[11] = MI.1.11 ;---- added xfer time (0.01 min)
MW[12] = MI.1.12 ;---- transfers (0+)
MW[13] = MI.1.13 ;---- drv acc time (0.01 min)
MW[14] = MI.1.14 ;---- drv acc distance (0.01 mile)
MW[15] = MI.1.15 ;---- pnr time (0.01 min)
MW[16] = MI.1.16 ;---- pnr cost (cents)

MW[17] = MI.1.17 ;---- metro board sta (1-150)
MW[18] = MI.1.18 ;---- metro alight sta (1-150)

JLOOP
; If the shortest path found in path building, which builds paths for all bus
; and Metro modes, has a Metro component = 0 or a bus component = 0,
; then this path will not be considered in the BM category
; and it will be equal to zero for this specific category,
; but will be included in AB or MR category.
IF ((MW[1] + MW[2] + MW[6] = 0 ) || (MW[3]+MW[5]=0))
MW[1] = 0
MW[2] = 0
MW[3] = 0

```

```

MW[4] = 0
MW[5] = 0
MW[6] = 0
MW[7] = 0
MW[8] = 0
MW[9] = 0
MW[10] = 0
MW[11] = 0
MW[12] = 0
MW[13] = 0
MW[14] = 0
MW[15] = 0
MW[16] = 0
MW[17] = 0
MW[18] = 0
ELSE
MW[12] = MW[12] - 1
IF (MW[16] = 1 ) MW[16] = 0
MW[15] = MW[15] - MW[16] * 6.0
IF (MW[17] < 0 || MW[17] > 150 ) MW[17] = 0
IF (MW[18] < 0 || MW[18] > 150 ) MW[18] = 0
ENDIF
ENDJLOOP

MW[100] =(MW[1] + MW[2] + MW[3] + MW[4] + MW[5] +
MW[6] + MW[7] + MW[8] + MW[9] + MW[10] +
MW[11] + MW[13]) * 0.01 ;; Total Real Transit Time in Whole Minutes
(not incl. PNR 'impedance')

ENDRUN

ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----

;-----
;Transit_Skims_CR.s
;MWCOG Version 2.2 Model
;
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Build Transit Skims by Time Period and Access Mode
; Input Files:
; TP+ Highway Network = ZONEHWY.NET
; Transit Line Files = MODE?_pp.TB
; Transit Network Data = MET *.TB, COM *.TB, BUS *.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Output Files:
; Walk and Drive Access Skims = pp_aa_mo.SKM
; Walk and Drive Station Data = pp_aa_mo.STA
;
; Step 1: AM Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.cr.skm
; Step 2: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.cr.skm
; Output Files: %_iter_%_AM_WK_CR.SKM, %_iter_%_AM_WK_CR.STA,
; Step 3: AM Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.cr.skm
; Step 4: Condition & Split Skims into Multiple Files

```

## 48 Transit\_Skims\_CR.s



## Appendix C Cube Voyager Scripts

```

; Input Files: transit.temp.cr.skm
; Output Files: %_iter_%_AM_DR_CR.SKM, %_iter_%_AM_DR_CR.STA,
; Step 5: Off Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.cr.skm
; Step 6: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.cr.skm
; Output Files: %_iter_%_OP_WK_CR.SKM, %_iter_%_OP_WK_CR.STA,
; Step 7: Off Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.cr.skm
; Step 8: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.cr.skm
; Output Files: %_iter_%_OP_DR_CR.SKM, %_iter_%_OP_DR_CR.STA
;
;-----
; rm 4/7/08 ;
; Added table #19 (Total Transit time in min.) to output transit.temp.cr.skm file
; create total transit time skims named:
; %_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.ttt
;
; useIdp = t (true) or f (false); this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=%useMdp%
;
; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion

;-----
; Loop through each period and access mode
;-----
pageheight=32767 ; Preclude header breaks
LOOP PERIOD = 1, 2

IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0

ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0

ENDIF

;---- start the access mode loop ----
LOOP ACCESS = 1, 2

IF (ACCESS = 1)
  ACCESS_MODE = 'WK'
  WALK_MODEL = ' '
  DRIVE_MODEL = ';'
ELSE
  ACCESS_MODE = 'DR'
  WALK_MODEL = ';'
  DRIVE_MODEL = ' '
ENDIF

;-----
; Step 1, 3, 5 & 7 Build Transit Path
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET
MATO = transit.temp.cr.skm
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

```

```

;--- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50
XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50

```

## Appendix C Cube Voyager Scripts

```

XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[12] = n, n, n, n, n, n, n, n, n, n, Y, Y, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, n, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, n, n, Y, n, n, n, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREQPERIOD = 1 ;--- Use the First Headway value
USERUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_@time_period@_@access_mode@_cr.asc modes=11-16 oneway=t
fixed=y
fileo nodeo = supn_@time_period@_@access_mode@_cr.dbf
fileo linko = trnl_@time_period@_@access_mode@_cr.dbf ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT,
XWAIT, WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D,
PNR_IMP, PNR_CST, I_STA, J_STA,
MW[1] = TIME(1,6,8), ;---- ivt-local bus (0.01 min)
MW[2] = TIME(2,7,9), ;---- ivt-exp bus (0.01 min)
MW[3] = TIME(3), ;---- ivt-metrorail (0.01 min)
MW[4] = TIME(4), ;---- ivt-commuter rail(0.01 min)
MW[5] = TIME(5), ;---- ivt-new rail mode(0.01 min)
MW[6] = TIME(10), ;---- ivt-new bus mode (0.01 min)
MW[7] = IWAIT, ;---- ini.wait time (0.01 min)
MW[8] = XWAIT(1,2,3,4,5,6,7,8,9,10), ;---- xfr wait time (0.01 min)

```

```

MW[9] = TIME(14,16), ;---- walk acc time (0.01 min)
MW[10] = TIME(12,13), ;---- other walk time (0.01 min)
MW[11] = XPEN, ;---- added xfer time (0.01 min)
MW[12] = BOARDS, ;---- boardings (1+)
MW[13] = TIME(11), ;---- drv acc time (0.01 min)
MW[14] = DIST(11), ;---- drv acc distance (0.01 mile)
MW[15] = TIME(15), ;---- pnr impedance (0.01 min)
MW[16] = DIST(15), ;---- pnr cost (cents)
MW[17] = NODE0(3) - 8000.0, ;---- metro board sta (1-150)
MW[18] = NODEL(3) - 8000.0 ;---- metro alight sta (1-150)

;---- Rail Stations & Links (modes 3 & 4) ----

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
READ FILE = com_node.tb ;---- Commuter Rail stations
READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
READ FILE = new_node.tb ;---- Model0 Stations
READ FILE = new_link.tb ;---- Model0 links
;---- Park and Ride Lots (mode 15) ----

;@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model0 PNR lots (nodes)

;@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@_pnr.tb ;---- Commuter Rail-PNR
connectors (links)
;@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model0-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
READ FILE = new_bus.tb ;--- Model0 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

;@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
;@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new_@TIME_PERIOD@_pnr.asc;--- drive to Model0

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail

```

## Appendix C Cube Voyager Scripts

```

READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

/* Transit path traces for select i/j pairs */
read file = ..\scripts\pathTrace.s

ENDRUN
;-----
;Step 2, 4, 6 & 8 Condition & Split Skims into Multiple Files
;-----
RUN PGM=MATRIX
; If we keep IDP here, we will need 16 cores, so we have commented it out
;@dp_token@DistributeIntrastep processId='mwcoq', ProcessList=%subnode%
MATI[1]=transit.temp.cr.skm
MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.SKM, MO = 1-16,
NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT, XWAIT,
WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D, PNR_IMP, PNR_CST

MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.STA, MO = 17-18,
NAME = I_STA, J_STA

MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.ttt, MO = 100,
NAME = sumtrntm

MW[1] = MI.1.1 ;---- ivt-local bus (0.01 min)
MW[2] = MI.1.2 ;---- ivt-exp bus (0.01 min)
MW[3] = MI.1.3 ;---- ivt-metrorail (0.01 min)
MW[4] = MI.1.4 ;---- ivt-commuter rail(0.01 min)
MW[5] = MI.1.5 ;---- ivt-new rail mode(0.01 min)
MW[6] = MI.1.6 ;---- ivt-new bus mode (0.01 min)
MW[7] = MI.1.7 ;---- ini.wait time (0.01 min)
MW[8] = MI.1.8 ;---- xfr wait time (0.01 min)
MW[9] = MI.1.9 ;---- walk acc time (0.01 min)
MW[10] = MI.1.10 ;---- other walk time (0.01 min)
MW[11] = MI.1.11 ;---- added xfer time (0.01 min)
MW[12] = MI.1.12 ;---- transfers (0+)
MW[13] = MI.1.13 ;---- drv acc time (0.01 min)
MW[14] = MI.1.14 ;---- drv acc distance (0.01 mile)
MW[15] = MI.1.15 ;---- pnr time (0.01 min)
MW[16] = MI.1.16 ;---- pnr cost (cents)

MW[17] = MI.1.17 ;---- metro board sta (1-150)
MW[18] = MI.1.18 ;---- metro alight sta (1-150)

;4
;

JLOOP
IF (MW[4] = 0 )
MW[1] = 0
MW[2] = 0
MW[3] = 0
MW[4] = 0
MW[5] = 0
MW[6] = 0
MW[7] = 0
MW[8] = 0
MW[9] = 0
MW[10] = 0
MW[11] = 0
MW[12] = 0
MW[13] = 0
MW[14] = 0
MW[15] = 0

```

```

MW[16] = 0
MW[17] = 0
MW[18] = 0
ELSE
MW[12] = MW[12] - 1
IF (MW[16] = 1 ) MW[16] = 0
MW[15] = MW[15] - MW[16] * 6.0
IF (MW[17] < 0 || MW[17] > 150 ) MW[17] = 0
IF (MW[18] < 0 || MW[18] > 150 ) MW[18] = 0
ENDIF
ENDJLOOP

MW[100] =(MW[1] + MW[2] + MW[3] + MW[4] + MW[5] +
MW[6] + MW[7] + MW[8] + MW[9] + MW[10] +
MW[11] + MW[13]) * 0.01 ;; Total Real Transit Time in Whole Minutes
(not incl. PNR 'impedance')

ENDRUN

ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----

```

## 49 Transit\_Skims\_MR.s

```

;-----
;Transit_Skims_MR.s
;MwCOG Version 2.2 Model
;
; - PATHSTYLE changed from 1 to 0 on 3.9.04 (RM)
; - iteration (_iter_) global variables used
;Build Transit Skims by Time Period and Access Mode
; Input Files:
; TP+ Highway Network = ZONEHWY.NET
; Transit Line Files = MODE?_pp.TB
; Transit Network Data = MET_*.TB, COM_*.TB, BUS_*.TB
; Walk and Drive Access = WALKACC.TB, *_PNR_pp.TB
; Walk Sidewalk Network = SIDEWALK.ASC
; Output Files:
; Walk and Drive Access Skims = pp_aa_mo.SKM
; Walk and Drive Station Data = pp_aa_mo.STA
;
; Step 1: AM Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 2: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter_%_AM_WK_MR.SKM, %_iter_%_AM_WK_MR.STA,
; Step 3: AM Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 4: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter_%_AM_DR_MR.SKM, %_iter_%_AM_DR_MR.STA,
; Step 5: AM Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 6: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter_%_AM_KR_MR.SKM, %_iter_%_AM_KR_MR.STA,
; Step 7: Off Peak Walk Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 8: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter_%_OP_WK_MR.SKM, %_iter_%_OP_WK_MR.STA,

```

## Appendix C Cube Voyager Scripts

```

; Step 9: Off Peak Drive Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 10: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter%_OP_DR_MR.SKM, %_iter%_OP_DR_MR.STA
; Step 11: Off Peak K/R Skims
; Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
; Output Files: transit.temp.mr.skm
; Step 12: Condition & Split Skims into Multiple Files
; Input Files: transit.temp.mr.skm
; Output Files: %_iter%_OP_KR_MR.SKM, %_iter%_OP_KR_MR.STA,
;
;-----
; rm 4/7/08 ;
; Added table #19 (Total Transit time in min.) to output transit.temp.mr.skm file
; create total transit time skims named:
; %_iter%_@TIME_PERIOD@_ACCESS_MODE@_MR.ttt
;
; useIdp = t (true) or f (false); this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=%useMdp%
;
; Read in time factors to increase local bus times
; based on increasing arterial hwy congestion
;-----
; Loop through each period and access mode
;-----
pageheight=32767 ; Preclude header breaks
LOOP PERIOD=1,2

IF (PERIOD = 1)
TIME_PERIOD = 'AM'
COMBINE = 5.0
ELSE
TIME_PERIOD = 'OP'
COMBINE = 10.0
ENDIF

;---- start the access mode loop ----

LOOP ACCESS=1,3

IF (ACCESS = 1)
ACCESS_MODE = 'WK'
WALK_MODEL = ';'
DRIVE_MODEL = ';'
KR_MODEL = ';'
ELSEIF (ACCESS = 2)
ACCESS_MODE = 'DR'
WALK_MODEL = ';'
DRIVE_MODEL = ';'
KR_MODEL = ';'
ELSE
ACCESS_MODE = 'KR'
WALK_MODEL = ';'
DRIVE_MODEL = ';'
KR_MODEL = ';'
ENDIF

;-----
; Step 1, 3, 5 & 7 Build Transit Path
;-----

RUN PGM=TRNBUILD
NETI = ZONEHWY.NET

```

```

MATO = transit.temp.mr.skm
maxnode = 60000

HWYTIME = @TIME_PERIOD@HTIME

;--- set default zone access and line parameters ----

ZONEACCESS GENERATE=N

@WALK_MODEL@ACCESSMODES = 14,16
@DRIVE_MODEL@ACCESSMODES = 11
@KR_MODEL@ACCESSMODES = 11

@WALK_MODEL@SKIPMODES = 11,15

PATHSTYLE = 0
USERRUNTIME = Y

;---- rules for combining multiple line and headways ----

COMBINE MAXDIFF[1] = 0.0, IF[1] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[2] = 0.0, IF[2] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[3] = 0.0, IF[3] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[4] = 0.0, IF[4] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[5] = 0.0, IF[5] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[6] = 0.0, IF[6] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[7] = 0.0, IF[7] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[8] = 0.0, IF[8] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[9] = 0.0, IF[9] = ((RUN - MINRUN) < @COMBINE@)
COMBINE MAXDIFF[10] = 0.0, IF[10] = ((RUN - MINRUN) < @COMBINE@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1] = 10*1.0 ;---- in-vehicle time
MODEFAC[11] = 1.50 ;---- drive access time
MODEFAC[12] = 2.00 ;---- transit transfer time
MODEFAC[13] = 2.00 ;---- walk network time
MODEFAC[14] = 2.00 ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50 ;---- park-&-ride transfer time
MODEFAC[16] = 2.00 ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0
XWAITMIN[1] = 2*4.0,0.0,4.0,0.0,3*4.0,10.0,4.0

;---- boarding and transfer penalties ----

XPEN[1]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[2]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[3]= 2*5.0, 0.0, 2*2.0,5*5.0, 6*0.0
XPEN[4]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[5]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[6]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[7]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[8]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[9]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[10]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[11]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[12]= 2*8.0,3*2.0,4*8.0,5.0, 6*0.0
XPEN[13]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[14]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[15]= 2*5.0,3*2.0,5*5.0, 6*0.0
XPEN[16]= 2*5.0,3*2.0,5*5.0, 6*0.0

XPENFAC[1]= 16*2.50

```

## Appendix C Cube Voyager Scripts

```

XPENFAC[2]= 16*2.50
XPENFAC[3]= 16*2.50
XPENFAC[4]= 16*2.50
XPENFAC[5]= 16*2.50
XPENFAC[6]= 16*2.50
XPENFAC[7]= 16*2.50
XPENFAC[8]= 16*2.50
XPENFAC[9]= 16*2.50
XPENFAC[10]= 16*2.50
XPENFAC[11]= 16*2.50
XPENFAC[12]= 16*2.50
XPENFAC[13]= 16*2.50
XPENFAC[14]= 16*2.50
XPENFAC[15]= 16*2.50
XPENFAC[16]= 16*2.50

;---- transfer prohibitions ----

;--- mode 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16
NOX[1] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[2] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[3] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[4] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[5] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[6] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[7] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[8] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[9] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[10] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[11] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[12] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[13] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[14] = n, n, n, n, n, n, n, n, Y, Y, n, Y, n, n, n, Y, n
NOX[15] = n, n, n, n, n, n, n, n, Y, Y, n, Y, Y, Y, Y, Y, Y
NOX[16] = n, n, n, n, n, n, n, n, Y, n, n, n, n, Y, Y, Y

;---- Parameters ----

LISTINPUT = N ;--- echo input files

MAXPATHTIME = 360.0 ;--- Kill any path with preceived time > 240 min.
FREPERIOD = 1 ;--- Use the First Headway value
USERRUNTIME = Y ;--- Ignore any RUNTIME or RT parameters on lines.
MAXRUNTIME = 240.0 ;--- Report lines with run times > 240 min.
;ONLINE = 100 ;--- Display every 100 lines

;WALKSPEED = 3.0 ;--- Set default walk speed to 3.0 mph
;XYFACTOR = 0.84401 ;--- Replicate MINUTP value
;WALKSPEED = 2.0 ;--- Added on 09/25
;XYFACTOR = 1.97 ;--- Added on 09/25

;-----
; write out support links for later viewing in VIPER
fileo supporto = supl_@time_period@_@access_mode@_mr.asc modes=11-16 oneway=t
fixed=y
fileo nodeo = supn_@time_period@_@access_mode@_mr.dbf
fileo linko = trnl_@time_period@_@access_mode@_mr.dbf ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT,
XWAIT, WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D,
PNR_IMP, PNR_CST, I_STA, J_STA,
MW[1] = TIME(1,6,8), ;---- ivt-local bus (0.01 min)
MW[2] = TIME(2,7,9), ;---- ivt-exp bus (0.01 min)

```

```

MW[3] = TIME(3), ;---- ivt-metrorail (0.01 min)
MW[4] = TIME(4), ;---- ivt-commuter rail(0.01 min)
MW[5] = TIME(5), ;---- ivt-new rail mode(0.01 min)
MW[6] = TIME(10), ;---- ivt-new bus mode (0.01 min)
MW[7] = IWAIT, ;---- ini.wait time (0.01 min)
MW[8] = XWAIT(1,2,3,4,5,6,7,8,9,10), ;---- xfr wait time (0.01 min)
MW[9] = TIME(14,16), ;---- walk acc time (0.01 min)
MW[10] = TIME(12,13), ;---- other walk time (0.01 min)
MW[11] = XPEN, ;---- added xfer time (0.01 min)
MW[12] = BOARDS, ;---- boardings (1+)
MW[13] = TIME(11), ;---- drv acc time (0.01 min)
MW[14] = DIST(11), ;---- drv acc distance (0.01 mile)
MW[15] = TIME(15), ;---- pnr impedance (0.01 min)
MW[16] = DIST(15), ;---- pnr cost (cents)
MW[17] = NODE0(3) - 8000.0, ;---- metro board sta (1-150)
MW[18] = NODEL(3) - 8000.0 ;---- metro alight sta (1-150)

```

;---- Rail Stations & Links (modes 3 & 4) ----

```

READ FILE = met_node.tb ;---- Metrorail stations
READ FILE = met_link.tb ;---- Metrorail links
;READ FILE = com_node.tb ;---- Commuter Rail stations
;READ FILE = com_link.tb ;---- Commuter Rail links
READ FILE = lrt_node.tb ;---- LRT stations
READ FILE = lrt_link.tb ;---- LRT links
;READ FILE = new_node.tb ;---- Model0 Stations
;READ FILE = new_link.tb ;---- Model0 links
;---- Park and Ride Lots (mode 15) ----

;@DRIVE_MODEL@ READ FILE = bus_pnrn.tb ;---- Bus PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = met_pnrn.tb ;---- Metro PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = com_pnrn.tb ;---- Commuter Rail PNR lots (nodes)
@DRIVE_MODEL@ READ FILE = lrt_pnrn.tb ;---- LRT PNR lots (nodes)
;@DRIVE_MODEL@ READ FILE = new_pnrn.tb ;---- Model0 PNR lots (nodes)

;@DRIVE_MODEL@ READ FILE = bus_@TIME_PERIOD@_pnr.tb ;---- Bus-PNR connectors
(links)
@DRIVE_MODEL@ READ FILE = met_@TIME_PERIOD@_pnr.tb ;---- Metro-PNR connectors
(links)
;@DRIVE_MODEL@ READ FILE = com_@TIME_PERIOD@_pnr.tb ;---- Commuter Rail-PNR
connectors (links)
@DRIVE_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)
;@DRIVE_MODEL@ READ FILE = new_@TIME_PERIOD@_pnr.tb ;---- Model0-PNR connectors
(links)

;---- Access Links (modes 11, 12 and 16) ----

READ FILE = met_bus.tb ;--- bus-metro links&xfer cards
;READ FILE = com_bus.tb ;--- bus-commuter rail links&xfer car
READ FILE = lrt_bus.tb ;--- bus-LRT links&xfer car
;READ FILE = new_bus.tb ;--- Model0 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = met_@TIME_PERIOD@_pnr.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = com_@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus_@TIME_PERIOD@_pnr.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt_@TIME_PERIOD@_pnr.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Model0

@KR_MODEL@READ FILE = met_@TIME_PERIOD@_knr.asc;--- k/r to metrorail
;KR_MODEL@READ FILE = bus_@TIME_PERIOD@_knr.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrt_@TIME_PERIOD@_knr.asc;--- k/r to LRT
;@KR_MODEL@READ FILE = new_@TIME_PERIOD@_knr.asc;--- k/r to Model0

@KR_MODEL@ READ FILE = lrt_@TIME_PERIOD@_pnr.tb ;---- LRT-PNR connectors (links)

```

## Appendix C Cube Voyager Scripts

```

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = sidewalk.asc;--- walk network for transfers

;---- Transit Line Cards (modes 1-10) ----

;READ FILE = MODE1@TIME_PERIOD@.TB ;---- M1- metrobus local
;READ FILE = MODE2@TIME_PERIOD@.TB ;---- M2- metrobus express
READ FILE = MODE3@TIME_PERIOD@.TB ;---- M3- metrorail
;READ FILE = MODE4@TIME_PERIOD@.TB ;---- M4- commuter rail
READ FILE = MODE5@TIME_PERIOD@.TB ;---- M5- other rail (future)
;READ FILE = MODE6@TIME_PERIOD@.TB ;---- M6- other local bus
;READ FILE = MODE7@TIME_PERIOD@.TB ;---- M7- other express bus
;READ FILE = MODE8@TIME_PERIOD@.TB ;---- M8- other local bus
;READ FILE = MODE9@TIME_PERIOD@.TB ;---- M9- other express bus
;READ FILE = MODE10@TIME_PERIOD@.TB ;---- M10- other bus (future)

/* Transit path traces for select i/j pairs */
read file = ..\scripts\pathTrace.s

ENDRUN
;-----
;Step 2, 4, 6 & 8 Condition & Split Skims into Multiple Files
;-----
RUN PGM=MATRIX
; If we keep IDP here, we will need 16 cores, so we have commented it out
;@dtp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
MATI[1]=transit.temp.mr.skm
MATO[1]=% iter % @TIME_PERIOD@_@ACCESS_MODE@_MR.SKM, MO = 1-16,
NAME = IVT_LBUS, IVT_XBUS, IVT_MET, IVT_CR, IVT_LRT, IVT_BRT, IWAIT, XWAIT,
WK_AC_T, OTH_WK_T, X_PEN, BOARDS, DR_AC_T, DR_AC_D, PNR_IMP, PNR_CST

MATO[2]=% iter % @TIME_PERIOD@_@ACCESS_MODE@_MR.STA, MO = 17-18,
NAME = I_STA, J_STA

MATO[3]=% iter % @TIME_PERIOD@_@ACCESS_MODE@_MR.ttt, MO = 100, ;
NAME = sumtrntm

MW[1] = MI.1.1 ;---- ivt-local bus (0.01 min)
MW[2] = MI.1.2 ;---- ivt-exp bus (0.01 min)
MW[3] = MI.1.3 ;---- ivt-metrorail (0.01 min)
MW[4] = MI.1.4 ;---- ivt-commuter rail(0.01 min)
MW[5] = MI.1.5 ;---- ivt-new rail mode(0.01 min)
MW[6] = MI.1.6 ;---- ivt-new bus mode (0.01 min)
MW[7] = MI.1.7 ;---- ini.wait time (0.01 min)
MW[8] = MI.1.8 ;---- xfr wait time (0.01 min)
MW[9] = MI.1.9 ;---- walk acc time (0.01 min)
MW[10] = MI.1.10 ;---- other walk time (0.01 min)
MW[11] = MI.1.11 ;---- added xfer time (0.01 min)
MW[12] = MI.1.12 ;---- transfers (0+)
MW[13] = MI.1.13 ;---- drv acc time (0.01 min)
MW[14] = MI.1.14 ;---- drv acc distance (0.01 mile)
MW[15] = MI.1.15 ;---- pnr time (0.01 min)
MW[16] = MI.1.16 ;---- pnr cost (cents)

MW[17] = MI.1.17 ;---- metro board sta (1-150)
MW[18] = MI.1.18 ;---- metro alight sta (1-150)

;4
;
;

JLOOP
IF ((MW[3]+MW[5] = 0) || (MW[1]+MW[2]+MW[6] > 0))
MW[1] = 0
MW[2] = 0

```

```

MW[3] = 0
MW[4] = 0
MW[5] = 0
MW[6] = 0
MW[7] = 0
MW[8] = 0
MW[9] = 0
MW[10] = 0
MW[11] = 0
MW[12] = 0
MW[13] = 0
MW[14] = 0
MW[15] = 0
MW[16] = 0
MW[17] = 0
MW[18] = 0
ELSE
MW[12] = MW[12] - 1
IF (MW[16] = 1 ) MW[16] = 0
MW[15] = MW[15] - MW[16] * 6.0
IF (MW[17] < 0 || MW[17] > 150 ) MW[17] = 0
IF (MW[18] < 0 || MW[18] > 150 ) MW[18] = 0
ENDIF
ENDJLOOP

MW[100] = (MW[1] + MW[2] + MW[3] + MW[4] + MW[5] +
MW[6] + MW[7] + MW[8] + MW[9] + MW[10] +
MW[11] + MW[13]) * 0.01 ;; Total Real Transit Time in Whole Minutes
(not incl. PNR 'impedance')

ENDRUN

ENDLOOP ;---- ACCESS ----
ENDLOOP ;---- PERIOD ----

50 Trip_Distribution_External.s

*del voya*.prn

; Trip_Distribution_External.s - Version 2.3 Trip Distribution for External Trips
;
ZONESIZE = 3722 ; Max. TAZ No. (Param)
LSTITAZ = 3675 ; Last Internal Zone No. (Param)

;; itr = '% iter %' ;;
;; IF (itr = 'pp')
;; AMSOVSKM = 'inputs\SOVppam.skm' ; AM HWY TIME SKIMS (Initial
iteration)
;; MDSOVSKM = 'inputs\SOVppmd.skm' ; MD HWY TIME SKIMS (Initial
iteration)
;; ELSE
;; AMSOVSKM = 'SOV_prev_%am.skm' ; AM HWY TIME SKIMS
;; MDSOVSKM = 'SOV_prev_%md.skm' ; MD HWY TIME SKIMS
;; ENDIF

AMSOVSKM = '%_prev_%_am_sov.skm' ; AM HWY TIME SKIMS
MDSOVSKM = '%_prev_%_md_sov.skm' ; MD HWY TIME SKIMS

ATYPFILE = 'AreaType_File.dbf' ; Zonal Area Type file (I/P file)
HWYTERM = 'ztermtm.asc' ; Zonal HWY TERMINAL TIME file (created
in THIS script)

```

## Appendix C Cube Voyager Scripts

```

AWTRNSKM = '%_iter_%_am_wk_MR.ttt'      ; AM WK (Metrorail only) ACC TRN TIME
SKIMS
ADTRNSKM = '%_iter_%_am_dr_MR.ttt'      ; AM DR (Metrorail Only) ACC TRN TIME
SKIMS

MWTRNSKM = '%_iter_%_op_wk_MR.ttt'      ; OP WK (Metrorail only) ACC TRN TIME
SKIMS
MDTRNSKM = '%_iter_%_op_dr_MR.ttt'      ; OP DR (Metrorail Only) ACC TRN TIME
SKIMS

; -----
; Equivalent minutes (min/'07$) by income level (for toll modeling)
toll_inc = '..\support\equiv_toll_min_by_inc.s' ; Equivalent minutes (min/'07$)
by period & income level (for toll modeling)

; Zonal K-factor Files
;
HBWK      = 'hbw_k.mat' ;
HBSK      = 'hbs_k.mat' ;
HBOK      = 'hbo_k.mat' ;
NHWK      = 'nhw_k.mat' ;
NHOK      = 'nho_k.mat' ;
;
; -----
;
FFsFile = '..\SUPPORT\ver23_f_factors.dbf' ; F-Factors for all modeled purposes
;;Variables in the dbf file:
; IMP  HBWINC1 HBWINC2 HBWINC3 HBWINC4 HWEI HBWEA ;
;      HBSINC1 HBSINC2 HBSINC3 HBSINC4 HBSEI HBSEA ;
;      HBOINC1 HBOINC2 HBOINC3 HBOINC4 HBOEI HBOEA ;
;      NHW   NHO   NHBEI  NHBEA ;
;      ICOM  IMTK  IHTK   EXTCOM  EXTMTK  EXTHTK ;
;
; Trip-End (P/A) Input Files:
;
AutoProds = '%_iter_%_Trip_Gen_Productions_Comp.dbf' ; Intl/Extl Auto
Productions
;;Variables in dbf file:
; TAZ  HBW_MTR_PS  HBW_NMT_PS  HBW_ALL_PS  HBWMTRP_I1  HBWMTRP_I2  HBWMTRP_I3
HBWMTRP_I4
;      HBS_MTR_PS  HBS_NMT_PS  HBS_ALL_PS  HBSMTRP_I1  HBSMTRP_I2  HBSMTRP_I3
HBSMTRP_I4
;      HBO_MTR_PS  HBO_NMT_PS  HBO_ALL_PS  HBOMTRP_I1  HBOMTRP_I2  HBOMTRP_I3
HBOMTRP_I4
;      NHW_MTR_PS  NHW_NMT_PS  NHW_ALL_PS  NHO_MTR_PS  NHO_NMT_PS  NHO_ALL_PS
;
AutoAttr = '%_iter_%_Trip_Gen_Attractions_Comp.dbf' ; Intl/Extl Auto Attractions
;;Variables in dbf file:
; TAZ  HBW_MTR_AS  HBW_NMT_AS  HBW_ALL_AS  HBWMTRA_I1  HBWMTRA_I2  HBWMTRA_I3
HBWMTRA_I4
;      HBS_MTR_AS  HBS_NMT_AS  HBS_ALL_AS  HBSMTRA_I1  HBSMTRA_I2  HBSMTRA_I3
HBSMTRA_I4
;      HBO_/TR_AS  HBO_NMT_AS  HBO_ALL_AS  HBOMTRA_I1  HBOMTRA_I2  HBOMTRA_I3
HBOMTRA_I4
;      NHW_MTR_AS  NHW_NMT_AS  NHW_ALL_AS  NHO_MTR_AS  NHO_NMT_AS  NHO_ALL_AS
;
ExtPsAs = '%_iter_%_Ext_Trip_Gen_PsAs.dbf' ; Extl Auto Ps, As
;;Variables in dbf file:
; TAZ  SHBW_Mtr_Ps  SHBS_Mtr_Ps  SHBO_Mtr_Ps  SNHW_Mtr_Ps  SNHO_Mtr_Ps
;      SHBW_Mtr_As  SHBS_Mtr_As  SHBO_Mtr_As  SNHW_Mtr_As  SNHO_Mtr_As
;
;
TruckEnds = '%_iter_%_ComVeh_Truck_Ends.dbf' ; Intl Comm.Veh/Truck TripEnds

```

```

;;Variables in dbf file:
; TAZ  ICOMM_VEH  IMED_TRUCK  IHVY_TRUCK
;
;
Ext_TrkEnds = '%_iter_%_Ext_CVTruck_Gen_PsAs.dbf'
;;Variables in dbf file:
; TAZ  SCOM_VEHPs  SMED_TRKPS  SHVY_TRKPS  SCOM_VEHAS  SMED_TRKAS
SHVY_TRKAS
;
; OUTPUT TRIP TABLES
HBWTDOUT = '%_iter_%_HBWext.PTT';
HBSTDOUT = '%_iter_%_HBsext.PTT';
HBOTDOUT = '%_iter_%_HBOext.PTT';
NHWTDOUT = '%_iter_%_NHWext.PTT';
NHOTDOUT = '%_iter_%_NHOext.PTT';
COMTDOUT = '%_iter_%_COMext.VTT';
MTKTDOUT = '%_iter_%_MTKext.VTT';
HTKTDOUT = '%_iter_%_HTKext.VTT';

; //////////////////////////////////////
; \\\\\\\\\\  BEGIN Composite Impedance, terminal time development \\\
; //////////////////////////////////////

RUN PGM=MATRIX
zones=1

;
FileI LOOKUPI[1] ="@atypfile@"
LOOKUP LOOKUPI=1, NAME=ZNAT,
      LOOKUP[1] = TAZ, RESULT=Atype, ;
      INTERPOLATE=N, FAIL= 0,0,0, LIST=N
; CREATE ZONAL ARRAY FOR EMPLOYMENT DENSITY

Loop M= 1,@ZONESIZE@

      Atype = ZNAT(1,M) ; Area Type
      If (Atype = 1 ) Termtm= 5.0
      if (Atype = 2 ) Termtm= 4.0
      if (Atype = 3 ) Termtm= 3.0
      if (Atype = 4 ) Termtm= 2.0
      if (Atype = 5 ) Termtm= 1.0
      if (Atype = 6 ) Termtm= 1.0
      if (Atype = 7 ) Termtm= 1.0

      if (M > @LSTITAZ@) Termtm = 0.0

; WRITE OUT ZONAL TERMINAL TIME FILE
      list = 'TAZ: ',M(4),' AT: ',_Atype(3),' Term. Time: ',
            termtm(3),file=@hwyterm@

ENDLOOP

ENDRUN

;
; //////////////////////////////////////
; \\\\\\\\\\  1) Add Highway Terminal Times to AM, Off-peak \\\\\\
; \\\\\\\\\\  SOV Skims \\\\\\
; //////////////////////////////////////

RUN PGM=MATRIX
Zones = 3722
; READ Highway terminal time file

```

## Appendix C Cube Voyager Scripts

```

ZDATI[1]= @hwyterm@, Z=6-9,hterm=31-33

; READ AM PEAK & Midday SOV TIME SKIM FILE (IN WHOLE MIN)

MATI[1] = @AMSOVSKM@ ; INPUT AM PK SKIM FILE
MATI[2] = @MDSOVSKM@ ; INPUT OFF-PK SKIM FILE

MW[1] = MI.1.1 ; INPUT AM PK Time (min) SKIM FILE
MW[2] = MI.2.1 ; INPUT OFF-PK Time (min) SKIM FILE

;
; Now add the terminal times to the AM/MD travel times below
; - terminal times added only to connected interchanges)
; - terminal times are added to both the i and j ends of the trip
;
JLOOP
  IF (MW[1] > 0)
    MW[3] = MW[1] + zi.1.hterm[I] + zi.1.hterm[J]
  ELSE
    MW[3] = MW[1]
  ENDF
  IF (MW[2] > 0)
    MW[4] = MW[2] + zi.1.hterm[I] + zi.1.hterm[J]
  ELSE
    MW[4] = MW[2]
  ENDF
ENDJLOOP

;
; Establish Intrazonal Values for Network Time Skims
; - Values equal to 85% of lowest nonzero interzonal value
; - Up from 50% used in Version 2.2
JLOOP
  IF (I=J)
    MW[3]=ROUND(0.50 * LOWEST(3,1,0.0001,99999.9))
    MW[4]=ROUND(0.50 * LOWEST(4,1,0.0001,99999.9))
  ENDF
ENDJLOOP
; WRITE OUT FINAL TIME SKIMS

MATO[1] = am_sov_termIntraTime.skf, MO=3; output am sov time(min) w/ o&d term&intra
times
MATO[2] = md_sov_termIntraTime.skf, MO=4; output md sov time(min) w/ o&d term&intra
times

; print row 1 of I/O matrices for checking

  IF (I =699)
    PRINTROW MW=1-4
  ENDF

ENDRUN

; ////////////////////////////////////////
; \\\\\\ 2) Compute Composite Impedances to by used in \\\
; \\\\\\ Trip Distribution for HBW, HBS, HBO, NHB Purposes \\\
; ////////////////////////////////////////

RUN PGM=MATRIX
Zones = 3722

; COMPUTATION OF COMPOSITE IMPEDANCES
; READ AM PEAK & OFF-PEAK SOV TIME SKIM FILE (IN WHOLE MIN)

MATI[1] = am_sov_termIntraTime.skf ; AM PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
MATI[2] = md_sov_termIntraTime.skf ; OFF-PK HWY TIME FILE W/ TERM&INTRAZNL VALUES

MATI[3] = @AWTRNSKM@ ; AM PK WALK ACC TRN (Metrorail Only) SKIM FILE
MATI[4] = @ADTRNSKM@ ; AM PK AUTO ACC TRN (Metrorail Only) SKIM FILE
MATI[5] = @MWTRNSKM@ ; Midday WALK ACC TRN (Metrorail Only) SKIM FILE
MATI[6] = @MDTRNSKM@ ; Midday AUTO ACC TRN (Metrorail Only) SKIM FILE

;$
MATI[7] = @AMSOVSKM@ ; INPUT AM PK tolls in '07 cents (on table 3)
MATI[8] = @MDSOVSKM@ ; INPUT Midday tolls in '07 cents (on table 3)
;
  READ FILE =@TOLL_INC@ ; READ in equivalent min/07$ by income group
;
;$

; ESTABLISH WORK MATRICES:

MW[1]=MI.1.1 ; AM PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
MW[2]=MI.2.1 ; OFF-PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
;
;-----; Make Sure interzonal (conn.or disconn.)
JLOOP
  IF (MW[1] = 0.0)
    MW[1] = 1.0
  ENDF
  IF (MW[2] = 0.0)
    MW[2] = 1.0
  ENDF
ENDJLOOP
;-----;
;
;$
;-
; add equivalent 'tolled' AM/OP highway time to normal times by income level
; AM pk normal + equivalent hwy time in work tables 61-64
; Offpk normal + equivalent hwy time in work tables 71-74

MW[61] = Round(MW[1] + ((MI.7.3/100.0) * i1PKEQM)) ;i1 AM hwy time w/eqv
MW[62] = Round(MW[1] + ((MI.7.3/100.0) * i2PKEQM)) ;i2 AM hwy time w/eqv
MW[63] = Round(MW[1] + ((MI.7.3/100.0) * i3PKEQM)) ;i3 AM hwy time w/eqv
MW[64] = Round(MW[1] + ((MI.7.3/100.0) * i4PKEQM)) ;i4 AM hwy time w/eqv

MW[71] = Round(MW[2] + ((MI.8.3/100.0) * i1MDEQM)) ;i1 MD hwy time w/eqv
MW[72] = Round(MW[2] + ((MI.8.3/100.0) * i2MDEQM)) ;i2 MD hwy time w/eqv
MW[73] = Round(MW[2] + ((MI.8.3/100.0) * i3MDEQM)) ;i3 MD hwy time w/eqv
MW[74] = Round(MW[2] + ((MI.8.3/100.0) * i4MDEQM)) ;i4 MD hwy time w/eqv

;
; Lines below convert tolls to time for distribution of external trips.
; Average factors from traffic assignment are used.
;
  MW[76] = Round(MW[1] + ((MI.7.3/100.0) * SVAMEQM)) ;X-I,I-X AM hwy time w/eqv
- added by DV 2/6/09
  MW[77] = Round(MW[2] + ((MI.8.3/100.0) * SVMDEQM)) ;X-I,I-X OP hwy time w/eqv
- added by DV 2/6/09
;
;
;
;
MW[3]=MI.3.1 ; AM PK WALK ACC TOTAL TRN TIME FILE
MW[4]=MI.4.1 ; AM PK AUTO ACC TOTAL TRN TIME FILE

MW[5]=MI.5.1 ; OFF-PK WALK ACC TOTAL TRN TIME FILE
MW[6]=MI.6.1 ; OFF-PK AUTO ACC TOTAL TRN TIME FILE

;FIRST, FIND 'BEST' WALK/AUTO TRANSIT TIME BOTH AM AND OFF-PK CONDITIONS

```



## Appendix C Cube Voyager Scripts

```

; BEST AM TRN TIME STORED IN MW11, BEST OP TRN TIME STORED IN MW12

JLOOP
  IF (MW[3] > 0 && MW[4] > 0) ; 'BEST' AM PK TRN TIME
    MW[11] = MIN(MW[3],MW[4]) ; WILL BE THE MINIMUM OF
  ELSE ; NON-ZERO WALK/AUTO TIMES OR
    MW[11] = MAX(MW[3],MW[4]) ; THE ONE THAT'S CONNECTED
  ENDIF

  IF (MW[5] > 0 && MW[6] > 0) ; SAME FOR OFF PEAK
    MW[12] = MIN(MW[5],MW[6]) ;
  ELSE ;
    MW[12] = MAX(MW[5],MW[6]) ;
  ENDIF
ENDJLOOP

; NOW COMPUTE HBW,HBS,HBO,NHB COMPOSITE IMPEDANCES
;
JLOOP
  IF (MW[11] = 0 || I = J)
    MW[15] = MW[61]
    MW[16] = MW[62]
    MW[17] = MW[63]
    MW[18] = MW[64]
  ELSE
    MW[15] = 1.0/((1.0/MW[61])+(0.1851/MW[11])) ; HBW -INC 1 CI MTX
    MW[16] = 1.0/((1.0/MW[62])+(0.1563/MW[11])) ; HBW -INC 2 CI MTX
    MW[17] = 1.0/((1.0/MW[63])+(0.1682/MW[11])) ; HBW -INC 3 CI MTX
    MW[18] = 1.0/((1.0/MW[64])+(0.1483/MW[11])) ; HBW -INC 4 CI MTX
  ENDIF

  IF (MW[12] = 0 || I = J)
    MW[20] = MW[71]
    MW[21] = MW[72]
    MW[22] = MW[73]
    MW[23] = MW[74]

    MW[25] = MW[71]
    MW[26] = MW[72]
    MW[27] = MW[73]
    MW[28] = MW[74]

    MW[50] = MW[72]
    MW[51] = MW[72]
  ELSE
    MW[20] = 1.0/((1.0/MW[71])+(0.0805/MW[12])) ; HBS -INC 1 CI MTX
    MW[21] = 1.0/((1.0/MW[72])+(0.0184/MW[12])) ; HBS -INC 2 CI MTX
    MW[22] = 1.0/((1.0/MW[73])+(0.0117/MW[12])) ; HBS -INC 3 CI MTX
    MW[23] = 1.0/((1.0/MW[74])+(0.0104/MW[12])) ; HBS -INC 4 CI MTX

    MW[25] = 1.0/((1.0/MW[71])+(0.1239/MW[12])) ; HBO -INC 1 CI MTX
    MW[26] = 1.0/((1.0/MW[72])+(0.0231/MW[12])) ; HBO -INC 2 CI MTX
    MW[27] = 1.0/((1.0/MW[73])+(0.0188/MW[12])) ; HBO -INC 3 CI MTX
    MW[28] = 1.0/((1.0/MW[74])+(0.0158/MW[12])) ; HBO -INC 4 CI MTX

    MW[50] = 1.0/((1.0/MW[72])+(0.0866/MW[12])) ; NHW
    MW[51] = 1.0/((1.0/MW[72])+(0.0224/MW[12])) ; NHO
  ENDIF
ENDJLOOP

MATO[1] = HBWC1_4.MAT, MO=15,16,17,18 ;HBW COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[2] = HBSC1_4.MAT, MO=20,21,22,23 ;HBS COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[3] = HBOC1_4.MAT, MO=25,26,27,28 ;HBO COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[4] = NHBC1.MAT, MO=50,51 ;NHW/NHO COMP.IMPEDANCES
MATO[5] = am_sov_termIntraTime_x.skf, MO=76 ; AM Peak X-I, I-X impedances
with tolls

```

```

MATO[6] = md_sov_termIntraTime_x.skf, MO=77 ; Off Peak X-I, I-X impedances
with tolls
;
; $
;
; NOW, WRITE OUT THE RESULTS OF SELECTED INTERCHANGES FOR CHECKING
; AND COMPARING
JLOOP INCLUDE=1 ; WILL PROCESS ONLY FOR J=1
  PRINT LIST = I(4), ' ', J(4), ' ', mw[15](5),mw[16](5),mw[17](5),mw[18](5),
    FILE =ci_hbw.txt
  PRINT LIST = I(4), ' ', J(4), ' ', mw[20](5),mw[21](5),mw[22](5),mw[23](5),
    FILE =ci_hbs.txt
  PRINT LIST = I(4), ' ', J(4), ' ', mw[25](5),mw[26](5),mw[27](5),mw[28](5),
    FILE =ci_hbo.txt
  PRINT LIST = I(4), ' ', J(4), ' ', mw[72](5),MW[12](5),mw[50](5),MW[51](5),
    FILE =ci_nhb.txt
ENDJLOOP
ENDRUN

; ////////////////////////////////////////////////////
; \\\ \\\ \\\ 3) Compute Impedance files to be used in the External \\\
; \\\ \\\ \\\ Trip Distribution processing \\\
; ////////////////////////////////////////////////////

RUN PGM=MATRIX
ZONES =3722
MATI[1] = am_sov_termIntraTime_x.skf ; AM PK HWY TIME FILE W/ TERM&INTRAZNL
VALUES
MATI[2] = md_sov_termIntraTime_x.skf ; Midday HWY TIME FILE W/ TERM&INTRAZNL
VALUES

MW[1]=MI.1.1 ; AM PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
MW[2]=MI.2.1 ; Midday HWY TIME FILE W/ TERM&INTRAZNL VALUES

; Development of Peak, Midday SOV Travel times to be used
; for External Trip distribution of Interstate and Arterial Trip Dist.
;
; 2 skim files will be written:
; MW[11] - AM Time Period, External ijs's
; MW[12] - Midday Period, External ijs's
;
; First, set work matrices equal to 'Full' AM, Off-peak time skims
;
MW[11] = MW[1] ; AM
MW[12] = MW[2] ; Midday

; next, put very large time value into all
; i-i and x-x ijs to preclude distributing externals in these cells

IF (I = 1-3675)
  MW[11] = 2000, INCLUDE= 1-3675 ; i-i ijs
  MW[12] = 2000, INCLUDE= 1-3675 ; i-i ijs
ELSE
  MW[11] = 2000, INCLUDE= 3675-3722 ; x-x ijs
  MW[12] = 2000, INCLUDE= 3675-3722 ; x-x ijs
ENDIF

; WRITE OUT EXTERNAL TRIP DISTRIBUTION IMPEDANCE TABLES

MATO[1] = am_sov_termIntraTime_e.skf, MO=11 ; AM -PK Time skims for Extl trip dist.
MATO[2] = md_sov_termIntraTime_e.skf, MO=12 ; Midday Time skims for Extl trip dist.
ENDRUN
;
; End of Composite Impedance Development

```





## Appendix C Cube Voyager Scripts

```

; T4 - NHO EXTL arterial facility FFactors
ENDRUN
; --ENB NHB Trip Dist---;
; |////////////////////|
; |//////// Start COM/TRK Trip Distribution Here: //|
; |////////////////////|

RUN PGM=TRIPDIST
MATI[1] = %prev % MD_SOV.SKM ; Off-Pk Time Imped. for COM
MATI[2] = %prev % MD_TRUCK.skm ; Off-Pk Truck Time for MTK/HTK
MATI[3] = md_sov_termIntraTime_e.skf ; Midday Time skims for Extl trip
dist.

; Put impedance matrices in work tables 11-12. Tab 11 is for COM
; trips; tab 12 is for MTK and HTK trips. All time values are in minutes.

MW[11] = MI.1.1 ; com veh los matix
MW[12] = MI.2.1 ; trk los matix
MW[13] = MI.3.1 ; extl los matix

ZDATI[1] = @TruckEnds@
ZDATI[2] = @Ext_TrkEnds@

; FFactors
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
LOOKUP[1] = IMP, RESULT=EXTCOM, ; Ext CV
LOOKUP[2] = IMP, RESULT=EXTMTK, ; Ext Mtk
LOOKUP[3] = IMP, RESULT=EXTHTK, ; Ext Htk
INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

; Establish production and attraction vectors here:

SETPA P[1]=ZI.2.SCOM_VEHPs, P[2]=ZI.2.SMED_TRKPS, P[3]=ZI.2.SHVY_TRKPS
SETPA A[1]=ZI.2.SCOM_VEHAS, A[2]=ZI.2.SMED_TRKAS, A[3]=ZI.2.SHVY_TRKAS

MAXITERS = 9 ; specify GM iterations
MAXRMSE = 0.0001

; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[13], FFACTORS= FF,losrange=2-250 ; COM External
GRAVITY PURPOSE = 2, LOS=MW[13], FFACTORS= FF,losrange=2-250 ; MTK External
GRAVITY PURPOSE = 3, LOS=MW[13], FFACTORS= FF,losrange=2-250 ; HTK External

MATO[1] = COMext.TEM,MO=1 ; Final COM trip tables: 1 = Extl
MATO[2] = MTKext.TEM,MO=2 ; Final MTK trip tables: 1 = Extl
MATO[3] = HTKext.TEM,MO=3 ; Final HTK trip tables: 1 = Extl

ENDRUN

; End COM/TRK Trip Distribution ---
;-----
; Now splice the external interstate/ external arterial matrices by purpose into
single external table -
;-----

RUN PGM=MATRIX
ZONES = @ZONESIZE@

MATI[1] = HBWext.TEM ; 2 HBW trip tables: Ext/InterstFFs, Extls/ArterFFs
MATI[2] = HBSext.TEM ; 2 HBS trip tables: Ext/InterstFFs, Extls/ArterFFs
MATI[3] = HBOext.TEM ; 2 HBO trip tables: Ext/InterstFFs, Extls/ArterFFs
MATI[4] = NHBext.TEM ; 4 NHB trip tables: NHW Extl/IntFFs,NHW Extl/ArtFFs,
NHO Extl/IntFFs,NHO Extl/ArtFFs
MATI[5] = COMext.TEM ; 1 Com trip tables: Extl
MATI[6] = MTKext.TEM ; 1 Mtk trip tables: Extl

```

```

MATI[7] = HTKext.TEM ; 1 Htk trip tables: Extl

FillMW MW[101]=mi.1.1,2 ; HBW external tabs in mw 101-102
FillMW MW[201]=mi.2.1,2 ; HBS external tabs in mw 201-202
FillMW MW[301]=mi.3.1,2 ; HBO external tabs in mw 301-302
FillMW MW[401]=mi.4.1,2,3,4 ; NHW,NHO external tabs in mw 401-404

FillMW MW[601]=mi.5.1 ; Com external tabs in mw 501
FillMW MW[701]=mi.6.1 ; Mtk external tabs in mw 601
FillMW MW[801]=mi.7.1 ; Htk external tabs in mw 701

; define external interstate, and external arterial station interchanges
; in mws 11, 22
MW[11]=0.0
MW[22]=0.0

; define External /Interstate rows, columns
if (I >= 1 && I <= @LstITaz@) mw[11] = 1.0, include =
3677,3680,3685,3687,3697,3702,3711,3713,3714,3715,3718,3722
if (I=3677 || I=3680 || I=3685 || I=3687 || I=3697 || I=3702 || I=3711 ||
I=3713 || I=3714 || I=3715 || I=3718 || I=3722)
mw[11] = 1.0
endif

; define External /Arterial rows, columns
if (I >= 1 && I <= @LstITaz@) mw[22] = 1.0, include =
3676,3678,3679,3681,3682,3683,3684,3686,3688,3689,3690,3691,3692,3693,3694,3695,
3696,3698,3699,3700,3701,3703,3704,3705,3706,3707,3708,3709,3710,3712,3716,3717,3719
,3720,3721
if (I=3676 || I=3678 || I=3679 || I=3681 || I=3682 || I=3683 || I=3684 || I=3686 ||
I=3688 || I=3689 || I=3690 || I=3691 || I=3692 ||
I=3693 || I=3694 || I=3695 || I=3696 || I=3698 || I=3699 || I=3700 || I=3701 ||
I=3703 || I=3704 || I=3705 || I=3706 || I=3707 ||
I=3708 || I=3709 || I=3710 || I=3712 || I=3716 || I=3717 || I=3719 || I=3720 ||
I=3721)
mw[22] = 1.0
endif

;
; Apply 'screen' matrices to separate external Int/Art matrices and combine in one
matrix
MW[107] = (MW[101] * mw[11]) + (MW[102] * mw[22]) ; Final HBW External trip
tables
MW[207] = (MW[201] * mw[11]) + (MW[202] * mw[22]) ; HBS External trip
tables
MW[307] = (MW[301] * mw[11]) + (MW[302] * mw[22]) ; HBO External trip
tables
MW[407] = (MW[401] * mw[11]) + (MW[402] * mw[22]) ; NHW External trip
tables
MW[507] = (MW[403] * mw[11]) + (MW[404] * mw[22]) ; NHO External trip
tables

;
; Compute Total ExtPsn Trips matrix
MW[108] = MW[107] ; Total external HBW Motorized Person Trip tabs
MW[208] = MW[207] ; Final external HBS Motorized Person Trip tabs
MW[308] = MW[307] ; Final external HBO Motorized Person Trip tabs
MW[408] = MW[407] ; Final external NHW Motorized Person Trip tabs
MW[508] = MW[507] ; Final external NHO Motorized Person Trip tabs
MW[608] = MW[601] ; Final external Commercial Vehicle Trips
MW[708] = MW[701] ; Final external Medium Truck Trips
MW[808] = MW[801] ; Final external Heavy Truck Trips

; write out final matrices comprehensive tabs

```

## Appendix C Cube Voyager Scripts

```

MATO[1] = @HBWTDOUT@ , MO=108,name=HBWExtpsn
MATO[2] = @HBSTDOUT@ , MO=208,name=HBSExtpsn
MATO[3] = @HBOTDOUT@ , MO=308,name=HBOExtpsn
MATO[4] = @NHWTDOUT@ , MO=408,name=NHWEExtpsn
MATO[5] = @NHOTDOUT@ , MO=508,name=NHOExtpsn
MATO[6] = @COMTDOUT@ , MO=608,name=COMExt
MATO[7] = @MTKTDOUT@ , MO=708,name=MTKExt
MATO[8] = @HTKTDOUT@ , MO=808,name=HTKExt

ENDRUN
;
;-----
;-----
;
; Standard 23x23 Summaries
; Trip Distribution (HBW,HBS,HBO,NHB,COM,MTK,HTK) and formats
; them in neat jurisdictional summaries (23x23)
;
;-----
;-----
;-----

COPY FILE=DJ.EQV
; -- Start of Jurisdiction-to-TAZ equivalency --
D 1=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; 0 DC Core
D 2=5,48,51,64,66-180,210-281,288-373,382-393 ; 0 DC Noncore
D 3=394-769 ; 1 Montgomery
D 4=771-776,778-1404 ; 2 Prince George
D 5=1471-1476, 1486-1489, 1495-1497 ; 3 ArlCore
D 6=1405-1470,1477-1485,1490-1494,1498-1545 ; 3 ArlNCore
D 7=1546-1610 ; 4 Alex
D 8=1611-2159 ; 5 FFX
D 9=2160-2441 ; 6 LDn
D 10=2442-2554,2556-2628,2630-2819 ; 7 PW
D 11=2820-2949 ; 9 Frd
D 12=3230-3265,3268-3287 ; 14 Car.
D 13=2950-3017 ; 10 How.
D 14=3018-3102,3104-3116 ; 11 AnnAr
D 15=3288-3334 ; 15 Calv
D 16=3335-3409 ; 16 StM
D 17=3117-3229 ; 12 Chs.
D 18=3604-3653 ; 21 Fau
D 19=3449-3477,3479-3481,3483-3494,3496-3541 ; 19 Stf.
D 20=3654-3662,3663-3675 ; 22/23 Clk,Jeff.
D 21=3435-3448,3542-3543,3545-3603 ; 18/20 Fbg,Spots
D 22=3410-3434 ; 17 KG.
D 23=3676-3722 ; Externals
; -- end of Jurisdiction-to-TAZ equivalency --
ENDCOPY

RUN PGM=MATRIX
ZONES=@ZONESIZE@
MATI[1]= @HBWTDOUT@
MATI[2]= @HBSTDOUT@
MATI[3]= @HBOTDOUT@
MATI[4]= @NHWTDOUT@
MATI[5]= @NHOTDOUT@
MATI[6]= @COMTDOUT@
MATI[7]= @MTKTDOUT@
MATI[8]= @HTKTDOUT@

MW[1] = MI.1.1 ; HBW TRIP TABLE/TAZ-LEVEL
MW[2] = MI.2.1 ; HBS TRIP TABLE/TAZ-LEVEL
MW[3] = MI.3.1 ; HBO TRIP TABLE/TAZ-LEVEL
MW[4] = MI.4.1 ; NHW TRIP TABLE/TAZ-LEVEL

MW[5] = MI.5.1 ; NHO TRIP TABLE/TAZ-LEVEL
MW[6] = MI.6.1 ; COM TRIP TABLE/TAZ-LEVEL
MW[7] = MI.7.1 ; MTK TRIP TABLE/TAZ-LEVEL
MW[8] = MI.8.1 ; HTK TRIP TABLE/TAZ-LEVEL

; -- PLACEMARKER TABLES - FUTURE WORK
MW[11] = 0 ; HBW TRIP TABLE/TAZ-LEVEL
MW[12] = 0 ; HBS TRIP TABLE/TAZ-LEVEL
MW[13] = 0 ; HBO TRIP TABLE/TAZ-LEVEL
MW[14] = 0 ; NHB TRIP TABLE/TAZ-LEVEL
MW[15] = 0 ; NHO TRIP TABLE/TAZ-LEVEL
MW[16] = 0 ; COM TRIP TABLE/TAZ-LEVEL
MW[17] = 0 ; MTK TRIP TABLE/TAZ-LEVEL
MW[18] = 0 ; HTK TRIP TABLE/TAZ-LEVEL

FILEO MATO[1] = HBW.SQZ MO=1,11 ; OUTPUT HBW TABLE(S), SQUEEZED
MATO[2] = HBS.SQZ MO=2,12 ; OUTPUT HBS TABLE(S), SQUEEZED
MATO[3] = HBO.SQZ MO=3,13 ; OUTPUT HBO TABLE(S), SQUEEZED
MATO[4] = NHW.SQZ MO=4,14 ; OUTPUT NHW TABLE(S), SQUEEZED
MATO[5] = NHO.SQZ MO=5,15 ; OUTPUT NHO TABLE(S), SQUEEZED
MATO[6] = COM.SQZ MO=6,16 ; OUTPUT COM TABLE(S), SQUEEZED
MATO[7] = MTK.SQZ MO=7,17 ; OUTPUT MTK TABLE(S), SQUEEZED
MATO[8] = HTK.SQZ MO=8,18 ; OUTPUT HTK TABLE(S), SQUEEZED

; renumber OUT.MAT according to DJ.EQV
RENUMBER FILE=DJ.EQV, MISSINGZI=M, MISSINGZO=W
ENDRUN

;
; LOOP PURP=1,8 ; Loop for Each Purpose
;
; Global Variables:
; SQFNAME Name of squeezed modal trip table(s)
; DESCRIPT Description
; PURPOSE Purpose
; MODE Mode
; DCML Decimal specification
; TABTYPE Table type(1/2), i.e.,-involves 1 or 2 trip tables
; SCALE=1 Scale factor to be applied (if desired)
; OPER='+' Operation(if tabtype=2) Tab1(?)Tab2=Result
;
;
; DESCRIPT = 'SIMULATION-%_iter_% Itr Year: %_year_% Alt: %_alt_%'
IF (PURP=1)
SQFNAME = 'HBW.SQZ'
PURPOSE = 'HBW'
MODE = 'External PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=2)
SQFNAME = 'HBS.SQZ'
PURPOSE = 'HBS'
MODE = 'External PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=3)
SQFNAME = 'HBO.SQZ'
PURPOSE = 'HBO'
MODE = 'External PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=4)

```





## Appendix C Cube Voyager Scripts

```

HBSK   = 'hbs_k.mat' ;
HBOK   = 'hbo_k.mat' ;
NHWK   = 'nhw_k.mat' ;
NHOK   = 'nho_k.mat' ;
;
;-----
;
;
FFsFile = '..\SUPPORT\ver23_f_factors.dbf' ; F-Factors for all modeled purposes
;;Variables in the dbf file:
; IMP   HBWINC1 HBWINC2 HBWINC3 HBWINC4 HBWEI HBWEA ;
;       HBSINC1 HBSINC2 HBSINC3 HBSINC4 HBSEI HBSEA ;
;       HBOINC1 HBOINC2 HBOINC3 HBOINC4 HBOEI HBOEA ;
;       NHW   NHO   NHBEI   NHBEA ;
;       ICOM  IMTK  IHTK   EXTCOM  EXTMK  EXHTK ;
;
;=====
; ALL Internal Motorized Ps AND As, by purpose
PsAs = '%_iter_%_Final_Int_Motor_PsAs.dbf'
;;Variables in dbf file:
; TAZ,   HBWIP,HBWIP1,HBWIP2,HBWIP3,HBWIP4,
;       HBWIA,HBWIA1,HBWIA2,HBWIA3,HBWIA4,
;       HBSIP,HBSIP1,HBSIP2,HBSIP3,HBSIP4,
;       HBSIA,HBSIA1,HBSIA2,HBSIA3,HBSIA4,
;       HBOIP,HBOIP1,HBOIP2,HBOIP3,HBOIP4,
;       HBOIA,HBOIA1,HBOIA2,HBOIA3,HBOIA4,
;       NHWIP,NHWIA, NHOIP, NHOIA,
;       COMIP,COMIA,
;       MTKIP,MTKIA,
;       HTKIP,HTKIA
;=====
; External trip tables, by purpose- developed in earlier trip distribution step
;
HBW_EXT_TRIPS = '%_iter_%_HBWext.PTT'
HBS_EXT_TRIPS = '%_iter_%_HBSext.PTT'
HBO_EXT_TRIPS = '%_iter_%_HBOext.PTT'
NHW_EXT_TRIPS = '%_iter_%_NHWext.PTT'
NHO_EXT_TRIPS = '%_iter_%_NHOext.PTT'
COM_EXT_TRIPS = '%_iter_%_COMext.VTT'
MTK_EXT_TRIPS = '%_iter_%_MTKext.VTT'
HTK_EXT_TRIPS = '%_iter_%_HTKext.VTT'
;
;; OUTPUT TRIP TABLES
HBWTDOUT = '%_iter_%_HBW.PTT';
HBSTDOUT = '%_iter_%_HBS.PTT';
HBOTDOUT = '%_iter_%_HBO.PTT';
NHWTDOUT = '%_iter_%_NHW.PTT';
NHOTDOUT = '%_iter_%_NHO.PTT';
COMTDOUT = '%_iter_%_COMMER.PTT';
MTKTDOUT = '%_iter_%_MTRUCK.PTT';
HTKTDOUT = '%_iter_%_HTRUCK.PTT';
;
;; OUTPUT matrices for mode choice model consumption
HBWforMC = '%_iter_%_hbw_NL.ptt' ; HBW Person Trips-4TABS (INC1..INC4)
HBSforMC = '%_iter_%_hbs_NL.ptt' ; HBS Person Trips-4TABS (INC1..INC4)
HBOforMC = '%_iter_%_hbo_NL.ptt' ; HBO Person Trips-4TABS (INC1..INC4)
NHWforMC = '%_iter_%_nhw_NL.ptt' ; NHW Person Trips-1TAB (INTERNAL)
NHOforMC = '%_iter_%_nho_NL.ptt' ; NHO Person Trips-1TAB (INTERNAL)

```

```

;-----
; Trip Distribution Model Calibration Process
;-----
;
; |////////////////////////////////////////////////////////////////|
; |/////////      Start HBW Trip Distribution Here:      |
; |////////////////////////////////////////////////////////////////|
;
RUN PGM=DISTRIBUTION
zones= 3722
MATI= HBWCI1_4.MAT, ; Composite Time Impedances HBW Inc.Levels 1-4 #1
@HBWK@, ; HBW Kfactors (Scaled by 1000.0) #2
am_sov_termIntraTime_e.skf, ; AM -PK Time skims for Extl trip dist.
#3
md_sov_termIntraTime_e.skf ; Midday Time skims for Extl trip dist.
#4
; Put income based impedance matrices in work tables 11-14
; tabs 11-14 are comp.time for inc.levels 1,2,3,4
; Put am, midday external impedances (hwy time) 21,31 respectively
FILLMW MW[11] = MI.1.1,2,3,4
MW[21] = MI.3.1
MW[31] = MI.4.1
; Put K-factor matrix in work table 20
; - K-factors are scaled by 1000s (eg, a mtx value of '1000'=1.0)
; - K-factors are applied across all HBW distributions
FILLMW MW[20] = MI.2.1
DUMMY = ROWFAC(20,0.001) ; scale k-factor's to 'true' units
ZDATI[1] = @PsAs@ ; internal productions file
; read friction factors file as lookup table
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
LOOKUP[1] = IMP, RESULT=HBWinc1, ;
LOOKUP[2] = IMP, RESULT=HBWinc2, ;
LOOKUP[3] = IMP, RESULT=HBWinc3, ;
LOOKUP[4] = IMP, RESULT=HBWinc4, ;
INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0
; Establish production and attraction vectors here:
SETPA P[1]=ZI.1.HBWMIP1, P[2]=ZI.1.HBWMIP2, P[3]=ZI.1.HBWMIP3, P[4]=ZI.1.HBWMIP4
SETPA A[1]=ZI.1.HBWMIA1, A[2]=ZI.1.HBWMIA2, A[3]=ZI.1.HBWMIA3, A[4]=ZI.1.HBWMIA4
MAXITERS = 15 ; specify GM iterations
MAXRMSE = 0.0001
; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[11], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 2, LOS=MW[12], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 3, LOS=MW[13], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 4, LOS=MW[14], FFACTORS= FF, KFACTORS = MW[20]
;REPORT ZDAT = Y
;REPORT ACOMP=1-4
MATO = HBW.TEM,MO=1-4 ; Final HBW trip table(s)
; T1 - HBW Inc. Level 1 (i-i)
; T2 - HBW Inc. Level 2 (i-i)
; T3 - HBW Inc. Level 3 (i-i)

```



## Appendix C Cube Voyager Scripts

```

;          T4 - HBW Inc. Level 4 (i-i)
ENDRUN

; --ENB HBW Trip Dist---;;
; |////////////////////////|
; |//////// Start HBS Trip Distribution Here: |
; |////////////////////////|

RUN PGM=DISTRIBUTION
zones= 3722
MATI= HBSCI1_4.MAT, ; Composite Time Impedances HBW Inc.Levels 1-4 #1
@HBSK@, ; HBW Kfactors (Scaled by 1000.0) #2
am_sov_termIntraTime_e.skf, ; AM -PK Time skims for Extl trip dist.
#3
md_sov_termIntraTime_e.skf ; Midday Time skims for Extl trip dist.
#4

; Put income based impedance matrices in work tables 11-14
; tabs 11-14 are comp.time for inc.levels 1,2,3,4
; Put am, midday external impedances (hwy time) 21,31 respectively

FILLMW MW[11] = MI.1.1,2,3,4
MW[21] = MI.3.1
MW[31] = MI.4.1

; Put K-factor matrix in work table 20
; - K-factors are scaled by 1000s (eg, a mtx value of '1000'=1.0)
; - K-factors are applied across all HBW distributions

FILLMW MW[20] = MI.2.1
DUMMY = ROWFAC(20,0.001) ; scale k-factor's to 'true' units

ZDATI[1] = @PsAs@ ; internal productions file

; read friction factors file as lookup table
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
LOOKUP[1] = IMP, RESULT=HBSinc1, ;
LOOKUP[2] = IMP, RESULT=HBSinc2, ;
LOOKUP[3] = IMP, RESULT=HBSinc3, ;
LOOKUP[4] = IMP, RESULT=HBSinc4, ;
INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

; Establish production and attraction vectors here:

SETPA P[1]=ZI.1.HBSMIP1, P[2]=ZI.1.HBSMIP2, P[3]=ZI.1.HBSMIP3, P[4]=ZI.1.HBSMIP4
SETPA A[1]=ZI.1.HBSMIA1, A[2]=ZI.1.HBSMIA2, A[3]=ZI.1.HBSMIA3, A[4]=ZI.1.HBSMIA4

MAXITERS = 27 ; specify GM iterations
MAXRMSE = 0.0001

; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[11], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 2, LOS=MW[12], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 3, LOS=MW[13], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 4, LOS=MW[14], FFACTORS= FF, KFACTORS = MW[20]

;REPORT ZDAT = Y
;REPORT ACOMP=1-4

MATO = HBS.TEM,MO=1-4 ; Final HBS trip table(s)
; T1 - HBS Inc. Level 1 (i-i)
; T2 - HBS Inc. Level 2 (i-i)
; T3 - HBS Inc. Level 3 (i-i)
; T4 - HBS Inc. Level 4 (i-i)

```

```

ENDRUN

; --ENB HBS Trip Dist---;;
; |////////////////////////|
; |//////// Start HBO Trip Distribution Here: |
; |////////////////////////|

RUN PGM=DISTRIBUTION
zones= 3722
MATI= HBOCI1_4.MAT, ; Composite Time Impedances HBW Inc.Levels 1-4 #1
@HBOK@, ; HBW Kfactors (Scaled by 1000.0) #2
am_sov_termIntraTime_e.skf, ; AM -PK Time skims for Extl trip dist.
#3
md_sov_termIntraTime_e.skf ; Midday Time skims for Extl trip dist.
#4

; Put income based impedance matrices in work tables 11-14
; tabs 11-14 are comp.time for inc.levels 1,2,3,4
; Put am, midday external impedances (hwy time) 21,31 respectively

FILLMW MW[11] = MI.1.1,2,3,4 ; comp. imp mw tabs 11-14
MW[21] = MI.3.1 ;
MW[31] = MI.4.1 ;

; Put K-factor matrix in work table 20
; - K-factors are scaled by 1000s (eg, a mtx value of '1000'=1.0)
; - K-factors are applied across all HBW distributions

FILLMW MW[20] = MI.2.1
DUMMY = ROWFAC(20,0.001) ; scale k-factor's to 'true' units

ZDATI[1] = @PsAs@ ; internal productions file

; read friction factors file as lookup table
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
LOOKUP[1] = IMP, RESULT=HBOinc1, ;
LOOKUP[2] = IMP, RESULT=HBOinc2, ;
LOOKUP[3] = IMP, RESULT=HBOinc3, ;
LOOKUP[4] = IMP, RESULT=HBOinc4, ;
INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

; Establish production and attraction vectors here:

SETPA P[1]=ZI.1.HBOMIP1, P[2]=ZI.1.HBOMIP2, P[3]=ZI.1.HBOMIP3, P[4]=ZI.1.HBOMIP4
SETPA A[1]=ZI.1.HBOMIA1, A[2]=ZI.1.HBOMIA2, A[3]=ZI.1.HBOMIA3, A[4]=ZI.1.HBOMIA4

MAXITERS = 27 ; specify GM iterations
MAXRMSE = 0.0001

; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[11], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 2, LOS=MW[12], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 3, LOS=MW[13], FFACTORS= FF, KFACTORS = MW[20]
GRAVITY PURPOSE = 4, LOS=MW[14], FFACTORS= FF, KFACTORS = MW[20]

;REPORT ZDAT = Y
;REPORT ACOMP=1-6

MATO = HBO.TEM,MO=1-4 ; Final HBO trip table(s)
; T1 - HBO Inc. Level 1 (i-i)
; T2 - HBO Inc. Level 2 (i-i)
; T3 - HBO Inc. Level 3 (i-i)
; T4 - HBO Inc. Level 4 (i-i)

```

## Appendix C Cube Voyager Scripts

```

ENDRUN

;; --ENB HBO Trip Dist---;;
;

; |////////////////////////////////////////////////////////////////|
; |//////// Start NHW/NHO Trip Distribution Here: |////////|
; |////////////////////////////////////////////////////////////////|

RUN PGM=DISTRIBUTION
zones= 3722
MATI= NHBCI.MAT, ; Composite Time Impedances NHW/NHO T1&2 file 1
      md_sov_termIntraTime_e.skf, ; Midday Time skims for Extl trip dist.
file 2
      @NHWK@, ; NHW Kfactors (Scaled by 1000.0) file 3
      @NHOK@ ; NHO Kfactors (Scaled by 1000.0) file 4

; Put nhw, nho impedance matrices in work tables 11-12
FILLMW MW[11] = MI.1.1,2
; Put extl impedance matrices in work tables 31
mw[31] = mi.2.1

; Put K-factor matrix in work table 20
; - K-factors are scaled by 1000s (eg, a mtx value of '1000'=1.0)
; - K-factors are applied across all HBS distributions

FILLMW MW[20] = MI.3.1
FILLMW MW[21] = MI.4.1
DUMMY = ROWFAC(20,0.001) ; scale k-factor's to 'true' units
DUMMY = ROWFAC(21,0.001) ; scale k-factor's to 'true' units
; Variables in the ZDATI files:
ZDATI[1] = @PsAs@ ; internal productions file

; read friction factors file as lookup table
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
      LOOKUP[1] = IMP, RESULT=NHW, ;
      LOOKUP[2] = IMP, RESULT=NHO, ;
      INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

; Establish production and attraction vectors here:

SETPA P[1]=ZI.1.NHWMIA, P[2]=ZI.1.NHOMIA
SETPA A[1]=ZI.1.NHWMIA, A[2]=ZI.1.NHOMIA

MAXITERS = 9 ; specify GM iterations
MAXRMSE = 0.0001

; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[11], FFACTORS= FF, KFACTORS = MW[20]
;; NHW INTL
GRAVITY PURPOSE = 2, LOS=MW[12], FFACTORS= FF, KFACTORS = MW[21]
;; NHO INTL

;REPORT ZDAT = Y
;REPORT ACOMP=1-2

MATO = NHB.TEM,MO=1-2 ; Final NHB trip table(s)
; ; T1 - NHW INTL
; ; T2 - NHO INTL

ENDRUN
;; --ENB NHB Trip Dist---;;
; |////////////////////////////////////////////////////////////////|
; |//////// Start COM/TRK Trip Distribution Here: |////////|
; |////////////////////////////////////////////////////////////////|

RUN PGM=TRIPDIST

```

```

MATI[1] = %_prev_% MD_SOV.SKM ; Off-Pk Time Imped. for COM
MATI[2] = %_prev_% MD_TRUCK.skm ; Off-Pk Truck Time for MTK/HTK
MATI[3] = md_sov_termIntraTime_e.skf ; Midday Time skims for Extl trip
dist.

; Put impedance matrices in work tables 11-12. Tab 11 is for COM
; trips; tab 12 is for MTK and HTK trips. All time values are in minutes.

MW[11] = MI.1.1 ; com veh los matix
MW[12] = MI.2.1 ; trk los matrix
MW[13] = MI.3.1 ; extl los matrix

ZDATI[1] = @PsAs@

; FFactors
FileI LOOKUPI[1] = "@FFsFile@"
LOOKUP LOOKUPI=1, NAME=FF,
      LOOKUP[1] = IMP, RESULT=ICOM, ; CVInt
      LOOKUP[2] = IMP, RESULT=IMTK, ; MTrk Intl
      LOOKUP[3] = IMP, RESULT=IHTK, ; HTrk Intl
      INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

; Establish production and attraction vectors here:

SETPA P[1]=ZI.1.COMIA, P[2]=ZI.1.MTKIA, P[3]=ZI.1.HTKIA
SETPA A[1]=ZI.1.COMIA, A[2]=ZI.1.MTKIA, A[3]=ZI.1.HTKIA

MAXITERS = 9 ; specify GM iterations
MAXRMSE = 0.0001

; Establish gravity model run files & parameters
GRAVITY PURPOSE = 1, LOS=MW[11], FFACTORS= FF ; COM I/I
GRAVITY PURPOSE = 2, LOS=MW[12], FFACTORS= FF ; MTK I/I
GRAVITY PURPOSE = 3, LOS=MW[12], FFACTORS= FF ; HTK I/I

MATO[1] = COM.TEM,MO=1 ; Final COM trip tables: 1 = I/I
MATO[2] = MTK.TEM,MO=2 ; Final MTK trip tables: 1 = I/I
MATO[3] = HTK.TEM,MO=3 ; Final HTK trip tables: 1 = I/I

ENDRUN

; End COM/TRK Trip Distribution ---
;-----
; Now splice the internal trip tables developed above with external trip tables
developed -
; in the earlier trip distribution step
-
;-----

RUN PGM=MATRIX
ZONES = @ZONESIZE@
;
; internal trip tables developed above
MATI[1] = HBW.TEM ; 4 HBW trip tables: Incl, ..., Inc4
MATI[2] = HBS.TEM ; 4 HBS trip tables: Incl, ..., Inc4
MATI[3] = HBO.TEM ; 4 HBO trip tables: Incl, ..., Inc4
MATI[4] = NHB.TEM ; 2 NHB trip tables: NHW I-I, NHO I-I
MATI[5] = COM.TEM ; 1 Com trip tables: I/I
MATI[6] = MTK.TEM ; 1 Mtk trip tables: I/I
MATI[7] = HTK.TEM ; 1 Htk trip tables: I/I

MATI[8] = @HBW_EXT_TRIPS@ ; external HBW trips developed previously
MATI[9] = @HBS_EXT_TRIPS@ ; external HBS trips
MATI[10] = @HBO_EXT_TRIPS@ ; external HBO trips
MATI[11] = @NHW_EXT_TRIPS@ ; external NHW trips
MATI[12] = @NHO_EXT_TRIPS@ ; external NHO trips
MATI[13] = @COM_EXT_TRIPS@ ; external COM trips

```

## Appendix C Cube Voyager Scripts

```

MATI[14] = @MTK_EXT_TRIPS@ ; external MTK trips
MATI[15] = @HTK_EXT_TRIPS@ ; external HTK trips
;

FillMW MW[101]=mi.1.1,2,3,4 ; HBW tabs in mw 101-104
FillMW MW[201]=mi.2.1,2,3,4 ; HBS tabs in mw 201-204
FillMW MW[301]=mi.3.1,2,3,4 ; HBO tabs in mw 301-304
FillMW MW[401]=mi.4.1 ; NHW tabs in mw 401
FillMW MW[501]=mi.4.2 ; NHO tabs in mw 402
MW[601] =mi.5.1 ; Com tabs in mw 601
MW[701] =mi.6.1 ; Mtk tabs in mw 701
MW[801] =mi.7.1 ; Htk tabs in mw 801

MW[105] =mi.8.1 ; Ext HBW tabs in mw 105
MW[205] =mi.9.1 ; Ext HBS tabs in mw 205
MW[305] =mi.10.1 ; Ext HBO tabs in mw 305
MW[405] =mi.11.1 ; Ext NHW tabs in mw 405
MW[505] =mi.12.1 ; Ext NHO tabs in mw 505
MW[605] =mi.13.1 ; Ext COM tabs in mw 605
MW[705] =mi.14.1 ; Ext MTK tabs in mw 705
MW[805] =mi.15.1 ; Ext HTK tabs in mw 805

;;
;;Compute Total Person Trips matrix
MW[901] = MW[101] + MW[102] + MW[103] + MW[104] + MW[105] ; Total HBW
Motorized Person Trip tabs (II,IX,XI)
MW[902] = MW[201] + MW[202] + MW[203] + MW[204] + MW[205] ; Final HBS
Motorized Person Trip tabs (II,IX,XI)
MW[903] = MW[301] + MW[302] + MW[303] + MW[304] + MW[305] ; Final HBO
Motorized Person Trip tabs (II,IX,XI)
MW[904] = MW[401] + MW[405] ; Final NHW
Motorized Person Trip tabs (II,IX,XI)
MW[905] = MW[501] + MW[505] ; Final NHO
Motorized Person Trip tabs (II,IX,XI)
MW[906] = MW[601] + MW[605] ; Final Commercial
Vehicle Trips (II,IX,XI)
MW[907] = MW[701] + MW[705] ; Final Medium Truck
Trips (II,IX,XI)
MW[908] = MW[801] + MW[805] ; Final Heavy Truck
Trips (II,IX,XI)

; ; write out final matrices comprehensive tabs
MATO[1] = @HBWTDOUT@ ,
MO=101,102,103,104,105,901,name=HBWI1Psn,HBWI2Psn,HBWI3Psn,HBWI4Psn,HBW_Ext,HBWAllPsn
MATO[2] = @HBSTDOUT@ ,
MO=201,202,203,204,205,902,name=HBSI1Psn,HBSI2Psn,HBSI3Psn,HBSI4Psn,HBS_Ext,HBSAllPsn
MATO[3] = @HBOTDOUT@ ,
MO=301,302,303,304,305,903,name=HBOI1Psn,HBOI2Psn,HBOI3Psn,HBOI4Psn,HBO_Ext,HBOAllPsn
MATO[4] = @NHWTDOUT@ , MO=401,405,904 ,name=NHWPsII,NHWPsExt
NHWAllPsn
MATO[5] = @NHOTDOUT@ , MO=501,505,905 ,name=NHOPsnII,NHOPsnExt
NHOAllPsn
MATO[6] = @COMTDOUT@ , MO=601,605,906 ,name=COM_Int ,COM_Ext,
COMAllVeh
MATO[7] = @MTKTDOUT@ , MO=701,705,907 ,name=MTK_Int ,MTK_Ext,
MTKAllVeh
MATO[8] = @HTKTDOUT@ , MO=801,805,908 ,name=HTK_Int ,HTK_Ext,
HTKAllVeh

; ; write out final matrices for mode choice model consumption

```

```

MATO[9] = @HBWforMC@ ,MO=101,102,103,104,name=HBWI1Psn,HBWI2Psn,HBWI3Psn,HBWI4Psn
; HBW Person Trips-4TABS (INCL..INC4)
MATO[10] = @HBSforMC@ ,MO=201,202,203,204,name=HBSI1Psn,HBSI2Psn,HBSI3Psn,HBSI4Psn
; HBS Person Trips-4TABS (INCL..INC4)
MATO[11] = @HBOforMC@ ,MO=301,302,303,304,name=HBOI1Psn,HBOI2Psn,HBOI3Psn,HBOI4Psn
; HBO Person Trips-4TABS (INCL..INC4)
MATO[12] = @NHWforMC@ ,MO=401 ,name=NHWPsII
; NHW Person Trips-1TAB (INTERNAL)
MATO[13] = @NHOforMC@ ,MO=501 ,name=NHOPsnII
; NHO Person Trips-1TAB (INTERNAL)

ENDRUN
;
;=====
;
;-----
;
; Standard 23x23 Summaries
; Trip Distribution (HBW,HBS,HBO,NHW,COM,MTK,HTK) and formats
; them in neat jurisdictional summaries (23x23)
;
;-----
;-----
;-----
COPY FILE=DJ.EQV
; -- Start of Jurisdiction-to-TAZ equivalency --
D 1=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ; 0 DC Core
D 2=5,48,51,64,66-180,210-281,288-373,382-393 ; 0 DC Noncore
D 3=394-769 ; 1 Montgomery
D 4=771-776,778-1404 ; 2 Prince George
D 5=1471-1476, 1486-1489, 1495-1497 ; 3 ArlCore
D 6=1405-1470,1477-1485,1490-1494,1498-1545 ; 3 ArlNCore
D 7=1546-1610 ; 4 Alex
D 8=1611-2159 ; 5 FFx
D 9=2160-2441 ; 6 LDn
D 10=2442-2554,2556-2628,2630-2819 ; 7 PW
D 11=2820-2949 ; 9 Frd
D 12=3230-3265,3268-3287 ; 14 Car.
D 13=2950-3017 ; 10 How.
D 14=3018-3102,3104-3116 ; 11 AnnAr
D 15=3288-3334 ; 15 Calv
D 16=3335-3409 ; 16 StM
D 17=3117-3229 ; 12 Chs.
D 18=3604-3653 ; 21 Fau
D 19=3449-3477,3479-3481,3483-3494,3496-3541 ; 19 Stf.
D 20=3654-3662,3663-3675 ; 22/23 Clk,Jeff.
D 21=3435-3448,3542-3543,3545-3603 ; 18/20 Fbg,Spots
D 22=3410-3434 ; 17 KG.
D 23=3676-3722 ; Externals
; -- end of Jurisdiction-to-TAZ equivalency --
ENDCOPY

RUN PGM=MATRIX
ZONES=@ZONESIZE@
MATI[1] = @HBWTDOUT@
MATI[2] = @HBSTDOUT@
MATI[3] = @HBOTDOUT@
MATI[4] = @NHWTDOUT@
MATI[5] = @NHOTDOUT@
MATI[6] = @COMTDOUT@
MATI[7] = @MTKTDOUT@
MATI[8] = @HTKTDOUT@

MW[1] = MI.1.6 ; HBW TRIP TABLE/TAZ-LEVEL
MW[2] = MI.2.6 ; HBS TRIP TABLE/TAZ-LEVEL
MW[3] = MI.3.6 ; HBO TRIP TABLE/TAZ-LEVEL

```

## Appendix C Cube Voyager Scripts

```

MW[4] = MI.4.3      ; NHW TRIP TABLE/TAZ-LEVEL
MW[5] = MI.5.3      ; NHO TRIP TABLE/TAZ-LEVEL
MW[6] = MI.6.3      ; COM TRIP TABLE/TAZ-LEVEL
MW[7] = MI.7.3      ; MTK TRIP TABLE/TAZ-LEVEL
MW[8] = MI.8.3      ; HTK TRIP TABLE/TAZ-LEVEL

; -- PLACEMARKER TABLES - FUTURE WORK
MW[11] = 0 ;          HBW TRIP TABLE/TAZ-LEVEL
MW[12] = 0 ;          HBS TRIP TABLE/TAZ-LEVEL
MW[13] = 0 ;          HBO TRIP TABLE/TAZ-LEVEL
MW[14] = 0 ;          NHB TRIP TABLE/TAZ-LEVEL
MW[15] = 0 ;          NHB TRIP TABLE/TAZ-LEVEL
MW[16] = 0 ;          COM TRIP TABLE/TAZ-LEVEL
MW[17] = 0 ;          MTK TRIP TABLE/TAZ-LEVEL
MW[18] = 0 ;          HTK TRIP TABLE/TAZ-LEVEL

FILEO MATO[1] = HBW.SQZ MO=1,11 ; OUTPUT HBW TABLE(S), SQUEEZED
MATO[2] = HBS.SQZ MO=2,12 ; OUTPUT HBS TABLE(S), SQUEEZED
MATO[3] = HBO.SQZ MO=3,13 ; OUTPUT HBO TABLE(S), SQUEEZED
MATO[4] = NHW.SQZ MO=4,14 ; OUTPUT NHW TABLE(S), SQUEEZED
MATO[5] = NHO.SQZ MO=5,15 ; OUTPUT NHO TABLE(S), SQUEEZED
MATO[6] = COM.SQZ MO=6,16 ; OUTPUT COM TABLE(S), SQUEEZED
MATO[7] = MTK.SQZ MO=7,17 ; OUTPUT MTK TABLE(S), SQUEEZED
MATO[8] = HTK.SQZ MO=8,18 ; OUTPUT HTK TABLE(S), SQUEEZED

; renumber OUT.MAT according to DJ.EQV
RENUMBER FILE=DJ.EQV, MISSINGZI=M, MISSINGZO=W
ENDRUN

;
LOOP PURP=1,8 ; Loop for Each Purpose
;
; Global Variables:
; SQFNAME Name of squeezed modal trip table(s)
; DESCRIPT Description
; PURPOSE Purpose
; MODE Mode
; DCML Decimal specification
; TABTYPE Table type(1/2), i.e.,-involves 1 or 2 trip tables
; SCALE=1 Scale factor to be applied (if desired)
; OPER='+' Operation(if tabtype=2) Tab1(?)Tab2=Result
;
DESCRIPT = 'SIMULATION-%_iter_% Itr Year: %_year_% Alt: %_alt_%'
IF (PURP=1)
SQFNAME = 'HBW.SQZ'
PURPOSE = 'HBW'
MODE = 'MOTORIZED PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=2)
SQFNAME = 'HBS.SQZ'
PURPOSE = 'HBS'
MODE = 'MOTORIZED PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=3)
SQFNAME = 'HBO.SQZ'
PURPOSE = 'HBO'
MODE = 'MOTORIZED PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=4)
SQFNAME = 'NHW.SQZ'
PURPOSE = 'NHW'
MODE = 'MOTORIZED PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=5)
SQFNAME = 'NHO.SQZ'
PURPOSE = 'NHO'
MODE = 'MOTORIZED PERSON'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=6)
SQFNAME = 'COM.SQZ'
PURPOSE = 'COM'
MODE = 'COMMERCIAL VEH'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=7)
SQFNAME = 'MTK.SQZ'
PURPOSE = 'MTK'
MODE = 'TRUCKS'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ELSEIF (PURP=8)
SQFNAME = 'HTK.SQZ'
PURPOSE = 'HTK'
MODE = 'TRUCKS'
DCML = 0
TABTYPE = 1
SCALE = 1
OPER = '+'
ENDIF
;
RUN PGM=MATRIX
PAGEheight=32000
ZONES=23
FILEI MATI=@SQFNAME@
ARRAY CSUM=23,CSUM1=23,CSUM2=23
; -----
; -- Table Cell Value decalaration or computation (in MW[1])
; -----
FILLMW MW[1]=MI.1.1,2 ; read input tables in MW 2,3
IF (@TABTYPE@ = 2)
FILLMW MW[2]=MI.1.1,2 ; read input tables in MW 2,3
ENDIF
IF (@TABTYPE@=2) ; Cell Value
JLOOP ; computed for
IF (MW[3][J]>0) MW[1]=MW[2]*@SCALE@@OPER@MW[3]; special summaries-
ENDJLOOP ; calculation in MW[1]
ENDIF
; -----
; --- ROW Marginal declaration or computation ---
; -----
RSUM = ROWSUM(1) ; 'normal' table- row summary value

```

## Appendix C Cube Voyager Scripts

```

IF (@TABTYPE@=2)
    RSUM = @SCALE@*ROWSUM(2)@OPER@ROWSUM(3) ; non-'normal' table
ENDIF ; compute the row marginal(%)

; -----
; --- COLUMN/Total Marginal Accumulation ---
; --- The computation (if necessary) is done below ---
; -----

JLOOP ; COL/Total Accumulation
CSUM[J] = CSUM[J] + MW[1][J] ; for 'normal' table
TOTAL = TOTAL + MW[1] ;
ENDJLOOP

IF (@TABTYPE@=2)
JLOOP ; COL/Total Accumulation
CSUM1[J] = CSUM1[J] + MW[2][J] ; for non-'normal' Table
TOTAL1 = TOTAL1 + MW[2] ;
CSUM2[J] = CSUM2[J] + MW[3][J] ;
TOTAL2 = TOTAL2 + MW[3] ;
ENDJLOOP
ENDIF

IF (I==1) ; print header

PRINT LIST='/bt ', '@DESCRIPT@'
PRINT LIST=' ', 'Purpose: ', '@PURPOSE@', ' MODE: ', '@MODE@'
PRINT LIST=' '

PRINT LIST=' DESTINATION'
PRINT LIST=' ORIGIN |',
' 1', ' 2', ' 3', ' 4',
' 5', ' 6', ' 7', ' 8', ' 9',
' 10', ' 11', ' 12', ' 13', ' 14',
' 15', ' 16', ' 17', ' 18', ' 19',
' 20', ' 21', ' 22', ' 23', ' | TOTAL'

PRINT LIST='=====',
'=====',
'=====',
'=====',
'=====',
'=====',
'=====',
'=====',

ENDIF

IF (I=1)
CURDIST=STR(I,2,1)+' DC CR'+ '| ' ; Make row header
ELSEIF (I=2)
CURDIST=STR(I,2,1)+' DC NC'+ '| ' ; Make row header
ELSEIF (I=3)
CURDIST=STR(I,2,1)+' MTG '+ '| ' ; Make row header
ELSEIF (I=4)
CURDIST=STR(I,2,1)+' PG '+ '| ' ; Make row header
ELSEIF (I=5)
CURDIST=STR(I,2,1)+' ARLCR'+ '| ' ; Make row header
ELSEIF (I=6)
CURDIST=STR(I,2,1)+' ARNCR'+ '| ' ; Make row header
ELSEIF (I=7)
CURDIST=STR(I,2,1)+' ALX '+ '| ' ; Make row header
ELSEIF (I=8)
CURDIST=STR(I,2,1)+' FFX '+ '| ' ; Make row header
ELSEIF (I=9)
CURDIST=STR(I,2,1)+' LDN '+ '| ' ; Make row header
ELSEIF (I=10)
CURDIST=STR(I,2,1)+' PW '+ '| ' ; Make row header

```

```

ELSEIF (I=11)
CURDIST=STR(I,2,1)+' FRD '+ '| ' ; Make row header
ELSEIF (I=12)
CURDIST=STR(I,2,1)+' CAR '+ '| ' ; Make row header
ELSEIF (I=13)
CURDIST=STR(I,2,1)+' HOW '+ '| ' ; Make row header
ELSEIF (I=14)
CURDIST=STR(I,2,1)+' AAR '+ '| ' ; Make row header
ELSEIF (I=15)
CURDIST=STR(I,2,1)+' CAL '+ '| ' ; Make row header
ELSEIF (I=16)
CURDIST=STR(I,2,1)+' STM '+ '| ' ; Make row header
ELSEIF (I=17)
CURDIST=STR(I,2,1)+' CHS '+ '| ' ; Make row header
ELSEIF (I=18)
CURDIST=STR(I,2,1)+' FAU '+ '| ' ; Make row header
ELSEIF (I=19)
CURDIST=STR(I,2,1)+' STA '+ '| ' ; Make row header
ELSEIF (I=20)
CURDIST=STR(I,2,1)+' CL/JF'+ '| ' ; Make row header
ELSEIF (I=21)
CURDIST=STR(I,2,1)+' SP/FB'+ '| ' ; Make row header
ELSEIF (I=22)
CURDIST=STR(I,2,1)+' KGEO '+ '| ' ; Make row header
ELSEIF (I=23)
CURDIST=STR(I,2,1)+' EXTL '+ '| ' ; Make row header
ELSE ; (I=24)
CURDIST=STR(I,2,1)+' TOTAL'+ '| ' ; Make row header
ENDIF

PRINT FORM=7.@DCML@ LIST=CURDIST, MW[1][1],MW[1][2],MW[1][3],MW[1][4],MW[1][5],
MW[1][6],MW[1][7],MW[1][8],MW[1][9],MW[1][10],
MW[1][11],MW[1][12],MW[1][13],MW[1][14],MW[1][15],
MW[1][16],MW[1][17],MW[1][18],MW[1][19],MW[1][20],
MW[1][21],MW[1][22],MW[1][23], ' | ',RSUM

IF (I==ZONES)
; Now at the end of Processed zone matrix
; Do final Column/Grand Total Computations
IF (@TABTYPE@=2)
LOOP IDX = 1,ZONES
IF (CSUM2[IDX] = 0)
CSUM[IDX] = 0
ELSE
CSUM[IDX] = @SCALE@* CSUM1[IDX] @OPER@ CSUM2[IDX]
ENDIF
ENDLOOP
ENDIF
ENDIF
IF (@TABTYPE@=2 )
IF (TOTAL2 = 0)
TOTAL = 0
ELSE
TOTAL = @SCALE@ *TOTAL1 @OPER@ TOTAL2
ENDIF
ENDIF

; End of final Column/Grand Total Computations

PRINT LIST='=====',
'=====',
'=====',
'=====',
'=====',

PRINT FORM=8.@DCML@,
LIST=' TOTAL ', ' ',CSUM[1], ' ',CSUM[3],
' ',CSUM[5], ' ',CSUM[7], ' ',CSUM[9],

```

## Appendix C Cube Voyager Scripts

```

'      ',CSUM[11],'      ',CSUM[13],'      ',CSUM[15],
'      ',CSUM[17],'      ',CSUM[19],'      ',CSUM[21],
'      ',CSUM[23],'      '|
PRINT FORM=8.@DCML@,
LIST=' /et      ',CSUM[2],
'      ',CSUM[4],'      ',CSUM[6],'      ',CSUM[8],
'      ',CSUM[10],'      ',CSUM[12],'      ',CSUM[14],
'      ',CSUM[16],'      ',CSUM[18],'      ',CSUM[20],
'      ',CSUM[22],'      ',TOTAL(9.@DCML@)

ENDIF
ENDRUN

ENDLOOP ; End Loop

```

## 52 Trip\_Generation.s

```

*del voya*.prn
;=====
; Trip_Generation.s
; Version 2.3, 3722 TAZ System - Trip Generation Process
;
; RM
; Date:      2011-02-15
;
;=====
; Note: Jurisdictional adjustment for P's A's added 2/8/11
; with nonmotorized fix 2/15/11
; Updated Area Type&juris marginal adjustments 9/14/2011
;=====
; Corrected application of non-motorized area type adjustments 1/04/13
;=====
; Modified scaling procedure such that Attr. scaling is based on both Motr&Non-Motr
Prod.s
; - Previously Attr scaling was done separately, for motorized P's and for non-
motorized P's
;
; - Increase non-work, non-motorized trip rates by 30 percent in area types 1 and 2
;
; 3/4/13 Changes:
; - Removed I-X extraction process
; - Truncated Trip Gen process to stop after COMPUTED trip attractions are
calculated
; Scaling will be handled, after External Trip Distribution process.

; 3/13/13 applied 15% and 17% reduction on Loudoun County P,A Trip rates
respectively
; 3/14/13 fixed conditions where the final non-motor P/A share could exceed 1
;=====
;Parameters and file specifications:
;=====
ZONESIZE      = 3722      ; No. of TAZs
LastIzn       = 3675     ; Last Internal TAZ no.

JrCl          = 24       ; No. of Juris. Classes (transformed)
JURIS. Code 0-23 becomes 1-24)
ArCl          = 6        ; No. of Area Classe (ATypes)
SzCl          = 4        ; No. of HH Size Classes
InCl          = 4        ; No. of Income Classes
VaCl          = 4        ; No. of Veh Avail Classes

```

```

PrCL          = 5        ; No. of Trip Purposes

ZNFILE_IN1   = 'inputs\ZONE.dbf'      ; Input Zonal Land Use File
Ext_PsAs     = 'inputs\Ext_PsAs.dbf'  ; External Ps/As

ZNFILE_IN3   = 'AreaType_File.dbf'    ; Input Zonal Area Type File
from network building
ZNFILE_IN4   = '%_iter_%_Demo_Models_HHbyISV.dbf' ; HHs by Income Size Vehs
Avail

ZNFILE_IN5   = 'TripGen_LUFile.dbf'    ; Consolidated zonal input
file (Intermediate I/O file)

ReportFile   = '%_iter_%_Trip_Generation.txt' ; Trip Gen. Report file
TripPros     = '%_iter_%_Trip_Gen_Productions_Comp.dbf' ; Zonal Trip productions -
Initial /Computed by purpose
TripAttsCom  = '%_iter_%_Trip_Gen_Attractions_Comp.dbf' ; Zonal Trip Attractions -
Initial /Computed by purpose

ZNFILE_IN2   = 'inputs\GIS_variables.DBF' ; Input Zonal GIS
variable File
Prate_IN     = '..\support\weighted_trip_rates.dbf' ; Trip Prod. rates
NMPrate_in   = '..\support\NMPrates.dbf' ; NonMotorized Prod share
model coeffs.
NMRate_in    = '..\support\NMRates.dbf' ; NonMotorized Attr share
model coeffs.
Attrate_in   = '..\support\AttrRates.dbf' ; Trip attraction rates
IncRat_in    = '..\support\HBINCRAT.dbf' ; HB income shares

;; Area Type-Based Trip End Adjustments BY PURPOSE AND AREA TYPE
;;
;; MOTORIZED PRODUCTIONS
MHBWPAdj1=1.1358 MHBWPAdj2=1.1180 MHBWPAdj3=1.0554 MHBWPAdj4=0.9175
MHBWPAdj5=0.9577 MHBWPAdj6=0.9307 ;
MHBSPAdj1=0.8092 MHBSPAdj2=0.9504 MHBSPAdj3=1.0793 MHBSPAdj4=0.9059
MHBSPAdj5=1.0751 MHBSPAdj6=0.8620 ;
MHBOPAdj1=1.1067 MHBOPAdj2=1.1181 MHBOPAdj3=1.0303 MHBOPAdj4=0.9647
MHBOPAdj5=1.0109 MHBOPAdj6=0.8324 ;
MNHWPAdj1=1.0000 MNHWPAdj2=1.0000 MNHWPAdj3=1.0000 MNHWPAdj4=1.0000
MNHWPAdj5=1.0000 MNHWPAdj6=1.0000 ;
MNHOPAdj1=1.0000 MNHOPAdj2=1.0000 MNHOPAdj3=1.0000 MNHOPAdj4=1.0000
MNHOPAdj5=1.0000 MNHOPAdj6=1.0000 ;

;; MOTORIZED ATTRACTIONS
MHBWAAdj1=1.0765 MHBWAAdj2=0.8478 MHBWAAdj3=0.9612 MHBWAAdj4=1.1045
MHBWAAdj5=0.9871 MHBWAAdj6=1.0383 ;
MHBAAAdj1=0.7952 MHBAAAdj2=1.0967 MHBAAAdj3=1.1577 MHBAAAdj4=0.8770
MHBAAAdj5=0.9437 MHBAAAdj6=0.5187 ;
MHBAAAdj1=1.1542 MHBAAAdj2=1.1304 MHBAAAdj3=0.9307 MHBAAAdj4=1.0635
MHBAAAdj5=1.0480 MHBAAAdj6=0.8032 ;
MNHWAAdj1=1.1457 MNHWAAdj2=0.8686 MNHWAAdj3=0.9843 MNHWAAdj4=1.5731
MNHWAAdj5=1.1860 MNHWAAdj6=1.0919 ;
MNHOOAdj1=0.7953 MNHOOAdj2=1.0652 MNHOOAdj3=1.0724 MNHOOAdj4=0.9180
MNHOOAdj5=1.0899 MNHOOAdj6=0.7224 ;

;; NONMOTORIZED PRODUCTIONS
;; original
;NHBWPAdj1=1.2600 NHBWPAdj2=1.0000 NHBWPAdj3=1.0000 NHBWPAdj4=1.0000
NHBWPAdj5=1.0000 NHBWPAdj6=1.0000 ;
;NHBSPAdj1=1.6700 NHBSPAdj2=1.4000 NHBSPAdj3=1.0000 NHBSPAdj4=1.0000
NHBSPAdj5=1.0000 NHBSPAdj6=1.0000 ;
;NHBOPAdj1=0.7000 NHBOPAdj2=1.0700 NHBOPAdj3=1.0000 NHBOPAdj4=1.0000
NHBOPAdj5=1.0000 NHBOPAdj6=1.0000 ;
;NNHWPAdj1=1.0000 NNHWPAdj2=1.0000 NNHWPAdj3=1.0000 NNHWPAdj4=1.0000
NNHWPAdj5=1.0000 NNHWPAdj6=1.0000 ;

```

## Appendix C Cube Voyager Scripts

```

;NHHOPAdj1=1.0000 NHHOPAdj2=1.0000 NHHOPAdj3=1.0000 NHHOPAdj4=1.0000
NHHOPAdj5=1.0000 NHHOPAdj6=1.0000 ;

;; revised - non-work rates in AT 1,2 raised by 30%
NHBWPAdj1=1.2600 NHBWPAdj2=1.0000 NHBWPAdj3=1.0000 NHBWPAdj4=1.0000
NHBWPAdj5=1.0000 NHBWPAdj6=1.0000 ;
NHBSPAdj1=2.1700 NHBSPAdj2=1.8200 NHBSPAdj3=1.0000 NHBSPAdj4=1.0000
NHBSPAdj5=1.0000 NHBSPAdj6=1.0000 ;
NHBOPAdj1=0.9100 NHBOPAdj2=1.3900 NHBOPAdj3=1.0000 NHBOPAdj4=1.0000
NHBOPAdj5=1.0000 NHBOPAdj6=1.0000 ;
NNHWPAdj1=1.3000 NNHWPAdj2=1.3000 NNHWPAdj3=1.0000 NNHWPAdj4=1.0000
NNHWPAdj5=1.0000 NNHWPAdj6=1.0000 ;
NNHOPAdj1=1.3000 NNHOPAdj2=1.3000 NNHOPAdj3=1.0000 NNHOPAdj4=1.0000
NNHOPAdj5=1.0000 NNHOPAdj6=1.0000 ;

;; NONMOTORIZED ATTRACTIONS
;; original
;NHBWAAdj1=1.0300 NHBWAAdj2=1.0000 NHBWAAdj3=1.1100 NHBWAAdj4=1.1100
NHBWAAdj5=1.1300 NHBWAAdj6=1.1000 ;
;NHBAAAdj1=1.8400 NHBAAAdj2=1.2900 NHBAAAdj3=1.0900 NHBAAAdj4=1.1000
NHBAAAdj5=1.0000 NHBAAAdj6=1.0000 ;
;NHBAAAdj1=0.6000 NHBAAAdj2=1.0600 NHBAAAdj3=1.1100 NHBAAAdj4=1.0900
NHBAAAdj5=1.1000 NHBAAAdj6=1.0800 ;
;NNHWAAdj1=1.0000 NNHWAAdj2=1.0000 NNHWAAdj3=1.0000 NNHWAAdj4=1.0000
NNHWAAdj5=1.0000 NNHWAAdj6=1.0000 ;
;NNHAAAdj1=1.6600 NNHAAAdj2=1.0000 NNHAAAdj3=0.7000 NNHAAAdj4=0.7000
NNHAAAdj5=0.7000 NNHAAAdj6=0.7000 ;

;; revised -non-work rates in AT 1,2 raised by 30%
NHBWAAdj1=1.0300 NHBWAAdj2=1.0000 NHBWAAdj3=1.1100 NHBWAAdj4=1.1100
NHBWAAdj5=1.1300 NHBWAAdj6=1.1000 ;
NHBAAAdj1=2.3900 NHBAAAdj2=1.6800 NHBAAAdj3=1.0900 NHBAAAdj4=1.1000
NHBAAAdj5=1.0000 NHBAAAdj6=1.0000 ;
NHBAAAdj1=0.7800 NHBAAAdj2=1.3800 NHBAAAdj3=1.1100 NHBAAAdj4=1.0900
NHBAAAdj5=1.1000 NHBAAAdj6=1.0800 ;
NNHWAAdj1=1.3000 NNHWAAdj2=1.3000 NNHWAAdj3=1.0000 NNHWAAdj4=1.0000
NNHWAAdj5=1.0000 NNHWAAdj6=1.0000 ;
NNHAAAdj1=2.1600 NNHAAAdj2=1.3000 NNHAAAdj3=0.7000 NNHAAAdj4=0.7000
NNHAAAdj5=0.7000 NNHAAAdj6=0.7000 ;

XNHW_Share = 0.41 ; Pct. of external NHB Auto Driver Trips that are NHW
(2007/08HTS)
XNHO_Share = 0.59 ; Pct. of external NHB Auto Driver Trips that are NHO
(2007/08HTS)

XOccHBW = 1.06 ; HBW External Auto occupancy assumption (2007/08HTS)
XOccHBS = 1.45 ; HBS External Auto occupancy assumption
XOccHBO = 1.63 ; HBO External Auto occupancy assumption
XOccNHW = 1.11 ; NHW External Auto occupancy assumption
XOccNHO = 1.50 ; NHO External Auto occupancy assumption

Ofmt = '(15.2)' ; Format of Output file data

;=====
;Program Steps
;=====
RUN PGM=MATRIX
ZONES=1
;=====
; Accumulate floating 0.5 mile block density for each TAZ
; Accumulation based on varying straightline distances between TAZ centroids
;
;=====
FILEO RECO[1] = "@ZNFfile_IN5@",
fields = TAZ(5),

```

```

HH(8.0), TOTPOP(8.0), TOTEMP(8.0),
RETEMP(8.0),NRETEMP(8.0),
OFFEMP(8.0), OTHEMP(8.0), INDEMP(8.0), HHPOP(8.0),
GQPOP(8.0),
LANDAREA(8.4), POP_10, EMP_10, AREA_10,
POPDEN10, EMPDEN10, ADISTTOX(5.2),
BLOCKS05(8.0), AREA05(15.4), BlockDen05(8.0),
jurcode(5.0), Atype(5.0)

; read XY coords from the ZONE file, as a zonal lookup table
FileI LOOKUPI[1] = "@ZNFfile_IN1@"
LOOKUP LOOKUPI=1, NAME=tazdata,
LOOKUP[1] = TAZ, RESULT=TAZXCRD, ;
LOOKUP[2] = TAZ, RESULT=TAZYCRD, ;
LOOKUP[3] = TAZ, RESULT=HH, ;
LOOKUP[4] = TAZ, RESULT=HHPOP, ;
LOOKUP[5] = TAZ, RESULT=GQPOP, ;
LOOKUP[6] = TAZ, RESULT=TOTPOP, ;
LOOKUP[7] = TAZ, RESULT=TOTEMP, ;
LOOKUP[8] = TAZ, RESULT=INDEMP, ;
LOOKUP[9] = TAZ, RESULT=RETEMP, ;
LOOKUP[10] = TAZ, RESULT=OFFEMP, ;
LOOKUP[11] = TAZ, RESULT=OTHEMP, ;
LOOKUP[12] = TAZ, RESULT=JURCODE, ;
LOOKUP[13] = TAZ, RESULT=LANDAREA, ;
LOOKUP[14] = TAZ, RESULT=ADISTTOX, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

; Read GIS File as a zonal lookup table
FileI LOOKUPI[2] = "@ZNFfile_IN2@"
LOOKUP LOOKUPI=2, NAME=gisdata,
LOOKUP[1] = TAZ, RESULT=BLOCKS,
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

; Read Area Type File as a zonal lookup table
FileI LOOKUPI[3] = "@ZNFfile_IN3@"
LOOKUP LOOKUPI=3, NAME=Atypedata,
LOOKUP[1] = TAZ, RESULT=pop_10,
LOOKUP[2] = TAZ, RESULT=emp_10,
LOOKUP[3] = TAZ, RESULT=Area_10,
LOOKUP[4] = TAZ, RESULT=popden,
LOOKUP[5] = TAZ, RESULT=empden,
LOOKUP[6] = TAZ, RESULT=atype,

INTERPOLATE=N, FAIL= 0,0,0, LIST=N
;; define zonal arrays for accumulating the variables
ARRAY BLOCKS05=3722, BLOCKDEN05=3722, AREA05=3722

LOOP M = 1,@LastIzn@ ; Loop through each zone, read coordinates

Xi = tazdata(1,M)
Yi = tazdata(2,M)
IF (Xi = 0.00) Continue

LOOP L = 1,@LastIzn@ ; Loop through all proximate zones, read coords.
Xj = tazdata(1,L)
Yj = tazdata(2,L)
IF (Xj = 0.00) Continue

Xdifff = abs(Xi-Xj) ; calc. airline distance
Ydifff = abs(Yi-Yj) ;
;
d_ft = sqrt(xdifff*xdifff + Ydifff*Ydifff) ;
d_mi = d_ft/5280.0 ;
;

```

## Appendix C Cube Voyager Scripts

```

;debug1
If (l=1)
  print form=10 list = l,m,xi,yi,xj,yj,d_ft,d_mi(6.2),
file=debug1.txt
endif
;end debug1

IF      (D_mi < 0.500)
  BLOCKS05[M] = BLOCKS05[M] + gisdata(1,L)
  Area05[M]   = Area05[M]   + tazdata(13,L)
ENDIF

ENDLOOP

ENDLOOP

;; All done reading, write out zonal results:

LOOP M= 1,@LastIzn@
  ro.TAZ           = M
  ro.Area05        = Area05[M]
  ro.BLOCKS05      = BLOCKS05[M]
  ro.BlockDen05    = 0
  IF (Area05[M] > 0)
    ro.BlockDen05  = BLOCKS05[M]/Area05[M]
  ENDIF

  ro.HH            = TAZdata(3,M)
  ro.HHPOP         = TAZdata(4,M)
  ro.GQPOP         = TAZdata(5,M)
  ro.TOTPOP        = TAZdata(6,M)
  ro.TOTEMP        = TAZdata(7,M)
  ro.RETEMP        = TAZdata(9,M)
  ro.NRETEMP       = TAZdata(7,M) - TAZdata(9,M)
  ro.INDEMP        = TAZdata(8,M)
  ro.OTHEMP        = TAZdata(11,M)
  ro.OFFEMP        = TAZdata(10,M)
  ro.JURCODE       = TAZdata(12,M)
  ro.LANDAREA      = TAZdata(13,M)
  ro.ADISTTOX      = TAZdata(14,M)

  ro.POP_10        = Atypedata(1,M)
  ro.EMP_10        = Atypedata(2,M)
  ro.Area_10       = Atypedata(3,M)
  ro.POPDEN10      = Atypedata(4,M)
  ro.EMPDEN10      = Atypedata(5,M)
  ro.ATYPE         = Atypedata(6,M)

  WRITE RECO= 1
ENDLOOP
endrun

RUN PGM=MATRIX
ZONES=@ZONESIZE@

FILEO PRINTO[1]   = "@ReportFile@"
pageheight=32767 ; Preclude header breaks

; Set up zone arrays for accumulating I/O variables
;

Array Proda        =@PrCl@,@InCl@,@SzCl@,@VaCl@
Array Zproda       =@PrCl@,@ZoneSize@

```

```

Array ZprodaInc    =@PrCl@,@InCl@,@ZoneSize@
Array MZprodaInc   =@PrCl@,@InCl@,@ZoneSize@
Array MZproda      =@PrCl@,@ZoneSize@
Array MTotProdInca =@PrCl@,@InCl@
Array NMZProda     =@PrCl@,@ZoneSize@
Array MZattra      =@PrCl@,@ZoneSize@
Array MZattraInc   =@PrCl@,@InCl@,@ZoneSize@
Array NMZattra     =@PrCl@,@ZoneSize@

Array AIncratio    =@InCl@,@ArCl@,@PrCl@
Array AIncShare    =@InCl@,@ArCl@,@PrCl@
Array IniAttr      =@InCl@,@PrCl@
Array FinAttr      =@InCl@,@PrCl@
Array IniAtot      =@PrCl@
Array FinAtot      =@PrCl@
Array Scaltot      =@PrCl@
Array Mscale       =@PrCl@
Array NMscale      =@PrCl@
Array MNMscale     =@PrCl@

Array HHa          =@InCl@,@SzCl@,@VaCl@
Array Prata        =@PrCl@,@InCl@,@SzCl@,@VaCl@
Array NMPrate      =10,@PrCl@,@ArCl@
Array NMArate      =10,@PrCl@,@ArCl@
Array Attrate      =10,@PrCl@,@ArCl@

Array I_proda      =@InCl@,@PrCl@
Array S_proda      =@SzCl@,@PrCl@
Array V_proda      =@VaCl@,@PrCl@
Array A_proda      =@ArCl@,@PrCl@
Array J_proda      =@JrCl@,@PrCl@

Array TotProda     =@PrCl@,
MTotProda          =@PrCl@,
MNMTotProda        =@PrCl@,
XMTotProda         =@PrCl@,
NMTotProda         =@PrCl@,
MTotAttr          =@PrCl@,
XMTotAttr          =@PrCl@,
MNMTotAttr        =@PrCl@,
NMTotAttr          =@PrCl@,

Atypea             =@zonesize@,

I_HHa              =@InCl@,
S_HHa              =@SzCl@,
V_HHa              =@VaCl@,
A_HHa              =@ArCl@,
J_HHa              =@JrCl@,

TotProdInca        =@InCl@,
TotProdSiza        =@SzCl@,
TotProdVeha        =@VaCl@,
TotProdAreaa       =@ArCl@,
TotProdJura        =@JrCl@,

HBWNMPro           =@zonesize@,
HBSNMPro           =@zonesize@,
HBONMPro           =@zonesize@,
NHWNMPro           =@zonesize@,
NHONMPro           =@zonesize@,

HBWNMAtt           =@zonesize@,
HBSNMAtt           =@zonesize@,
HBONMAtt           =@zonesize@,
NHWNMAtt           =@zonesize@,
NHONMAtt           =@zonesize@,

```





## Appendix C Cube Voyager Scripts

```

" 150, 1.00, 1.00, 1.12, 1.00, 1.00, " ;;cal
" 160, 1.36, 1.00, 1.00, 1.00, 1.00, " ;;stm
" 170, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;kg
" 180, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;fbg
" 190, 1.00, 1.14, 0.86, 1.00, 1.00, " ;;sta
" 200, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;spt
" 210, 1.00, 1.00, 0.88, 1.00, 1.00, " ;;fau
" 220, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;clk
" 230, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;jef
;
;
LOOKUP NAME=A_JurAdj, ;
LOOKUP[1] = 1, RESULT=2, ; HBW Attraction Adjustment
LOOKUP[2] = 1, RESULT=3, ; HBS Attraction Adjustment
LOOKUP[3] = 1, RESULT=4, ; HBO Attraction Adjustment
LOOKUP[4] = 1, RESULT=5, ; NHW Attraction Adjustment
LOOKUP[5] = 1, RESULT=6, ; NHO Attraction Adjustment
INTERPOLATE=N, FAIL= 1.0,1.0,1.0,
;
;
;
R=" HBWAs HBSAs HBOAs NNWAs NHOAs
0, 1.10, 0.60, 0.90, 1.10, 0.80, " ;;dc NONCORE
" 1, 1.10, 0.60, 0.90, 1.10, 0.80, " ;;dc CORE
" 10, 1.02, 1.07, 1.10, 0.90, 1.13, " ;;mtg
" 20, 1.08, 0.78, 0.77, 1.00, 0.77, " ;;pg
" 30, 1.22, 0.87, 0.95, 1.00, 0.60, " ;;arl NONCORE
" 31, 1.22, 0.87, 0.95, 1.00, 0.60, " ;;arl CORE
" 40, 0.77, 0.85, 1.00, 1.00, 1.14, " ;;alx
" 50, 1.07, 1.05, 1.00, 0.95, 0.95, " ;;ffx
" 60, 0.89, 1.07, 0.87, 0.85, 1.00, " ;;ldn
" 70, 1.11, 1.05, 0.96, 1.00, 1.00, " ;;pw
" 80, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;
" 90, 1.00, 1.00, 0.83, 0.88, 1.14, " ;;frd
" 100, 0.82, 1.18, 0.87, 0.78, 1.00, " ;;how
" 110, 0.86, 1.00, 0.85, 0.89, 0.94, " ;;aa
" 120, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;chs
" 130, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;
" 140, 1.00, 1.51, 0.94, 1.00, 1.24, " ;;car
" 150, 1.00, 0.78, 1.29, 1.00, 1.00, " ;;cal
" 160, 1.40, 1.00, 0.80, 1.49, 1.00, " ;;stm
" 170, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;kg
" 180, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;fbg
" 190, 1.00, 1.72, 1.00, 1.00, 1.00, " ;;sta
" 200, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;spt
" 210, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;fau
" 220, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;clk
" 230, 1.00, 1.00, 1.00, 1.00, 1.00, " ;;jef
;
;
;
* P_JurAdj(1,jurcode)
* A_JurAdj(1,jurcode)
;
;
;
;
; End Jurisdiction Motorized Production, Attraction Adjustment Lookups
;
;
; Read in Production rates, fill in production rate array
FILEI DBI[1] = "@Prate in@"
LOOP K = 1,dbi.1.NUMRECORDS
x = DBIReadRecord(1,k)
count = dbi.1.recno
Prata[1][di.1.Inc][di.1.Siz][di.1.Veh] = di.1.HBW
Prata[2][di.1.Inc][di.1.Siz][di.1.Veh] = di.1.HBS
Prata[3][di.1.Inc][di.1.Siz][di.1.Veh] = di.1.HBO
Prata[4][di.1.Inc][di.1.Siz][di.1.Veh] = di.1.NHW

```

```

Prata[5][di.1.Inc][di.1.Siz][di.1.Veh] = di.1.NHO
ENDLOOP

; Read in NMproduction model
; rates arrayed as: variables (1-4) - 1/constant, 2/1-mi float.pop den.,3/1-mi
float emp. den.,4/0.5mi. float. block density
; purpose (1-5)
; area type (1-6)
FILEI DBI[2] = "@NMPrate_in@"
LOOP K = 1,dbi.2.NUMRECORDS
x = DBIReadRecord(2,k)
NMPrate[dbi.2.recno][1][1] = di.2.HBW1
NMPrate[dbi.2.recno][1][2] = di.2.HBW2
NMPrate[dbi.2.recno][1][3] = di.2.HBW3
NMPrate[dbi.2.recno][1][4] = di.2.HBW4
NMPrate[dbi.2.recno][1][5] = di.2.HBW5
NMPrate[dbi.2.recno][1][6] = di.2.HBW6

NMPrate[dbi.2.recno][2][1] = di.2.HBS1
NMPrate[dbi.2.recno][2][2] = di.2.HBS2
NMPrate[dbi.2.recno][2][3] = di.2.HBS3
NMPrate[dbi.2.recno][2][4] = di.2.HBS4
NMPrate[dbi.2.recno][2][5] = di.2.HBS5
NMPrate[dbi.2.recno][2][6] = di.2.HBS6

NMPrate[dbi.2.recno][3][1] = di.2.HBO1
NMPrate[dbi.2.recno][3][2] = di.2.HBO2
NMPrate[dbi.2.recno][3][3] = di.2.HBO3
NMPrate[dbi.2.recno][3][4] = di.2.HBO4
NMPrate[dbi.2.recno][3][5] = di.2.HBO5
NMPrate[dbi.2.recno][3][6] = di.2.HBO6

NMPrate[dbi.2.recno][4][1] = di.2.NHW1
NMPrate[dbi.2.recno][4][2] = di.2.NHW2
NMPrate[dbi.2.recno][4][3] = di.2.NHW3
NMPrate[dbi.2.recno][4][4] = di.2.NHW4
NMPrate[dbi.2.recno][4][5] = di.2.NHW5
NMPrate[dbi.2.recno][4][6] = di.2.NHW6

NMPrate[dbi.2.recno][5][1] = di.2.NHO1
NMPrate[dbi.2.recno][5][2] = di.2.NHO2
NMPrate[dbi.2.recno][5][3] = di.2.NHO3
NMPrate[dbi.2.recno][5][4] = di.2.NHO4
NMPrate[dbi.2.recno][5][5] = di.2.NHO5
NMPrate[dbi.2.recno][5][6] = di.2.NHO6
ENDLOOP

; Read in NMattraction model
; rates arrayed as: Ind.Variable (1-4) - 1/constant, 2/1-mi float.pop den.,3/1-mi
float emp. den.,4/0.5mi. float. block density
; purpose (1-5)
; area type (1-6)
FILEI DBI[3] = "@NMArate_in@"
LOOP K = 1,dbi.3.NUMRECORDS
x = DBIReadRecord(3,k)
NMArate[dbi.3.recno][1][1] = di.3.HBW1
NMArate[dbi.3.recno][1][2] = di.3.HBW2
NMArate[dbi.3.recno][1][3] = di.3.HBW3
NMArate[dbi.3.recno][1][4] = di.3.HBW4
NMArate[dbi.3.recno][1][5] = di.3.HBW5
NMArate[dbi.3.recno][1][6] = di.3.HBW6

NMArate[dbi.3.recno][2][1] = di.3.HBS1
NMArate[dbi.3.recno][2][2] = di.3.HBS2
NMArate[dbi.3.recno][2][3] = di.3.HBS3

```

## Appendix C Cube Voyager Scripts

```

NMArate[dbi.3.recno][2][4] = di.3.HBS4
NMArate[dbi.3.recno][2][5] = di.3.HBS5
NMArate[dbi.3.recno][2][6] = di.3.HBS6

NMArate[dbi.3.recno][3][1] = di.3.HBO1
NMArate[dbi.3.recno][3][2] = di.3.HBO2
NMArate[dbi.3.recno][3][3] = di.3.HBO3
NMArate[dbi.3.recno][3][4] = di.3.HBO4
NMArate[dbi.3.recno][3][5] = di.3.HBO5
NMArate[dbi.3.recno][3][6] = di.3.HBO6

NMArate[dbi.3.recno][4][1] = di.3.NHW1
NMArate[dbi.3.recno][4][2] = di.3.NHW2
NMArate[dbi.3.recno][4][3] = di.3.NHW3
NMArate[dbi.3.recno][4][4] = di.3.NHW4
NMArate[dbi.3.recno][4][5] = di.3.NHW5
NMArate[dbi.3.recno][4][6] = di.3.NHW6

NMArate[dbi.3.recno][5][1] = di.3.NHO1
NMArate[dbi.3.recno][5][2] = di.3.NHO2
NMArate[dbi.3.recno][5][3] = di.3.NHO3
NMArate[dbi.3.recno][5][4] = di.3.NHO4
NMArate[dbi.3.recno][5][5] = di.3.NHO5
NMArate[dbi.3.recno][5][6] = di.3.NHO6

ENDLOOP

;; Read in Attraction rates
;; rates arrayed as: Ind.Variables - 1/TotalEmp.,2/Total
Pop.,3/Ret.Emp.,4/Off.Emp.,5/OtherEmp.,6/Non-retail Emp.
;;           purpose (1-5)
;;           area type (1-6)
FILEI DBI[4]      ="@Attrate_in@"
LOOP K = 1,dbi.4.NUMRECORDS
  x = DBIReadRecord(4,k)
  ATTrate[dbi.4.recno][1][1] = di.4.HBW1
  ATTrate[dbi.4.recno][1][2] = di.4.HBW2
  ATTrate[dbi.4.recno][1][3] = di.4.HBW3
  ATTrate[dbi.4.recno][1][4] = di.4.HBW4
  ATTrate[dbi.4.recno][1][5] = di.4.HBW5
  ATTrate[dbi.4.recno][1][6] = di.4.HBW6

  ATTrate[dbi.4.recno][2][1] = di.4.HBS1
  ATTrate[dbi.4.recno][2][2] = di.4.HBS2
  ATTrate[dbi.4.recno][2][3] = di.4.HBS3
  ATTrate[dbi.4.recno][2][4] = di.4.HBS4
  ATTrate[dbi.4.recno][2][5] = di.4.HBS5
  ATTrate[dbi.4.recno][2][6] = di.4.HBS6

  ATTrate[dbi.4.recno][3][1] = di.4.HBO1
  ATTrate[dbi.4.recno][3][2] = di.4.HBO2
  ATTrate[dbi.4.recno][3][3] = di.4.HBO3
  ATTrate[dbi.4.recno][3][4] = di.4.HBO4
  ATTrate[dbi.4.recno][3][5] = di.4.HBO5
  ATTrate[dbi.4.recno][3][6] = di.4.HBO6

  ATTrate[dbi.4.recno][4][1] = di.4.NHW1
  ATTrate[dbi.4.recno][4][2] = di.4.NHW2
  ATTrate[dbi.4.recno][4][3] = di.4.NHW3
  ATTrate[dbi.4.recno][4][4] = di.4.NHW4
  ATTrate[dbi.4.recno][4][5] = di.4.NHW5
  ATTrate[dbi.4.recno][4][6] = di.4.NHW6

  ATTrate[dbi.4.recno][5][1] = di.4.NHO1
  ATTrate[dbi.4.recno][5][2] = di.4.NHO2
  ATTrate[dbi.4.recno][5][3] = di.4.NHO3
  ATTrate[dbi.4.recno][5][4] = di.4.NHO4
  ATTrate[dbi.4.recno][5][5] = di.4.NHO5

```

```

ATTrate[dbi.4.recno][5][6] = di.4.NHO6

ENDLOOP

;; Read in Income/Area Type - Attraction Shares
;; rates arrayed as: Income, AreaType
;;
;;
FILEI DBI[5]      ="@Incrat_in@"
LOOP K = 1,dbi.5.NUMRECORDS
  x = DBIReadRecord(5,k)
  AIncRatio[di.5.income][di.5.Atype][1] = di.5.HBWRat
  AIncShare[di.5.income][di.5.Atype][1] = di.5.HBWSHare

  AIncRatio[di.5.income][di.5.Atype][2] = di.5.HBSRat
  AIncShare[di.5.income][di.5.Atype][2] = di.5.HBSShare

  AIncRatio[di.5.income][di.5.Atype][3] = di.5.HBORat
  AIncShare[di.5.income][di.5.Atype][3] = di.5.HBOShare

ENDLOOP

If (I <= @LastIZN@)

; Read in HHS by Income , Size, Vehs. Avail
ZDATI[2] = @ZNFILIN4@ ;; variables in DBF file:
HHSISV114      ;; HHSISV111      HHSISV112      HHSISV113
                ;; HHSISV211      HHSISV212      HHSISV213
HHSISV214      ;; HHSISV311      HHSISV312      HHSISV313
                ;; HHSISV411      HHSISV412      HHSISV413
HHSISV314      ;; HHSISV121      HHSISV122      HHSISV123
                ;; HHSISV221      HHSISV222      HHSISV223
HHSISV414      ;; HHSISV321      HHSISV322      HHSISV323
                ;; HHSISV421      HHSISV422      HHSISV423
HHSISV124      ;; HHSISV131      HHSISV132      HHSISV133
                ;; HHSISV231      HHSISV232      HHSISV233
HHSISV224      ;; HHSISV331      HHSISV332      HHSISV333
                ;; HHSISV431      HHSISV432      HHSISV433
HHSISV324      ;; HHSISV141      HHSISV142      HHSISV143
                ;; HHSISV241      HHSISV242      HHSISV243
HHSISV424      ;; HHSISV341      HHSISV342      HHSISV343
                ;; HHSISV441      HHSISV442      HHSISV443
HHSISV134      ;;
HHSISV234      ;;
HHSISV334      ;;
HHSISV434      ;;
HHSISV144      ;;
HHSISV244      ;;
HHSISV344      ;;
HHSISV444      ;;

;; store current TAZ HHS in Array
Hha[1][1][1] = zi.2.HHSISV111 Hha[1][1][2] = zi.2.HHSISV112 Hha[1][1][3] =
zi.2.HHSISV113 Hha[1][1][4] = zi.2.HHSISV114
Hha[2][1][1] = zi.2.HHSISV211 Hha[2][1][2] = zi.2.HHSISV212 Hha[2][1][3] =
zi.2.HHSISV213 Hha[2][1][4] = zi.2.HHSISV214
Hha[3][1][1] = zi.2.HHSISV311 Hha[3][1][2] = zi.2.HHSISV312 Hha[3][1][3] =
zi.2.HHSISV313 Hha[3][1][4] = zi.2.HHSISV314
Hha[4][1][1] = zi.2.HHSISV411 Hha[4][1][2] = zi.2.HHSISV412 Hha[4][1][3] =
zi.2.HHSISV413 Hha[4][1][4] = zi.2.HHSISV414

```

## Appendix C Cube Voyager Scripts

```

Hha[1][2][1] = zi.2.HHSISV121 Hha[1][2][2] = zi.2.HHSISV122 Hha[1][2][3] =
zi.2.HHSISV123 Hha[1][2][4] = zi.2.HHSISV124
Hha[2][2][1] = zi.2.HHSISV221 Hha[2][2][2] = zi.2.HHSISV222 Hha[2][2][3] =
zi.2.HHSISV223 Hha[2][2][4] = zi.2.HHSISV224
Hha[3][2][1] = zi.2.HHSISV321 Hha[3][2][2] = zi.2.HHSISV322 Hha[3][2][3] =
zi.2.HHSISV323 Hha[3][2][4] = zi.2.HHSISV324
Hha[4][2][1] = zi.2.HHSISV421 Hha[4][2][2] = zi.2.HHSISV422 Hha[4][2][3] =
zi.2.HHSISV423 Hha[4][2][4] = zi.2.HHSISV424
Hha[1][3][1] = zi.2.HHSISV131 Hha[1][3][2] = zi.2.HHSISV132 Hha[1][3][3] =
zi.2.HHSISV133 Hha[1][3][4] = zi.2.HHSISV134
Hha[2][3][1] = zi.2.HHSISV231 Hha[2][3][2] = zi.2.HHSISV232 Hha[2][3][3] =
zi.2.HHSISV233 Hha[2][3][4] = zi.2.HHSISV234
Hha[3][3][1] = zi.2.HHSISV331 Hha[3][3][2] = zi.2.HHSISV332 Hha[3][3][3] =
zi.2.HHSISV333 Hha[3][3][4] = zi.2.HHSISV334
Hha[4][3][1] = zi.2.HHSISV431 Hha[4][3][2] = zi.2.HHSISV432 Hha[4][3][3] =
zi.2.HHSISV433 Hha[4][3][4] = zi.2.HHSISV434
Hha[1][4][1] = zi.2.HHSISV141 Hha[1][4][2] = zi.2.HHSISV142 Hha[1][4][3] =
zi.2.HHSISV143 Hha[1][4][4] = zi.2.HHSISV144
Hha[2][4][1] = zi.2.HHSISV241 Hha[2][4][2] = zi.2.HHSISV242 Hha[2][4][3] =
zi.2.HHSISV243 Hha[2][4][4] = zi.2.HHSISV244
Hha[3][4][1] = zi.2.HHSISV341 Hha[3][4][2] = zi.2.HHSISV342 Hha[3][4][3] =
zi.2.HHSISV343 Hha[3][4][4] = zi.2.HHSISV344
Hha[4][4][1] = zi.2.HHSISV441 Hha[4][4][2] = zi.2.HHSISV442 Hha[4][4][3] =
zi.2.HHSISV443 Hha[4][4][4] = zi.2.HHSISV444

Jr = zi.1.Jurcode + 1.0 ; Initialize Jur code index
At = zi.1.Atype ; Initialize Area Type index

loop in=1,4
loop Si=1,4
loop Ve=1,4

TotHHa = TotHHa + HHa[in][Si][Ve]
I_HHa[in] = I_HHa[in] + HHa[in][Si][Ve]
; HHa by Inc
S_HHa[Si] = S_HHa[Si] + HHa[in][Si][Ve]
; by Size
V_HHa[Ve] = V_HHa[Ve] + HHa[in][Si][Ve]
; by Vehs.
A_HHa[At] = A_HHa[At] + HHa[in][Si][Ve]
; by Area Type
J_HHa[Jr] = J_HHa[Jr] + HHa[in][Si][Ve]
; by Juris.
TotHH = TotHH + HHa[in][Si][Ve]
; Sum of all HHa

loop pu=1,5

Proda[pu][in][Si][Ve] = HHa[in][Si][Ve] * Prata[pu][in][Si][Ve]
; Compute Motorized/NonMotorized productions
Zproda[pu][i] = Zproda[pu][i] +
Proda[pu][in][Si][Ve] ; Zonal Motor/NonMotor productions by purp
ZprodaInc[pu][in][i] = ZprodaInc[pu][in][i] +
Proda[pu][in][Si][Ve] ; Zonal Motor/NonMotor productions by purp&Inc

TotProda[pu] = TotProda[pu] + Proda[pu][in][Si][Ve]
; Accumulate total M/NM productions by purpose

; Accumualte M/NM summary arrays
I_proda[in][pu] = I_proda[in][pu] + Proda[pu][in][Si][Ve]
; Productions by Inc and Purpose
S_proda[Si][pu] = S_proda[Si][pu] + Proda[pu][in][Si][Ve]
; Productions by Size and Purpose
V_proda[Ve][pu] = V_proda[Ve][pu] + Proda[pu][in][Si][Ve]
; Productions by Vehs. and Purpose
A_proda[At][pu] = A_proda[At][pu] + Proda[pu][in][Si][Ve]
; Productions by Area Tp. and Purpose

```

```

J_proda[Jr][pu] = J_proda[Jr][pu] + Proda[pu][in][Si][Ve]
; Productions by Juris. and Purpose

TotProdInca[in] = TotProdInca[in] + Proda[pu][in][Si][Ve]
; Total Productions by Inc.
TotProdSiza[Si] = TotProdSiza[Si] + Proda[pu][in][Si][Ve]
; Total Productions by Size
TotProdVeha[Ve] = TotProdVeha[Ve] + Proda[pu][in][Si][Ve]
; Total Productions by Vehs.
TotProdAreaa[At] = TotProdAreaa[At] + Proda[pu][in][Si][Ve]
; Total Productions by Area Tp.
TotProdJura[Jr] = TotProdJura[At] + Proda[pu][in][Si][Ve]
; Total Productions by Juris.

endloop
endloop
endloop
endloop

;; I-X Model is voided out
;; Compute Internal Motorized / NonMotorized productions here:
;;
;; H.Humeida's NM Model - 10/14/10
;; original model (single curve: IX_ShareHBW = 0.1786 * (exp(-0.1435 *
zi.1.ADISTTOX))
;; updated model

;; Default Curves
IX_ShareHBW = 0.2133 * (exp(-0.1950 * zi.1.ADISTTOX))
IX_ShareHBS = 0.2133 * (exp(-0.1950 * zi.1.ADISTTOX))
IX_ShareHBO = 0.2133 * (exp(-0.1950 * zi.1.ADISTTOX))
IX_ShareNHW = 0.2133 * (exp(-0.1950 * zi.1.ADISTTOX))
IX_ShareNHO = 0.2133 * (exp(-0.1950 * zi.1.ADISTTOX))

;; Baltimore area curves:
;; If (zi.1.jurcode = 10 || zi.1.jurcode = 11 || zi.1.jurcode = 14 )
;; IX_ShareHBW = 0.3348 * (exp(-0.0938 * zi.1.ADISTTOX))
;; IX_ShareHBS = 0.1766 * (exp(-0.1957 * zi.1.ADISTTOX))
;; IX_ShareHBO = 0.1766 * (exp(-0.1957 * zi.1.ADISTTOX))
;; IX_ShareNHW = 0.1766 * (exp(-0.1957 * zi.1.ADISTTOX))
;; IX_ShareNHO = 0.1766 * (exp(-0.1957 * zi.1.ADISTTOX))
;; endif
;-----

;; Compute "raw" Non-Motorized shares here:
NMP_ShareHBW = NMPrate[1][1][zi.1.atype] +
NMPrate[2][1][zi.1.atype] * zi.1.POPDEN10 +
NMPrate[3][1][zi.1.atype] * zi.1.EMPDEN10 +
NMPrate[4][1][zi.1.atype] * zi.1.Blockden05

NMP_ShareHBS = NMPrate[1][2][zi.1.atype] +
NMPrate[2][2][zi.1.atype] * zi.1.POPDEN10 +
NMPrate[3][2][zi.1.atype] * zi.1.EMPDEN10 +
NMPrate[4][2][zi.1.atype] * zi.1.Blockden05

NMP_ShareHBO = NMPrate[1][3][zi.1.atype] +
NMPrate[2][3][zi.1.atype] * zi.1.POPDEN10 +
NMPrate[3][3][zi.1.atype] * zi.1.EMPDEN10 +
NMPrate[4][3][zi.1.atype] * zi.1.Blockden05

NMP_ShareNHW = NMPrate[1][4][zi.1.atype] +
NMPrate[2][4][zi.1.atype] * zi.1.POPDEN10 +
NMPrate[3][4][zi.1.atype] * zi.1.EMPDEN10 +
NMPrate[4][4][zi.1.atype] * zi.1.Blockden05

NMP_ShareNHO = NMPrate[1][5][zi.1.atype] +

```

## Appendix C Cube Voyager Scripts

```

NMPRate[2][5][zi.1.atype] * zi.1.POPDEN10 +
NMPRate[3][5][zi.1.atype] * zi.1.EMPDEN10 +
NMPRate[4][5][zi.1.atype] * zi.1.Blockden05

;; Compute Non-Motorized shares with area type adjustments here (make sure shares do
not exceed 1.00):

NMP_ShareHBW_adj = MIN(1.00, (NMP_ShareHBW * NPro_Adj[1][At]))
NMP_ShareHBS_adj = MIN(1.00, (NMP_ShareHBS * NPro_Adj[2][At]))
NMP_ShareHBO_adj = MIN(1.00, (NMP_ShareHBO * NPro_Adj[3][At]))
NMP_ShareNHW_adj = MIN(1.00, (NMP_ShareNHW * NPro_Adj[4][At]))
NMP_ShareNHO_adj = MIN(1.00, (NMP_ShareNHO * NPro_Adj[5][At]))

;; compute Internal Motor/NonMotor productions by purpose
jurcode2 = jurcode* 10 + coreflag ;; establish juris/core-noncore index for P-/A-
mods

;-----HBW-----
MZProda[1][i] = Zproda[1][i] * (1.0 - NMP_ShareHBW_adj) * MPro_Adj[1][At] *
P_JurAdj(1,jurcode2) ;; compute internal HBW Motorized productions
NMZProda[1][i] = Zproda[1][i] * NMP_ShareHBW_adj * MPro_Adj[1][At] *
P_JurAdj(1,jurcode2) ;; compute internal HBW Non-Motorized productions

IF (Zproda[1][i]>0)
;; Pr In zone; Pr Zn Pr Zn in pr zn
;; compute internal HBW Motorized productions by Income level
MZProdaInc[1][1][i] = (MZProda[1][i]/Zproda[1][i]) * ZprodaInc[1][1][i]
MZProdaInc[1][2][i] = (MZProda[1][i]/Zproda[1][i]) * ZprodaInc[1][2][i]
MZProdaInc[1][3][i] = (MZProda[1][i]/Zproda[1][i]) * ZprodaInc[1][3][i]
MZProdaInc[1][4][i] = (MZProda[1][i]/Zproda[1][i]) * ZprodaInc[1][4][i]
ENDIF

;-----HBS-----
MZProda[2][i] = Zproda[2][i] * (1.0 - NMP_ShareHBS_adj) * MPro_Adj[2][At] *
P_JurAdj(2,jurcode2) ;; compute internal HBS Motorized productions
NMZProda[2][i] = Zproda[2][i] * NMP_ShareHBS_adj * MPro_Adj[2][At] *
P_JurAdj(2,jurcode2) ;; compute internal HBS Non-Motorized productions

IF (Zproda[2][i]>0)
;; Pr In zone; Pr Zn Pr Zn in pr zn
;; compute internal HBS Motorized productions by Income level
MZProdaInc[2][1][i] = (MZProda[2][i]/Zproda[2][i]) * ZprodaInc[2][1][i]
MZProdaInc[2][2][i] = (MZProda[2][i]/Zproda[2][i]) * ZprodaInc[2][2][i]
MZProdaInc[2][3][i] = (MZProda[2][i]/Zproda[2][i]) * ZprodaInc[2][3][i]
MZProdaInc[2][4][i] = (MZProda[2][i]/Zproda[2][i]) * ZprodaInc[2][4][i]
ENDIF

;-----HBO-----
MZProda[3][i] = Zproda[3][i] * (1.0 - NMP_ShareHBO_adj) * MPro_Adj[3][At] *
P_JurAdj(3,jurcode2) ;; compute internal HBO Motorized productions
NMZProda[3][i] = Zproda[3][i] * NMP_ShareHBO_adj * MPro_Adj[3][At] *
P_JurAdj(3,jurcode2) ;; compute internal HBO Non-Motorized productions

IF (Zproda[3][i]>0)
;; Pr In zone; Pr Zn Pr Zn in pr zn
;; compute internal HBO Motorized productions by Income level
MZProdaInc[3][1][i] = (MZProda[3][i]/Zproda[3][i]) * ZprodaInc[3][1][i]
MZProdaInc[3][2][i] = (MZProda[3][i]/Zproda[3][i]) * ZprodaInc[3][2][i]
MZProdaInc[3][3][i] = (MZProda[3][i]/Zproda[3][i]) * ZprodaInc[3][3][i]
MZProdaInc[3][4][i] = (MZProda[3][i]/Zproda[3][i]) * ZprodaInc[3][4][i]
ENDIF

;-----NHW-----
MZProda[4][i] = Zproda[4][i] * (1.0 - NMP_ShareNHW_adj) * MPro_Adj[4][At] *
P_JurAdj(4,jurcode2) ;; compute internal NHW Motorized productions

```

```

NMZProda[4][i] = Zproda[4][i] * NMP_ShareNHW_adj * MPro_Adj[4][At] *
P_JurAdj(4,jurcode2) ;; compute internal NHW Non-Motorized productions

;-----NHO-----
MZProda[5][i] = Zproda[5][i] * (1.0 - NMP_ShareNHO_adj) * MPro_Adj[5][At] *
P_JurAdj(5,jurcode2) ;; compute internal NHO Motorized productions
NMZProda[5][i] = Zproda[5][i] * NMP_ShareNHO_adj * MPro_Adj[5][At] *
P_JurAdj(5,jurcode2) ;; compute internal NHO Non-Motorized productions

;--;; Debug productions calculations
if (I < 300)
print form=6.4 list='taz: ',i,' NMP shares by purp: ', NMP_ShareHBW,' ',
NMP_ShareHBS,' ', NMP_ShareHBO,' ', NMP_ShareNHW,' ', NMP_ShareNHO, file=
debug_P_Shares.txt
print form=6.4 list='taz: ',i,' NMP shares w/NMMODS: ', NMP_ShareHBW_adj,' ',
NMP_ShareHBS_adj,' ', NMP_ShareHBO_adj,' ', NMP_ShareNHW_adj,' ', NMP_ShareNHO_adj,
file= debug_P_Shares.txt
print form=6.2 list='taz: ',i,' Total Prods by purp: ', ZProda[1][i],'
,ZProda[2][i],' ', ZProda[3][i],' ', ZProda[4][i],' ',ZProda[5][i],
file= debug_P_Shares.txt
print form=6.2 list='taz: ',i,' Motr Prods by purp: ',MZProda[1][i],'
',MZProda[2][i],' ',MZProda[3][i],' ',MZProda[4][i],' ',MZProda[5][i],
file= debug_P_Shares.txt
print form=6.2 list='taz: ',i,' NMtr Prods by purp: ',NMZProda[1][i],'
',NMZProda[2][i],' ',NMZProda[3][i],' ',NMZProda[4][i],' ',NMZProda[5][i],
file= debug_P_Shares.txt

endif
;;
;; write out dbf files for Trip Productions by purpose and mode
;; Pr In zone;
;;
MZProdaInc[1][1][i]
FILEO RECO[1] = "@TripPros",fields =
TAZ(5),
HBW_Mtr_Ps@ofmt@, HBW_Nmt_Ps@ofmt@,
HBW_All_Ps@ofmt@,HBWMtrP_I1@ofmt@,HBWMtrP_I2@ofmt@,HBWMtrP_I3@ofmt@,HBWMtrP_I4@ofmt@
',
HBS_Mtr_Ps@ofmt@, HBS_Nmt_Ps@ofmt@,
HBS_All_Ps@ofmt@,HBSMtrP_I1@ofmt@,HBSMtrP_I2@ofmt@,HBSMtrP_I3@ofmt@,HBSMtrP_I4@ofmt@
',
HBO_Mtr_Ps@ofmt@, HBO_Nmt_Ps@ofmt@,
HBO_All_Ps@ofmt@,HBOMtrP_I1@ofmt@,HBOMtrP_I2@ofmt@,HBOMtrP_I3@ofmt@,HBOMtrP_I4@ofmt@
',
NHW_Mtr_Ps@ofmt@, NHW_Nmt_Ps@ofmt@, NHW_All_Ps@ofmt@,
NHO_Mtr_Ps@ofmt@, NHO_Nmt_Ps@ofmt@, NHO_All_Ps@ofmt@

ro.TAZ = i
ro.HBW_Mtr_Ps = MZProda[1][i]
ro.HBW_Nmt_Ps = NMZProda[1][i]
ro.HBW_All_Ps = MZProda[1][i] + NMZProda[1][i] ;ZProda[1][i]

ro.HBWMtrP_I1 = MZProdaInc[1][1][i]
ro.HBWMtrP_I2 = MZProdaInc[1][2][i]
ro.HBWMtrP_I3 = MZProdaInc[1][3][i]
ro.HBWMtrP_I4 = MZProdaInc[1][4][i]

ro.HBS_Mtr_Ps = MZProda[2][i]
ro.HBS_Nmt_Ps = NMZProda[2][i]
ro.HBS_All_Ps = MZProda[2][i] + NMZProda[2][i] ;ZProda[2][i]

ro.HBSMtrP_I1 = MZProdaInc[2][1][i]
ro.HBSMtrP_I2 = MZProdaInc[2][2][i]
ro.HBSMtrP_I3 = MZProdaInc[2][3][i]
ro.HBSMtrP_I4 = MZProdaInc[2][4][i]

```

## Appendix C Cube Voyager Scripts

```

ro.HBO_Mtr_Ps = MZProda[3][i]
ro.HBO_NMT_Ps = NMZProda[3][i]
ro.HBO_All_Ps = MZProda[3][i] + NMZProda[3][i] ;ZProda[3][i]

ro.HBOMtrP_I1 = MZProdaInc[3][1][i]
ro.HBOMtrP_I2 = MZProdaInc[3][2][i]
ro.HBOMtrP_I3 = MZProdaInc[3][3][i]
ro.HBOMtrP_I4 = MZProdaInc[3][4][i]

ro.NHW_Mtr_Ps = MZProda[4][i]
ro.NHW_NMT_Ps = NMZProda[4][i]
ro.NHW_All_Ps = MZProda[4][i] + NMZProda[4][i] ;ZProda[4][i]

ro.NHO_Mtr_Ps = MZProda[5][i]
ro.NHO_NMT_Ps = NMZProda[5][i]
ro.NHO_All_Ps = MZProda[5][i] + NMZProda[5][i] ;ZProda[5][i]

WRITE RECO=1

;; Accumulate Regional Motor/NonMotor Totals by purpose

MTotProda[1] = MTotProda[1] + MZProda[1][i]
;; accum. internal HBW Motorized productions
MTotProdInca[1][1] = MTotProdInca[1][1] + MZProdaInc[1][1][i]
;; accum. internal HBW Motorized productions by inc.
MTotProdInca[1][2] = MTotProdInca[1][2] + MZProdaInc[1][2][i]
;; accum. internal HBW Motorized productions by inc.
MTotProdInca[1][3] = MTotProdInca[1][3] + MZProdaInc[1][3][i]
;; accum. internal HBW Motorized productions by inc.
MTotProdInca[1][4] = MTotProdInca[1][4] + MZProdaInc[1][4][i]
;; accum. internal HBW Motorized productions by inc.
NMTotProda[1] = NMTotProda[1] + NMZProda[1][i]
;; accum. internal HBW Non-Motorized productions
MNMTotProda[1] = MNMTotProda[1] + MZProda[1][i] +
NMZProda[1][i] ;; accum. internal HBW Motorized&Non-Motorized productions

MTotProda[2] = MTotProda[2] + MZProda[2][i]
;; accum.internal HBS Motorized productions
MTotProdInca[2][1] = MTotProdInca[2][1] + MZProdaInc[2][1][i]
;; accum. internal HBS Motorized productions by inc.
MTotProdInca[2][2] = MTotProdInca[2][2] + MZProdaInc[2][2][i]
;; accum. internal HBS Motorized productions by inc.
MTotProdInca[2][3] = MTotProdInca[2][3] + MZProdaInc[2][3][i]
;; accum. internal HBS Motorized productions by inc.
MTotProdInca[2][4] = MTotProdInca[2][4] + MZProdaInc[2][4][i]
;; accum. internal HBS Motorized productions by inc.
NMTotProda[2] = NMTotProda[2] + NMZProda[2][i]
;; accum.internal HBS Non-Motorized productions
MNMTotProda[2] = MNMTotProda[2] + MZProda[2][i] +
NMZProda[2][i] ;; accum.internal HBS Motorized&Non-Motorized productions

MTotProda[3] = MTotProda[3] + MZProda[3][i]
;; accum.internal HBO Motorized productions
MTotProdInca[3][1] = MTotProdInca[3][1] + MZProdaInc[3][1][i]
;; accum. internal HBO Motorized productions by inc.
MTotProdInca[3][2] = MTotProdInca[3][2] + MZProdaInc[3][2][i]
;; accum. internal HBO Motorized productions by inc.
MTotProdInca[3][3] = MTotProdInca[3][3] + MZProdaInc[3][3][i]
;; accum. internal HBO Motorized productions by inc.
MTotProdInca[3][4] = MTotProdInca[3][4] + MZProdaInc[3][4][i]
;; accum. internal HBO Motorized productions by inc.
NMTotProda[3] = NMTotProda[3] + NMZProda[3][i]
;; accum.internal HBO Non-Motorized productions
MNMTotProda[3] = MNMTotProda[3] + MZProda[3][i] +
NMZProda[3][i] ;; accum.internal HBO Motorized&Non-Motorized productions

```

```

MTotProda[4] = MTotProda[4] + MZProda[4][i]
;; accum.internal NHW Motorized productions
NMTotProda[4] = NMTotProda[4] + NMZProda[4][i]
;; accum.internal NHW Non-Motorized productions
MNMTotProda[4] = MNMTotProda[4] + MZProda[4][i] +
NMZProda[4][i] ;; accum.internal NHW Motorized&Non-Motorized productions

MTotProda[5] = MTotProda[5] + MZProda[5][i]
;; accum.internal NHO Motorized productions
NMTotProda[5] = NMTotProda[5] + NMZProda[5][i]
;; accum.internal NHO Non-Motorized productions
MNMTotProda[5] = MNMTotProda[5] + MZProda[5][i] +
NMZProda[5][i] ;; accum.internal NHO Motorized&Non-Motorized productions

;; Accumulate Regional Motor/NonMotor/ summed Totals
MTotProd = MTotProd + MZProda[1][i] + MZProda[2][i] +
MZProda[3][i] + MZProda[4][i] + MZProda[5][i]
NMTotProd = NMTotProd + NMZProda[1][i] + NMZProda[2][i] +
NMZProda[3][i] + NMZProda[4][i] + NMZProda[5][i]

MNMTotProd = MNMTotProd + MZProda[1][i] + MZProda[2][i] +
MZProda[3][i] + MZProda[4][i] + MZProda[5][i] +
NMZProda[1][i] + NMZProda[2][i] +
NMZProda[3][i] + NMZProda[4][i] + NMZProda[5][i]
;=====
;=====

HBWCompATT[i] =(Attrate[1][1][zi.1.atype] * zi.1.TOTEMP +
Attrate[2][1][zi.1.atype] * zi.1.TOTPOP +
Attrate[3][1][zi.1.atype] * zi.1.RETEMP +
Attrate[4][1][zi.1.atype] * zi.1.OFFEMP +
Attrate[5][1][zi.1.atype] * zi.1.OTHEMP +
Attrate[6][1][zi.1.atype] * zi.1.NRETEMP)

HBSCompATT[i] =(Attrate[1][2][zi.1.atype] * zi.1.TOTEMP +
Attrate[2][2][zi.1.atype] * zi.1.TOTPOP +
Attrate[3][2][zi.1.atype] * zi.1.RETEMP +
Attrate[4][2][zi.1.atype] * zi.1.OFFEMP +
Attrate[5][2][zi.1.atype] * zi.1.OTHEMP +
Attrate[6][2][zi.1.atype] * zi.1.NRETEMP)

HBOCompATT[i] =(Attrate[1][3][zi.1.atype] * zi.1.TOTEMP +
Attrate[2][3][zi.1.atype] * zi.1.TOTPOP +
Attrate[3][3][zi.1.atype] * zi.1.RETEMP +
Attrate[4][3][zi.1.atype] * zi.1.OFFEMP +
Attrate[5][3][zi.1.atype] * zi.1.OTHEMP +
Attrate[6][3][zi.1.atype] * zi.1.NRETEMP)

NHWCompATT[i] =(Attrate[1][4][zi.1.atype] * zi.1.TOTEMP +
Attrate[2][4][zi.1.atype] * zi.1.TOTPOP +
Attrate[3][4][zi.1.atype] * zi.1.RETEMP +
Attrate[4][4][zi.1.atype] * zi.1.OFFEMP +
Attrate[5][4][zi.1.atype] * zi.1.OTHEMP +
Attrate[6][4][zi.1.atype] * zi.1.NRETEMP)

```

## Appendix C Cube Voyager Scripts

```

NHOCmpATT[i] = (Attrate[1][5][zi.1.atype] * zi.1.TOTEMP +
Attrate[2][5][zi.1.atype] * zi.1.TOTPOP +
Attrate[3][5][zi.1.atype] * zi.1.RETEMP +
Attrate[4][5][zi.1.atype] * zi.1.OFFEMP +
Attrate[5][5][zi.1.atype] * zi.1.OTHEMP +
Attrate[6][5][zi.1.atype] * zi.1.NRETEMP)

TOTHBWCompATT = TOTHBWCompATT + HBWCompATT[I]
TOTHBSCompATT = TOTHBSCompATT + HBSCompATT[I]
TOTHBOCompATT = TOTHBOCompATT + HBOCompATT[I]
TOTNHWCompATT = TOTNHWCompATT + NHWCompATT[I]
TOTNHOCompATT = TOTNHOCompATT + NHOCompATT[I]

;;
;; Compute Internal Motorized / NonMotorized ATTRACTIONS here:
;;

NMA_ShareHBW = NMArate[1][1][zi.1.atype] +
NMArate[2][1][zi.1.atype] * zi.1.POPDEN10 +
NMArate[3][1][zi.1.atype] * zi.1.EMPDEN10 +
NMArate[4][1][zi.1.atype] * zi.1.Blockden05

NMA_ShareHBS = NMArate[1][2][zi.1.atype] +
NMArate[2][2][zi.1.atype] * zi.1.POPDEN10 +
NMArate[3][2][zi.1.atype] * zi.1.EMPDEN10 +
NMArate[4][2][zi.1.atype] * zi.1.Blockden05

NMA_ShareHBO = NMArate[1][3][zi.1.atype] +
NMArate[2][3][zi.1.atype] * zi.1.POPDEN10 +
NMArate[3][3][zi.1.atype] * zi.1.EMPDEN10 +
NMArate[4][3][zi.1.atype] * zi.1.Blockden05

NMA_ShareNHW = NMArate[1][4][zi.1.atype] +
NMArate[2][4][zi.1.atype] * zi.1.POPDEN10 +
NMArate[3][4][zi.1.atype] * zi.1.EMPDEN10 +
NMArate[4][4][zi.1.atype] * zi.1.Blockden05

NMA_ShareNHO = NMArate[1][5][zi.1.atype] +
NMArate[2][5][zi.1.atype] * zi.1.POPDEN10 +
NMArate[3][5][zi.1.atype] * zi.1.EMPDEN10 +
NMArate[4][5][zi.1.atype] * zi.1.Blockden05

;; Compute Non-Motorized shares with area type adjustments here (make sure shares do
not exceed 1.00):
NMA_ShareHBW_adj = MIN(1.00, (NMA_ShareHBW * NAtt_Adj[1][At]))
NMA_ShareHBS_adj = MIN(1.00, (NMA_ShareHBS * NAtt_Adj[2][At]))
NMA_ShareHBO_adj = MIN(1.00, (NMA_ShareHBO * NAtt_Adj[3][At]))
NMA_ShareNHW_adj = MIN(1.00, (NMA_ShareNHW * NAtt_Adj[4][At]))
NMA_ShareNHO_adj = MIN(1.00, (NMA_ShareNHO * NAtt_Adj[5][At]))

;; compute Internal Motor/NonMotor ATTRACTIONS by purpose

MZAttr[1][i] = HBWCompATT[i] * (1.0 - NMA_ShareHBW_adj) * Matt_Adj[1][at]
* A_JurAdj(1,jurcode2) ;; compute internal HBW Motorized attractions
NMZAttr[1][i] = HBWCompATT[i] * NMA_ShareHBW_adj * Matt_Adj[1][at]
* A_JurAdj(1,jurcode2) ;; compute internal HBW Non-Motorized attractions

MZAttr[2][i] = HBSCompATT[i] * (1.0 - NMA_ShareHBS_adj) * Matt_Adj[2][at]
* A_JurAdj(2,jurcode2) ;; compute internal HBS Motorized attractions
NMZAttr[2][i] = HBSCompATT[i] * NMA_ShareHBS_adj * Matt_Adj[2][at]
* A_JurAdj(2,jurcode2) ;; compute internal HBS Non-Motorized attractions

```

```

MZAttr[3][i] = HBOCompATT[i] * (1.0 - NMA_ShareHBO_adj) * Matt_Adj[3][at]
* A_JurAdj(3,jurcode2) ;; compute internal HBO Motorized attractions
NMZAttr[3][i] = HBOCompATT[i] * NMA_ShareHBO_adj * Matt_Adj[3][at]
* A_JurAdj(3,jurcode2) ;; compute internal HBO Non-Motorized attractions

MZAttr[4][i] = NHWCompATT[i] * (1.0 - NMA_ShareNHW_adj) * Matt_Adj[4][at]
* A_JurAdj(4,jurcode2) ;; compute internal NHW Motorized attractions
NMZAttr[4][i] = NHWCompATT[i] * NMA_ShareNHW_adj * Matt_Adj[4][at]
* A_JurAdj(4,jurcode2) ;; compute internal NHW Non-Motorized attractions

MZAttr[5][i] = NHOCompATT[i] * (1.0 - NMA_ShareNHO_adj) * Matt_Adj[5][at]
* A_JurAdj(5,jurcode2) ;; compute internal NHO Motorized attractions
NMZAttr[5][i] = NHOCompATT[i] * NMA_ShareNHO_adj * Matt_Adj[5][at]
* A_JurAdj(5,jurcode2) ;; compute internal NHO Non-Motorized attractions

;; Accumulate Regional Motor/NonMotor Totals by purpose

MTotAttr[1] = MTotAttr[1] + MZAttr[1][i]
;; compute internal HBW Motorized attractions
NMTotAttr[1] = NMTotAttr[1] + NMZAttr[1][i]
;; compute internal HBW Non-Motorized attractions
MNMTotAttr[1] = MNMTotAttr[1] + MZAttr[1][i] + NMZAttr[1][i]
;; compute internal HBW Motorized&Non-Motorized attractions

MTotAttr[2] = MTotAttr[2] + MZAttr[2][i]
;; compute internal HBS Motorized attractions
NMTotAttr[2] = NMTotAttr[2] + NMZAttr[2][i]
;; compute internal HBS Non-Motorized attractions
MNMTotAttr[2] = MNMTotAttr[2] + MZAttr[2][i] + NMZAttr[2][i]
;; compute internal HBS Motorized&Non-Motorized attractions

MTotAttr[3] = MTotAttr[3] + MZAttr[3][i]
;; compute internal HBO Motorized attractions
NMTotAttr[3] = NMTotAttr[3] + NMZAttr[3][i]
;; compute internal HBO Non-Motorized attractions
MNMTotAttr[3] = MNMTotAttr[3] + MZAttr[3][i] + NMZAttr[3][i]
;; compute internal HBO Motorized&Non-Motorized attractions

MTotAttr[4] = MTotAttr[4] + MZAttr[4][i]
;; compute internal NHW Motorized attractions
NMTotAttr[4] = NMTotAttr[4] + NMZAttr[4][i]
;; compute internal NHW Non-Motorized attractions
MNMTotAttr[4] = MNMTotAttr[4] + MZAttr[4][i] + NMZAttr[4][i]
;; compute internal NHW Motorized&Non-Motorized attractions

MTotAttr[5] = MTotAttr[5] + MZAttr[5][i]
;; compute internal NHO Motorized attractions
NMTotAttr[5] = NMTotAttr[5] + NMZAttr[5][i]
;; compute internal NHO Non-Motorized attractions
MNMTotAttr[5] = MNMTotAttr[5] + MZAttr[5][i] + NMZAttr[5][i]
;; compute internal NHO Motorized&Non-Motorized attractions

;; Accumulate Regional Motor/NonMotor/Summed Totals
MTotAttr = MTotAttr + MZAttr[1][i] + MZAttr[2][i] +
MZAttr[3][i] + MZAttr[4][i] + MZAttr[5][i]
NMTotAttr = NMTotAttr + NMZAttr[1][i] + NMZAttr[2][i] +
NMZAttr[3][i] + NMZAttr[4][i] + NMZAttr[5][i]
MNMTotAttr = MNMTotAttr + MZAttr[1][i] + MZAttr[2][i] +
MZAttr[3][i] + MZAttr[4][i] + MZAttr[5][i] +
NMZAttr[1][i] + NMZAttr[2][i] +
NMZAttr[3][i] + NMZAttr[4][i] + NMZAttr[5][i]

```

## Appendix C Cube Voyager Scripts

```

=====
;; debug1
if (i=1 )

    print list= ' TAZ ', ' NMP_ShHW ', ' NMP_ShHS ', ' NMP_ShHO ', ' NMP_ShNW ', '
NMP_ShNO ',
                ' NMA_ShHW ', ' NMA_ShHS ', ' NMA_ShHO ', ' NMA_ShNW ', '
NMA_ShNO ',
                ' AttrsHBW ', ' AttrsHBS ', ' AttrsHBO ', ' AttrsNHW ', '
AttrsnHO ',
                file= debug.txt
endif
    print form=10.4,list= I(5),NMP_ShareHBW,      NMP_ShareHBS,
NMP_ShareHBO,      NMP_ShareNHW,      NMP_ShareNHO,
NMA_ShareHBO,      NMA_ShareNHW,      NMA_ShareHBS,
NMA_ShareHBO,      NMA_ShareNHW,      NMA_ShareNHO,

HBWCOMPATT[i](10),HBSCOMPATT[i](10),HBOCOMPATT[i](10),NHWCOMPATT[i](10),NHOCOMPATT[i]
](10),
                file= debug.txt

    print form=10.4,list= I(5),MZAttr[1][i](10), HBWATTInca[i][1](10),
HBWATTInca[i][2](10),HBWATTInca[i][3](10),HBWATTInca[i][4](10),
                file= debugHBWAs.txt
    print form=10.4,list= I(5),MZAttr[2][i](10), HBSATTInca[i][1](10),
HBSATTInca[i][2](10),HBSATTInca[i][3](10),HBSATTInca[i][4](10),
                file= debugHBSAs.txt
    print form=10.4,list= I(5),MZAttr[3][i](10), HBOATTInca[i][1](10),
HBOATTInca[i][2](10),HBOATTInca[i][3](10),HBOATTInca[i][4](10),
                file= debugHBOAs.txt

=====
;; Disaggregate Motorized Attractions by Income
=====

Loop Pr = 1,3
  IniAtot[Pr] = 0
  FinAtot[Pr] = 0
  Scaltot[Pr] = 0
  Loop In = 1,4
    IniAttr[In][Pr]= MZAttr[Pr][i]* AincRatio[In][zi.1.Atype][Pr] *
AincShare[In][zi.1.Atype][Pr] ;; compute initial attractions by income
    IniAtot[Pr] = IniAtot[Pr] + IniAttr[In][Pr]
  ;; accum. initial attractions by purpose
  EndLoop
EndLoop

Loop Pr = 1,3
  if (IniAtot[Pr] = 0)
    Scaltot[Pr] = 0
  else
    Scaltot[Pr] = MZAttr[Pr][i] / IniAtot[Pr] ;; compute scaling
  factor by purpose
  endif

  Loop In = 1,4
    FinAttr[In][Pr]= 0
    FinAttr[In][Pr]= IniAttr[In][Pr] * Scaltot[Pr] ;; compute final
    attractions by purp/income level (apply scaling factor)

    FinAtot[Pr] = FinAtot[Pr] + FinAttr[In][Pr] ;; accumu. final
    attractions by income level
  EndLoop
EndLoop
;; ---
print list = 'comp HBW attractions ',MZAttr[1][i], ' Area Type: ', zi.1.Atype
,file= debug_incdisagg.txt
print list = 'Initial HBW attractions ',IniAttr[1][1], IniAttr[2][1]
,IniAttr[3][1],IniAttr[4][1], ' Sum: ', IniAtot[1],file= debug_incdisagg.txt
print list = 'HBW scale ',scaltot[1](8.6)
,file= debug_incdisagg.txt
print list = 'FINAL HBW attractions ',FinAttr[1][1], FinAttr[2][1]
,FinAttr[3][1],FinAttr[4][1], ' Sum: ', FinAtot[1],file= debug_incdisagg.txt

;; in pr
HBWATTInca[i][1] =FinAttr[1][1]
HBWATTInca[i][2] =FinAttr[2][1]
HBWATTInca[i][3] =FinAttr[3][1]
HBWATTInca[i][4] =FinAttr[4][1]

;; in pr
HBSATTInca[i][1] =FinAttr[1][2]
HBSATTInca[i][2] =FinAttr[2][2]
HBSATTInca[i][3] =FinAttr[3][2]
HBSATTInca[i][4] =FinAttr[4][2]

;; in pr
HBOATTInca[i][1] =FinAttr[1][3]
HBOATTInca[i][2] =FinAttr[2][3]
HBOATTInca[i][3] =FinAttr[3][3]
HBOATTInca[i][4] =FinAttr[4][3]

=====
ENDIF ; if I <= last internal zone
=====
;;
=====
IF (I=@Zonesize@) ;; If at last TAZ

=====
;; Now at the end of the internal TAZs--
;; Write out the attractions here
;; then read external Ps& As, and write to the P,A files
=====

;; Write out zonal dbf files for Computed Trip Attractions by purpose and mode
;; The NHB and NHO attractions will be scaled to match the production totals here!

LOOP zz= 1,@LastIZn@

  FILEO RECO[2] = "@TripAttsCom@",fields =
    TAZ(5),
    HBW_Mtr_As@ofmt@, HBW_NMt_As@ofmt@, HBW_All_As@ofmt@,
    HBWMtrA_I1@ofmt@, HBWMtrA_I2@ofmt@, HBWMtrA_I3@ofmt@,
    HBWMtrA_I4@ofmt@,

    HBS_Mtr_As@ofmt@, HBS_NMt_As@ofmt@, HBS_All_As@ofmt@,
    HBSMtrA_I1@ofmt@, HBSMtrA_I2@ofmt@, HBSMtrA_I3@ofmt@,
    HBSMtrA_I4@ofmt@,

```



## Appendix C Cube Voyager Scripts

```

HBO_Mtr_As@ofmt@, HBO_NMt_As@ofmt@, HBO_All_As@ofmt@,
HBOmtrA_I1@ofmt@, HBOmtrA_I2@ofmt@, HBOmtrA_I3@ofmt@,
HBOmtrA_I4@ofmt@,

NHW_Mtr_As@ofmt@, NHW_NMt_As@ofmt@, NHW_All_As@ofmt@,

NHO_Mtr_As@ofmt@, NHO_NMt_As@ofmt@, NHO_All_As@ofmt@

ro.TAZ = zz
ro.HBW_Mtr_As = MZAttr[1][zz]
ro.HBW_NMt_As = NMZAttr[1][zz]
ro.HBW_All_As = MZAttr[1][zz] + NMZAttr[1][zz]

ro.HBWMtrA_I1 = HBWAttInca[zz][1]
ro.HBWMtrA_I2 = HBWAttInca[zz][2]
ro.HBWMtrA_I3 = HBWAttInca[zz][3]
ro.HBWMtrA_I4 = HBWAttInca[zz][4]

ro.HBS_Mtr_As = MZAttr[2][zz]
ro.HBS_NMt_As = NMZAttr[2][zz]
ro.HBS_All_As = MZAttr[2][zz] + NMZAttr[2][zz]

ro.HBSMtrA_I1 = HBSAttInca[zz][1]
ro.HBSMtrA_I2 = HBSAttInca[zz][2]
ro.HBSMtrA_I3 = HBSAttInca[zz][3]
ro.HBSMtrA_I4 = HBSAttInca[zz][4]

ro.HBO_Mtr_As = MZAttr[3][zz]
ro.HBO_NMt_As = NMZAttr[3][zz]
ro.HBO_All_As = MZAttr[3][zz] + NMZAttr[3][zz]

ro.HBOMtrA_I1 = HBOAttInca[zz][1]
ro.HBOMtrA_I2 = HBOAttInca[zz][2]
ro.HBOMtrA_I3 = HBOAttInca[zz][3]
ro.HBOMtrA_I4 = HBOAttInca[zz][4]

ro.NHW_Mtr_As = MZAttr[4][zz]
ro.NHW_NMt_As = NMZAttr[4][zz]
ro.NHW_All_As = MZAttr[4][zz] + NMZAttr[4][zz]

ro.NHO_Mtr_As = MZAttr[5][zz]
ro.NHO_NMt_As = NMZAttr[5][zz]
ro.NHO_All_As = MZAttr[5][zz] + NMZAttr[5][zz]

WRITE RECO=2
ENDLOOP

FILEI DBI[6]      ="@Ext_PsAs@" ;; variables in file: TAZ   HBW_XI  HBS_XI
HBO_XI  NHB_XI   HBW_IX  HBS_IX  HBO_IX  NHB_IX

LOOP K = 1,dbi.6.NUMRECORDS
  x = DBIReadRecord(6,k)
  MZProda[1][di.6.TAZ] = di.6.HBW_XI * @XOccHBW@
  MZProda[2][di.6.TAZ] = di.6.HBS_XI * @XOccHBS@
  MZProda[3][di.6.TAZ] = di.6.HBO_XI * @XOccHBO@
  NHBProds              = di.6.NHB_XI

  MZProda[4][di.6.TAZ] = NHBProds * @XNHW_Share@ * @XOccNHW@
  MZProda[5][di.6.TAZ] = NHBProds * @XNHO_Share@ * @XOccNHO@

  MZAttr[1][di.6.TAZ] = di.6.HBW_IX * @XOccHBW@
  MZAttr[2][di.6.TAZ] = di.6.HBS_IX * @XOccHBS@
  MZAttr[3][di.6.TAZ] = di.6.HBO_IX * @XOccHBO@
  NHBAttr[1][di.6.TAZ] = di.6.NHB_IX

  MZAttr[4][di.6.TAZ] = NHBAttr[1][di.6.TAZ] * @XNHW_Share@ * @XOccNHW@

```

```

MZAttr[5][di.6.TAZ] = NHBAttr[1][di.6.TAZ] * @XNHO_Share@ * @XOccNHO@

;; Accumulate external P's As by purpose
XMTotProda[1] = XMTotProda[1] + MZProda[1][di.6.TAZ]
XMTotProda[2] = XMTotProda[2] + MZProda[2][di.6.TAZ]
XMTotProda[3] = XMTotProda[3] + MZProda[3][di.6.TAZ]
XMTotProda[4] = XMTotProda[4] + MZProda[4][di.6.TAZ]
XMTotProda[5] = XMTotProda[5] + MZProda[5][di.6.TAZ]

XMTotAttr[1] = XMTotAttr[1] + MZAttr[1][di.6.TAZ]
XMTotAttr[2] = XMTotAttr[2] + MZAttr[2][di.6.TAZ]
XMTotAttr[3] = XMTotAttr[3] + MZAttr[3][di.6.TAZ]
XMTotAttr[4] = XMTotAttr[4] + MZAttr[4][di.6.TAZ]
XMTotAttr[5] = XMTotAttr[5] + MZAttr[5][di.6.TAZ]

;; Write extl Motorized Ps out to the zonal production file (Internals were
written out previously)
;; the extl Motorized As will be written out below, along with the scaled
attractions
ro.TAZ = di.6.TAZ
ro.HBW_Mtr_Ps = MZProda[1][di.6.taz]
ro.HBS_Mtr_Ps = MZProda[2][di.6.taz]
ro.HBO_Mtr_Ps = MZProda[3][di.6.taz]
ro.NHW_Mtr_Ps = MZProda[4][di.6.taz]
ro.NHO_Mtr_Ps = MZProda[5][di.6.taz]

;; all external Ps are motorized, zero out all other external P-data
(Non-motorized Ps, Ps by Income, etc.)
ro.HBW_NMT_PS = 0
ro.HBS_NMT_PS = 0
ro.HBO_NMT_PS = 0
ro.NHW_NMT_PS = 0
ro.NHO_NMT_PS = 0

ro.HBW_ALL_PS = MZProda[1][di.6.taz]
ro.HBS_ALL_PS = MZProda[2][di.6.taz]
ro.HBO_ALL_PS = MZProda[3][di.6.taz]
ro.NHW_ALL_PS = MZProda[4][di.6.taz]
ro.NHO_ALL_PS = MZProda[5][di.6.taz]

ro.HBWMTRP_I1 = 0
ro.HBSMTRP_I1 = 0
ro.HBOMTRP_I1 = 0

ro.HBWMTRP_I2 = 0
ro.HBSMTRP_I2 = 0
ro.HBOMTRP_I2 = 0

ro.HBWMTRP_I3 = 0
ro.HBSMTRP_I3 = 0
ro.HBOMTRP_I3 = 0

ro.HBWMTRP_I4 = 0
ro.HBSMTRP_I4 = 0
ro.HBOMTRP_I4 = 0

WRITE RECO=1

ro.TAZ = di.6.taz
ro.HBW_Mtr_As = MZAttr[1][di.6.taz]
ro.HBW_NMt_As = 0.0
ro.HBW_All_As = MZAttr[1][di.6.taz]

ro.HBWMtrA_I1 = 0.0
ro.HBWMtrA_I2 = 0.0

```

## Appendix C Cube Voyager Scripts

```
ro.HBWMtrA_I3 = 0.0
ro.HBWMtrA_I4 = 0.0

ro.HBS_Mtr_As = MZAttr[2][di.6.taz]
ro.HBS_Nmt_As = 0.0
ro.HBS_All_As = MZAttr[2][di.6.taz]

ro.HBSMtrA_I1 = 0.0
ro.HBSMtrA_I2 = 0.0
ro.HBSMtrA_I3 = 0.0
ro.HBSMtrA_I4 = 0.0

ro.HBO_Mtr_As = MZAttr[3][di.6.taz]
ro.HBO_Nmt_As = 0.0
ro.HBO_All_As = MZAttr[3][di.6.taz]

ro.HBOMtrA_I1 = 0.0
ro.HBOMtrA_I2 = 0.0
ro.HBOMtrA_I3 = 0.0
ro.HBOMtrA_I4 = 0.0

ro.NHW_Mtr_As = MZAttr[4][di.6.taz]
ro.NHW_Nmt_As = 0.0
ro.NHW_All_As = MZAttr[4][di.6.taz]

ro.NHO_Mtr_As = MZAttr[5][di.6.taz]
ro.NHO_Nmt_As = 0.0
ro.NHO_All_As = MZAttr[5][di.6.taz]

WRITE RECO=2

ENDLOOP

TotExtPs = XMTotProda[1] + XMTotProda[2] + XMTotProda[3] + XMTotProda[4] +
XMTotProda[5]
TotExtAs = XMTotAttr[1] + XMTotAttr[2] + XMTotAttr[3] + XMTotAttr[4] +
XMTotAttr[5]

;=====
; Print out computed trip productions:
;-----

PRINT PRINTO=1 form=10.0csv list = ' Regional TOTAL (II,IX) COMPUTED PERSON TRIP
PRODUCTIONS SUMMARY - INTERNAL TAZs '
PRINT PRINTO=1 form=10.0csv list = '                HBW        HBS        HBO
NHW        NHO        TOTAL '
PRINT PRINTO=1 form=10.0csv list = ' -----'
PRINT PRINTO=1 form=10.0csv list = ' Motorized ', MtotProda[1],', ',
MtotProda[2],', ', MtotProda[3],', ', MtotProda[4],', ', MtotProda[5],', ', MtotProda
PRINT PRINTO=1 form=10.0csv list = ' Non-Motor. ', NMtotProda[1],', ',
NMtotProda[2],', ', NMtotProda[3],', ', NMtotProda[4],', ', NMtotProda[5],', ',
NMtotProda
PRINT PRINTO=1 form=10.0csv list = ' Total ',MNMtotProda[1],',
',MNMtotProda[2],', ',MNMtotProda[3],', ',MNMtotProda[4],', ',MNMtotProda[5],',
',MNMtotProda
PRINT PRINTO=1 form=10.0csv list = ' '
PRINT PRINTO=1 form=10.0csv list = ' '
; end
```

```
;-----
; print out Total (I-I and I-X, Motorized, NonMotorized) Attractions Tables- by
Income

PRINT PRINTO=1 form=10.0csv list = ' Regional TOTAL (II,XI) COMPUTED PERSON TRIP
ATTRATIONS SUMMARY - INTERNAL TAZs '
PRINT PRINTO=1 form=10.0csv list = '                HBW        HBS        HBO
NHW        NHO        TOTAL '
PRINT PRINTO=1 form=10.0csv list = ' -----'
PRINT PRINTO=1 form=10.0csv list = ' Motorized ', MtotAttr[1],', ',
MtotAttr[2],', ', MtotAttr[3],', ', MtotAttr[4],', ', MtotAttr[5],', ', MtotAttr
PRINT PRINTO=1 form=10.0csv list = ' Non-Motor. ', NMtotAttr[1],', ',
NMtotAttr[2],', ', NMtotAttr[3],', ', NMtotAttr[4],', ', NMtotAttr[5],', ',
NMtotAttr
PRINT PRINTO=1 form=10.0csv list = ' Total ',MNMtotAttr[1],',
',MNMtotAttr[2],', ',MNMtotAttr[3],', ',MNMtotAttr[4],', ',MNMtotAttr[5],',
',MNMtotAttr
PRINT PRINTO=1 form=10.0csv list = ' '
PRINT PRINTO=1 form=10.0csv list = ' '
; end
; Print out computed Exteranl Trips from the External File :

;-----

PRINT PRINTO=1 form=10.0csv list = ' EXTERNAL PERSON TRIPS FROM THE EXTERNAL TRIP
FILE INPUT '
PRINT PRINTO=1 form=10.0csv list = '                HBW        HBS        HBO
NHW        NHO        TOTAL '
PRINT PRINTO=1 form=10.0csv list = ' -----'
PRINT PRINTO=1 form=10.0csv list = ' X-I Trips ',XMTotProda[1],',
',XMTotProda[2],', ',XMTotProda[3],', ',XMTotProda[4],', ',XMTotProda[5],', ',TotExtPs
PRINT PRINTO=1 form=10.0csv list = ' I-X Trips ',XMTotAttr[1],',
',XMTotAttr[2],', ',XMTotAttr[3],', ',XMTotAttr[4],', ',XMTotAttr[5],', ',TotExtAs

print list = ' HBW attrs ', TotHBWCompAtt,
' HBS attrs ', TotHBSCompAtt,
' HBO attrs ', TotHBOCompAtt,
' NHW attrs ', TotNHWCompAtt,
' NHO attrs ', TotNHOCompAtt

;-----
; print out Total (I-I and I-X, Motorized, NonMotorized) Productions Tables- by
Income

PRINT PRINTO=1 form=10.0csv list = ' Regional Total (I-I, I-X & Motorized,
NonMotorized) Trip Productions Summary by Income '
PRINT PRINTO=1 form=10.0csv list = '                Income_1   Income_2   Income_3
Income_4   Sum '
PRINT PRINTO=1 form=10.0csv list = ' -----'
PRINT PRINTO=1 form=10.0csv list = ' HHS: ', I_HHa[1],', ',I_HHa[2],
', ',I_HHa[3],', ',I_HHa[4],', ',TOTHha
PRINT PRINTO=1 form=10.0csv list = ' HBW Trips: ',I_Proda[1][1],',
',I_Proda[2][1],', ',I_Proda[3][1],', ',I_Proda[4][1],', ',TOTProda[1]
PRINT PRINTO=1 form=10.0csv list = ' HBS Trips: ',I_Proda[1][2],',
',I_Proda[2][2],', ',I_Proda[3][2],', ',I_Proda[4][2],', ',TOTProda[2]
```

## Appendix C Cube Voyager Scripts

```

PRINT PRINTO=1 form=10.0csv list = ' HBO Trips: ',I_Proda[1][3],
',I_Proda[2][3],',I_Proda[3][3],',I_Proda[4][3],',TOTProda[3]
PRINT PRINTO=1 form=10.0csv list = ' NHW Trips: ',I_Proda[1][4],
',I_Proda[2][4],',I_Proda[3][4],',I_Proda[4][4],',TOTProda[4]
PRINT PRINTO=1 form=10.0csv list = ' NHO Trips: ',I_Proda[1][5],
',I_Proda[2][5],',I_Proda[3][5],',I_Proda[4][5],',TOTProda[5]
PRINT PRINTO=1 form=10.0csv list = '
-----
PRINT PRINTO=1 form=10.0csv list = '
;; end

;; print out Total Productions Tables- by Size

PRINT PRINTO=1 form=10.0csv list = ' Regional Total (I-I,I-X & Motorized,
NonMotorized) Trip Productions Summary by Size
PRINT PRINTO=1 form=10.0csv list = '          Size_1   Size_2   Size_3
Size_4   Sum
PRINT PRINTO=1 form=10.0csv list = '
-----
PRINT PRINTO=1 form=10.0csv list = ' HHS:          ', S_HHa[1]  ', ',S_HHa[2]
', ',S_HHa[3]  ', ',S_HHa[4]  ', ',TOTHHa
PRINT PRINTO=1 form=10.0csv list = ' HBW Trips: ',S_Proda[1][1],
',S_Proda[2][1],',S_Proda[3][1],',S_Proda[4][1],',TOTProda[1]
PRINT PRINTO=1 form=10.0csv list = ' HBS Trips: ',S_Proda[1][2],
',S_Proda[2][2],',S_Proda[3][2],',S_Proda[4][2],',TOTProda[2]
PRINT PRINTO=1 form=10.0csv list = ' HBO Trips: ',S_Proda[1][3],
',S_Proda[2][3],',S_Proda[3][3],',S_Proda[4][3],',TOTProda[3]
PRINT PRINTO=1 form=10.0csv list = ' NHW Trips: ',S_Proda[1][4],
',S_Proda[2][4],',S_Proda[3][4],',S_Proda[4][4],',TOTProda[4]
PRINT PRINTO=1 form=10.0csv list = ' NHO Trips: ',S_Proda[1][5],
',S_Proda[2][5],',S_Proda[3][5],',S_Proda[4][5],',TOTProda[5]
PRINT PRINTO=1 form=10.0csv list = '
-----
PRINT PRINTO=1 form=10.0csv list = '
;; end

;; print out Total Productions Tables- by Size

PRINT PRINTO=1 form=10.0csv list = ' Regional Total (I-I,I-X & Motorized,
NonMotorized) Trip Productions Summary by Vehicles
PRINT PRINTO=1 form=10.0csv list = '          0_Vehs   1_Veh   2_Vehs
3+Vehs   Sum
PRINT PRINTO=1 form=10.0csv list = '
-----
PRINT PRINTO=1 form=10.0csv list = ' HHS:          ', V_HHa[1]  ', ',V_HHa[2]
', ',V_HHa[3]  ', ',V_HHa[4]  ', ',TOTHHa
PRINT PRINTO=1 form=10.0csv list = ' HBW Trips: ',V_Proda[1][1],
',V_Proda[2][1],',V_Proda[3][1],',V_Proda[4][1],',TOTProda[1]
PRINT PRINTO=1 form=10.0csv list = ' HBS Trips: ',V_Proda[1][2],
',V_Proda[2][2],',V_Proda[3][2],',V_Proda[4][2],',TOTProda[2]
PRINT PRINTO=1 form=10.0csv list = ' HBO Trips: ',V_Proda[1][3],
',V_Proda[2][3],',V_Proda[3][3],',V_Proda[4][3],',TOTProda[3]
PRINT PRINTO=1 form=10.0csv list = ' NHW Trips: ',V_Proda[1][4],
',V_Proda[2][4],',V_Proda[3][4],',V_Proda[4][4],',TOTProda[4]
PRINT PRINTO=1 form=10.0csv list = ' NHO Trips: ',V_Proda[1][5],
',V_Proda[2][5],',V_Proda[3][5],',V_Proda[4][5],',TOTProda[5]
PRINT PRINTO=1 form=10.0csv list = '
-----
PRINT PRINTO=1 form=10.0csv list = '
;; end

print list = ' idx          ', HHInc',', IncPs',', Irate',', HHSiz',
sizPs',', Srate',', HHVeh',', VehPs',', Vrate', file=dud.dat
loop m= 1,4
irate= TotProdInca[m]/I_HHa[m] srate= TotProdSiza[m]/S_HHa[m] vrate=
TotProdVeha[m]/V_HHa[m]

```

```

print form = 10.0 list = m, I_HHa[m], TotProdInca[m], irate(10.2),
S_HHa[m],TotProdSiza[m], srate(10.2), V_HHa[m], TotProdVeha[m],vrate=dud.dat
endloop

;;
;; -----
;;
;;
ENDIF ;; If at last TAZ

ENDRUN
*copy voya*.prn mod2.rpt

```

## 53 Trip\_Generation\_Summary.s

```

*del Voya*.prn
;
ReportFile = '%_iter_%_Trip_Generation_Summary.txt'

;-----
; Trip_Generation_Summary.s - Summarize demographics and trip ends by purpose at the
juris. level ("cores" broken out)
; and at area type level.
;
;-----
;----- Create Juris.TAZ Range Lookup -----
; file include jur index(1-23), 8 TAZ 'Low/High' ranges, and jur name (Some juris.
categories have more than one TAZ range)
;
;
COPY File = JurCore.lkp
1, 1, 4, 6, 47, 49, 50, 52, 63, 65, 65, 181, 209, 282,
287, 374, 381, DC_Core ,
2, 5, 5, 48, 48, 51, 51, 64, 64, 66, 180, 210, 281, 288,
373, 382, 393, DC_Noncore ,
3, 394, 769, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Mtg ,
4, 771, 776, 778, 1404, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, PGeo ,
5, 1471, 1476, 1486, 1489, 1495, 1497, 0, 0, 0, 0, 0, 0,
0, 0, 0, ArlCore ,
6, 1405, 1470, 1477, 1485, 1490, 1494, 1498, 1545, 0, 0, 0, 0, 0,
0, 0, 0, ArlNCore ,
7, 1546, 1610, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, ALX ,
8, 1611, 2159, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, FFX ,
9, 2160, 2441, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, LDN ,
10, 2442, 2554, 2556, 2628, 2630, 2819, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, PW ,
11, 2820, 2949, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Frd ,
12, 3230, 3265, 3268, 3287, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Car ,
13, 2950, 3017, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, How ,
14, 3018, 3102, 3104, 3116, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, AnnAr ,

```

**Appendix C Cube Voyager Scripts**

```

15, 3288, 3334, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Calv, , , , , , , , , , , ,
16, 3335, 3409, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, STM, , , , , , , , , , ,
17, 3117, 3229, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Chs, , , , , , , , , , ,
18, 3604, 3653, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Fau, , , , , , , , , , ,
19, 3449, 3477, 3479, 3481, 3483, 3494, 3496, 3541, 0, 0, 0, 0, 0,
0, 0, 0, Stf, , , , , , , , , , ,
20, 3654, 3662, 3663, 3675, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Clk_Jeff, , , , , , , , , , ,
21, 3435, 3448, 3542, 3543, 3545, 3603, 0, 0, 0, 0, 0, 0,
0, 0, 0, Fbg_Spots, , , , , , , , , , ,
22, 3410, 3434, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, KGeo, , , , , , , , , , ,
23, 3676, 3722, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, Externals, , , , , , , , , , ,
ENDCOPY
;-----
; Put Juris-TAZ lookup into a DBF file
;
;
RUN PGM=MATRIX
ZONES=1
FILEI RECI = JurCore.lkp,
  Jno = 1,
  LoTAZ1 = 2,
  HiTAZ1 = 3,
  LoTAZ2 = 4,
  HiTAZ2 = 5,
  LoTAZ3 = 6,
  HiTAZ3 = 7,
  LoTAZ4 = 8,
  HiTAZ4 = 9,
  LoTAZ5 = 10,
  HiTAZ5 = 11,
  LoTAZ6 = 12,
  HiTAZ6 = 13,
  LoTAZ7 = 14,
  HiTAZ7 = 15,
  LoTAZ8 = 16,
  HiTAZ8 = 17,
  JName(c) = 18,
  DELIMITER[1]=","
  n=n+1
  RECO[1] = "JurCore.dbf",
  Fields = RECI.ALLFIELDS ;
  WRITE RECO=1
endrun
;
;-----
; now summarize demographic data and trip end data files
;-----
RUN PGM=MATRIX
ZONES =1
; zone file input
FILEI DBI[1] = "TripGen_LUfile.dbf"
;variables in file: TAZ HH HHPOP GQPOP TOTPOP TOTEMP INDEMP RETEMP
OFFEMP OTHEMP JURCODE LANDAREA

;; Juris.-TAZ lookup (core broken out)
FILEI DBI[2] = "JurCore.dbf"

;; Zonal trip productions
FILEI DBI[3] = "%_iter_%_Trip_Gen Productions_Comp.dbf"

```

```

;; variables in file:
;;TAZ HBW_MTR_PS HBW_NMT_PS HBW_ALL_PS HBWMTRP_I1 HBWMTRP_I2
HBWMTRP_I3 HBWMTRP_I4
;; HBS_MTR_PS HBS_NMT_PS HBS_ALL_PS HBSMTRP_I1 HBSMTRP_I2
HBSMTRP_I3 HBSMTRP_I4
;; HBO_MTR_PS HBO_NMT_PS HBO_ALL_PS HBOMTRP_I1 HBOMTRP_I2
HBOMTRP_I3 HBOMTRP_I4
;; NHW_MTR_PS NHW_NMT_PS NHW_ALL_PS NHO_MTR_PS NHO_NMT_PS
NHO_ALL_PS

;;Zonal final/scaled trip attractions
FILEI DBI[4] = "%_iter_%_Trip_Gen Attractions_Comp.dbf"
; variables in file:
;TAZ HBW_MTR_AS HBW_NMT_AS HBW_ALL_AS HBWMTRA_I1 HBWMTRA_I2
HBWMTRA_I3 HBWMTRA_I4
; HBS_MTR_AS HBS_NMT_AS HBS_ALL_AS HBSMTRA_I1 HBSMTRA_I2
HBSMTRA_I3 HBSMTRA_I4
; HBO_MTR_AS HBO_NMT_AS HBO_ALL_AS HBOMTRA_I1 HBOMTRA_I2
HBOMTRA_I3 HBOMTRA_I4
; NHW_MTR_AS NHW_NMT_AS NHW_ALL_AS NHO_MTR_AS NHO_NMT_AS
NHO_ALL_AS

PRINTO[1] = "@ReportFile@"
; juris and area type arrays:
ARRAY HH_Ja=25, HHPOP_Ja=25, GQPOP_Ja=25, TotPOP_Ja=25, TotEmp_Ja=25,
IndEmp_Ja=25, RetEmp_Ja=25, OffEmp_Ja=25, OthEmp_Ja=25, LArea_Ja= 25
ARRAY HH_Aa=6, HHPOP_Aa=6, GQPOP_Aa=6, TotPOP_Aa=6, TotEmp_Aa=6, IndEmp_Aa=6,
RetEmp_Aa=6, OffEmp_Aa=6, OthEmp_Aa=6, LArea_Aa= 6

ARRAY AT_Za=3675 ; zonal area type array

ARRAY MTR_Pro_Ja=5,25 ; jurisdictional motor. productions
ARRAY NMT_Pro_Ja=5,25 ; nonmot productions
ARRAY MTR_PTot_Ja=25 ; productions
ARRAY NMT_PTot_Ja=25 ; productions
ARRAY MTR_PTot_Aa=6 ; productions
ARRAY NMT_PTot_Aa=6 ; productions

ARRAY MTR_Att_Ja=5,25 ; motor. attractions
ARRAY NMT_Att_Ja=5,25 ; nonmot attractions
ARRAY MTR_ATot_Ja=25 ; motor. attractions
ARRAY NMT_ATot_Ja=25 ; nonmot attractions
ARRAY MTR_ATot_Aa=6 ; motor. attractions
ARRAY NMT_ATot_Aa=6 ; nonmot attractions

ARRAY MTR_ProInc_Ja=5,4,25 ; jurisdictional motor. productions by income group
ARRAY MTR_AttInc_Ja=5,4,25 ; motor. attractions by income group

ARRAY MTR_Pro_Aa=5,6 ; area type motor. productions
ARRAY NMT_Pro_Aa=5,6 ; nonmot productions
ARRAY MTR_Att_Aa=5,6 ; motor. attractions
ARRAY NMT_Att_Aa=5,6 ; nonmot attractions

ARRAY MTR_ProInc_Aa=5,4,6 ; area type motor. productions by income group
ARRAY MTR_AttInc_Aa=5,4,6 ; motor. attractions by income group

ARRAY HHPrate_pj=5,25
Array HHTPrate_j=25,HHTPrate_p=5

ARRAY EMPArate_pj=5,25
Array EMPTArate_j=25,EMPTArate_p=5

;#####
;=====
; process land use file first =

```

## Appendix C Cube Voyager Scripts

```

=====
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)

  ; Define input variables in zone file
  _TAZ      = di.1.TAZ
  _HH       = di.1.HH
  _HHPOP    = di.1.HHPOP
  _GQPOP    = di.1.GQPOP
  _TotPOP   = di.1.TotPOP
  _TotEmp   = di.1.TotEmp
  _IndEmp   = di.1.IndEmp
  _RetEmp   = di.1.RetEmp
  _OffEmp   = di.1.OffEmp
  _OthEmp   = di.1.OthEmp
  _LArea    = di.1.Landarea
  _At       = di.1.Atype

  AT_Za[_TAZ] = _At ; zonal area type array to be used later with trip
prod/attr summaries

  ;; Slot TAZ into a jurisdiction -----
  ;; JDX = 25 ; begin with assumed unknown juris
  Loop KK = 1,dbi.2.numrecords
    xx = DBIReadRecord(2,kk)
    IF ((_TAZ >= di.2.LoTAZ1 && _TAZ <= di.2.HiTAZ1) ||
        (_TAZ >= di.2.LoTAZ2 && _TAZ <= di.2.HiTAZ2) ||
        (_TAZ >= di.2.LoTAZ3 && _TAZ <= di.2.HiTAZ3) ||
        (_TAZ >= di.2.LoTAZ4 && _TAZ <= di.2.HiTAZ4) ||
        (_TAZ >= di.2.LoTAZ5 && _TAZ <= di.2.HiTAZ5) ||
        (_TAZ >= di.2.LoTAZ6 && _TAZ <= di.2.HiTAZ6) ||
        (_TAZ >= di.2.LoTAZ7 && _TAZ <= di.2.HiTAZ7) ||
        (_TAZ >= di.2.LoTAZ8 && _TAZ <= di.2.HiTAZ8))
      JDX = di.2.Jno
    ENDIF
  ENDLOOP

  ;; ----- Array accumulation for weighted HHs and trips by purpose---
  -----
  HH_Ja[jdx] = HH_Ja[jdx] + di.1.HH
  HHPOP_Ja[jdx] = HHPOP_Ja[jdx] + di.1.HHPOP
  GQPOP_Ja[jdx] = GQPOP_Ja[jdx] + di.1.GQPOP
  TotPOP_Ja[jdx] = TotPOP_Ja[jdx] + di.1.TotPOP
  TotEmp_Ja[jdx] = TotEmp_Ja[jdx] + di.1.TotEmp
  IndEmp_Ja[jdx] = IndEmp_Ja[jdx] + di.1.IndEmp
  RetEmp_Ja[jdx] = RetEmp_Ja[jdx] + di.1.RetEmp
  OffEmp_Ja[jdx] = OffEmp_Ja[jdx] + di.1.OffEmp
  OthEmp_Ja[jdx] = OthEmp_Ja[jdx] + di.1.OthEmp
  LArea_Ja[jdx] = LArea_Ja[jdx] + di.1.Landarea

  HH_Aa[_At] = HH_Aa[_At] + di.1.HH
  HHPOP_Aa[_At] = HHPOP_Aa[_At] + di.1.HHPOP
  GQPOP_Aa[_At] = GQPOP_Aa[_At] + di.1.GQPOP
  TotPOP_Aa[_At] = TotPOP_Aa[_At] + di.1.TotPOP
  TotEmp_Aa[_At] = TotEmp_Aa[_At] + di.1.TotEmp
  IndEmp_Aa[_At] = IndEmp_Aa[_At] + di.1.IndEmp
  RetEmp_Aa[_At] = RetEmp_Aa[_At] + di.1.RetEmp
  OffEmp_Aa[_At] = OffEmp_Aa[_At] + di.1.OffEmp
  OthEmp_Aa[_At] = OthEmp_Aa[_At] + di.1.OthEmp
  LArea_Aa[_At] = LArea_Aa[_At] + di.1.Landarea

  HH_Tot = HH_Tot + di.1.HH
  HHPOP_Tot = HHPOP_Tot + di.1.HHPOP
  GQPOP_Tot = GQPOP_Tot + di.1.GQPOP
  TotPOP_Tot = TotPOP_Tot + di.1.TotPOP
  TotEmp_Tot = TotEmp_Tot + di.1.TotEmp

```

```

IndEmp_Tot = IndEmp_Tot + di.1.IndEmp
RetEmp_Tot = RetEmp_Tot + di.1.RetEmp
OffEmp_Tot = OffEmp_Tot + di.1.OffEmp
OthEmp_Tot = OthEmp_Tot + di.1.OthEmp
LArea_Tot = LArea_Tot + di.1.Landarea

  ;; ----- End of Array accumulation -----
  ENDLOOP

  ;; ===== Printout Reports =====
  ;; -----
  Print PRINTO=1 LIST= '\n','\n',' Land Activity by Jurisdiction ', '\n','\n'
  Print PRINTO=1 LIST= ' Jurisdiction ', ' Households', ' HH_Pop', '
GQ_Pop', ' Tot_Pop', ' Total_Emp', ' IND_Emp', ' RET_Emp', ' Off_Emp', '
Oth_Emp', ' LandArea', ' HH_Size', ' JobHHRatio ', '\n'

  Loop KK = 1,dbi.2.numrecords
    xx = DBIReadRecord(2,kk)
    jdx = di.2.Jno

    HH_Size = 0
    JobHHRat = 0
    if (HH_Ja[jdx] > 0) HH_Size = HHPOP_Ja[jdx] / HH_Ja[jdx]
    if (TotEmp_Ja[jdx] > 0) JobHHRat = TotEmp_Ja[jdx] / HH_Ja[jdx]
    Print form=10csv PRINTO=1 LIST=
      di.2.JNAME(c25), ' ', HH_Ja[jdx], ' ', HHPOP_Ja[jdx], ' ', GQPOP_Ja[jdx], '
', TotPop_Ja[jdx], ' ', TotEmp_Ja[jdx], ' ',
      INDEmp_Ja[jdx], ' ', RetEmp_Ja[jdx], ' ', OffEmp_Ja[jdx], '
', OthEmp_Ja[jdx], ' ', LArea_Ja[jdx] (10.3csv),
      HH_Size (10.3csv) ' ', JobHHRat (10.3csv)

  ENDLOOP

  HH_Size = 0
  JobHHRat = 0
  if (HH_Tot > 0) HH_Size = HHPOP_Tot / HH_Tot
  if (TotEmp_Tot > 0) JobHHRat = TotEmp_Tot / HH_Tot

  Print PRINTO=1 LIST= '
Print form=10csv PRINTO=1 LIST= ' TOTAL ' ', HH_Tot, ' ',
HHPOP_Tot, ' ', GQPOP_Tot, ' ', TotPop_Tot, ' ', TotEMP_Tot, ' ',
INDEmp_Tot, '
', RetEmp_Tot, ' ', OffEmp_Tot, ' ', OthEmp_Tot, ' ', LArea_Tot (10.3csv),
HH_Size (10.3csv) ',
', JobHHRat (10.3csv)

  ;; -----
  Print PRINTO=1 LIST= '\n','\n',' Land Activity by Area Type ',
'\n','\n'
  Print PRINTO=1 LIST= ' Area Type ', ' Households', ' HH_Pop', '
GQ_Pop', ' Tot_Pop', ' Total_Emp', ' IND_Emp', ' RET_Emp', ' Off_Emp', '
Oth_Emp', ' LandArea', ' HH_Size', ' JobHHRatio ', '\n'

  Loop KK = 1,6

  Adx = kk

  HH_Size = 0

```

# Appendix C Cube Voyager Scripts

```

JobHHRat = 0
if (HH_Aa[Adx] > 0) HH_Size = HHPop_Aa[Adx] / HH_Aa[Adx]
if (TotEmp_Aa[Adx] > 0) JobHHRat = TotEmp_Aa[Adx] / HH_Aa[Adx]
Print form=10csv PRINTO=1 LIST=
  Adx, ' ', HH_Aa[Adx], ' ', HHPOP_Aa[Adx], '
',GQPOP_Aa[Adx], ' ',TotPop_Aa[Adx], ' ',TotEMP_Aa[Adx], ' ',
  INDEmp_Aa[Adx], ' ',RetEmp_Aa[Adx], '
',OffEmp_Aa[Adx], ' ',OthEmp_Aa[Adx], ' ',Larea_Aa[Adx](10.3csv),
  HH_Size(10.3csv) ' ',JobHHRat(10.3csv)
ENDLOOP

HH_Size = 0
JobHHRat = 0
if (HH_Tot > 0) HH_Size = HHPop_Tot / HH_Tot
if (TotEmp_Tot > 0) JobHHRat = TotEmp_Tot / HH_Tot

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST=' TOTAL ' ', HH_Tot, ' ',
HHPOP_Tot, ' ',GQPOP_Tot, ' ',TotPop_Tot, ' ',TotEMP_Tot, ' ',
  INDEmp_Tot, '
',RetEmp_Tot, ' ',OffEmp_Tot, ' ',OthEmp_Tot, ' ',Larea_Tot(10.3csv),
  HH_Size(10.3csv) '
',JobHHRat(10.3csv)
;#####
;#####
;=====
; process trip productions next =
;=====

LOOP K = 1,dbi.3.NUMRECORDS
x = DBIReadRecord(3,k)
if (K <= 3675)
; Define input variables in production zone file

_TAZ = di.3.TAZ
_HBW_MTR_PS = di.3.HBW_MTR_PS
_HBW_NMT_PS = di.3.HBW_NMT_PS
_HBW_ALL_PS = di.3.HBW_ALL_PS
_HBWMTRP_I1 = di.3.HBWMTRP_I1
_HBWMTRP_I2 = di.3.HBWMTRP_I2
_HBWMTRP_I3 = di.3.HBWMTRP_I3
_HBWMTRP_I4 = di.3.HBWMTRP_I4
_HBS_MTR_PS = di.3.HBS_MTR_PS
_HBS_NMT_PS = di.3.HBS_NMT_PS
_HBS_ALL_PS = di.3.HBS_ALL_PS
_HBSMTRP_I1 = di.3.HBSMTRP_I1
_HBSMTRP_I2 = di.3.HBSMTRP_I2
_HBSMTRP_I3 = di.3.HBSMTRP_I3
_HBSMTRP_I4 = di.3.HBSMTRP_I4
_HBO_MTR_PS = di.3.HBO_MTR_PS
_HBO_NMT_PS = di.3.HBO_NMT_PS
_HBO_ALL_PS = di.3.HBO_ALL_PS
_HBOMTRP_I1 = di.3.HBOMTRP_I1
_HBOMTRP_I2 = di.3.HBOMTRP_I2
_HBOMTRP_I3 = di.3.HBOMTRP_I3
_HBOMTRP_I4 = di.3.HBOMTRP_I4
_NHW_MTR_PS = di.3.NHW_MTR_PS
_NHW_NMT_PS = di.3.NHW_NMT_PS
_NHW_ALL_PS = di.3.NHW_ALL_PS
_NHO_MTR_PS = di.3.NHO_MTR_PS
_NHO_NMT_PS = di.3.NHO_NMT_PS
_NHO_ALL_PS = di.3.NHO_ALL_PS

ADX = AT_Za[_TAZ] ; slot cuurent taz into an area
type
; Slot TAZ into a jurisdiction -----

```

```

Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
IF ((_TAZ >= di.2.LoTAZ1 && _TAZ <= di.2.HiTAZ1) ||
(_TAZ >= di.2.LoTAZ2 && _TAZ <= di.2.HiTAZ2) ||
(_TAZ >= di.2.LoTAZ3 && _TAZ <= di.2.HiTAZ3) ||
(_TAZ >= di.2.LoTAZ4 && _TAZ <= di.2.HiTAZ4) ||
(_TAZ >= di.2.LoTAZ5 && _TAZ <= di.2.HiTAZ5) ||
(_TAZ >= di.2.LoTAZ6 && _TAZ <= di.2.HiTAZ6) ||
(_TAZ >= di.2.LoTAZ7 && _TAZ <= di.2.HiTAZ7) ||
(_TAZ >= di.2.LoTAZ8 && _TAZ <= di.2.HiTAZ8))
  JDX = di.2.Jno
ENDIF
ENDLOOP

; ----- Array accumulation for productions -----

; total Ps
Mtr_Pro_ja[1][jdx] = Mtr_Pro_ja[1][jdx] + di.3.HBW_MTR_Ps
Mtr_Pro_Aa[1][adx] = Mtr_Pro_Aa[1][adx] + di.3.HBW_MTR_Ps
Mtr_Pro_ja[2][jdx] = Mtr_Pro_ja[2][jdx] + di.3.HBS_MTR_Ps
Mtr_Pro_Aa[2][adx] = Mtr_Pro_Aa[2][adx] + di.3.HBS_MTR_Ps
Mtr_Pro_ja[3][jdx] = Mtr_Pro_ja[3][jdx] + di.3.HBO_MTR_Ps
Mtr_Pro_Aa[3][adx] = Mtr_Pro_Aa[3][adx] + di.3.HBO_MTR_Ps
Mtr_Pro_ja[4][jdx] = Mtr_Pro_ja[4][jdx] + di.3.NHW_MTR_Ps
Mtr_Pro_Aa[4][adx] = Mtr_Pro_Aa[4][adx] + di.3.NHW_MTR_Ps
Mtr_Pro_ja[5][jdx] = Mtr_Pro_ja[5][jdx] + di.3.NHO_MTR_Ps
Mtr_Pro_Aa[5][adx] = Mtr_Pro_Aa[5][adx] + di.3.NHO_MTR_Ps

MTR_PTot_Ja[jdx] = MTR_PTot_Ja[jdx] + di.3.HBW_MTR_Ps + di.3.HBS_MTR_Ps +
di.3.HBO_MTR_Ps + di.3.NHW_MTR_Ps + di.3.NHO_MTR_Ps
MTR_PTot_Aa[adx] = MTR_PTot_Aa[adx] + di.3.HBW_MTR_Ps + di.3.HBS_MTR_Ps +
di.3.HBO_MTR_Ps + di.3.NHW_MTR_Ps + di.3.NHO_MTR_Ps

NMT_Pro_ja[1][jdx] = NMT_Pro_ja[1][jdx] + di.3.HBW_NMT_Ps
NMT_Pro_Aa[1][adx] = NMT_Pro_Aa[1][adx] + di.3.HBW_NMT_Ps
NMT_Pro_ja[2][jdx] = NMT_Pro_ja[2][jdx] + di.3.HBS_NMT_Ps
NMT_Pro_Aa[2][adx] = NMT_Pro_Aa[2][adx] + di.3.HBS_NMT_Ps
NMT_Pro_ja[3][jdx] = NMT_Pro_ja[3][jdx] + di.3.HBO_NMT_Ps
NMT_Pro_Aa[3][adx] = NMT_Pro_Aa[3][adx] + di.3.HBO_NMT_Ps
NMT_Pro_ja[4][jdx] = NMT_Pro_ja[4][jdx] + di.3.NHW_NMT_Ps
NMT_Pro_Aa[4][adx] = NMT_Pro_Aa[4][adx] + di.3.NHW_NMT_Ps
NMT_Pro_ja[5][jdx] = NMT_Pro_ja[5][jdx] + di.3.NHO_NMT_Ps
NMT_Pro_Aa[5][adx] = NMT_Pro_Aa[5][adx] + di.3.NHO_NMT_Ps

NMT_PTot_Ja[jdx] = NMT_PTot_Ja[jdx] + di.3.HBW_NMT_Ps + di.3.HBS_NMT_Ps +
di.3.HBO_NMT_Ps + di.3.NHW_NMT_Ps + di.3.NHO_NMT_Ps
NMT_PTot_Aa[adx] = NMT_PTot_Aa[adx] + di.3.HBW_NMT_Ps + di.3.HBS_NMT_Ps +
di.3.HBO_NMT_Ps + di.3.NHW_NMT_Ps + di.3.NHO_NMT_Ps

; total HB motorized Ps by income
Mtr_ProInc_ja[1][1][jdx] = Mtr_ProInc_ja[1][1][jdx] + di.3.HBWMTRP_I1
Mtr_ProInc_Aa[1][1][adx] = Mtr_ProInc_Aa[1][1][adx] + di.3.HBWMTRP_I1
Mtr_ProInc_ja[2][1][jdx] = Mtr_ProInc_ja[2][1][jdx] + di.3.HBSMTRP_I1
Mtr_ProInc_Aa[2][1][adx] = Mtr_ProInc_Aa[2][1][adx] + di.3.HBSMTRP_I1
Mtr_ProInc_ja[3][1][jdx] = Mtr_ProInc_ja[3][1][jdx] + di.3.HBOMTRP_I1
Mtr_ProInc_Aa[3][1][adx] = Mtr_ProInc_Aa[3][1][adx] + di.3.HBOMTRP_I1

Mtr_ProInc_ja[1][2][jdx] = Mtr_ProInc_ja[1][2][jdx] + di.3.HBWMTRP_I2
Mtr_ProInc_Aa[1][2][adx] = Mtr_ProInc_Aa[1][2][adx] + di.3.HBWMTRP_I2
Mtr_ProInc_ja[2][2][jdx] = Mtr_ProInc_ja[2][2][jdx] + di.3.HBSMTRP_I2
Mtr_ProInc_Aa[2][2][adx] = Mtr_ProInc_Aa[2][2][adx] + di.3.HBSMTRP_I2
Mtr_ProInc_ja[3][2][jdx] = Mtr_ProInc_ja[3][2][jdx] + di.3.HBOMTRP_I2
Mtr_ProInc_Aa[3][2][adx] = Mtr_ProInc_Aa[3][2][adx] + di.3.HBOMTRP_I2

Mtr_ProInc_ja[1][3][jdx] = Mtr_ProInc_ja[1][3][jdx] + di.3.HBWMTRP_I3
Mtr_ProInc_Aa[1][3][adx] = Mtr_ProInc_Aa[1][3][adx] + di.3.HBWMTRP_I3

```

## Appendix C Cube Voyager Scripts

```

Mtr_ProInc_ja[2][3][jdx] = Mtr_ProInc_ja[2][3][jdx] + di.3.HBSMTRP_I3
Mtr_ProInc_Aa[2][3][adx] = Mtr_ProInc_Aa[2][3][adx] + di.3.HBSMTRP_I3
Mtr_ProInc_ja[3][3][jdx] = Mtr_ProInc_ja[3][3][jdx] + di.3.HBOMTRP_I3
Mtr_ProInc_Aa[3][3][adx] = Mtr_ProInc_Aa[3][3][adx] + di.3.HBOMTRP_I3

Mtr_ProInc_ja[1][4][jdx] = Mtr_ProInc_ja[1][4][jdx] + di.3.HBWMTRP_I4
Mtr_ProInc_Aa[1][4][adx] = Mtr_ProInc_Aa[1][4][adx] + di.3.HBWMTRP_I4
Mtr_ProInc_ja[2][4][jdx] = Mtr_ProInc_ja[2][4][jdx] + di.3.HBSMTRP_I4
Mtr_ProInc_Aa[2][4][adx] = Mtr_ProInc_Aa[2][4][adx] + di.3.HBSMTRP_I4
Mtr_ProInc_ja[3][4][jdx] = Mtr_ProInc_ja[3][4][jdx] + di.3.HBOMTRP_I4
Mtr_ProInc_Aa[3][4][adx] = Mtr_ProInc_Aa[3][4][adx] + di.3.HBOMTRP_I4

;; totals

TotHBWMtrPs = TotHBWMtrPs + di.3.HBW_MTR_Ps   TotHBWNmtPs = TotHBWNmtPs +
di.3.HBW_NMT_Ps
TotHBSMtrPs = TotHBSMtrPs + di.3.HBS_MTR_Ps   TotHBSNmtPs = TotHBSNmtPs +
di.3.HBS_NMT_Ps
TotHBOMtrPs = TotHBOMtrPs + di.3.HBO_MTR_Ps   TotHBONmtPs = TotHBONmtPs +
di.3.HBO_NMT_Ps
TotNHWMtrPs = TotNHWMtrPs + di.3.NHW_MTR_Ps   TotNHWNmtPs = TotNHWNmtPs +
di.3.NHW_NMT_Ps
TotNHOMtrPs = TotNHOMtrPs + di.3.NHO_MTR_Ps   TotNHONmtPs = TotNHONmtPs +
di.3.NHO_NMT_Ps

TotMtrPs = TotMtrPs + di.3.HBW_MTR_Ps + di.3.HBS_MTR_Ps + di.3.HBO_MTR_Ps
+ di.3.NHW_MTR_Ps + di.3.NHO_MTR_Ps
TotNmtPs = TotNmtPs + di.3.HBW_NMT_Ps + di.3.HBS_NMT_Ps + di.3.HBO_NMT_Ps
+ di.3.NHW_NMT_Ps + di.3.NHO_NMT_Ps

TotHBWMtrPs_I1 = TotHBWMtrPs_I1 + di.3.HBWMTRP_I1
TotHBSMtrPs_I1 = TotHBSMtrPs_I1 + di.3.HBSMTRP_I1
TotHBOMtrPs_I1 = TotHBOMtrPs_I1 + di.3.HBOMTRP_I1

TotHBWMtrPs_I2 = TotHBWMtrPs_I2 + di.3.HBWMTRP_I2
TotHBSMtrPs_I2 = TotHBSMtrPs_I2 + di.3.HBSMTRP_I2
TotHBOMtrPs_I2 = TotHBOMtrPs_I2 + di.3.HBOMTRP_I2

TotHBWMtrPs_I3 = TotHBWMtrPs_I3 + di.3.HBWMTRP_I3
TotHBSMtrPs_I3 = TotHBSMtrPs_I3 + di.3.HBSMTRP_I3
TotHBOMtrPs_I3 = TotHBOMtrPs_I3 + di.3.HBOMTRP_I3

TotHBWMtrPs_I4 = TotHBWMtrPs_I4 + di.3.HBWMTRP_I4
TotHBSMtrPs_I4 = TotHBSMtrPs_I4 + di.3.HBSMTRP_I4
TotHBOMtrPs_I4 = TotHBOMtrPs_I4 + di.3.HBOMTRP_I4

ENDIF      ;; -----      End of Array accumulation      ---
-----
ENDLOOP

Loop Jdx = 1,25
  if (HH_ja[jdx] > 0) HHPrate_pj[1][Jdx] = Mtr_Pro_ja[1][jdx] / HH_ja[jdx]
  if (HH_ja[jdx] > 0) HHPrate_pj[2][Jdx] = Mtr_Pro_ja[2][jdx] / HH_ja[jdx]
  if (HH_ja[jdx] > 0) HHPrate_pj[3][Jdx] = Mtr_Pro_ja[3][jdx] / HH_ja[jdx]
  if (HH_ja[jdx] > 0) HHPrate_pj[4][Jdx] = Mtr_Pro_ja[4][jdx] / HH_ja[jdx]
  if (HH_ja[jdx] > 0) HHPrate_pj[5][Jdx] = Mtr_Pro_ja[5][jdx] / HH_ja[jdx]

  if (HH_ja[jdx] > 0) HHTPrate_j[jdx] = MTR_PTot_Ja[jdx] / HH_ja[jdx]
ENDLOOP

  if (HH_Tot > 0) HHTPrate_p[1] = TotHBWMtrPs / HH_Tot
  if (HH_Tot > 0) HHTPrate_p[2] = TotHBSMtrPs / HH_Tot
  if (HH_Tot > 0) HHTPrate_p[3] = TotHBOMtrPs / HH_Tot
  if (HH_Tot > 0) HHTPrate_p[4] = TotNHWMtrPs / HH_Tot
  if (HH_Tot > 0) HHTPrate_p[5] = TotNHOMtrPs / HH_Tot

```

```

  if (HH_Tot>0) TotRATESALL =
(TotHBWMtrPs+TotHBSMtrPs+TotHBOMtrPs+TotNHWMtrPs+TotNHOMtrPs) / HH_Tot

; ===== Printout Trip Production Reports
-----
; -----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Productions by Purpose and
Jurisdiction ', '\n','\n'
Print PRINTO=1 LIST= ' Jurisdiction ', ' HBW', ' HBS', '
HBO', ' NHW', ' NHO', ' Total', '\n'

Loop KK = 1,dbi.2.numrecords
  xx = DBIReadRecord(2,kk)
  jdx = di.2.Jno

  Print form=10csv PRINTO=1 LIST=
  di.2.JNAME(c25), ' ',Mtr_Pro_Ja[1][jdx], ' ',Mtr_Pro_Ja[2][jdx],
',Mtr_Pro_Ja[3][jdx], ' ',Mtr_Pro_Ja[4][jdx], ' ',
Mtr_Pro_Ja[5][jdx], ' ',Mtr_Ptot_Ja[jdx]
ENDLOOP

Print PRINTO=1 LIST= '
Print form=10csv PRINTO=1 LIST= ' TOTAL '
', TOTHBWMTRPs, ' ',TOTHBSMTRPs, ' ',TOTHBOMTRPs, '
',TOTNHWMTRPs, ' ',TOTNHOMTRPs, ' ',TOTMTRPs

; -----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Productions per Household by
Purpose and Jurisdiction ', '\n','\n'
Print PRINTO=1 LIST= ' Jurisdiction ', ' HBW', ' HBS', '
HBO', ' NHW', ' NHO', ' Total', '\n'

Loop KK = 1,dbi.2.numrecords
  xx = DBIReadRecord(2,kk)
  jdx = di.2.Jno

  Print form=10.2csv PRINTO=1 LIST=
  di.2.JNAME(c25), ' ',HHPrate_pj[1][jdx], ' ',HHPrate_pj[2][jdx],
',HHPrate_pj[3][jdx], ' ',HHPrate_pj[4][jdx], ' ',
HHPrate_pj[5][jdx], ' ',HHTPrate_j[jdx]
ENDLOOP

Print PRINTO=1 LIST= '
Print form=10.2csv PRINTO=1 LIST= ' TOTAL '
', HHTPrate_P[1], ' ',HHTPrate_P[2], '
',HHTPrate_P[3], ' ',HHTPrate_P[4], ' ',HHTPrate_P[5], ' ',TOTRATESALL

;; -----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Productions by Purpose and Area
Type ', '\n','\n'
Print PRINTO=1 LIST= ' Jurisdiction ', ' HBW', ' HBS', '
HBO', ' NHW', ' NHO', ' Total', '\n'

Loop KK = 1,6

  Adx = kk

  Print form=10csv PRINTO=1 LIST= ' Area Type: ',Adx(5),
' ',Mtr_Pro_Aa[1][adx], ' ',Mtr_Pro_Aa[2][adx],
',Mtr_Pro_Aa[3][adx], ' ',Mtr_Pro_Aa[4][adx], ' ',
Mtr_Pro_Aa[5][adx], ' ',Mtr_Ptot_Aa[adx]

```

## Appendix C Cube Voyager Scripts

```

ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          '
          TOHBMWTRPs, ' ',TOHBSMTRPs, ' ',TOHBOMTRPs, '
',TOTNHWMTRPs, ' ',TOTNHOMTRPs, ' ',TOTMTRPs
; -----
Print PRINTO=1 LIST= '\n','\n',' NonMotorized Trip Productions by Purpose and
Jurisdiction ' '\n','\n'
Print PRINTO=1 LIST='          Jurisdiction          ' '          HBW',' '          HBS',' '
HBO',' '          NHW',' '          NHO',' '          Total','\n'

Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
jdx = di.2.Jno

Print form=10csv PRINTO=1 LIST=
di.2.JNAME(c25), ' ',Nmt_Pro_Ja[1][jdx], ' ',Nmt_Pro_Ja[2][jdx], '
',Nmt_Pro_Ja[3][jdx], ' ',Nmt_Pro_Ja[4][jdx], ' ',
Nmt_Pro_Ja[5][jdx], ' ',Nmt_Ptot_Ja[jdx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          '
          TOHBNMTRPs, ' ',TOHBSNMTRPs, ' ',TOHBONMTRPs, '
',TOTNHWNMTRPs, ' ',TOTNHONMTRPs, ' ',TOTNMTRPs
; -----
Print PRINTO=1 LIST= '\n','\n',' NonMotorized Trip Productions by Purpose and Area
Type ' '\n','\n'
Print PRINTO=1 LIST='          Jurisdiction          ' '          HBW',' '          HBS',' '
HBO',' '          NHW',' '          NHO',' '          Total','\n'

Loop KK = 1,6
Adx = kk

Print form=10csv PRINTO=1 LIST='          Area Type:          'Adx(5),
          ' ',Nmt_Pro_Aa[1][adx], ' ',Nmt_Pro_Aa[2][adx], '
',Nmt_Pro_Aa[3][adx], ' ',Nmt_Pro_Aa[4][adx], ' ',
Nmt_Pro_Aa[5][adx], ' ',Nmt_Ptot_Aa[adx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          '
          TOHBNMTRPs, ' ',TOHBSNMTRPs, ' ',TOHBONMTRPs, '
',TOTNHWNMTRPs, ' ',TOTNHONMTRPs, ' ',TOTNMTRPs
; -----
Print PRINTO=1 LIST= '\n','\n',' Home-Based Motorized Trip Productions by Purpose,
Income, and Jurisdiction ' '\n','\n'
Print PRINTO=1 LIST='          Jurisdiction          ' '          HBW_Incl',' '          HBW_Inc2',' '
HBW_Inc3',' '          HBW_Inc4','
          '          HBS_Incl',' '          HBS_Inc2',' '
HBS_Inc3',' '          HBS_Inc4','
          '          HBO_Incl',' '          HBO_Inc2',' '
HBO_Inc3',' '          HBO_Inc4',' '\n'

Loop KK = 1,dbi.2.numrecords

```

```

xx = DBIReadRecord(2,kk)
jdx = di.2.Jno

Print form=10csv PRINTO=1 LIST=
di.2.JNAME(c25), ' ',Mtr_ProInc_ja[1][1][jdx], ' ',Mtr_ProInc_ja[1][2][jdx], '
',Mtr_ProInc_ja[1][3][jdx], ' ',Mtr_ProInc_ja[1][4][jdx], ' ',
Mtr_ProInc_ja[2][1][jdx], ' ',Mtr_ProInc_ja[2][2][jdx], '
',Mtr_ProInc_ja[2][3][jdx], ' ',Mtr_ProInc_ja[2][4][jdx], ' ',
Mtr_ProInc_ja[3][1][jdx], ' ',Mtr_ProInc_ja[3][2][jdx], '
',Mtr_ProInc_ja[3][3][jdx], ' ',Mtr_ProInc_ja[3][4][jdx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          '
          TOHBMWTRPs_i1, ' ',TOHBMWTRPs_i2, '
',TOHBMWTRPs_i3, ' ',TOHBMWTRPs_i4, ' ',
          TOHBSMTRPs_i1, ' ',TOHBSMTRPs_i2, '
',TOHBSMTRPs_i3, ' ',TOHBSMTRPs_i4, ' ',
          TOHBOMTRPs_i1, ' ',TOHBOMTRPs_i2, '
',TOHBOMTRPs_i3, ' ',TOHBOMTRPs_i4
; -----
; -----
Print PRINTO=1 LIST= '\n','\n',' Home-Based Motorized Trip Productions by Purpose,
Income, and Area Type ' '\n','\n'
Print PRINTO=1 LIST='          Area Type          ' '          HBW_Incl',' '          HBW_Inc2',' '
HBW_Inc3',' '          HBW_Inc4','
          '          HBS_Incl',' '          HBS_Inc2',' '
HBS_Inc3',' '          HBS_Inc4','
          '          HBO_Incl',' '          HBO_Inc2',' '
HBO_Inc3',' '          HBO_Inc4',' '\n'

Loop KK = 1,6
Adx = kk

Print form=10csv PRINTO=1 LIST='          Area Type:          'Adx(5),
          ' ',Mtr_ProInc_Aa[1][1][Adx], ' ',Mtr_ProInc_Aa[1][2][Adx], '
',Mtr_ProInc_Aa[1][3][Adx], ' ',Mtr_ProInc_Aa[1][4][Adx], ' ',
Mtr_ProInc_Aa[2][1][Adx], ' ',Mtr_ProInc_Aa[2][2][Adx], '
',Mtr_ProInc_Aa[2][3][Adx], ' ',Mtr_ProInc_Aa[2][4][Adx], ' ',
Mtr_ProInc_Aa[3][1][Adx], ' ',Mtr_ProInc_Aa[3][2][Adx], '
',Mtr_ProInc_Aa[3][3][Adx], ' ',Mtr_ProInc_Aa[3][4][Adx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          '
          TOHBMWTRPs_i1, ' ',TOHBMWTRPs_i2, '
',TOHBMWTRPs_i3, ' ',TOHBMWTRPs_i4, ' ',
          TOHBSMTRPs_i1, ' ',TOHBSMTRPs_i2, '
',TOHBSMTRPs_i3, ' ',TOHBSMTRPs_i4, ' ',
          TOHBOMTRPs_i1, ' ',TOHBOMTRPs_i2, '
',TOHBOMTRPs_i3, ' ',TOHBOMTRPs_i4
; =====
; =====
; process Trip Attractions next =
; =====
LOOP K = 1,dbi.4.NUMRECORDS
x = DBIReadRecord(4,k)

```



## Appendix C Cube Voyager Scripts

```

if (K <= 3675)
; Define input variables in ATTRACTION zone file

_TAZ = di.4.TAZ
_HBW_MTR_AS = di.4.HBW_MTR_AS
_HBW_NMT_AS = di.4.HBW_NMT_AS
_HBW_ALL_AS = di.4.HBW_ALL_AS
_HBWMTRA_I1 = di.4.HBWMTRA_I1
_HBWMTRA_I2 = di.4.HBWMTRA_I2
_HBWMTRA_I3 = di.4.HBWMTRA_I3
_HBWMTRA_I4 = di.4.HBWMTRA_I4
_HBS_MTR_AS = di.4.HBS_MTR_AS
_HBS_NMT_AS = di.4.HBS_NMT_AS
_HBS_ALL_AS = di.4.HBS_ALL_AS
_HBSMTRA_I1 = di.4.HBSMTRA_I1
_HBSMTRA_I2 = di.4.HBSMTRA_I2
_HBSMTRA_I3 = di.4.HBSMTRA_I3
_HBSMTRA_I4 = di.4.HBSMTRA_I4
_HBO_MTR_AS = di.4.HBO_MTR_AS
_HBO_NMT_AS = di.4.HBO_NMT_AS
_HBO_ALL_AS = di.4.HBO_ALL_AS
_HBOMTRA_I1 = di.4.HBOMTRA_I1
_HBOMTRA_I2 = di.4.HBOMTRA_I2
_HBOMTRA_I3 = di.4.HBOMTRA_I3
_HBOMTRA_I4 = di.4.HBOMTRA_I4
_NHW_MTR_AS = di.4.NHW_MTR_AS
_NHW_NMT_AS = di.4.NHW_NMT_AS
_NHW_ALL_AS = di.4.NHW_ALL_AS
_NHO_MTR_AS = di.4.NHO_MTR_AS
_NHO_NMT_AS = di.4.NHO_NMT_AS
_NHO_ALL_AS = di.4.NHO_ALL_AS

ADX = AT_Za[_TAZ] ; slot cuurent taz into an area

type

;; Slot TAZ into a jurisdiction -----
Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
IF ((_TAZ >= di.2.LoTAZ1 && _TAZ <= di.2.HiTAZ1) ||
(_TAZ >= di.2.LoTAZ2 && _TAZ <= di.2.HiTAZ2) ||
(_TAZ >= di.2.LoTAZ3 && _TAZ <= di.2.HiTAZ3) ||
(_TAZ >= di.2.LoTAZ4 && _TAZ <= di.2.HiTAZ4) ||
(_TAZ >= di.2.LoTAZ5 && _TAZ <= di.2.HiTAZ5) ||
(_TAZ >= di.2.LoTAZ6 && _TAZ <= di.2.HiTAZ6) ||
(_TAZ >= di.2.LoTAZ7 && _TAZ <= di.2.HiTAZ7) ||
(_TAZ >= di.2.LoTAZ8 && _TAZ <= di.2.HiTAZ8))
JDX = di.2.Jno
ENDIF
ENDLOOP

;; ----- Array accumulation for productions-----

;; total As
Mtr_Att_ja[1][jdx] = Mtr_Att_ja[1][jdx] + di.4.HBW_MTR_As
Mtr_Att_Aa[1][adx] = Mtr_Att_Aa[1][adx] + di.4.HBW_MTR_As
Mtr_Att_ja[2][jdx] = Mtr_Att_ja[2][jdx] + di.4.HBS_MTR_As
Mtr_Att_Aa[2][adx] = Mtr_Att_Aa[2][adx] + di.4.HBS_MTR_As
Mtr_Att_ja[3][jdx] = Mtr_Att_ja[3][jdx] + di.4.HBO_MTR_As
Mtr_Att_Aa[3][adx] = Mtr_Att_Aa[3][adx] + di.4.HBO_MTR_As
Mtr_Att_ja[4][jdx] = Mtr_Att_ja[4][jdx] + di.4.NHW_MTR_As
Mtr_Att_Aa[4][adx] = Mtr_Att_Aa[4][adx] + di.4.NHW_MTR_As
Mtr_Att_ja[5][jdx] = Mtr_Att_ja[5][jdx] + di.4.NHO_MTR_As
Mtr_Att_Aa[5][adx] = Mtr_Att_Aa[5][adx] + di.4.NHO_MTR_As

MTR_ATot_Ja[jdx] = MTR_ATot_Ja[jdx] + di.4.HBW_MTR_As + di.4.HBS_MTR_As +
di.4.HBO_MTR_As + di.4.NHW_MTR_As + di.4.NHO_MTR_As
MTR_ATot_Aa[adx] = MTR_ATot_Aa[adx] + di.4.HBW_MTR_As + di.4.HBS_MTR_As +
di.4.HBO_MTR_As + di.4.NHW_MTR_As + di.4.NHO_MTR_As

```

```

NMT_Att_ja[1][jdx] = NMT_Att_ja[1][jdx] + di.4.HBW_NMT_As
NMT_Att_Aa[1][adx] = NMT_Att_Aa[1][adx] + di.4.HBW_NMT_As
NMT_Att_ja[2][jdx] = NMT_Att_ja[2][jdx] + di.4.HBS_NMT_As
NMT_Att_Aa[2][adx] = NMT_Att_Aa[2][adx] + di.4.HBS_NMT_As
NMT_Att_ja[3][jdx] = NMT_Att_ja[3][jdx] + di.4.HBO_NMT_As
NMT_Att_Aa[3][adx] = NMT_Att_Aa[3][adx] + di.4.HBO_NMT_As
NMT_Att_ja[4][jdx] = NMT_Att_ja[4][jdx] + di.4.NHW_NMT_As
NMT_Att_Aa[4][adx] = NMT_Att_Aa[4][adx] + di.4.NHW_NMT_As
NMT_Att_ja[5][jdx] = NMT_Att_ja[5][jdx] + di.4.NHO_NMT_As
NMT_Att_Aa[5][adx] = NMT_Att_Aa[5][adx] + di.4.NHO_NMT_As

NMT_ATot_Ja[jdx] = NMT_ATot_Ja[jdx] + di.4.HBW_NMT_As + di.4.HBS_NMT_As +
di.4.HBO_NMT_As + di.4.NHW_NMT_As + di.4.NHO_NMT_As
NMT_ATot_Aa[adx] = NMT_ATot_Aa[adx] + di.4.HBW_NMT_As + di.4.HBS_NMT_As +
di.4.HBO_NMT_As + di.4.NHW_NMT_As + di.4.NHO_NMT_As

;; total HB motorized As by income
Mtr_AttInc_ja[1][1][jdx] = Mtr_AttInc_ja[1][1][jdx] + di.4.HBWMTRA_I1
Mtr_AttInc_Aa[1][1][adx] = Mtr_AttInc_Aa[1][1][adx] + di.4.HBWMTRA_I1
Mtr_AttInc_ja[2][1][jdx] = Mtr_AttInc_ja[2][1][jdx] + di.4.HBSMTRA_I1
Mtr_AttInc_Aa[2][1][adx] = Mtr_AttInc_Aa[2][1][adx] + di.4.HBSMTRA_I1
Mtr_AttInc_ja[3][1][jdx] = Mtr_AttInc_ja[3][1][jdx] + di.4.HBOMTRA_I1
Mtr_AttInc_Aa[3][1][adx] = Mtr_AttInc_Aa[3][1][adx] + di.4.HBOMTRA_I1

Mtr_AttInc_ja[1][2][jdx] = Mtr_AttInc_ja[1][2][jdx] + di.4.HBWMTRA_I2
Mtr_AttInc_Aa[1][2][adx] = Mtr_AttInc_Aa[1][2][adx] + di.4.HBWMTRA_I2
Mtr_AttInc_ja[2][2][jdx] = Mtr_AttInc_ja[2][2][jdx] + di.4.HBSMTRA_I2
Mtr_AttInc_Aa[2][2][adx] = Mtr_AttInc_Aa[2][2][adx] + di.4.HBSMTRA_I2
Mtr_AttInc_ja[3][2][jdx] = Mtr_AttInc_ja[3][2][jdx] + di.4.HBOMTRA_I2
Mtr_AttInc_Aa[3][2][adx] = Mtr_AttInc_Aa[3][2][adx] + di.4.HBOMTRA_I2

Mtr_AttInc_ja[1][3][jdx] = Mtr_AttInc_ja[1][3][jdx] + di.4.HBWMTRA_I3
Mtr_AttInc_Aa[1][3][adx] = Mtr_AttInc_Aa[1][3][adx] + di.4.HBWMTRA_I3
Mtr_AttInc_ja[2][3][jdx] = Mtr_AttInc_ja[2][3][jdx] + di.4.HBSMTRA_I3
Mtr_AttInc_Aa[2][3][adx] = Mtr_AttInc_Aa[2][3][adx] + di.4.HBSMTRA_I3
Mtr_AttInc_ja[3][3][jdx] = Mtr_AttInc_ja[3][3][jdx] + di.4.HBOMTRA_I3
Mtr_AttInc_Aa[3][3][adx] = Mtr_AttInc_Aa[3][3][adx] + di.4.HBOMTRA_I3

Mtr_AttInc_ja[1][4][jdx] = Mtr_AttInc_ja[1][4][jdx] + di.4.HBWMTRA_I4
Mtr_AttInc_Aa[1][4][adx] = Mtr_AttInc_Aa[1][4][adx] + di.4.HBWMTRA_I4
Mtr_AttInc_ja[2][4][jdx] = Mtr_AttInc_ja[2][4][jdx] + di.4.HBSMTRA_I4
Mtr_AttInc_Aa[2][4][adx] = Mtr_AttInc_Aa[2][4][adx] + di.4.HBSMTRA_I4
Mtr_AttInc_ja[3][4][jdx] = Mtr_AttInc_ja[3][4][jdx] + di.4.HBOMTRA_I4
Mtr_AttInc_Aa[3][4][adx] = Mtr_AttInc_Aa[3][4][adx] + di.4.HBOMTRA_I4

;; totals

TotHBWMtrAs = TotHBWMtrAs + di.4.HBW_MTR_As TotHBWNmtAs = TotHBWNmtAs +
di.4.HBW_NMT_As
TotHBSMtrAs = TotHBSMtrAs + di.4.HBS_MTR_As TotHBSNmtAs = TotHBSNmtAs +
di.4.HBS_NMT_As
TotHBOMtrAs = TotHBOMtrAs + di.4.HBO_MTR_As TotHBOntAs = TotHBOntAs +
di.4.HBO_NMT_As
TotNHWmtrAs = TotNHWmtrAs + di.4.NHW_MTR_As TotNHWnmtAs = TotNHWnmtAs +
di.4.NHW_NMT_As
TotNHOmtrAs = TotNHOmtrAs + di.4.NHO_MTR_As TotNHOntAs = TotNHOntAs +
di.4.NHO_NMT_As

TotMtrAs = TotMtrAs + di.4.HBW_MTR_As + di.4.HBS_MTR_As + di.4.HBO_MTR_As
+ di.4.NHW_MTR_As + di.4.NHO_MTR_As
TotNmtAs = TotNmtAs + di.4.HBW_NMT_As + di.4.HBS_NMT_As + di.4.HBO_NMT_As
+ di.4.NHW_NMT_As + di.4.NHO_NMT_As

TotHBWMtrAs_I1 = TotHBWMtrAs_I1 + di.4.HBWMTRA_I1
TotHBSMtrAs_I1 = TotHBSMtrAs_I1 + di.4.HBSMTRA_I1

```

## Appendix C Cube Voyager Scripts

```

TotHBOMtrAs_I1 = TotHBOMtrAs_I1 + di.4.HBOMTRA_I1

TotHBWMtrAs_I2 = TotHBWMtrAs_I2 + di.4.HBWMTRA_I2
TotHBSMtrAs_I2 = TotHBSMtrAs_I2 + di.4.HBSMTRA_I2
TotHBOMtrAs_I2 = TotHBOMtrAs_I2 + di.4.HBOMTRA_I2

TotHBWMtrAs_I3 = TotHBWMtrAs_I3 + di.4.HBWMTRA_I3
TotHBSMtrAs_I3 = TotHBSMtrAs_I3 + di.4.HBSMTRA_I3
TotHBOMtrAs_I3 = TotHBOMtrAs_I3 + di.4.HBOMTRA_I3

TotHBWMtrAs_I4 = TotHBWMtrAs_I4 + di.4.HBWMTRA_I4
TotHBSMtrAs_I4 = TotHBSMtrAs_I4 + di.4.HBSMTRA_I4
TotHBOMtrAs_I4 = TotHBOMtrAs_I4 + di.4.HBOMTRA_I4

ENDIF      ;; -----      End of Array accumulation      ---
-----
ENDLOOP

Loop Jdx = 1,25
  if (TotEMP_ja[jdx] > 0) EMPARate_pj[1][Jdx] = Mtr_Att_ja[1][jdx] /
TotEMP_ja[jdx]
  if (TotEMP_ja[jdx] > 0) EMPARate_pj[2][Jdx] = Mtr_Att_ja[2][jdx] /
TotEMP_ja[jdx]
  if (TotEMP_ja[jdx] > 0) EMPARate_pj[3][Jdx] = Mtr_Att_ja[3][jdx] /
TotEMP_ja[jdx]
  if (TotEMP_ja[jdx] > 0) EMPARate_pj[4][Jdx] = Mtr_Att_ja[4][jdx] /
TotEMP_ja[jdx]
  if (TotEMP_ja[jdx] > 0) EMPARate_pj[5][Jdx] = Mtr_Att_ja[5][jdx] /
TotEMP_ja[jdx]

  if (TotEMP_ja[jdx] > 0) EMPATARate_j[jdx] = MTR_ATot_Ja[jdx] /
TotEMP_ja[jdx]
ENDLOOP

  if (TotEMP_Tot > 0) EMPATARate_p[1] = TotHBWMtrAs / TotEMP_Tot
  if (TotEMP_Tot > 0) EMPATARate_p[2] = TotHBSMtrAs / TotEMP_Tot
  if (TotEMP_Tot > 0) EMPATARate_p[3] = TotHBOMtrAs / TotEMP_Tot
  if (TotEMP_Tot > 0) EMPATARate_p[4] = TotNHWmtrAs / TotEMP_Tot
  if (TotEMP_Tot > 0) EMPATARate_p[5] = TotNHOMtrAs / TotEMP_Tot

  if (TotEMP_Tot>0) TotRATESALL =
(TotHBWMtrAs+TotHBSMtrAs+TotHBOMtrAs+TotNHWmtrAs+TotNHOMtrAs) / TotEMP_Tot

; ===== Printout Trip Production Reports
=====
;
-----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Attractions by Purpose and
Jurisdiction ', '\n','\n'
Print PRINTO=1 LIST=' Jurisdiction ', ' HBW',' HBS','
HBO',' NHW',' NHO',' Total','\n'

Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
jdx = di.2.Jno

Print form=10csv PRINTO=1 LIST=
di.2.JNAME(c25), ' ',Mtr_Att_Ja[1][jdx], ' ',Mtr_Att_Ja[2][jdx], '
',Mtr_Att_Ja[3][jdx], ' ',Mtr_Att_Ja[4][jdx], ' ',
Mtr_Att_Ja[5][jdx], ' ',Mtr_Atot_Ja[jdx]
ENDLOOP

Print PRINTO=1 LIST='

```

```

Print form=10csv PRINTO=1 LIST=' TOTAL ',
TOTHBWMTRAs, ' ',TOTHBSMTRAs, ' ',TOTHBOMTRAs,
',TOTNHWMTRAs, ' ',TOTNHOMTRAs, ' ',TOTMTRAs

; -----
-----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Attractions per Job by Purpose and
Jurisdiction ', '\n','\n'
Print PRINTO=1 LIST=' Jurisdiction ', ' HBW',' HBS','
HBO',' NHW',' NHO',' Total','\n'

Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
jdx = di.2.Jno

Print form=10.2csv PRINTO=1 LIST=
di.2.JNAME(c25), ' ',EMPARate_pj[1][jdx], ' ',EMPARate_pj[2][jdx], '
',EMPARate_pj[3][jdx], ' ',EMPARate_pj[4][jdx], ' ',
EMPARate_pj[5][jdx], ' ',EMPTARate_j[jdx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10.2csv PRINTO=1 LIST=' TOTAL ',
EMPTARate_P[1], ' ',EMPTARate_P[2], '
',EMPTARate_P[3], ' ',EMPTARate_P[4], ' ',EMPTARate_P[5], ' ',TOTRatesall

; -----
-----
Print PRINTO=1 LIST= '\n','\n',' Motorized Trip Attractions by Purpose and Area
Type ', '\n','\n'
Print PRINTO=1 LIST=' Jurisdiction ', ' HBW',' HBS','
HBO',' NHW',' NHO',' Total','\n'

Loop KK = 1,6

Adx = kk

Print form=10csv PRINTO=1 LIST=' Area Type: ',Adx(5),
' ',Mtr_Att_Aa[1][adx], ' ',Mtr_Att_Aa[2][adx], '
',Mtr_Att_Aa[3][adx], ' ',Mtr_Att_Aa[4][adx], ' ',
Mtr_Att_Aa[5][adx], ' ',Mtr_Atot_Aa[adx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST=' TOTAL ',
TOTHBWMTRAs, ' ',TOTHBSMTRAs, ' ',TOTHBOMTRAs,
',TOTNHWMTRAs, ' ',TOTNHOMTRAs, ' ',TOTMTRAs

; -----
-----
Print PRINTO=1 LIST= '\n','\n',' NonMotorized Trip Attractions by Purpose and
Jurisdiction ', '\n','\n'
Print PRINTO=1 LIST=' Jurisdiction ', ' HBW',' HBS','
HBO',' NHW',' NHO',' Total','\n'

Loop KK = 1,dbi.2.numrecords
xx = DBIReadRecord(2,kk)
jdx = di.2.Jno

Print form=10csv PRINTO=1 LIST=
di.2.JNAME(c25), ' ',Nmt_Att_Ja[1][jdx], ' ',Nmt_Att_Ja[2][jdx], '
',Nmt_Att_Ja[3][jdx], ' ',Nmt_Att_Ja[4][jdx], ' ',
Nmt_Att_Ja[5][jdx], ' ',Nmt_Atot_Ja[jdx]
ENDLOOP

```

## Appendix C Cube Voyager Scripts

```

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          ',
          TOTHBWNMTAs, ' ',TOTHBNSMTAs,' ',TOTHBONMTAs,'
',TOTNHNWMTAs, ' ',TOTNHONMTAs,' ',TOTNMTAs

;; -----
Print PRINTO=1 LIST= '\n','\n',' NonMotorized Trip Attractions by Purpose and Area
Type          ', '\n','\n'
Print PRINTO=1 LIST='          Jurisdiction          ', '          HBW','          HBS','
HBO','          NHW','          NHO','          Total','\n'

Loop KK = 1,6

  Adx = kk

  Print form=10csv PRINTO=1 LIST='          Area Type:          ',Adx(5),
          ',Nmt_Att_Aa[1][Adx],', ' ',Nmt_Att_Aa[2][Adx],',
',Nmt_Att_Aa[3][Adx],', ' ',Nmt_Att_Aa[4][Adx],', ' ',
          Nmt_Att_Aa[5][Adx],', ' ',Nmt_Atot_Aa[Adx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          ',
          TOTHBWNMTAs, ' ',TOTHBNSMTAs,' ',TOTHBONMTAs,'
',TOTNHNWMTAs, ' ',TOTNHONMTAs,' ',TOTNMTAs

; -----
Print PRINTO=1 LIST= '\n','\n',' Home-Based Motorized Trip Attractions by Purpose,
Income, and Jurisdiction          ', '\n','\n'
Print PRINTO=1 LIST='          Jurisdiction          ', '          HBW_Inc1','          HBW_Inc2','
HBW_Inc3','          HBW_Inc4',
          '          HBS_Inc1','          HBS_Inc2','
          HBS_Inc3','          HBS_Inc4',
          '          HBO_Inc1','          HBO_Inc2','
          HBO_Inc3','          HBO_Inc4', '\n'

Loop KK = 1,dbi.2.numrecords
  xx = DBIReadRecord(2,kk)
  jdx = di.2.Jno

  Print form=10csv PRINTO=1 LIST=
  di.2.JNAME(c25),', ',Mtr_AttInc_ja[1][1][jdx],', ',Mtr_AttInc_ja[1][2][jdx],',
',Mtr_AttInc_ja[1][3][jdx],', ',Mtr_AttInc_ja[1][4][jdx],', ' ',
          Mtr_AttInc_ja[2][1][jdx],', ',Mtr_AttInc_ja[2][2][jdx],',
',Mtr_AttInc_ja[2][3][jdx],', ',Mtr_AttInc_ja[2][4][jdx],', ' ',
          Mtr_AttInc_ja[3][1][jdx],', ',Mtr_AttInc_ja[3][2][jdx],',
',Mtr_AttInc_ja[3][3][jdx],', ',Mtr_AttInc_ja[3][4][jdx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          ',
          TOTHBWMTRAs_i3, ' ',TOTHBWMTRAs_i4, ' ',
          TOTHBWMTRAs_i1, ' ',TOTHBWMTRAs_i2, '
',TOTHBSMTRAs_i3, ' ',TOTHBSMTRAs_i4, ' ',
          TOTHBSMTRAs_i1, ' ',TOTHBSMTRAs_i2, '
',TOTHBOMTRAs_i3, ' ',TOTHBOMTRAs_i4

ENDRUN
*copy voya*.prn Juris_Trip_Rate_summary.rpt

```

```

;; -----
Print PRINTO=1 LIST= '\n','\n',' Home-Based Motorized Trip Attractions by Purpose,
Income, and Area Type          ', '\n','\n'
Print PRINTO=1 LIST='          Area Type          ', '          HBW_Inc1','          HBW_Inc2','
HBW_Inc3','          HBW_Inc4',
          '          HBS_Inc1','          HBS_Inc2','
          HBS_Inc3','          HBS_Inc4',
          '          HBO_Inc1','          HBO_Inc2','
          HBO_Inc3','          HBO_Inc4', '\n'

Loop KK = 1,6

  Adx = kk

  Print form=10csv PRINTO=1 LIST='          Area Type:          ',Adx(5),
          ',Mtr_AttInc_Aa[1][1][Adx],', ' ',Mtr_AttInc_Aa[1][2][Adx],',
',Mtr_AttInc_Aa[1][3][Adx],', ' ',Mtr_AttInc_Aa[1][4][Adx],', ' ',
          Mtr_AttInc_Aa[2][1][Adx],', ',Mtr_AttInc_Aa[2][2][Adx],',
',Mtr_AttInc_Aa[2][3][Adx],', ' ',Mtr_AttInc_Aa[2][4][Adx],', ' ',
          Mtr_AttInc_Aa[3][1][Adx],', ',Mtr_AttInc_Aa[3][2][Adx],',
',Mtr_AttInc_Aa[3][3][Adx],', ' ',Mtr_AttInc_Aa[3][4][Adx]
ENDLOOP

Print PRINTO=1 LIST='
Print form=10csv PRINTO=1 LIST='          TOTAL          ',
          TOTHBWMTRAs_i1, ' ',TOTHBWMTRAs_i2, '
',TOTHBWMTRAs_i3, ' ',TOTHBWMTRAs_i4, ' ',
          TOTHBSMTRAs_i1, ' ',TOTHBSMTRAs_i2, '
',TOTHBSMTRAs_i3, ' ',TOTHBSMTRAs_i4, ' ',
          TOTHBOMTRAs_i1, ' ',TOTHBOMTRAs_i2, '
',TOTHBOMTRAs_i3, ' ',TOTHBOMTRAs_i4

ENDRUN
*copy voya*.prn Juris_Trip_Rate_summary.rpt

```

## 54 Truck\_Com\_Trip\_Generation.s

```

*del voya*.prn
;-----
; Truck_Com_Trip_Generation.s
; Version 2.3, 3722 TAZ System - Truck and Commercial Vehicle Trip Generation
Process
;
; RM
; Date:      12/08/10
;
;-----
;Parameters and file specifications:
;-----
ZONESIZE      = 3722                ; No. of TAZs
LastIZn       = 3675                ; Last Internal TAZ no.

JrCl          = 24                  ; No. of Juris. Classes (transformed
JURIS. Code 0-23 becomes 1-24)
ArCl          = 6                   ; No. of Area Classe (ATypes)
VeCl         = 3                    ; No. of Vehicle Classes (1/Medium Truck,
2/ Heavy Truck, 3, Comm. Vehicle

```

## Appendix C Cube Voyager Scripts

```

ZNFILE_IN1 = 'inputs\ZONE.dbf' ; Input Zonal Land Use File
ZNFILE_IN2 = 'AreaType_File.dbf' ; Input Zonal Area Type File
from network building
Ext_PsAs = 'inputs\Ext_PsAs.dbf' ; External Ps, As
ZoneConnect = '%_prev_%_skimtot.txt' ; Zone file showing TAZs
without Truck Access (generation is suppressed)

ZnFile_Ou1 = '%_iter_%_ComVeh_Truck_Ends.dbf' ; output comm, med trk, hvy
truck trip ends
ZnFile_Ou2 = '%_iter_%_ComVeh_Truck_dbg.dbf' ; output debug file- zonal
inputs and outputs

Rates_in = '..\support\Truck_Com_Trip_Rates.DBF' ; Truck, Comm.Veh trip rates
reportfile = '%_iter_%_Truck_Com_Trip_Generation.txt'

;=====
;Program Steps
;=====
RUN PGM=MATRIX
ZONES=1
ARRAY OFFRateA = 3,6 ; trip rates arrayed as 3 types (Med, Hvy, CV) by 6 area
types
ARRAY RETRateA = 3,6 ;
ARRAY INDRateA = 3,6 ;
ARRAY OTHRateA = 3,6 ;
ARRAY HH_RateA = 3,6 ;

ARRAY MHC_JurA =3,24 ; jurisdictional arrays 3 TYPES (Med, Hvy, CV) by juris. code
1 to 24 (0-23)
ARRAY MHC_AtpA =3,24 ; Area Type arrays 3 TYPES (Med, Hvy, CV) by Area Type
(1-6)

;=====
; Define Zonal Land activity as a zonal lookup table
FileI LOOKUPI[1] = "@ZNFILE_IN1@"
LOOKUP LOOKUPI=1, NAME=tazlu,
LOOKUP[1] = TAZ, RESULT=OFFEMP, ;
LOOKUP[2] = TAZ, RESULT=RETEMP, ;
LOOKUP[3] = TAZ, RESULT=INDEMP, ;
LOOKUP[4] = TAZ, RESULT=OTHEMP, ;
LOOKUP[5] = TAZ, RESULT=HH, ;
LOOKUP[6] = TAZ, RESULT=JURCODE, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

; Define Zonal Truck Access indicator (sum of truck time skims to/from each TAZ)
LOOKUP NAME=trkskims,
LOOKUP[1] = 1, RESULT=2, ; row sum of truck skims
LOOKUP[2] = 1, RESULT=3, ; col sum of truck skims
INTERPOLATE=N, FAIL= 10000000.0, 10000000.0, 10000000.0, LIST=N,file
=@zoneconnect@

; Define special truck generator TAZs - as defined in the original calibration work
; Lookup table to identify "truck zones" for 2005 (new TAZs)
LOOKUP NAME=tzone,
LOOKUP[1] = 2, RESULT=1, ; row sum of truck skims
interpolate = n, fail = 0,0,0,
R=
'1 213',
'1 218',
'1 519',
'1 520',
'1 527',
'1 531',
'1 864',

```

```

'1 865',
'1 870',
'1 1018',
'1 1021',
'1 1022',
'1 1031',
'1 1088',
'1 1119',
'1 1120',
'1 1230',
'1 1249',
'1 1511',
'1 1652',
'1 1800',
'1 1973',
'1 1983',
'1 1985',
'1 1987',
'1 1988',
'1 2014',
'1 2116',
'1 2321',
'1 2326',
'1 2327',
'1 2383',
'1 2386',
'1 2388',
'1 2527',
'1 2542',
'1 2547',
'1 2834',
'1 2835',
'1 2837',
'1 2838',
'1 2839',
'1 2840',
'1 2841',
'1 2842',
'1 2921',
'1 2922',
'1 2923',
'1 2930',
'1 2931',
'1 2937',
'1 2940',
'1 2943',
'1 2990',
'1 2992',
'1 2999',
'1 3002',
'1 3003',
'1 3004',
'1 3005',
'1 3036',
'1 3233',
'1 3234',
'1 3235',
'1 3236',
'1 3237',
'1 3238',
'1 3239',
'1 3245',
'1 3572',
'1 3573',
'1 3574',
'1 3575',
'1 3580',
'1 3585'

```

## Appendix C Cube Voyager Scripts

```

;;; end

; Define zonal Area Type File as a zonal lookup table
FileI LOOKUPI[2] = "@ZNFIL IN2@"
LOOKUP LOOKUPI=2, NAME=TAZat,
LOOKUP[1] = TAZ, RESULT=atype,
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

; Define External trip end file as a zonal lookup table
FileI LOOKUPI[3] = "@Ext PsAs@"
LOOKUP LOOKUPI=3, NAME=ExtTAZdat,
LOOKUP[1] = TAZ, RESULT=CV_XI,
LOOKUP[2] = TAZ, RESULT=MTK_XI,
LOOKUP[3] = TAZ, RESULT=HTK_XI,
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

; Read in Trip rates, fill in rate array
FILEI DBI[1] = "@Rates in@"
LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  count = dbi.1.recno
  OFFRateA[di.1.Vtype][di.1.ATYPE] = di.1.OFFRATE ;;VNAME VTYPE
  ATYPE OFFRATE RETRATE INDRATE OTHRATE HHRATE
  RETRateA[di.1.Vtype][di.1.ATYPE] = di.1.RETRATE ;;VNAME VTYPE
  ATYPE OFFRATE RETRATE INDRATE OTHRATE HHRATE
  INDRateA[di.1.Vtype][di.1.ATYPE] = di.1.INDRATE ;;VNAME VTYPE
  ATYPE OFFRATE RETRATE INDRATE OTHRATE HHRATE
  OTHRRateA[di.1.Vtype][di.1.ATYPE] = di.1.OTHRATE ;;VNAME VTYPE
  ATYPE OFFRATE RETRATE INDRATE OTHRATE HHRATE
  HH_RateA[di.1.Vtype][di.1.ATYPE] = di.1.HHRATE ;;VNAME VTYPE
  ATYPE OFFRATE RETRATE INDRATE OTHRATE HHRATE
ENDLOOP

;; define output DBF file name and variables
;; output trip file here:
FILEO RECO[1] = "@ZNFIL ou1@",
fields = TAZ(5), Comm_Veh(12.2), Med_Truck(12.2),
Hvy_Truck(12.2), ;; <-- All(Int/Ext) trip ends
IComm_Veh(12.2), IMed_Truck(12.2),
IHvy_Truck(12.2) ;; <-- Internal ONLY Trip ends

;; output debug file here (all zonal inputs and outputs):
FILEO RECO[2] = "@ZNFIL ou2@",
fields = TAZ(5), Atype(3.0),
Comm_Veh(8.0), Med_Truck(8.0), Hvy_Truck(8.0),
Off(8.0),Ret(8.0),Ind(8.0),Oth(8.0),HH(8.0),
COff_Rate(8.5), CRet_Rate(8.5), CInd_Rate(8.5), COth_Rate(8.5),
CHH_Rate(8.5), MOff_Rate(8.5), MRet_Rate(8.5), MInd_Rate(8.5), MOth_Rate(8.5),
MHH_Rate(8.5), HOff_Rate(8.5), HRet_Rate(8.5), HInd_Rate(8.5), HOth_Rate(8.5),
HHH_Rate(8.5),
tzfactm(8.2),tzfacth(8.2),supressed(4)

;; All done reading, now compute the trips and write out zonal results:
LOOP M= 1,@LastIZN@
  _ATYPE = TAZat(1,M) ;;CURRENT Area
  type _Jur = TAZlu(6,M) + 1.0 ;;CURRENT Jur
  index (=jurcode + 1, so 0-23 becomes 1-24)
  _Comm_Veh = TAZlu(1,M) * OFFRATEA[3][_ATYPE] + ;; compute
  commercial trips TAZlu(2,M) * RETRATEA[3][_ATYPE] +

```

```

TAZlu(3,M) * INDRATEA[3][_ATYPE] +
TAZlu(4,M) * OTHRATEA[3][_ATYPE] +
TAZlu(5,M) * HH_RATEA[3][_ATYPE]

_Med_Truck = TAZlu(1,M) * OFFRATEA[1][_ATYPE] + ;; compute
Medium Truck trips

TAZlu(2,M) * RETRATEA[1][_ATYPE] +
TAZlu(3,M) * INDRATEA[1][_ATYPE] +
TAZlu(4,M) * OTHRATEA[1][_ATYPE] +
TAZlu(5,M) * HH_RATEA[1][_ATYPE]

_Hvy_Truck = TAZlu(1,M) * OFFRATEA[2][_ATYPE] + ;; compute
Heavy truck trips

TAZlu(2,M) * RETRATEA[2][_ATYPE] +
TAZlu(3,M) * INDRATEA[2][_ATYPE] +
TAZlu(4,M) * OTHRATEA[2][_ATYPE] +
TAZlu(5,M) * HH_RATEA[2][_ATYPE]

; If zone is not truck-accessible, zero out all truck trips.
ro.supressed = 0.0
skimout = trkskims(1,M)
skimin = trkskims(2,M)

IF (SKIMOUT/@ZONESIZE@ > 2000.0 || SKIMIN/@ZONESIZE@ > 2000.0)
  _Med_Truck = 0
  _Hvy_Truck = 0
  ro.supressed = 1.0
ENDIF

; Incorporate truck zone adjustment factors

TZFACTM = 1.0
TZFACTH = 1.0
IF (TZONE(1,M) > 0.0)
  TZFACTM = 2.7
  TZFACTH = 5.3
ENDIF
_Med_Truck = _Med_Truck * TZFACTM
_Hvy_Truck = _Hvy_Truck * TZFACTH

ro.TAZ = M ; define
current zonal output vars
ro.ATYPE = _ATYPE ;
; ; com/trk trips will be written out along with extls
ro.Comm_Veh = _Comm_Veh ;
comm trips ro.Med_Truck = _Med_Truck ;
medtk trips ro.Hvy_Truck = _Hvy_Truck ;
hvytk trips ;;Internal com/trk trips will also be explicitly written for trip dist.
ro.IComm_Veh = _Comm_Veh ;
comm trips ro.IMed_Truck = _Med_Truck ;
medtk trips ro.IHvy_Truck = _Hvy_Truck ;
hvytk trips

ro.Off = TAZlu(1,M) ; land
activity ro.Ret = TAZlu(2,M) ;
ro.Ind = TAZlu(3,M) ;
ro.Oth = TAZlu(4,M) ;
ro.HH = TAZlu(5,M) ;

```

## Appendix C Cube Voyager Scripts

```

; CV trip rates
ro.COFF_Rate = OFFRATE[3][_ATYPE] ;
ro.CRET_Rate = RETRATE[3][_ATYPE] ;
ro.CIND_Rate = INDRATE[3][_ATYPE] ;
ro.COTH_Rate = OTHRATE[3][_ATYPE] ;
ro.CHH_Rate = HH_RATEA[3][_ATYPE] ;

ro.MOFF_Rate = OFFRATE[1][_ATYPE] ro.HOFF_Rate =
OFFRATE[2][_ATYPE] ; truck rates
ro.MRET_Rate = RETRATE[1][_ATYPE] ro.HRET_Rate =
RETRATE[2][_ATYPE] ;
ro.MIND_Rate = INDRATE[1][_ATYPE] ro.HIND_Rate =
INDRATE[2][_ATYPE] ;
ro.MOTH_Rate = OTHRATE[1][_ATYPE] ro.HOTH_Rate =
OTHRATE[2][_ATYPE] ;
ro.MHH_Rate = HH_RATEA[1][_ATYPE] ro.HHH_Rate =
HH_RATEA[2][_ATYPE] ;
ro.TZFACTM = TZFACTM
ro.TZFACTH = TZFACTH

WRITE RECO=1 ; write out
current record
WRITE RECO=2 ; write out
current record

;; accumulate Area type trip totals for reporting/checking
MHC_AtpA[1][_Atype] = MHC_AtpA[1][_Atype] + _Med_Truck
MHC_AtpA[2][_Atype] = MHC_AtpA[2][_Atype] + _Hvy_Truck
MHC_AtpA[3][_Atype] = MHC_AtpA[3][_Atype] + _Comm_Veh

;; accumulate juris trip totals for reporting/checking
MHC_JurA[1][_jur] = MHC_JurA[1][_jur] + _Med_Truck
MHC_JurA[2][_jur] = MHC_JurA[2][_jur] + _Hvy_Truck
MHC_JurA[3][_jur] = MHC_JurA[3][_jur] + _Comm_Veh

;; accumulate internal totals for reporting/checking
Tot_CVs = Tot_CVs + _Comm_Veh
Tot_MTs = Tot_MTs + _Med_Truck
Tot_HTs = Tot_HTs + _Hvy_Truck
Tot_OFF = Tot_OFF + TAZlu(1,M)
Tot_RET = Tot_RET + TAZlu(2,M)
Tot_IND = Tot_IND + TAZlu(3,M)
Tot_OTH = Tot_OTH + TAZlu(4,M)
Tot_HHs = Tot_HHs + TAZlu(5,M)

ENDLOOP

;; finally, write out external trips from extl file

; Read in External trip file:
firstExtl= @LastIzn@ + 1
LOOP K = firstExtl,@zonesize@

station) ro.TAZ = k ; TAZ (extl
ro.Comm_Veh = ExtTAZdat(1,k) ; comm trips
ro.Med_Truck = ExtTAZdat(2,k) ; medtk trips
ro.Hvy_Truck = ExtTAZdat(3,k) ; hvytk trips
;; Also write out null values for intl only trips to be used in trip
distribution

ro.IComm_Veh = 0.0 ; int comm trips
ro.IMed_Truck = 0.0 ; int medtk
trips

```

```

trips ro.IHvy_Truck = 0.0 ; int hvytk
write RECO = 1

;; accumulate total externals for reporting/checking
Tot_ExtCVs = Tot_ExtCVs + ExtTAZdat(1,k)
Tot_ExtMTs = Tot_ExtMTs + ExtTAZdat(2,k)
Tot_ExtHTs = Tot_ExtHTs + ExtTAZdat(3,k)

ENDLOOP

;; sum up total internals / externals for reporting/checking
Tot_IntExtCVs = Tot_ExtCVs + Tot_CVs
Tot_IntExtMTs = Tot_ExtMTs + Tot_MTs
Tot_IntExtHTs = Tot_ExtHTs + Tot_HTs

Total_Emp = Tot_Off + Tot_Ret + Tot_Ind + Tot_Oth
;; Print report and we're done

FILEO PRINTO[1] = "@Reportfile@"
PRINT PRINTO=1 form=12.0csv list = '
'
PRINT PRINTO=1 form=12.0csv list = ' Regional Total Truck and Commercial Trip-
Ends '
PRINT PRINTO=1 form=12.0csv list = ' Internal
External ALL '
PRINT PRINTO=1 form=12.0csv list = ' ----- --
PRINT PRINTO=1 form=12.0csv list = ' Commercial Vehicle Trips: ', Tot_CVs , ' ',
Tot_ExtCVs , ' ',Tot_IntExtCVs
PRINT PRINTO=1 form=12.0csv list = ' Medium Truck Trips : ', Tot_MTs , ' ',
Tot_ExtMTs , ' ',Tot_IntExtMTs
PRINT PRINTO=1 form=12.0csv list = ' Heavy Truck Trips : ', Tot_HTs , ' ',
Tot_ExtHTs , ' ',Tot_IntExtHTs

PRINT PRINTO=1 form=12.0csv list = '
'
PRINT PRINTO=1 form=12.0csv list = ' Land Activity Totals
'
PRINT PRINTO=1 form=12.0csv list = ' HHs : ', Tot_HHs
PRINT PRINTO=1 form=12.0csv list = ' Office Emp. : ', Tot_OFF
PRINT PRINTO=1 form=12.0csv list = ' Retail Emp. : ', Tot_Ret
PRINT PRINTO=1 form=12.0csv list = ' Industrial Emp. : ', Tot_Ind
PRINT PRINTO=1 form=12.0csv list = ' Other Emp. : ', Tot_Oth
PRINT PRINTO=1 form=12.0csv list = ' Total Emp. : ', Total_Emp

PRINT PRINTO=1 form=12.0csv list = '
'
PRINT PRINTO=1 form=12.0csv list = ' Truck and Comm. Veh. Internal Trip Totals by
Area Type '
PRINT PRINTO=1 form=12.0csv list = ' ATYPE Medium Trk Heavy Trk Comm. Veh.
'
PRINT PRINTO=1 form=12.0csv list = '-----
'
Loop K= 1,6
PRINT PRINTO=1 form=12.0csv list = K(8), MHC_AtpA[1][K], MHC_AtpA[2][K],
MHC_AtpA[3][K]
ENDLOOP
PRINT PRINTO=1 form=12.0csv list = ' Total ', Tot_MTs, Tot_HTs, Tot_CVs

PRINT PRINTO=1 form=12.0csv list = '
'
PRINT PRINTO=1 form=12.0csv list = ' Truck and Comm. Veh. Internal Trip Totals by
Jurisdiction '

```

## Appendix C Cube Voyager Scripts

```

PRINT PRINTO=1 form=12.0csv list = ' JurCode Medium Trk Heavy Trk Comm. Veh.
,
PRINT PRINTO=1 form=12.0csv list = '-----
Loop K= 1,24
  kk = k-1.0
  PRINT PRINTO=1 form=12.0csv list =  kK(8), MHC_JurA[1][K], MHC_JurA[2][K],
MHC_JurA[3][K]
ENDLOOP
  PRINT PRINTO=1 form=12.0csv list = ' Total ', Tot_MTs, Tot_HTs, Tot_Cvs

ENDRUN
*copy voya*.prn Truck_Com_Trip_Generation.rpt

```

## 55 unbuild\_net.s

```

*del ttpl*.prn
;-----
; unbuild_net.s
; Unbuilds a highway network (converts from TP+ binary to DBF format)
; Output files are in the format needed for the Version 2.3 travel model
;-----
pageheight=32767 ; Set the page height to a large value to minimize page breaks

basepath = 'V:\CGV2_3_70_Conformity2016_MD_ETL\Network_Development'
inhwy = 'zonehwy_2016Val_SecondRun.net'
out_link = 'Link.dbf'
out_node = 'Node.dbf'

run pgm = hwynet

neti = @basepath@\@inhwy@

/* Write out link file */

linko= @basepath@\@out_link@,
format=DBF,

include=a(5),b(5),distance(7.2),spdclass(7),capclass(7),jur(7),Screen(5),ftype(7),to
ll(9),tollgrp(5),

amlane(3),amlimit(3),pmlane(3),pmlimit(3),oplane(3),oplimit(3),edgeid(10),linkid(10)
,netyear(8),Shape_Leng(7.2),
projectid(10)

/* Write out node file */

nodeo= @basepath@\@out_node@,
format=DBF,
include=n(6),x(8),y(8)

endrun

*copy ttpl*.prn unbuild_net.rpt

```

## 56 V2.3\_Highway\_Build.s

```

*del voya*.prn
;=====
; HIGHWAY_BUILD_TOLL.S
;
; MWCOCG Version 2.3 Model - Highway Network Building Program
; Toll-DBF lookup file used
;=====
; 12/4/12 rm added 'timepen' variable to output network
; 3/13/13 timepen = 11 if screen=20 or screen = 36 zero otherwise (Bridge Ks
removed in set_factors.s)
; (These are the potomac river bridges from WWB to the Harpers Ferry
Bridge)
;=====

; PARAMETERS :
ZONESIZE = 3722 ; Max. TAZ No. (Param)
LSTITAZ = 3675 ; Last Internal Zone No. (Param)
FstHwyNode = 20000 ; First Highway node (Param)

; I/O Files :
NODEFILE = 'inputs\NODE.dbf' ; Node X/Y File (I/P file)
LINKFILE = 'inputs\LINK.dbf' ; Link File (I/P file)
ZONEFILE = 'inputs\ZONE.dbf' ; Zonal Land Use File (I/P file)
;AT_OVR = 'AREAOVER.ASC' ; Area Type Override file (I/P file)
ATYPFILE = 'AreaType_File.dbf' ; Zonal Area Type file (I/P file)

AMSPD = '..\support\AM_SPD_LKP.txt' ; AM Speed lookup ATxFT (I/P
file)
MDSPD = '..\support\MD_SPD_LKP.txt' ; Midday Speed lookup ATxFT (I/P
file)

TOLL_Esc = 'inputs\TOLL_Esc.dbf' ; INPUT Toll Escalation Param file
HWY_Defl = 'HWY_DeflatoR.txt' ; INPUT Default Highway Deflator (I/P
file)
LKTAZFILE = 'LinkTAZ.DBF' ; Nearest Taz to each link file(O/P file)
OU_BSNET = 'ZONEHWY.NET' ; OUTPUT BUILT network FILE

;-----
; Associate each link in the network to its nearest TAZ
RUN PGM=MATRIX
ZONES=1

FILEI DBI[1] = "@LINKFILE@" ; highway links
FILEO RECO[1] = "@LKTAZFILE@",fields = A(8),B(8),AB(15),TAZ(8) ; output a/b &
nearest TAZ

FileI LOOKUPI[1] = "@nodefile@"
LOOKUP LOOKUPI=1, NAME=nodexys,
LOOKUP[1] = N, RESULT=x, ;
LOOKUP[2] = N, RESULT=y, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

LOOP L= 1,dbi.1.NUMRECORDS
y=DBIReadRecord(1,L)
A = di.1.A
B = di.1.B

;
; The TAZ designated for the link is that with the minimum distance
; to either the A-node or the B-node
;

```

## Appendix C Cube Voyager Scripts

```

If      (A <= @ZONESIZE@)
  TAZ =A
elseif (B <= @ZONESIZE@)
  TAZ =B
else
  Ax =nodexys(1,A)
  Ay =nodexys(2,A)
  bx =nodexys(1,B)
  by =nodexys(2,B)
  TAZ= 0

  IF (AX > 0 && BX > 0)
    midx = (Ax+ Bx)/2.0
    midy = (Ay+ By)/2.0
    mindist = 9999999.
    TAZ=0
    loop tdx=1,@LstITAZ@
      CURDIST= SQRT( (midx - nodexys(1,tdx))**2 + (midy -
nodexys(2,tdx))**2 )/ 5280.
      if (curdist < mindist)
        mindist = curdist
        TAZ = TDX
      endif
    endloop
  endif
  Endif
  ;;Let's check this
  if (L= 1-10, 10000-10100,30000-30100)
    print form=10 list = A, B, TAZ, ';;; A XY: ',Ax,Ay,' B XY: ', Bx,By,'
MidXY: ', midx,midy, file= Link_Taz_Check.txt
endif

  ro.A   = A
  ro.B   = B
  ro.AB  = A*100000 + B
  ro.TAZ = TAZ
  WRITE RECO= 1
ENDLOOP
endrun

;
;=====
; Highway Building Part 1 - Develop Area type, Spdclass/CapClass Vars
;
;=====
RUN PGM = NETWORK
ZONES=@ZONESIZE@

; Node Coordinate File
; XY Units are NAD83 (in whole feet)
FILEI NODEI=@Nodefile@
; Node
; X Crd
; Y Crd

; Highway Links
FILEI LINKI=@LINKFILE@
; A-Node Number
; B-Node Number
; Distance in whole miles (xx.xx)
; Speed Class(optional)
; Capacity Class(optional)
; Observed AAWDT in 1000's
; Count Type 0,1,2,6,7
; Jurisdiction Code (0-23)
; Screenline Code (1-36)

```

```

; Facility Type Code (0-6)
; Current year Toll Value in cents
; Toll Group code (1-10)
; AM Peak Prd. No. of Lanes
; AM Peak Period Operation Code (0-9)
; PM Peak Prd. No. of Lanes
; PM Peak Period Operation Code (0-9)
; Off-Peak Prd. No. of Lanes
; Off-Peak Period Operation Code (0-9)
; EDGEID
; Project ID String
; Code

; Note:
; The Standard SPDCLASS(1-67), CAPCLASS(1-67), & TAZ defined below
;
NETO=TEMP.NET ; TEMPORARY NETWORK TO BE PASSED ONTO NEXT STEP

;-----
; Develop Link Area type/ Spdclass/ Capclass Attributes -
;-----

;
; Zonal Area Type Lookup (produced above)
;
FileI LOOKUPI[1] = "@atypfile@"
LOOKUP LOOKUPI=1, NAME=ZNAT,
LOOKUP[1] = TAZ, RESULT=AType, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[2] = "@lktazfile@"
LOOKUP LOOKUPI=2, NAME=lktaz,
LOOKUP[1] = ab, RESULT=TAZ, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

FileI LOOKUPI[3] = "@TOLL_ESC@"
LOOKUP LOOKUPI=3, NAME=Toll_Esc,
LOOKUP[1]= TOLLGrp, result=EscFAC, ;
LOOKUP[2]= TOLLGrp, result=DstFAC, ;
LOOKUP[3]= TOLLGrp, result=AM_TFtr, ;
LOOKUP[4]= TOLLGrp, result=PM_TFtr, ;
LOOKUP[5]= TOLLGrp, result=OP_TFtr, ;
LOOKUP[6]= TOLLGrp, result=AT_Min, ; x
LOOKUP[7]= TOLLGrp, result=AT_Max, ; x
LOOKUP[8]= TOLLGrp, result=TollType, ;
INTERPOLATE=N, FAIL= 0,0,0, LIST=N

_ABJoined = A*100000 + B

;; Ensure Centroids have lanes coded

IF (A<= 3722 || B <= 3722)
  SCREEN =0 ; Screenline Code (1-36)
  FTYPE =0 ; Facility Type Code (0-6)
  TOLL =0 ; Current year Toll Value in cents
  TOLLGRP =0 ; Toll Group code (1-10)
  AMLANE =7 ; AM Peak Prd. No. of Lanes
  AMLIMIT =0 ; AM Peak Period Operation Code (0-9)
  PMLANE =7 ; PM Peak Prd. No. of Lanes
  PMLIMIT =0 ; PM Peak Period Operation Code (0-9)
  OPLANE =7 ; Off-Peak Prd. No. of Lanes
  OPLIMIT =0 ; Off-Peak Period Operation Code (0-9)
ENDIF

TAZ = LKTAZ(1,_ABJOINED)
AType = ZNAT(1,TAZ) ; Area Type

```



## Appendix C Cube Voyager Scripts

```

;
;
; Here we will override the standard default Area Type code for any link with a
; TOLLGRP code - user's option
; area type override range (Min, Max)
; (via TG_ATOVR lookup table in the TOLL Group lookup file)

      _TG_ATMin = Toll_Esc(6,tollgrp)
      _TG_ATMax = Toll_Esc(7,tollgrp)
      _DefaultAT = AType

      IF (_TG_ATMin > 0 && _DefaultAT < _TG_ATMin) AType = _TG_ATMin
      IF (_TG_ATMax > 0 && _DefaultAT > _TG_ATMax) AType = _TG_ATMax

;
;
;;      IF (AType < 1 || AType > 7)
;;          print list= 'A: ',A(5),' B: ',B(5),' TAZ: ',TAZ: ',TAZ(3),' Area
Type: ', AType(3)
;;      ABORT
;;      ENDIf
;
;
; With the TAZ designated, now the speed/capacity class is defined as
; a two-digit code-- facility type & areatype
;
      SPDCLASS = FTYPE*10 + AType      ; Speed Class
      CAPCLASS = FTYPE*10 + AType      ; Capacity Class
;
;
; Check that TOLLGRP is coded for any link coded with a TOLL value-
; IF TOLLGRP is not coded with non-zero value, then give it a default
; value of '1.0'
;
      IF (TOLL > 0.0 && TOLLGRP = 0.0)
          TOLLGRP = 1.0
      ENDIF
;
;
; Set the Night (NT) and Midday (MD) lanes, limits equal to the Off-peak
; values read in on the link
      MDLANE = OPLANE
      MDLIMIT = OPLIMIT

      NTLANE = OPLANE
      NTLIMIT = OPLIMIT
;
;
ENDRUN

;=====
;
; Highway Building Part 2 - develop deflated highway tolls and
; pump prime speeds
;
;=====
;
RUN PGM = NETWORK

ZONES=@ZONESIZE@

NETI=TEMP.NET
; output network in TP+ format
NETO = zonehwy.net
;
; Compute AM, PM, Off-Peak Tolls
; The tolls are read in as undeflated, based on the coded TOLL value on the

```

```

; link and/or as a function of a distance based rate;
; The deflation is handled below. If the 'escfac' lookup (in the TOLL_Esc.dbf
file)
; is non-zero, then it is used to deflate. If it is zero, then the default
; highway deflator 'DEFLATION' (calculated in the SET_Factors.s script) is used.
; The recommended approach is to set the 'escfac' lookup array to zero and use
; HWY_Deflator
;
FileI LOOKUPI[1]= "@TOLL_ESC@"
LOOKUP LOOKUPI=1, NAME=Toll_Esc,
      LOOKUP[1]= TOLLGrp, result=EscFAC, ; x
      LOOKUP[2]= TOLLGrp, result=DstFAC, ; x
      LOOKUP[3]= TOLLGrp, result=AM_TFtr, ; x
      LOOKUP[4]= TOLLGrp, result=PM_TFtr, ; x
      LOOKUP[5]= TOLLGrp, result=OP_TFtr, ; x
      LOOKUP[6]= TOLLGrp, result=AT_Min, ;
      LOOKUP[7]= TOLLGrp, result=AT_Max, ;
      LOOKUP[8]= TOLLGrp, result=TollType, ; x
      INTERPOLATE=N, FAIL= 0,0,0, LIST=N

READ FILE=@HWY_Defl@

; deflated toll based on escfac:
AMTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(3,tollgrp)*Toll_Esc(1,tollgrp)
PMTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(4,tollgrp)*Toll_Esc(1,tollgrp)
OPTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(5,tollgrp)*Toll_Esc(1,tollgrp)

; if escfac set to zero then deflate based on HWY_Deflator:
IF (AMTOLL = 0)
      AMTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(3,tollgrp)*DEFLATIONFTR
ENDIF
IF (PMTOLL = 0)
      PMTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(4,tollgrp)*DEFLATIONFTR
ENDIF
IF (OPTOLL = 0)
      OPTOLL=(TOLL+(Toll_Esc(2,tollgrp)*DISTANCE))*Toll_Esc(5,tollgrp)*DEFLATIONFTR
ENDIF
; -----
; 1/25/08/ rm Changes made to develop special travel times/tolls for the MC
; program regarding variably priced facilities

      AMTOLL_VP = 0
      PMTOLL_VP = 0
      OPTOLL_VP = 0

; Check that coded tolls have a TOLLTYPE designation
; then define tolls on variably priced facilities ONLY
      TOLLTP = Toll_Esc(8,tollgrp) ;
      IF ((AMTOLL > 0 || PMTOLL > 0 || OPTOLL>0) && _TOLLTP = 0)
          LIST=' non-zero TOLL exists on a link has a zero TOLLTYPE code'
          abort
      ELSEIF (_TOLLTP = 2)
          AMTOLL_VP = AMTOLL
          PMTOLL_VP = PMTOLL
          OPTOLL_VP = OPTOLL
      ENDIF
; -----
;
; AM and Off-peak Initial Speed Lookup Tables...
;
; Use two lookups for AM/OP period by Facility type and Area type for now.
;
lookup name = amspd, ; AM Initial Speeds Atype x Ftype

```

## Appendix C Cube Voyager Scripts

```

lookup[1] = 1,result=2, ; AM CentConn Speeds (mph)
lookup[2] = 1,result=3, ; AM Freeway Speeds (mph)
lookup[3] = 1,result=4, ; AM Maj Art Speeds (mph)
lookup[4] = 1,result=5, ; AM Min Art Speeds (mph)
lookup[5] = 1,result=6, ; AM Collect Speeds (mph)
lookup[6] = 1,result=7, ; AM Exprway Speeds (mph)
lookup[7] = 1,result=8, ; AM Ramp Speeds (mph)
interpolate=N,fail=0,0,0,file=@AMSPD@

lookup name = opspd, ; Off-Pk Initial Speeds Atype x Ftype
lookup[1] = 1,result=2, ; Off-pk CentConn Speeds (mph)
lookup[2] = 1,result=3, ; Off-pk Freeway Speeds (mph)
lookup[3] = 1,result=4, ; Off-pk Maj Art Speeds (mph)
lookup[4] = 1,result=5, ; Off-pk Min Art Speeds (mph)
lookup[5] = 1,result=6, ; Off-pk Collect Speeds (mph)
lookup[6] = 1,result=7, ; Off-pk Exprway Speeds (mph)
lookup[7] = 1,result=8, ; Off-pk Ramp Speeds (mph)
interpolate=N,fail=0,0,0,file=@MDSFD@

_IDX = FTYPE + 1
PPAMSPD= AMSPD(_IDX,Atype)
PPOPSPD= OPSPD(_IDX,Atype)

;
; ESTABLISH AM/PM/MD/NT Highway Times (for the transit Network)
;
PPMSPD = PPAMSPD ; assume PM spd is equal to AM
IF (PPAMSPD != 0 )
AMHTIME = (DISTANCE/PPAMSPD)*60.00
PMHTIME = (DISTANCE/PPMSPD)*60.00
ELSE
AMHTIME = 0.01
PMHTIME = 0.01
ENDIF

IF (PPOPSPD != 0 )
OPHTIME = (DISTANCE/PPOPSPD)*60.00
ELSE
OPHTIME = 0.01
ENDIF

MDTOLL = OPTOLL
MDTOLL_VP = OPTOLL_VP
PFMDSPD = PPOPSPD
MDHTIME = OPHTIME

NTTOLL = OPTOLL
NTTOLL_VP = OPTOLL_VP
PNTSPD = PPOPSPD
NTHTIME = OPHTIME

; ** Create timepen variable here
; ; 11 minute perceived time penalty at the Potomac River
; ; from WVB to the Harpers Ferry Bridge
timepen =0.0
IF (screen = 20 || screen = 36)

timepen = 11.0

ENDIF

; ** end timepen variable section

; CREATE SOME FREQUENCY-CROSSTABS FOR CHECKING

```

```

_CNT= 1
_AMLANEMI= AMLANE*DISTANCE
_OPLANEMI= OPLANE*DISTANCE
_PMLANEMI= PMLANE*DISTANCE

CROSSTAB VAR=_AMLANEMI,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=ATYPE, RANGE=1-7-1, 1-7
CROSSTAB VAR=_OPLANEMI,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=ATYPE, RANGE=1-7-1, 1-7
CROSSTAB VAR=_PMLANEMI,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=ATYPE, RANGE=1-7-1, 1-7

CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=AMLANE, RANGE=1-7-1, 1-7
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=OPLANE, RANGE=1-7-1, 1-7
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=PMLANE, RANGE=1-7-1, 1-7

CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=AMLIMIT,RANGE=0-9-1,0-9
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=OPLIMIT,RANGE=0-9-1,0-9
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1,,1-7, COL=PMLIMIT,RANGE=0-9-1,0-9

;
;

ENDRUN

```

## 57 walkacc.s

```

*del voya*.prn
; ; Walkacc.s - walk access link development - based on walkacc.for from AECOM
; ; 5/10/11 - changed script so HBWV2A1.dbf is read in from the working SD, not the
; ; inputs SD
; ; 06/11/14 - Changed line 116 so that TAZ number from the input file is used (_TAZ)
; ; as
; ; opposed to using the sequence number as TAZ (K), as
; ; notified by AECOM - RJM
; ; 04/17/17 - Corrected the distance notes: "tenths" to "hundredths" - DQN
; ; Dimensions:
NodeSize = 60000 ; ; Highway node size
TAZSTASize = 7999 ; ; TAZ/Sta dimensions
ITAZSize = 3675 ; ; Internal TAZ dimensions
XLinkSize = 1000 ; ; Max. no. of user-defined Add/Del links

; ; Input Files:
NodeF = 'inputs\node.dbf'
AreaF = 'HBWV2A1.dbf' ; ; used to be in \inputs SD, now created below
and read as an input
XtraF = 'inputs\Xtrawalk.dbf'
LinkF = 'WalkAcc_Links.dbf'

; ; Output Files:
sidewalkF = 'sidewalk.asc'
walkaccF = 'walkacc.asc'
supportF = 'support.asc'
; ; -----
; ; =====
; ;
; ;
; ; Create area walk percentage files first (This script section was formerly
; ; Create_HBWV2A1.s) ; ;
; ; Read: 1) a short/long walk area file created by the GIS-based
; ; buffering procedures ; ;
; ; 2) a standard zonal land use filecreate two file
; ;
; ; ...and create two files to be copied to the \inputs
; ; subdirectory for the appropriate year ; ;

```

## Appendix C Cube Voyager Scripts

```

;;
;;          1) HEWV2A1.dbf - file used by the WALKACC.s  program
the generate walk-access transit links ;;
;;
;;          2) NLwalkPct.txt- file used by the PrefareV23.s program
to generate zonal walks pct's to      ;;
;;
;;          Metrorail for the MFARE2 process, and
a zonal file for the NL Mode Choice model ;;
=====
;;Input Files:
PctWalkF   = 'inputs\Areawalk.txt'   ; zonal walk percentage file from the GIS
process
ZoneF      = 'inputs\zone.dbf'       ; standard zonal attribute input file
;; Outputs

NL_Pct_wk  = 'NLwalkPCT.txt'

; Convert zonal walk area file to dbf
RUN PGM=MATRIX
ZONES=1
FILEI reci = @PctWalkF@ ,
          TAZ      =1,      ; TAZ
          TAZAREA  =2,      ; TAZ area (sq mi)
          MtrShort =3,      ; Area within short walk range to a Metro Station
          MtrLong  =4,      ; Area within long walk range to a Metro Station
(incl short walk)
          AmShort  =5,      ; Area within short walk range to AM Prd Transit of
any kind
          AmLong   =6,      ; Area within long walk range to AM Prd Transit of
any kind
          OPShort  =7,      ; Area within short walk range to Offpk Prd Transit
of any kind
          OPLong   =8,      ; Area within long walk range to Of pk Prd Transit
of any kind
          sort=TAZ        ;

IF (reci.RECNO>1) ; Skip first record which has no data, only variable names
n=n+1             ; n record counter

; write out TAZ level dbf file
FILEO RECO[1] ="AreaWlk.dbf", Fields = RECI.ALLFIELDS
WRITE RECO=1
ENDIF
endrun

=====
RUN PGM=MATRIX

ZONES=1
FileI DBI[1]  = "AreaWlk.dbf"
FILEO RECO[1] ="HBWV2A1.dbf", Fields = TAZ(8), Pctwksh(8), Pctwklg(8), Area(10.4)
FILEO PRINTO[1] ="@NL_PCT_Wk@"

LOOP K = 1,dbi.1.NUMRECORDS
x = DBIReadRecord(1,k)
  _TAZ      = di.1.TAZ      ;
  _Area     = di.1.TAZArea ;
  _AMShort  = di.1.AMShort ;
  _AMLong   = di.1.AMLong  ;
  _MtrShort = di.1.MtrShort ; Area within short walk range to a Metro
Station
  _MtrLong  = di.1.MtrLong ; Area within long walk range to a Metro
Station (incl short walk)

```

```

          _OPShort = di.1.OPShort ; Area within short walk range to Offpk
Prd Transit of any kind
          _OPLong  = di.1.OPLong ; Area within long walk range to Of pk
Prd Transit of any kind

;; evaluate pct walks for reasonability (for future checking)
;;IF (_TAZ < 1 || _TAZ > 3675) abort MSG=' TAZs < 1 or > 3675 , I quit)
;;IF (_Area < 0 ) abort MSG='Zonal area < zero , I quit)
;;IF (_AMShort > _Area ) abort MSG='AMShort area > Area , I quit)
;;IF (_AMLong > _Area ) abort MSG='AMLong area > Area , I quit)
;;IF (_AMShort < 0 ) abort MSG='AMShort area < zero , I quit)
;;IF (_AMLong < 0 ) abort MSG='AMLong area < zero , I quit)

_Pctwksh = MIN(( _AMShort/_Area * 100.00),100.00)
_Pctwksh_lg = MIN(( _AMLong /_Area * 100.00),100.00)
_Pctwklg = _Pctwksh_lg - _Pctwksh

_Proamwksh = MIN(( _AMShort/_Area ),1.00) ; proportion of TAZ
that is in AM short service area
_Proamwksh_lg = MIN(( _AMLong /_Area ),1.00) ; proportion of TAZ
that is in AM long service area
_Proopwksh = MIN(( _OPShort/_Area ),1.00) ; proportion of TAZ
that is in OP short service area
_Proopwksh_lg = MIN(( _OPLong /_Area ),1.00) ; proportion of TAZ
that is in OP long service area
_Prometwksh = MIN(( _MtrShort/_Area ),1.00) ; proportion of TAZ
that is in Metrorail short service area
_Prometwksh_lg = MIN(( _MtrLong /_Area ),1.00) ; proportion of TAZ
that is in Metrorail long service area

ro.TAZ = _TAZ
ro.Pctwksh = _Pctwksh
ro.Pctwklg = _Pctwklg
ro.Area = _Area
write reco=1
if (K=1)
print printo=1 list = ' TAZ MetSht MetShLg AMSht AMShLg OPSht
OPShLg '
endif
;;print printo=1 list = K(8), _Prometwksh(8.2),_Prometwksh_lg(8.2),
print printo=1 list = _TAZ(8),
_Prometwksh(8.2),_Prometwksh_lg(8.2),
_Proamwksh(8.2), _Proamwksh_lg(8.2),
_Proopwksh(8.2), _Proopwksh_lg(8.2)

ENDLOOP
ENDRUN

;;
;; Now begin walk access link process
=====
RUN PGM=MATRIX

ZONES=1
FileI DBI[1] = "@nodef@"
FILEI DBI[2] = "@Xtraf@"
FileI DBI[3] = "@areaf@"
FILEI DBI[4] = "@Linkf@"

FILEO PRINTO[1] =@sidewalkf@
FILEO PRINTO[2] =@walkaccf@
FILEO PRINTO[3] =@supportf@

```

## Appendix C Cube Voyager Scripts

```

;ARRAY Type=c1 AD = @Xlinksize@

ARRAY nx      = @nodesize@,
ny          = @nodesize@,
use        = @nodesize@,
Delal     = @Xlinksize@,
Delbl     = @Xlinksize@,
Dela2     = @Xlinksize@,
Delb2     = @Xlinksize@,
DeltAZ    = @TAZSTASize@,
Tazdist   = @TAZSTASize@,
Tazarea   = @TAZSTASize@,
Tazpctw   = @TAZSTASize@

; Fill node XY Array
Maxnode = 0.0

LOOP K = 1,dbi.1.NUMRECORDS
  x = DBIReadRecord(1,k)
  N      = di.1.N
  NX[N]  = di.1.X
  NY[N]  = di.1.Y
  IF (N > Maxnode) Maxnode = N
ENDLOOP

; Fill xtra node Array
LOOP K = 1,dbi.2.NUMRECORDS
  x = DBIReadRecord(2,k)
  AD      = di.2.AD
  AD_A    = di.2.AD_A
  AD_B    = di.2.AD_B

  if (AD_A <=@TAZSTASIZE@ || AD_B <=@TAZSTASIZE@)
    ip = 16
  else
    ip = 13
  endif

  if (AD = '-' && ip = 13)
    Ndel1 = Ndel1 + 1.0
    Delal[Ndel1] = AD_A
    Delbl[Ndel1] = AD_B
  endif

  if (AD = '-' && ip = 16)
    Ndel2 = Ndel2 + 1.0
    Dela2[Ndel2] = AD_A
    Delb2[Ndel2] = AD_B
    IF (AD_A <= @TAZSTASIZE@) DelTAZ[AD_A] = 1.0
    IF (AD_B <= @TAZSTASIZE@) DelTAZ[AD_B] = 1.0
  endif

  IF (AD = '+')
    Ndel2 = Ndel2 + 1.0
    Dela2[Ndel2] = AD_A
    Delb2[Ndel2] = AD_B
    Xdist = abs(NX[AD_A] - NX[AD_B])
    Ydist = abs(NY[AD_A] - NY[AD_B])
    Distft = ((Xdist*Xdist) + (Ydist*Ydist))*0.50
    Dist = Round(Distft/52.80) ; distance in hundredths of miles

    IF (IP = 13)
      Print PRINTO=1 list = 'SUPPORT N=',AD_A(6),'-',AD_B(6),
        ' MODE=13 SPEED=3 ONEWAY=Y DIST = ',
DIST(6)

```

```

      Print PRINTO=3 list = 'SUPPLINK N=',AD_A(6),'-',AD_B(6),
        ' MODE=13 SPEED=3 ONEWAY=Y DIST = ',
DIST(6)
    ELSE
      Print PRINTO=2 list = 'SUPPORT N=',AD_A(6),'-',AD_B(6),
        ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)

      Print PRINTO=3 list = 'SUPPLINK N=',AD_A(6),'-',AD_B(6),
        ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)
    ENDIF
  ENDIF
ENDLOOP
;;

LOOP K = 1,dbi.3.NUMRECORDS
  x = DBIReadRecord(3,k)
  TAZ      = di.3.TAZ
  Pctwksh  = di.3.pctwksh
  Pctwklg  = di.3.pctwklg
  area     = di.3.area

  Tazarea[TAZ] = area
  Tazdist[TAZ] = min(100.0, ( 75*((area)**0.5) ) )
  Tazpctw[TAZ] = Pctwksh + Pctwklg
  print list = TAZ, Pctwksh,Pctwklg,TAZarea[TAZ],Tazdist[TAZ],Tazpctw[TAZ],
file = zonal.asc
ENDLOOP

LOOP K = 1,dbi.4.NUMRECORDS
  x = DBIReadRecord(4,k)
  A      = di.4.A
  B      = di.4.B
  hdist  = di.4.distance
  htaz   = di.4.TAZ
  ftype  = di.4.ftype

  LOOP L=1, Ndel1
    IF (A = Delal[L] && B = Delbl[L]) GOTO SKIP
    IF (B = Delal[L] && A = Delbl[L]) GOTO SKIP
  ENDLOOP

  IF (ftype <= 1 || ftype=5 || ftype =6 ) GOTO SKIP
  IF (TAZPctw[hTAZ] = 0.0 ) GOTO SKIP

  USE[A] = 1.0
  USE[B] = 1.0

  tdist = Round(hdist*100.00)
  Print PRINTO=1 list = 'SUPPORT N=',A(6),'-',B(6),
    ' MODE=13 SPEED=3 ONEWAY=Y DIST = ', TDIST(6)

  Print PRINTO=3 list = 'SUPPLINK N=',A(6),'-',B(6),
    ' MODE=13 SPEED=3 ONEWAY=Y DIST = ', TDIST(6)

;SKIP
ENDLOOP

;; END of Sidewalk Links ;;

;; debug
LOOP NN = NNode,Maxnode
  print list = NN, USE[NN] ,file= uselist.asc
ENDLOOP
;;

```

## Appendix C Cube Voyager Scripts

```

;; BEGIN zonal access link development
;-----
; Find all access links within the 1.00 mile search radius:
;-----
LOOP ZZ = 1,@ITAZSize@
  Find = 0
  If (TAZPctw[ZZ] = 0.0) GOTO NextTAZ
  If (NX[ZZ] = 0.0) GOTO NextTAZ

  NNode = @ITAZSize@ + 1.0
  LOOP NN = NNode,Maxnode
    IF (DelTAZ[ZZ] = 0.0) GOTO SkipDLst
    LOOP M=1, Ndel1
      IF (ZZ = Dela2[M] && NN = Delb2[M]) GOTO NextNode
      IF (NN = Dela2[M] && ZZ = Delb2[M]) GOTO NextNode
    ENDLOOP
  :SkipDLst

  IF (Use[NN] = 0) GOTO NextNode
  IF (NX[NN] = 0) GOTO NextNode

  Xdist = abs(NX[zz] - NX[nn])
  Ydist = abs(NY[zz] - NY[nn])
  search = 5280.0
  IF (Xdist > search) GOTO NextNode
  IF (Ydist > search) GOTO NextNode
  IF (Xdist = 0.0 && YDist = 0.0) GOTO NextNode

  Distft = ((Xdist*Xdist) + (Ydist*Ydist))*0.50
  Dist = (Distft/52.80) ; distance in hundredths of miles
  ;;--
  ; IF (ZZ=190) ;; debug section
  ; print list = ' zz', ' NN',' TAZarea',' TAZPctw',' XDist','
  YDist ',' Search ',' Distft ',' Dist',' TAZdist', file = dud.asc
  ; print form=8.2 list = zz, NN,TAZarea[ZZ](8.4), TAZPctw[ZZ],XDist, YDist,
  search, Distft, Dist,TAZdist[ZZ], file = dud.asc
  ; endif
  ;;--
  IF (Dist > TAZdist[ZZ]) GOTO NextNode

  Print PRINTO=2 list = 'SUPPORT N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6), ';; search =
',search

  Print PRINTO=3 list = 'SUPPLINK N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)

  Find = Find + 1.0

:NextNode
ENDLOOP

;-----
; Expand search radius to 1.25 * TAZDist if no access links found thus far
;-----
IF (Find > 0) GOTO NEXTTAZ

LOOP NN = NNode,Maxnode

  LOOP M=1, Ndel1
    IF (ZZ = Dela2[M] && NN = Delb2[M]) GOTO NextNode1
    IF (NN = Dela2[M] && ZZ = Delb2[M]) GOTO NextNode1
  ENDLOOP

  IF (Use[NN] = 0) GOTO NextNode2
  IF (NX[NN] = 0) GOTO NextNode2

  Xdist = abs(NX[zz] - NX[nn])
  Ydist = abs(NY[zz] - NY[nn])
  search = 1.25 * 52.80 * TAZdist[ZZ]
  IF (Xdist > search) GOTO NextNode2
  IF (Ydist > search) GOTO NextNode2
  IF (Xdist = 0.0 && YDist = 0.0) GOTO NextNode2

  Distft = ((Xdist*Xdist) + (Ydist*Ydist))*0.50
  Dist = (Distft/52.80) ; distance in hundredths of miles
  IF (Dist > 100.0) GOTO NextNode2

  Print PRINTO=2 list = 'SUPPORT N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6), ';; search =
',search

  Print PRINTO=3 list = 'SUPPLINK N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)

  Find = Find + 1.0

```

```

ENDLOOP

IF (Use[NN] = 0) GOTO NextNode1
IF (NX[NN] = 0) GOTO NextNode1

Xdist = abs(NX[zz] - NX[nn])
Ydist = abs(NY[zz] - NY[nn])
search = 1.25 * 52.80 * TAZdist[ZZ]
IF (Xdist > search) GOTO NextNode1
IF (Ydist > search) GOTO NextNode1
IF (Xdist = 0.0 && YDist = 0.0) GOTO NextNode1

Distft = ((Xdist*Xdist) + (Ydist*Ydist))*0.50
Dist = (Distft/52.80) ; distance in hundredths of miles

IF (Dist > 100.0) GOTO NextNode1

Print PRINTO=2 list = 'SUPPORT N=',ZZ(6),'-',NN(6),
  ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6), ';; search =
',search

Print PRINTO=3 list = 'SUPPLINK N=',ZZ(6),'-',NN(6),
  ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)

Find = Find + 1.0

:NextNode1
ENDLOOP

;-----
; Expand search radius to 1.50 * TAZDist if no access links found thus far
;-----
IF (Find > 0) GOTO NEXTTAZ

LOOP NN = NNode,Maxnode

  LOOP M=1, Ndel1
    IF (ZZ = Dela2[M] && NN = Delb2[M]) GOTO NextNode2
    IF (NN = Dela2[M] && ZZ = Delb2[M]) GOTO NextNode2
  ENDLOOP

  IF (Use[NN] = 0) GOTO NextNode2
  IF (NX[NN] = 0) GOTO NextNode2

  Xdist = abs(NX[zz] - NX[nn])
  Ydist = abs(NY[zz] - NY[nn])
  search = 1.50 * 52.80 * TAZdist[ZZ]
  IF (Xdist > search) GOTO NextNode2
  IF (Ydist > search) GOTO NextNode2
  IF (Xdist = 0.0 && YDist = 0.0) GOTO NextNode2

  Distft = ((Xdist*Xdist) + (Ydist*Ydist))*0.50
  Dist = (Distft/52.80) ; distance in hundredths of miles
  IF (Dist > 100.0) GOTO NextNode2

  Print PRINTO=2 list = 'SUPPORT N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6), ';; search =
',search

  Print PRINTO=3 list = 'SUPPLINK N=',ZZ(6),'-',NN(6),
    ' ONEWAY=N MODE=16 SPEED= 3 DIST= ', DIST(6)

  Find = Find + 1.0

```

## Appendix C Cube Voyager Scripts

---

```
:NextNode2  
ENDLOOP
```

```
:NextTAZ  
ENDLOOP  
ENDRUN  
*copy voya*.prn WalkAcc.rpt
```

---

## Appendix D. AEMS and LINESUM Control files

1	hbo_nl_mc.ctl .....	D-1
2	hbs_nl_mc.ctl.....	D-21
3	hbw_nl_mc.ctl.....	D-42
4	lineSum_MR_access.ctl .....	D-62
5	lineSum_MR_line.ctl .....	D-63
6	LineSum_Volume.ctl .....	D-63
7	nho_nl_mc.ctl .....	D-64
8	nhw_nl_mc.ctl.....	D-83
9	station_names.dbf .....	D-103





Appendix D: AEMS Fortran Control Files

# 1 hbo\_nl\_mc.ctf

```

HBO OP NESTED LOGIT MC - #DATE: 9/17/2011 #VER: 21
CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
*
*
*LOGIT COEFFICIENTS BY CHOICE FOR EACH SKIM (NO INPUT SKIM IS
*EQUIVALENT TO A CONSTANT)
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
COEF01:IVTT      1>-0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322  -0.02322
SKIM01:IVTT      1>DAIV      S2IV      S3IV      WCIV      WBIV      WTIV      WMIV      PCIV      KCIV      PBIV      KBIV      PTIV      KTIV      PMIV      KMIV
COEF02:AUTO ACC  1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
COEF02:AUTO ACC  1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
SKIM02:AUTO ACC  1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
COEF03:TERM/OVTT 1>-0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805  -0.05805
SKIM03:TERM/OVTT 1>DATE      S2TE      S3TE      WCOV      WBOV      WTOV      WMOV      PCOV      KCOV      PBOV      KBOV      PTOV      KTOV      PMOV      KMOV
* LIMIT COEF 04 TO PURPOSE 1
COEF PURP04      >1
COEF04:COST INC1 1>-0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202  -0.00202
SKIM04:COST INC1 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 05 TO PURPOSE 2
COEF PURP05      >2
COEF05:COST INC2 1>-0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101  -0.00101
SKIM05:COST INC2 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 06 TO PURPOSE 3
COEF PURP06      >3
COEF06:COST INC3 1>-0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067  -0.00067
SKIM06:COST INC3 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF PURP07      >4
* LIMIT COEF 07 TO PURPOSE 4
COEF07:COST INC4 1>-0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051  -0.00051
SKIM07:COST INC4 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF08:TRN XFERS 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
SKIM08:TRN XFERS 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
COEF09:TRN BRDPEN 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
SKIM09:TRN BRDPEN 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
*WALK WEIGHT
COEF10:TRN WLKWT 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
SKIM10:TRN WLKWT 1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>          1>
*SYNTAX TO LIMIT UTILITY ELEMENT TO A PARTICULAR WALK SEGMENT IN THIS EXAMPLE
* COEF 18 APPLIES ONLY TO WALK SEGMENT 1
*COEF WLKSEG18   >1
*
* ASSUMED MATRIX ORGANIZATION
* FILE 1 TRIP TABLE (SEPARATE FOR EACH PURPOSE)
* 1 INCOME 1 (HOME-BASED)/ALL NHB TRIPS
* 2 INCOME 2 (HOME-BASED)
* 3 INCOME 3 (HOME-BASED)
* 4 INCOME 4 (HOME-BASED)
*
* FILE 2 HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 SOV TIME (MIN)
* 2 SOV DIST (0.1 MILES)
* 3 SOV TOLL (2007 CENTS)
* 4 HOV2 TIME (MIN)
* 5 HOV2 DIST (0.1 MILES)
* 6 HOV2 TOLL (2007 CENTS)
* 7 HOV3+ TIME (MIN)
* 8 HOV3+ DIST (0.1 MILES)
* 9 HOV3+ TOLL (2007 CENTS)
*
* FILE 3=COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 4=BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 5=METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)

```

## Appendix D: AEMS Fortran Control Files

```
* FILE 6=BUS+METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)
* 2 WLK OTHER (.01 MIN) 16 PNR OTHER (.01 MIN) 34 KNR OTHER (.01 MIN)
* 3 WLK IWAIT (.01 MIN) 17 PNR IWAIT (.01 MIN) 35 KNR IWAIT (.01 MIN)
* 4 WLK XWAIT (.01 MIN) 18 PNR XWAIT (.01 MIN) 36 KNR XWAIT (.01 MIN)
* 5 WLK IVTT TOT (.01 MIN) 19 PNR IVTT TOT (.01 MIN) 37 KNR IVTT TOT (.01 MIN)
* 6 WLK IVTT CR (.01 MIN) 20 PNR IVTT CR (.01 MIN) 38 KNR IVTT CR (.01 MIN)
* 7 WLK IVTT XB (.01 MIN) 21 PNR IVTT XB (.01 MIN) 39 KNR IVTT XB (.01 MIN)
* 8 WLK IVTT MR (.01 MIN) 22 PNR IVTT MR (.01 MIN) 40 KNR IVTT MR (.01 MIN)
* 9 WLK IVTT NM (.01 MIN) 23 PNR IVTT NM (.01 MIN) 41 KNR IVTT NM (.01 MIN)
* 10 WLK IVTT NM2 (.01 MIN) 24 PNR IVTT NM2 (.01 MIN) 42 KNR IVTT NM2 (.01 MIN)
* 11 WLK IVTT LB (.01 MIN) 25 PNR IVTT LB (.01 MIN) 43 KNR IVTT LB (.01 MIN)
* 12 WLK #XFERS (NUMBER ) 26 PNR #XFERS (NUMBER ) 44 KNR #XFERS (NUMBER )
* 13 WLK COST (.07CENTS) 27 PNR COST (.07CENTS) 45 KNR COST (.07CENTS)
* 14 WLK XPEN (.01 MIN) 28 PNR XPEN (.01 MIN) 46 KNR XPEN (.01 MIN)
* 29 PNR ACC TIME(.01 MIN) 47 KNR ACC TIME(.01 MIN)
* 30 PNR ACC DIST(.01 MIL) 48 KNR ACC DIST(.01 MIL)
* 31 PNR ACC COST(07CENTS)
* 32 PNR STA TERM(.01 MIN)
*
* FILE 8=ZDATA
* 1 HBW PARK COST (2007 CENTS)
* 2 HBS PARK COST (2007 CENTS)
* 3 HBO PARK COST (2007 CENTS)
* 4 NHB PARK COST (2007 CENTS)
* 5 TERMINAL TIME (HOME BASED) (MINUTES)
* 6 TERMINAL TIME (NON HOME BASED) (MINUTES)
* 7 ARC VIEW SHORT WALK PERCENT TO METRO
* 8 ARC VIEW LONG WALK PERCENT TO METRO
* 9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
* 10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT
* 11 ARC VIEW SHORT WALK PERCENT TO ALL OP TRANSIT
* 12 ARC VIEW LONG WALK PERCENT TO ALL OP TRANSIT
* 13 AREA TYPE
* 1=DC CORE
* 2=VA CORE
* 3=DC URBAN
* 4=MD URBAN
* 5=VA URBAN
* 6=MD OTHER
* 7=VA OTHER
*
* PARAMETERS
*=====
* AUTO OPERATING COSTS IN CENTS/mile
* COMPUTE AUOP >10
* AUTO OCCUPANCY FOR 3+ Reduced from 3.5 to 3.35 on 3/1/07 rm
* COMPUTE OCC3 >3.35
*
* TERMINAL TIMES, USE i/j805 FOR HBW, HBS, AND HBO. USE i/j806 FOR NHB
* HBW/HBS/HBO
* COMPUTE TERI >i805
* COMPUTE TERJ >j805
* NHB
* COMPUTE TERI >i806
* COMPUTE TERJ >j806
*
* PARK COSTS, USE i/j801 802 803 804 FOR HBW, HBS, HBO, NHB RESPECTIVELY
* HBW
* COMPUTE PRKC >j801/2.
* HBS
* COMPUTE PRKC >j802/2.
* HBO
* COMPUTE PRKC >j803/2.
* NHB
* COMPUTE PRKC >j804
*
* Percent of productions in long-walk area that are assumed to walk = 25% (i.e., 75% drive)
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE PCLM      >0.25
COMPUTE PCLT      >0.25
* PERCENT WALKS-METRO RAIL ONLY
COMPUTE PCMI      >(i807+PCLM*(i808-i807))/100.
COMPUTE PCMJ      >(j807+PCLM*(j808-j807))/100.
* PERCENT WALKS-PEAK
*COMPUTE PCTI     >(i809+PCLT*(i810-i809))/100.
*COMPUTE PCTJ     >(j809+PCLT*(j810-j809))/100.
* PERCENT WALKS-OFFPEAK
COMPUTE PCTI     >(i811+PCLT*(i812-i811))/100.
COMPUTE PCTJ     >(j811+PCLT*(j812-j811))/100.
COMPUTE PCMI     >MAX(PCMI,0)
COMPUTE PCMI     >MIN(PCMI,1)
COMPUTE PCMJ     >MAX(PCMJ,0)
COMPUTE PCMJ     >MIN(PCMJ,1)
COMPUTE PCTI     >MAX(PCTI,PCMI)
COMPUTE PCTI     >MIN(PCTI,1)
COMPUTE PCTJ     >MAX(PCTJ,PCMJ)
COMPUTE PCTJ     >MIN(PCTJ,1)
*
* DO TRIP SUBDIVISIONS
*
* HOME BASED ALTERNATIVES
COMPUTE TRP1     >m101
COMPUTE TRP2     >m102
COMPUTE TRP3     >m103
COMPUTE TRP4     >m104
* NON-HOME BASED
*COMPUTE TRP1     >0.25*m101
*COMPUTE TRP2     >0.25*m101
*COMPUTE TRP3     >0.25*m101
*COMPUTE TRP4     >0.25*m101
*
* BE SURE TO UPDATE THE IVTT COEFFICIENT IN FTA SECTION FOR EACH PURPOSE
*
*=====
*INITIALIZING ALL VARIABLES WITHIN IF STATEMENTS TO ZERO
COMPUTE DAIV     >0
COMPUTE DACS     >0
COMPUTE DATE     >0
COMPUTE S2IV     >0
COMPUTE S2CS     >0
COMPUTE S2TE     >0
COMPUTE S3IV     >0
COMPUTE S3CS     >0
COMPUTE S3TE     >0
COMPUTE WKIV     >0
COMPUTE WKOV     >0
COMPUTE WKXF     >0
COMPUTE WKCS     >0
COMPUTE WKXP     >0
COMPUTE WBIV     >0
COMPUTE WBOV     >0
COMPUTE WBXF     >0
COMPUTE WBCS     >0
COMPUTE WBXP     >0
COMPUTE WTIV     >0
COMPUTE WTOV     >0
COMPUTE WTXF     >0
COMPUTE WTCS     >0
COMPUTE WTXP     >0
COMPUTE WMIV     >0
COMPUTE WMOV     >0
COMPUTE WMXF     >0
COMPUTE WMCS     >0
COMPUTE WMXP     >0
COMPUTE PCIV     >0
COMPUTE PCAA     >0
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE PCOV      >0
COMPUTE PCXF      >0
COMPUTE PCCS      >0
COMPUTE PCXP      >0
COMPUTE PBIV      >0
COMPUTE PBAA      >0
COMPUTE PBOV      >0
COMPUTE PBXF      >0
COMPUTE PBCS      >0
COMPUTE PBXP      >0
COMPUTE PTIV      >0
COMPUTE PTAA      >0
COMPUTE PTOV      >0
COMPUTE PTXF      >0
COMPUTE PTCS      >0
COMPUTE PTXP      >0
COMPUTE PMIV      >0
COMPUTE PMAA      >0
COMPUTE PMOV      >0
COMPUTE PMXF      >0
COMPUTE PMCS      >0
COMPUTE PMXP      >0
COMPUTE KCIV      >0
COMPUTE KCAA      >0
COMPUTE KCOV      >0
COMPUTE KCXF      >0
COMPUTE KCCS      >0
COMPUTE KCXP      >0
COMPUTE KBIV      >0
COMPUTE KBAA      >0
COMPUTE KBOV      >0
COMPUTE KBXF      >0
COMPUTE KBCS      >0
COMPUTE KBXP      >0
COMPUTE KTIV      >0
COMPUTE KTA      >0
COMPUTE KTOV      >0
COMPUTE KTXF      >0
COMPUTE KTCS      >0
COMPUTE KTXP      >0
COMPUTE KMIV      >0
COMPUTE KMAA      >0
COMPUTE KMOV      >0
COMPUTE KMXF      >0
COMPUTE KMCS      >0
COMPUTE KMXP      >0

COMPUTE WCWK      >0
COMPUTE WBWK      >0
COMPUTE WTWK      >0
COMPUTE WMWK      >0
COMPUTE PCWK      >0
COMPUTE KCWK      >0
COMPUTE PBWK      >0
COMPUTE KBWK      >0
COMPUTE PTWK      >0
COMPUTE KTWK      >0
COMPUTE PMWK      >0
COMPUTE KMWK      >0

* SKIM VALUES, Divide distances by 10 to convert tenths of miles to whole miles
* DRIVE ALONE
COMPUTE           >IF (m201>0)
COMPUTE DAIV      >m201
COMPUTE DACS      >m202/10*AUOP+m203+PRKC
COMPUTE DATE      >TERI+TERJ
COMPUTE           >ENDIF
```

## Appendix D: AEMS Fortran Control Files

---

```
* SHARED RIDE 2
COMPUTE >IF (m204>0)
COMPUTE S2IV >m204
COMPUTE S2CS >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 3
COMPUTE >IF (m207>0)
COMPUTE S3IV >m207
COMPUTE S3CS >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE >TERI+TERJ
COMPUTE >ENDIF

* Assign Intrazonal trips to Autos (mj11/04/05)
COMPUTE >IF (P()==Q())
COMPUTE DAIV >1
COMPUTE DACS >m202/10*AUOP+m203+PRKC
COMPUTE DATE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 2
COMPUTE >IF (P()==Q())
COMPUTE S2IV >1
COMPUTE S2CS >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 3
COMPUTE >IF (P()==Q())
COMPUTE S3IV >1
COMPUTE S3CS >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE >TERI+TERJ
COMPUTE >ENDIF

*End of Intrazonal trips

* WALK COMMUTER RAIL
COMPUTE >IF (m305>0)
COMPUTE WCIV >m305/100.
COMPUTE WCOV >(m303+m304)/100.
COMPUTE WCXF >m312
COMPUTE WCCS >m313
COMPUTE WCXP >m314/100.
COMPUTE WCWK >(m301+m302)/100.
COMPUTE >ENDIF

* WALK BUS
COMPUTE >IF (m405>0)
COMPUTE WBIV >m405/100.
COMPUTE WBOV >(m403+m404)/100.
COMPUTE WBXF >m412
COMPUTE WBCS >m413
COMPUTE WBXP >m414/100.
COMPUTE WBWK >(m401+m402)/100.
COMPUTE >ENDIF

* WALK BUS/METRORAIL (TRANSIT)
COMPUTE >IF (m605>0)
COMPUTE WTIV >m605/100.
COMPUTE WTOV >(m603+m604)/100.
COMPUTE WTXF >m612
COMPUTE WTCS >m613
COMPUTE WTXP >m614/100.
COMPUTE WTWK >(m601+m602)/100.
COMPUTE >ENDIF
```

## Appendix D: AEMS Fortran Control Files

---

```
* WALK METRORAIL
COMPUTE >IF (m505>0)
COMPUTE WMIV >m505/100.
COMPUTE WMOV >(m503+m504)/100.
COMPUTE WMXF >m512
COMPUTE WMCS >m513
COMPUTE WMXF >m514/100.
COMPUTE WMWK >(m501+m502)/100.
COMPUTE >ENDIF

* PNR COMMUTER RAIL
COMPUTE >IF (m319>0)
COMPUTE PCIV >m319/100.
COMPUTE PCAA >m329/100.
COMPUTE PCOV >(m317+m318+m332)/100.
COMPUTE PCXF >m326
COMPUTE PCCS >m327+m331+m330/100*AUOP
COMPUTE PCXP >m328/100.
COMPUTE PCWK >(m315+m316)/100.
COMPUTE >ENDIF

* PNR BUS
COMPUTE >IF (m419>0)
COMPUTE PBIV >m419/100.
COMPUTE PBAA >m429/100.
COMPUTE PBOV >(m417+m418+m432)/100.
COMPUTE PBXF >m426
COMPUTE PBXS >m427+m431+m430/100*AUOP
COMPUTE PBXP >m428/100.
COMPUTE PBWK >(m415+m416)/100.
COMPUTE >ENDIF

* PNR BUS/METRORAIL (TRANSIT)
COMPUTE >IF (m619>0)
COMPUTE PTIV >m619/100.
COMPUTE PTAA >m629/100.
COMPUTE PTOV >(m617+m618+m632)/100.
COMPUTE PTXF >m626
COMPUTE PTCS >m627+m631+m630/100*AUOP
COMPUTE PTXP >m628/100.
COMPUTE PTWK >(m615+m616)/100.
COMPUTE >ENDIF

* PNR METRORAIL
COMPUTE >IF (m519>0)
COMPUTE PMIV >m519/100.
COMPUTE PMAA >m529/100.
COMPUTE PMOV >(m517+m518+m532)/100.
COMPUTE PMXF >m526
COMPUTE PMCS >m527+m531+m530/100*AUOP
COMPUTE PMXP >m528/100.
COMPUTE PMWK >(m515+m516)/100.
COMPUTE >ENDIF

* KNR COMMUTER RAIL
COMPUTE >IF (m319>0)
COMPUTE KCIV >m319/100.
COMPUTE KCAA >m329/100.
COMPUTE KCOV >(m317+m318)/100.
COMPUTE KCXF >m326
COMPUTE KCCS >m327+m330/100*AUOP
COMPUTE KCXP >m328/100.
COMPUTE KCWK >(m315+m316)/100.
COMPUTE >ENDIF
```

## Appendix D: AEMS Fortran Control Files

```

* KNR BUS
COMPUTE >IF (m437>0)
COMPUTE KBIV >m437/100.
COMPUTE KBAA >m447/100.
COMPUTE KBOV >(m435+m436)/100.
COMPUTE KBXF >m444
COMPUTE KBCS >m445+m448/100*AUOP
COMPUTE KBXP >m446/100.
COMPUTE KBWK >(m433+m434)/100.
COMPUTE >ENDIF

```

```

* KNR BUS/METRORAIL (TRANSIT)
COMPUTE >IF (m637>0)
COMPUTE KTIV >m637/100.
COMPUTE KTAA >m647/100.
COMPUTE KTOV >(m635+m636)/100.
COMPUTE KTXF >m644
COMPUTE KTCS >m645+m648/100*AUOP
COMPUTE KTXP >m646/100.
COMPUTE KTWK >(m633+m634)/100.
COMPUTE >ENDIF

```

```

* KNR METRORAIL
COMPUTE >IF (m537>0)
COMPUTE KMIV >m537/100.
COMPUTE KMAA >m547/100.
COMPUTE KMOV >(m535+m536)/100.
COMPUTE KMXF >m544
COMPUTE KMCS >m545+m548/100*AUOP
COMPUTE KMXP >m546/100.
COMPUTE KMWK >(m533+m534)/100.
COMPUTE >ENDIF

```

\*CONSTANTS BY CHOICE FOR EACH PURPOSE

*CHOICE	1>DR	ALONE	SR2	SR3+	WK-CR	WK-BUS	WK-BU/MR	WK-MR	PNR-CR	KNR-CR	PNR-BUS	KNR-BUS	PNR-BU/MR	KNR-BU/MR	PNR-MR	KNR-MR
PURP01 1INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 1INC 2	1>															
PURP03 1INC 3	1>															
PURP04 1INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 2INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 2INC 2	1>															
PURP03 2INC 3	1>															
PURP04 2INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 3INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 3INC 2	1>															
PURP03 3INC 3	1>															
PURP04 3INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 4INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 4INC 2	1>															
PURP03 4INC 3	1>															
PURP04 4INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 5INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 5INC 2	1>															
PURP03 5INC 3	1>															
PURP04 5INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 6INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 6INC 2	1>															
PURP03 6INC 3	1>															
PURP04 6INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 7INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 7INC 2	1>															
PURP03 7INC 3	1>															
PURP04 7INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								
PURP01 8INC 1	1>				2.000000	2.000000	2.000000	2.000000								
PURP02 8INC 2	1>															
PURP03 8INC 3	1>															
PURP04 8INC 4	1>				-2.000000	-2.000000	-2.000000	-2.000000								

**Appendix D: AEMS Fortran Control Files**

```

PURP01 9INC 1      1>                2.000000  2.000000  2.000000  2.000000
PURP02 9INC 2      1>
PURP03 9INC 3      1>
PURP04 9INC 4      1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0110INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0210INC 2     1>
PURP0310INC 3     1>
PURP0410INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0111INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0211INC 2     1>
PURP0311INC 3     1>
PURP0411INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0112INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0212INC 2     1>
PURP0312INC 3     1>
PURP0412INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0113INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0213INC 2     1>
PURP0313INC 3     1>
PURP0413INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0114INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0214INC 2     1>
PURP0314INC 3     1>
PURP0414INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0115INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0215INC 2     1>
PURP0315INC 3     1>
PURP0415INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0116INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0216INC 2     1>
PURP0316INC 3     1>
PURP0416INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0117INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0217INC 2     1>
PURP0317INC 3     1>
PURP0417INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0118INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0218INC 2     1>
PURP0318INC 3     1>
PURP0418INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0119INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0219INC 2     1>
PURP0319INC 3     1>
PURP0419INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000
PURP0120INC 1     1>                2.000000  2.000000  2.000000  2.000000
PURP0220INC 2     1>
PURP0320INC 3     1>
PURP0420INC 4     1>                -2.000000 -2.000000 -2.000000 -2.000000

```

```

TRIPIN01          >TRP1
TRIPIN02          >TRP2
TRIPIN03          >TRP3
TRIPIN04          >TRP4
TRIPIFACT01       >tfi1
TRIPIFACT02       >tfi2
TRIPIFACT03       >tfi3
TRIPIFACT04       >tfi4
COMPUTE tfi1      >1.0
COMPUTE tfi2      >1.0
COMPUTE tfi3      >1.0
COMPUTE tfi4      >1.0

```

```

*
*OUTPUT MATRICES AND OUTPUT FACTORS BY CHOICE FOR EACH PURPOSE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
TRIPOUT01        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT01        1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT02        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914

```



**Appendix D: AEMS Fortran Control Files**

```

TRIPFACT02      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT03      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT03      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT04      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT04      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
**
**P AND A WALK PERCENTS BY CHOICE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
WALK SEG CW 1 PCT 1>WSWM
WALK SEG CW 1 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 2 PCT 1>WSW1
WALK SEG CW 2 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 3 PCT 1>WSW2
WALK SEG CW 3 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 6 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
*SYNTAX OF COMMAND TO ADD A COMPONENT TO A SPECIFIC WALK SEGMENT IF DESIRED
*WALK SEG CW 1 COEF1>      -0.04747  -0.04747  -0.04747  -0.04747  -0.04747  -0.04747
*WALK SEG CW 1 VAR 1>      WTSS      DTSS      DISS      WRSS      DRSS      DJSS
COMPUTE WSWM      >PCMI*PCMJ
COMPUTE WSW1      >(PCTI-PCMI)*PCMJ
COMPUTE WSW2      >(PCTI-PCMI)*(PCTJ-PCMJ)
COMPUTE WSW3      >PCMI*(PCTJ-PCMJ)
COMPUTE WSM1      >(1-PCTI)*PCMJ
COMPUTE WSM2      >(1-PCTI)*(PCTJ-PCMJ)
COMPUTE WSNT      >1-WSWM-WSW1-WSW2-WSW3-WSM1-WSM2

*NEST DEFINITIONS BY CHOICE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
NEST 1,1=        1>Y      Y      Y
NEST 1,2=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,1=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,2=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,3=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,1=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,2=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,3=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,4=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,1=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,2=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,3=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,4=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,1=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,2=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,3=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,4=        1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 6,1=        1>Y      Y      Y
NEST 6,2=        1>      Y      Y
NEST 7,1=        1>      Y      Y
NEST 7,2=        1>      Y      Y

IGRP DEFINITION  >i813
JGRP DEFINITION  >j813
* 1 DC CORE/URBAN-DC CORE
SEGMENT 1      > 1 1
SEGMENT 1      > 3 1
* 2 DC CORE/URBAN-VA CORE
SEGMENT 2      > 1 2
SEGMENT 2      > 3 2
* 3 DC CORE/URBAN-URBAN
SEGMENT 3      > 1 3
SEGMENT 3      > 3 3

```

## Appendix D: AEMS Fortran Control Files

---

```

SEGMENT 3      >  1  4
SEGMENT 3      >  3  4
SEGMENT 3      >  1  5
SEGMENT 3      >  3  5
* 4 DC CORE/URBAN-OTHER
SEGMENT 4      >  1  6
SEGMENT 4      >  3  6
SEGMENT 4      >  1  7
SEGMENT 4      >  3  7
* 5 MD URBAN-DC CORE
SEGMENT 5      >  4  1
* 6 MD URBAN-VA CORE
SEGMENT 6      >  4  2
* 7 MD URBAN-URBAN
SEGMENT 7      >  4  3
SEGMENT 7      >  4  4
SEGMENT 7      >  4  5
* 8 MD URBAN-OTHER
SEGMENT 8      >  4  6
SEGMENT 8      >  4  7
* 9 VA CORE/URBAN-DC CORE
SEGMENT 9      >  2  1
SEGMENT 9      >  5  1
*10 VA CORE/URBAN-VA CORE
SEGMENT 10     >  2  2
SEGMENT 10     >  5  2
*11 VA CORE/URBAN-URBAN
SEGMENT 11     >  2  3
SEGMENT 11     >  5  3
SEGMENT 11     >  2  4
SEGMENT 11     >  5  4
SEGMENT 11     >  2  5
SEGMENT 11     >  5  5
*12 VA CORE/URBAN-OTHER
SEGMENT 12     >  2  6
SEGMENT 12     >  5  6
SEGMENT 12     >  2  7
SEGMENT 12     >  5  7
*13 MD OTHER-DC CORE
SEGMENT 13     >  6  1
*14 MD OTHER-VA CORE
SEGMENT 14     >  6  2
*15 MD OTHER-URBAN
SEGMENT 15     >  6  3
SEGMENT 15     >  6  4
SEGMENT 15     >  6  5
*16 MD OTHER-OTHER
SEGMENT 16     >  6  6
SEGMENT 16     >  6  7
*17 VA OTHER-DC CORE
SEGMENT 17     >  7  1
*18 VA OTHER-VA CORE
SEGMENT 18     >  7  2
*19 VA OTHER-URBAN
SEGMENT 19     >  7  3
SEGMENT 19     >  7  4
SEGMENT 19     >  7  5
*20 VA OTHER-OTHER
SEGMENT 20     >  7  6
SEGMENT 20     >  7  7

* SEGMENT 1
NSTC 10 1GRND TOTAL>
NSTC 11 1AUTO      >    0.5  0.00000
NSTC 12 1TRANSIT  >    0.5  0.45317
NSTC 20 1TOTAL TRN >
NSTC 21 1WALK ACC >    0.5  0.00000
NSTC 22 1PNR ACC  >    0.5 -2.50663

```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 23 1KNR ACC > 0.5 -6.06114
NSTC 30 1WLK TRN
NSTC 31 1WLK CR > 1.0 0.05410
NSTC 32 1WLK BUS > 1.0 0.01251
NSTC 33 1WLK BU/MR > 1.0 0.45714
NSTC 34 1WLK METRO > 1.0 0.00000
NSTC 40 1PNR TRN
NSTC 41 1PNR CR > 1.0 -0.22736
NSTC 42 1PNR BUS > 1.0 -2.64918
NSTC 43 1PNR BU/MR > 1.0 -1.15032
NSTC 44 1PNR METRO > 1.0 0.00000
NSTC 50 1KNR TRN
NSTC 51 1KNR CR > 1.0 1.33122
NSTC 52 1KNR BUS > 1.0 4.59386
NSTC 53 1KNR BU/MR > 1.0 8.79511
NSTC 54 1KNR METRO > 1.0 0.00000
NSTC 60 1AUTO
NSTC 61 1LOV > 1.0 0.00000
NSTC 62 1HOV > 0.5 0.00689
NSTC 70 1HOV
NSTC 71 1HOV2 > 1.0 0.00000
NSTC 72 1HOV3+ > 1.0 -0.73183
* SEGMENT 2
NSTC 10 2GRND TOTAL>
NSTC 11 2AUTO > 0.5 0.00000
NSTC 12 2TRANSIT > 0.5 0.27914
NSTC 20 2TOTAL TRN >
NSTC 21 2WALK ACC > 0.5 0.00000
NSTC 22 2PNR ACC > 0.5 -2.32206
NSTC 23 2KNR ACC > 0.5 -4.38172
NSTC 30 2WLK TRN
NSTC 31 2WLK CR > 1.0 -0.65743
NSTC 32 2WLK BUS > 1.0 -7.72372
NSTC 33 2WLK BU/MR > 1.0 -2.20894
NSTC 34 2WLK METRO > 1.0 0.00000
NSTC 40 2PNR TRN
NSTC 41 2PNR CR > 1.0 0.00001
NSTC 42 2PNR BUS > 1.0 0.00001
NSTC 43 2PNR BU/MR > 1.0 0.00001
NSTC 44 2PNR METRO > 1.0 0.00000
NSTC 50 2KNR TRN
NSTC 51 2KNR CR > 1.0 0.08545
NSTC 52 2KNR BUS > 1.0 4.26724
NSTC 53 2KNR BU/MR > 1.0 0.08545
NSTC 54 2KNR METRO > 1.0 0.00000
NSTC 60 2AUTO
NSTC 61 2LOV > 1.0 0.00000
NSTC 62 2HOV > 0.5 -0.42368
NSTC 70 2HOV
NSTC 71 2HOV2 > 1.0 0.00000
NSTC 72 2HOV3+ > 1.0 -1.03353
* SEGMENT 3
NSTC 10 3GRND TOTAL>
NSTC 11 3AUTO > 0.5 0.00000
NSTC 12 3TRANSIT > 0.5 -0.31828
NSTC 20 3TOTAL TRN >
NSTC 21 3WALK ACC > 0.5 0.00000
NSTC 22 3PNR ACC > 0.5 -3.45783
NSTC 23 3KNR ACC > 0.5 -5.80534
NSTC 30 3WLK TRN
NSTC 31 3WLK CR > 1.0 0.23247
NSTC 32 3WLK BUS > 1.0 0.47662
NSTC 33 3WLK BU/MR > 1.0 -0.22459
NSTC 34 3WLK METRO > 1.0 0.00000
NSTC 40 3PNR TRN
NSTC 41 3PNR CR > 1.0 -0.52712
NSTC 42 3PNR BUS > 1.0 -3.70830
NSTC 43 3PNR BU/MR > 1.0 0.81313
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 44 3PNR METRO > 1.0 0.00000
NSTC 50 3KNR TRN
NSTC 51 3KNR CR > 1.0 0.67000
NSTC 52 3KNR BUS > 1.0 2.01709
NSTC 53 3KNR BU/MR > 1.0 1.54875
NSTC 54 3KNR METRO > 1.0 0.00000
NSTC 60 3AUTO
NSTC 61 3LOV > 1.0 0.00000
NSTC 62 3HOV > 0.5 -0.18691
NSTC 70 3HOV
NSTC 71 3HOV2 > 1.0 0.00000
NSTC 72 3HOV3+ > 1.0 -0.25745
* SEGMENT 4
NSTC 10 4GRND TOTAL>
NSTC 11 4AUTO > 0.5 0.00000
NSTC 12 4TRANSIT > 0.5 -0.18863
NSTC 20 4TOTAL TRN >
NSTC 21 4WALK ACC > 0.5 0.00000
NSTC 22 4PNR ACC > 0.5 -2.42793
NSTC 23 4KNR ACC > 0.5 -3.47893
NSTC 30 4WLK TRN
NSTC 31 4WLK CR > 1.0 0.70193
NSTC 32 4WLK BUS > 1.0 -3.07899
NSTC 33 4WLK BU/MR > 1.0 -3.24678
NSTC 34 4WLK METRO > 1.0 0.00000
NSTC 40 4PNR TRN
NSTC 41 4PNR CR > 1.0 -5.00273
NSTC 42 4PNR BUS > 1.0 -0.51498
NSTC 43 4PNR BU/MR > 1.0 -5.00273
NSTC 44 4PNR METRO > 1.0 0.00000
NSTC 50 4KNR TRN
NSTC 51 4KNR CR > 1.0 11.36994
NSTC 52 4KNR BUS > 1.0 -1.49215
NSTC 53 4KNR BU/MR > 1.0 -1.94597
NSTC 54 4KNR METRO > 1.0 0.00000
NSTC 60 4AUTO
NSTC 61 4LOV > 1.0 0.00000
NSTC 62 4HOV > 0.5 -0.63104
NSTC 70 4HOV
NSTC 71 4HOV2 > 1.0 0.00000
NSTC 72 4HOV3+ > 1.0 -0.09180
* SEGMENT 5
NSTC 10 5GRND TOTAL>
NSTC 11 5AUTO > 0.5 0.00000
NSTC 12 5TRANSIT > 0.5 1.30468
NSTC 20 5TOTAL TRN >
NSTC 21 5WALK ACC > 0.5 0.00000
NSTC 22 5PNR ACC > 0.5 -5.65379
NSTC 23 5KNR ACC > 0.5 -7.69645
NSTC 30 5WLK TRN
NSTC 31 5WLK CR > 1.0 -1.59482
NSTC 32 5WLK BUS > 1.0 -5.05964
NSTC 33 5WLK BU/MR > 1.0 -3.34682
NSTC 34 5WLK METRO > 1.0 0.00000
NSTC 40 5PNR TRN
NSTC 41 5PNR CR > 1.0 -0.20546
NSTC 42 5PNR BUS > 1.0 -0.20546
NSTC 43 5PNR BU/MR > 1.0 9.62388
NSTC 44 5PNR METRO > 1.0 0.00000
NSTC 50 5KNR TRN
NSTC 51 5KNR CR > 1.0 0.24425
NSTC 52 5KNR BUS > 1.0 0.24425
NSTC 53 5KNR BU/MR > 1.0 11.06296
NSTC 54 5KNR METRO > 1.0 0.00000
NSTC 60 5AUTO
NSTC 61 5LOV > 1.0 0.00000
NSTC 62 5HOV > 0.5 -0.48182
NSTC 70 5HOV
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 71 5HOV2 > 1.0 0.00000
NSTC 72 5HOV3+ > 1.0 -1.21036
* SEGMENT 6
NSTC 10 6GRND TOTAL>
NSTC 11 6AUTO > 0.5 0.00000
NSTC 12 6TRANSIT > 0.5 6.08282
NSTC 20 6TOTAL TRN >
NSTC 21 6WALK ACC > 0.5 0.00000
NSTC 22 6PNR ACC > 0.5 -8.05437
NSTC 23 6KNR ACC > 0.5 -8.05437
NSTC 30 6WLK TRN
NSTC 31 6WLK CR > 1.0 -0.83154
NSTC 32 6WLK BUS > 1.0 -4.38261
NSTC 33 6WLK BU/MR > 1.0 0.68570
NSTC 34 6WLK METRO > 1.0 0.00000
NSTC 40 6PNR TRN
NSTC 41 6PNR CR > 1.0 0.00001
NSTC 42 6PNR BUS > 1.0 0.00001
NSTC 43 6PNR BU/MR > 1.0 0.00001
NSTC 44 6PNR METRO > 1.0 0.00000
NSTC 50 6KNR TRN
NSTC 51 6KNR CR > 1.0 0.00001
NSTC 52 6KNR BUS > 1.0 0.00001
NSTC 53 6KNR BU/MR > 1.0 0.00001
NSTC 54 6KNR METRO > 1.0 0.00000
NSTC 60 6AUTO
NSTC 61 6LOV > 1.0 0.00000
NSTC 62 6HOV > 0.5 0.00001
NSTC 70 6HOV
NSTC 71 6HOV2 > 1.0 0.00000
NSTC 72 6HOV3+ > 1.0 0.00001
* SEGMENT 7
NSTC 10 7GRND TOTAL>
NSTC 11 7AUTO > 0.5 0.00000
NSTC 12 7TRANSIT > 0.5 0.38378
NSTC 20 7TOTAL TRN >
NSTC 21 7WALK ACC > 0.5 0.00000
NSTC 22 7PNR ACC > 0.5 -6.97210
NSTC 23 7KNR ACC > 0.5 -9.32075
NSTC 30 7WLK TRN
NSTC 31 7WLK CR > 1.0 -2.32569
NSTC 32 7WLK BUS > 1.0 -3.27955
NSTC 33 7WLK BU/MR > 1.0 -3.67036
NSTC 34 7WLK METRO > 1.0 0.00000
NSTC 40 7PNR TRN
NSTC 41 7PNR CR > 1.0 0.72325
NSTC 42 7PNR BUS > 1.0 2.00047
NSTC 43 7PNR BU/MR > 1.0 -0.33653
NSTC 44 7PNR METRO > 1.0 0.00000
NSTC 50 7KNR TRN
NSTC 51 7KNR CR > 1.0 0.06216
NSTC 52 7KNR BUS > 1.0 4.09049
NSTC 53 7KNR BU/MR > 1.0 0.06216
NSTC 54 7KNR METRO > 1.0 0.00000
NSTC 60 7AUTO
NSTC 61 7LOV > 1.0 0.00000
NSTC 62 7HOV > 0.5 0.02044
NSTC 70 7HOV
NSTC 71 7HOV2 > 1.0 0.00000
NSTC 72 7HOV3+ > 1.0 -0.35894
* SEGMENT 8
NSTC 10 8GRND TOTAL>
NSTC 11 8AUTO > 0.5 0.00000
NSTC 12 8TRANSIT > 0.5 -0.86870
NSTC 20 8TOTAL TRN >
NSTC 21 8WALK ACC > 0.5 0.00000
NSTC 22 8PNR ACC > 0.5 -3.56771
NSTC 23 8KNR ACC > 0.5 -6.82549
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 30 8WLK TRN
NSTC 31 8WLK CR > 1.0 -1.54482
NSTC 32 8WLK BUS > 1.0 -1.53220
NSTC 33 8WLK BU/MR > 1.0 -2.64989
NSTC 34 8WLK METRO > 1.0 0.00000
NSTC 40 8PNR TRN
NSTC 41 8PNR CR > 1.0 0.00001
NSTC 42 8PNR BUS > 1.0 8.69455
NSTC 43 8PNR BU/MR > 1.0 0.00001
NSTC 44 8PNR METRO > 1.0 0.00000
NSTC 50 8KNR TRN
NSTC 51 8KNR CR > 1.0 0.00001
NSTC 52 8KNR BUS > 1.0 7.60861
NSTC 53 8KNR BU/MR > 1.0 2.63773
NSTC 54 8KNR METRO > 1.0 0.00000
NSTC 60 8AUTO
NSTC 61 8LOV > 1.0 0.00000
NSTC 62 8HOV > 0.5 0.06779
NSTC 70 8HOV
NSTC 71 8HOV2 > 1.0 0.00000
NSTC 72 8HOV3+ > 1.0 -0.72493
* SEGMENT 9
NSTC 10 9GRND TOTAL>
NSTC 11 9AUTO > 0.5 0.00000
NSTC 12 9TRANSIT > 0.5 3.92523
NSTC 20 9TOTAL TRN >
NSTC 21 9WALK ACC > 0.5 0.00000
NSTC 22 9PNR ACC > 0.5 -8.30451
NSTC 23 9KNR ACC > 0.5 -11.17724
NSTC 30 9WLK TRN
NSTC 31 9WLK CR > 1.0 -2.46668
NSTC 32 9WLK BUS > 1.0 -10.60038
NSTC 33 9WLK BU/MR > 1.0 -9.10348
NSTC 34 9WLK METRO > 1.0 0.00000
NSTC 40 9PNR TRN
NSTC 41 9PNR CR > 1.0 0.02007
NSTC 42 9PNR BUS > 1.0 0.02007
NSTC 43 9PNR BU/MR > 1.0 1.47201
NSTC 44 9PNR METRO > 1.0 0.00000
NSTC 50 9KNR TRN
NSTC 51 9KNR CR > 1.0 2.24601
NSTC 52 9KNR BUS > 1.0 2.24601
NSTC 53 9KNR BU/MR > 1.0 17.43966
NSTC 54 9KNR METRO > 1.0 0.00000
NSTC 60 9AUTO
NSTC 61 9LOV > 1.0 0.00000
NSTC 62 9HOV > 0.5 0.05993
NSTC 70 9HOV
NSTC 71 9HOV2 > 1.0 0.00000
NSTC 72 9HOV3+ > 1.0 -0.36571
* SEGMENT 10
NSTC 1010GRND TOTAL>
NSTC 1110AUTO > 0.5 0.00000
NSTC 1210TRANSIT > 0.5 -0.68407
NSTC 2010TOTAL TRN >
NSTC 2110WALK ACC > 0.5 0.00000
NSTC 2210PNR ACC > 0.5 -6.99975
NSTC 2310KNR ACC > 0.5 -6.99975
NSTC 3010WLK TRN
NSTC 3110WLK CR > 1.0 -1.98870
NSTC 3210WLK BUS > 1.0 -10.86321
NSTC 3310WLK BU/MR > 1.0 -9.00427
NSTC 3410WLK METRO > 1.0 0.00000
NSTC 4010PNR TRN
NSTC 4110PNR CR > 1.0 0.00001
NSTC 4210PNR BUS > 1.0 0.00001
NSTC 4310PNR BU/MR > 1.0 0.00001
NSTC 4410PNR METRO > 1.0 0.00000
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5010KNR TRN
NSTC 5110KNR CR > 1.0 0.00001
NSTC 5210KNR BUS > 1.0 0.00001
NSTC 5310KNR BU/MR > 1.0 0.00001
NSTC 5410KNR METRO > 1.0 0.00000
NSTC 6010AUTO
NSTC 6110LOV > 1.0 0.00000
NSTC 6210HOV > 0.5 0.05002
NSTC 7010HOV
NSTC 7110HOV2 > 1.0 0.00000
NSTC 7210HOV3+ > 1.0 -0.16315
* SEGMENT 11
NSTC 1011GRND TOTAL>
NSTC 1111AUTO > 0.5 0.00000
NSTC 1211TRANSIT > 0.5 0.22501
NSTC 2011TOTAL TRN >
NSTC 2111WALK ACC > 0.5 0.00000
NSTC 2211PNR ACC > 0.5 -6.47383
NSTC 2311KNR ACC > 0.5 -8.46553
NSTC 3011WLK TRN
NSTC 3111WLK CR > 1.0 -3.68540
NSTC 3211WLK BUS > 1.0 -6.32010
NSTC 3311WLK BU/MR > 1.0 -1.97074
NSTC 3411WLK METRO > 1.0 0.00000
NSTC 4011PNR TRN
NSTC 4111PNR CR > 1.0 -5.51823
NSTC 4211PNR BUS > 1.0 -5.51823
NSTC 4311PNR BU/MR > 1.0 3.05786
NSTC 4411PNR METRO > 1.0 0.00000
NSTC 5011KNR TRN
NSTC 5111KNR CR > 1.0 -2.13576
NSTC 5211KNR BUS > 1.0 2.56385
NSTC 5311KNR BU/MR > 1.0 -2.13576
NSTC 5411KNR METRO > 1.0 0.00000
NSTC 6011AUTO
NSTC 6111LOV > 1.0 0.00000
NSTC 6211HOV > 0.5 -0.04226
NSTC 7011HOV
NSTC 7111HOV2 > 1.0 0.00000
NSTC 7211HOV3+ > 1.0 -0.32995
* SEGMENT 12
NSTC 1012GRND TOTAL>
NSTC 1112AUTO > 0.5 0.00000
NSTC 1212TRANSIT > 0.5 0.64554
NSTC 2012TOTAL TRN >
NSTC 2112WALK ACC > 0.5 0.00000
NSTC 2212PNR ACC > 0.5 -6.04737
NSTC 2312KNR ACC > 0.5 -6.12088
NSTC 3012WLK TRN
NSTC 3112WLK CR > 1.0 -5.23554
NSTC 3212WLK BUS > 1.0 -9.87643
NSTC 3312WLK BU/MR > 1.0 -8.57162
NSTC 3412WLK METRO > 1.0 0.00000
NSTC 4012PNR TRN
NSTC 4112PNR CR > 1.0 -10.58905
NSTC 4212PNR BUS > 1.0 -10.58905
NSTC 4312PNR BU/MR > 1.0 -10.58905
NSTC 4412PNR METRO > 1.0 0.00000
NSTC 5012KNR TRN
NSTC 5112KNR CR > 1.0 -4.93712
NSTC 5212KNR BUS > 1.0 -6.20552
NSTC 5312KNR BU/MR > 1.0 -4.93712
NSTC 5412KNR METRO > 1.0 0.00000
NSTC 6012AUTO
NSTC 6112LOV > 1.0 0.00000
NSTC 6212HOV > 0.5 -0.21246
NSTC 7012HOV
NSTC 7112HOV2 > 1.0 0.00000
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 7212HOV3+ > 1.0 -0.23617
* SEGMENT 13
NSTC 1013GRND TOTAL>
NSTC 1113AUTO > 0.5 0.00000
NSTC 1213TRANSIT > 0.5 0.63282
NSTC 2013TOTAL TRN >
NSTC 2113WALK ACC > 0.5 0.00000
NSTC 2213PNR ACC > 0.5 -2.80539
NSTC 2313KNR ACC > 0.5 -5.16069
NSTC 3013WLK TRN
NSTC 3113WLK CR > 1.0 10.96184
NSTC 3213WLK BUS > 1.0 -0.75120
NSTC 3313WLK BU/MR > 1.0 -2.94234
NSTC 3413WLK METRO > 1.0 0.00000
NSTC 4013PNR TRN
NSTC 4113PNR CR > 1.0 3.48708
NSTC 4213PNR BUS > 1.0 5.68887
NSTC 4313PNR BU/MR > 1.0 9.01923
NSTC 4413PNR METRO > 1.0 0.00000
NSTC 5013KNR TRN
NSTC 5113KNR CR > 1.0 1.90441
NSTC 5213KNR BUS > 1.0 0.77602
NSTC 5313KNR BU/MR > 1.0 5.51745
NSTC 5413KNR METRO > 1.0 0.00000
NSTC 6013AUTO
NSTC 6113LOV > 1.0 0.00000
NSTC 6213HOV > 0.5 -0.29387
NSTC 7013HOV
NSTC 7113HOV2 > 1.0 0.00000
NSTC 7213HOV3+ > 1.0 -0.07599
* SEGMENT 14
NSTC 1014GRND TOTAL>
NSTC 1114AUTO > 0.5 0.00000
NSTC 1214TRANSIT > 0.5 0.41169
NSTC 2014TOTAL TRN >
NSTC 2114WALK ACC > 0.5 0.00000
NSTC 2214PNR ACC > 0.5 -1.21265
NSTC 2314KNR ACC > 0.5 -1.29053
NSTC 3014WLK TRN
NSTC 3114WLK CR > 1.0 -2.41491
NSTC 3214WLK BUS > 1.0 -2.41491
NSTC 3314WLK BU/MR > 1.0 -4.24053
NSTC 3414WLK METRO > 1.0 0.00000
NSTC 4014PNR TRN
NSTC 4114PNR CR > 1.0 0.66791
NSTC 4214PNR BUS > 1.0 0.66791
NSTC 4314PNR BU/MR > 1.0 10.70197
NSTC 4414PNR METRO > 1.0 0.00000
NSTC 5014KNR TRN
NSTC 5114KNR CR > 1.0 -0.36013
NSTC 5214KNR BUS > 1.0 -0.36013
NSTC 5314KNR BU/MR > 1.0 1.98230
NSTC 5414KNR METRO > 1.0 0.00000
NSTC 6014AUTO
NSTC 6114LOV > 1.0 0.00000
NSTC 6214HOV > 0.5 -0.58845
NSTC 7014HOV
NSTC 7114HOV2 > 1.0 0.00000
NSTC 7214HOV3+ > 1.0 0.91729
* SEGMENT 15
NSTC 1015GRND TOTAL>
NSTC 1115AUTO > 0.5 0.00000
NSTC 1215TRANSIT > 0.5 -0.61522
NSTC 2015TOTAL TRN >
NSTC 2115WALK ACC > 0.5 0.00000
NSTC 2215PNR ACC > 0.5 -2.94896
NSTC 2315KNR ACC > 0.5 -4.10451
NSTC 3015WLK TRN
```



## Appendix D: AEMS Fortran Control Files

---

NSTC 3115WLK CR >	1.0	1.45835
NSTC 3215WLK BUS >	1.0	1.95937
NSTC 3315WLK BU/MR >	1.0	0.43874
NSTC 3415WLK METRO >	1.0	0.00000
NSTC 4015PNR TRN		
NSTC 4115PNR CR >	1.0	2.28001
NSTC 4215PNR BUS >	1.0	0.72610
NSTC 4315PNR BU/MR >	1.0	1.72976
NSTC 4415PNR METRO >	1.0	0.00000
NSTC 5015KNR TRN		
NSTC 5115KNR CR >	1.0	3.77376
NSTC 5215KNR BUS >	1.0	3.91105
NSTC 5315KNR BU/MR >	1.0	2.95518
NSTC 5415KNR METRO >	1.0	0.00000
NSTC 6015AUTO		
NSTC 6115LOV >	1.0	0.00000
NSTC 6215HOV >	0.5	-0.25379
NSTC 7015HOV		
NSTC 7115HOV2 >	1.0	0.00000
NSTC 7215HOV3+ >	1.0	-0.31082
* SEGMENT 16		
NSTC 1016GRND TOTAL>		
NSTC 1116AUTO >	0.5	0.00000
NSTC 1216TRANSIT >	0.5	-1.37850
NSTC 2016TOTAL TRN >		
NSTC 2116WALK ACC >	0.5	0.00000
NSTC 2216PNR ACC >	0.5	-5.46387
NSTC 2316KNR ACC >	0.5	-5.49947
NSTC 3016WLK TRN		
NSTC 3116WLK CR >	1.0	1.03027
NSTC 3216WLK BUS >	1.0	2.78982
NSTC 3316WLK BU/MR >	1.0	1.67218
NSTC 3416WLK METRO >	1.0	0.00000
NSTC 4016PNR TRN		
NSTC 4116PNR CR >	1.0	-0.71616
NSTC 4216PNR BUS >	1.0	2.92869
NSTC 4316PNR BU/MR >	1.0	-27.59178
NSTC 4416PNR METRO >	1.0	0.00000
NSTC 5016KNR TRN		
NSTC 5116KNR CR >	1.0	-0.09418
NSTC 5216KNR BUS >	1.0	3.60157
NSTC 5316KNR BU/MR >	1.0	1.07478
NSTC 5416KNR METRO >	1.0	0.00000
NSTC 6016AUTO		
NSTC 6116LOV >	1.0	0.00000
NSTC 6216HOV >	0.5	0.05097
NSTC 7016HOV		
NSTC 7116HOV2 >	1.0	0.00000
NSTC 7216HOV3+ >	1.0	-0.20668
* SEGMENT 17		
NSTC 1017GRND TOTAL>		
NSTC 1117AUTO >	0.5	0.00000
NSTC 1217TRANSIT >	0.5	3.14858
NSTC 2017TOTAL TRN >		
NSTC 2117WALK ACC >	0.5	0.00000
NSTC 2217PNR ACC >	0.5	-7.83922
NSTC 2317KNR ACC >	0.5	-9.27002
NSTC 3017WLK TRN		
NSTC 3117WLK CR >	1.0	-4.08443
NSTC 3217WLK BUS >	1.0	-5.64167
NSTC 3317WLK BU/MR >	1.0	-6.68884
NSTC 3417WLK METRO >	1.0	0.00000
NSTC 4017PNR TRN		
NSTC 4117PNR CR >	1.0	0.66782
NSTC 4217PNR BUS >	1.0	4.22864
NSTC 4317PNR BU/MR >	1.0	5.96479
NSTC 4417PNR METRO >	1.0	0.00000
NSTC 5017KNR TRN		

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5117KNR CR > 1.0 0.44610
NSTC 5217KNR BUS > 1.0 0.44610
NSTC 5317KNR BU/MR > 1.0 3.81861
NSTC 5417KNR METRO > 1.0 0.00000
NSTC 6017AUTO
NSTC 6117LOV > 1.0 0.00000
NSTC 6217HOV > 0.5 -0.66508
NSTC 7017HOV
NSTC 7117HOV2 > 1.0 0.00000
NSTC 7217HOV3+ > 1.0 -1.87496
* SEGMENT 18
NSTC 1018GRND TOTAL>
NSTC 1118AUTO > 0.5 0.00000
NSTC 1218TRANSIT > 0.5 0.97062
NSTC 2018TOTAL TRN >
NSTC 2118WALK ACC > 0.5 0.00000
NSTC 2218PNR ACC > 0.5 -5.88272
NSTC 2318KNR ACC > 0.5 -8.53952
NSTC 3018WLK TRN
NSTC 3118WLK CR > 1.0 -4.60790
NSTC 3218WLK BUS > 1.0 -8.29148
NSTC 3318WLK BU/MR > 1.0 -8.78040
NSTC 3418WLK METRO > 1.0 0.00000
NSTC 4018PNR TRN
NSTC 4118PNR CR > 1.0 -0.19921
NSTC 4218PNR BUS > 1.0 -5.26481
NSTC 4318PNR BU/MR > 1.0 3.61636
NSTC 4418PNR METRO > 1.0 0.00000
NSTC 5018KNR TRN
NSTC 5118KNR CR > 1.0 -0.05071
NSTC 5218KNR BUS > 1.0 -0.05071
NSTC 5318KNR BU/MR > 1.0 8.13127
NSTC 5418KNR METRO > 1.0 0.00000
NSTC 6018AUTO
NSTC 6118LOV > 1.0 0.00000
NSTC 6218HOV > 0.5 0.10237
NSTC 7018HOV
NSTC 7118HOV2 > 1.0 0.00000
NSTC 7218HOV3+ > 1.0 -0.15589
* SEGMENT 19
NSTC 1019GRND TOTAL>
NSTC 1119AUTO > 0.5 0.00000
NSTC 1219TRANSIT > 0.5 1.47913
NSTC 2019TOTAL TRN >
NSTC 2119WALK ACC > 0.5 0.00000
NSTC 2219PNR ACC > 0.5 -6.61721
NSTC 2319KNR ACC > 0.5 -8.10023
NSTC 3019WLK TRN
NSTC 3119WLK CR > 1.0 -6.02278
NSTC 3219WLK BUS > 1.0 -9.01987
NSTC 3319WLK BU/MR > 1.0 -4.61014
NSTC 3419WLK METRO > 1.0 0.00000
NSTC 4019PNR TRN
NSTC 4119PNR CR > 1.0 -1.22267
NSTC 4219PNR BUS > 1.0 -2.60941
NSTC 4319PNR BU/MR > 1.0 -0.71393
NSTC 4419PNR METRO > 1.0 0.00000
NSTC 5019KNR TRN
NSTC 5119KNR CR > 1.0 -0.12802
NSTC 5219KNR BUS > 1.0 1.35590
NSTC 5319KNR BU/MR > 1.0 1.82296
NSTC 5419KNR METRO > 1.0 0.00000
NSTC 6019AUTO
NSTC 6119LOV > 1.0 0.00000
NSTC 6219HOV > 0.5 -0.23707
NSTC 7019HOV
NSTC 7119HOV2 > 1.0 0.00000
NSTC 7219HOV3+ > 1.0 -0.50686
```

## Appendix D: AEMS Fortran Control Files

---

```

* SEGMENT 20
NSTC 1020GRND TOTAL>
NSTC 1120AUTO > 0.5 0.00000
NSTC 1220TRANSIT > 0.5 -0.60785
NSTC 2020TOTAL TRN >
NSTC 2120WALK ACC > 0.5 0.00000
NSTC 2220PNR ACC > 0.5 -26.03037
NSTC 2320KNR ACC > 0.5 -26.35298
NSTC 3020WLK TRN
NSTC 3120WLK CR > 1.0 -2.28782
NSTC 3220WLK BUS > 1.0 -1.65126
NSTC 3320WLK BU/MR > 1.0 -8.00663
NSTC 3420WLK METRO > 1.0 0.00000
NSTC 4020PNR TRN
NSTC 4120PNR CR > 1.0 -52.10533
NSTC 4220PNR BUS > 1.0 -44.19218
NSTC 4320PNR BU/MR > 1.0 -64.23509
NSTC 4420PNR METRO > 1.0 0.00000
NSTC 5020KNR TRN
NSTC 5120KNR CR > 1.0 -52.97657
NSTC 5220KNR BUS > 1.0 -43.88001
NSTC 5320KNR BU/MR > 1.0 -61.20523
NSTC 5420KNR METRO > 1.0 0.00000
NSTC 6020AUTO
NSTC 6120LOV > 1.0 0.00000
NSTC 6220HOV > 0.5 0.05703
NSTC 7020HOV
NSTC 7120HOV2 > 1.0 0.00000
NSTC 7220HOV3+ > 1.0 -0.10912

*DOWNTOWN=8
*SELI > 8

*UNION STATION=64
*SELI > 64

* =122
*SELI > 122

*BETHESDA=345
*SELI > 345

*SILVER SPRING=362
*SELI > 362

*N.SILVER SPRING=464
*SELI > 464

* =475
*SELI > 475

*SHADY GROVE RD=578
*SELI > 578

* =787
*SELI > 787

*ANDREWS AFB=829
*SELI > 829

*NEW CARROLTON=927
*SELI > 927

*BRISTOL=972
*SELI > 972

*FREDERICK=1043
*SELI > 1043

```

## Appendix D: AEMS Fortran Control Files

---

```
*JESSUP=1080
*SELI      >    1080

*SCAGGSVILLE=1091
*SELI      >    1091

*WALDORF=1216
*SELI      >    1216

*PENTAGON=1231
*SELI      >    1231

*ROSSLYN=1236
*SELI      >    1236

*ALEXANDRIA=1337
*SELI      >    1337

* =1455
*SELI      >    1455

*SPRINGFIELD=1502
*SELI      >    1502

* =1511
*SELI      >    1511

*TYSONS CRNR=1537
*SELI      >    1537

*FT BELVOIR=1554
*SELI      >    1554

*VIENNA=1619
*SELI      >    1619

*DULES AP=1698
*SELI      >    1698

*RESTON=1716
*SELI      >    1716

*LEESBURG=1842
*SELI      >    1842

*BRUNSWICK=1863
*SELI      >    1863

*DALE CITY=1942
*SELI      >    1942

*MANASSAS=1967
*SELI      >    1967

*SPOTSYLVANIA=2110
*SELI      >    2110

* =2055
*SELI      >    2055

*SELJ      >      8
*SELJ      >     63
*SELJ      >     64
*SELJ      >     77
*SELJ      >    100
*SELJ      >    344
*SELJ      >    345
```

## Appendix D: AEMS Fortran Control Files

```

*SELJ      >      362
*SELJ      >      1231
*SELJ      >      1236
*SELJ      >      1265
*SELJ      >      1337
*SELJ      >      1537
*SELI      >      523
*SELJ      >      9

TRACE      >      0
* OUTPUT % >
*PROCSSEL  >
PRINT MS   >HBO_NL_MC.PRN
INPUT PRINT FILE >HBO_NL_MC.PRN
INPUT GOALS >HBO_NL_MC.GOL
INFILE 1   >hbo_income.ptt
INFILE 2   >hwyop.skm
INFILE 3   >TRNOP_CR.SKM
INFILE 4   >TRNOP_AB.SKM
INFILE 5   >TRNOP_MR.SKM
INFILE 6   >TRNOP_BM.SKM
ZINFILE 8  >ZONEV2.A2F
OUTFILE 9  >HBO_NL_MC.MTT

* FTA USER BENEFITS SPECIFICATIONS
*FTA RESULTS FILE >HBO_NL_MC.BEN
FTA TRANSIT COEFF >-0.02322
FTA AUTO COEFF >-0.02322
FTA PURPOSE NAME >HBO
FTA PERIOD NAME >ALLDAY
FTA ALTER. NAME >CALIB
*CHOICE      1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
FTA AUTO NEST >      1      1
FTA MOTORIZED? >Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
FTA TRANSIT?  >      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y

```

## 2 hbs\_nl\_mc.cti

```

HBS OP NESTED LOGIT MC - #DATE: 9/17/2011 #VER: 21
CHOICE      1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
*
*
*LOGIT COEFFICIENTS BY CHOICE FOR EACH SKIM (NO INPUT SKIM IS
*EQUIVALENT TO A CONSTANT)
*CHOICE      1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
COEF01:IVTT  1>-0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168 -0.02168
SKIM01:IVTT  1>DAIV  S2IV  S3IV  WCIV  WBIV  WTIV  WMIV  PCIV  KCIV  PBIV  KBIV  PTIV  KTIV  PMIV  KMIV
COEF02:AUTO ACC 1>
SKIM02:AUTO ACC 1>
COEF03:TERM/OVTT 1>-0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420
SKIM03:TERM/OVTT 1>DATE S2TE S3TE WCOV WBOV WTOV WMOV PCOV KCOV PBOV KBOV PTOV KTOV PMOV KMOV
* LIMIT COEF 04 TO PURPOSE 1
COEF PURP04 >1
COEF04:COST INC1 1>-0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202 -0.00202
SKIM04:COST INC1 1>DACS S2CS S3CS WCCS WBCS WTCS WMCS PCCS KCCS PBCS KBCS PTCS KTCS PMCS KMCS
* LIMIT COEF 05 TO PURPOSE 2
COEF PURP05 >2
COEF05:COST INC2 1>-0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101 -0.00101
SKIM05:COST INC2 1>DACS S2CS S3CS WCCS WBCS WTCS WMCS PCCS KCCS PBCS KBCS PTCS KTCS PMCS KMCS
* LIMIT COEF 06 TO PURPOSE 3
COEF PURP06 >3
COEF06:COST INC3 1>-0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067 -0.00067
SKIM06:COST INC3 1>DACS S2CS S3CS WCCS WBCS WTCS WMCS PCCS KCCS PBCS KBCS PTCS KTCS PMCS KMCS

```

## Appendix D: AEMS Fortran Control Files

```

COEF PURP07      >4
* LIMIT COEF 07 TO PURPOSE 4
COEF07:COST INC4 1>-0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051 -0.00051
SKIM07:COST INC4 1>DACS S2CS S3CS WCCS WBCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS WPCS
COEF08:TRN XFERS 1> -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000
SKIM08:TRN XFERS 1> WCXF WBXF WTXF WMXF PCXF KCXF KBXF PTXF KTXF PMXF KMXF
COEF09:TRN BRDPEN 1> -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420 -0.05420
SKIM09:TRN BRDPEN 1> WCXP WBXP WTXP WMXP PCXP KCXP PBXP KBXP PTXP KTXP PMXP KMXF
*WALK WEIGHT
COEF10:TRN WLKWT 1> -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336 -0.04336
SKIM10:TRN WLKWT 1> WCWK WBWK WTWK WMWK PCWK KCWK PBWK KBWK PTWK KTWK PMWK KMWK

*SYNTAX TO LIMIT UTILITY ELEMENT TO A PARTICULAR WALK SEGMENT IN THIS EXAMPLE
* COEF 18 APPLIES ONLY TO WALK SEGMENT 1
*COEF WLKSEG18 >1

* ASSUMED MATRIX ORGANIZATION
* FILE 1 TRIP TABLE (SEPARATE FOR EACH PURPOSE)
* 1 INCOME 1 (HOME-BASED)/ALL NHB TRIPS
* 2 INCOME 2 (HOME-BASED)
* 3 INCOME 3 (HOME-BASED)
* 4 INCOME 4 (HOME-BASED)
*
* FILE 2 HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 SOV TIME (MIN)
* 2 SOV DIST (0.1 MILES)
* 3 SOV TOLL (2007 CENTS)
* 4 HOV2 TIME (MIN)
* 5 HOV2 DIST (0.1 MILES)
* 6 HOV2 TOLL (2007 CENTS)
* 7 HOV3+ TIME (MIN)
* 8 HOV3+ DIST (0.1 MILES)
* 9 HOV3+ TOLL (2007 CENTS)
*
* FILE 3=COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 4=BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 5=METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 6=BUS+METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)
* 2 WLK OTHER (.01 MIN) 16 PNR OTHER (.01 MIN) 34 KNR OTHER (.01 MIN)
* 3 WLK IWAIT (.01 MIN) 17 PNR IWAIT (.01 MIN) 35 KNR IWAIT (.01 MIN)
* 4 WLK XWAIT (.01 MIN) 18 PNR XWAIT (.01 MIN) 36 KNR XWAIT (.01 MIN)
* 5 WLK IVTT TOT (.01 MIN) 19 PNR IVTT TOT (.01 MIN) 37 KNR IVTT TOT (.01 MIN)
* 6 WLK IVTT CR (.01 MIN) 20 PNR IVTT CR (.01 MIN) 38 KNR IVTT CR (.01 MIN)
* 7 WLK IVTT XB (.01 MIN) 21 PNR IVTT XB (.01 MIN) 39 KNR IVTT XB (.01 MIN)
* 8 WLK IVTT MR (.01 MIN) 22 PNR IVTT MR (.01 MIN) 40 KNR IVTT MR (.01 MIN)
* 9 WLK IVTT NM (.01 MIN) 23 PNR IVTT NM (.01 MIN) 41 KNR IVTT NM (.01 MIN)
* 10 WLK IVTT NM2 (.01 MIN) 24 PNR IVTT NM2 (.01 MIN) 42 KNR IVTT NM2 (.01 MIN)
* 11 WLK IVTT LB (.01 MIN) 25 PNR IVTT LB (.01 MIN) 43 KNR IVTT LB (.01 MIN)
* 12 WLK #XFERS (NUMBER ) 26 PNR #XFERS (NUMBER ) 44 KNR #XFERS (NUMBER )
* 13 WLK COST (07CENTS) 27 PNR COST (07CENTS) 45 KNR COST (07CENTS)
* 14 WLK XPEN (.01 MIN) 28 PNR XPEN (.01 MIN) 46 KNR XPEN (.01 MIN)
* 29 PNR ACC TIME(.01 MIN) 47 KNR ACC TIME(.01 MIN)
* 30 PNR ACC DIST(.01 MIL) 48 KNR ACC DIST(.01 MIL)
* 31 PNR ACC COST(07CENTS)
* 32 PNR STA TERM(.01 MIN)
*
* FILE 8=ZDATA
* 1 HBW PARK COST (2007 CENTS)
* 2 HBS PARK COST (2007 CENTS)
* 3 HBO PARK COST (2007 CENTS)
* 4 NHB PARK COST (2007 CENTS)
* 5 TERMINAL TIME (HOME BASED) (MINUTES)
* 6 TERMINAL TIME (NON HOME BASED) (MINUTES)
* 7 ARC VIEW SHORT WALK PERCENT TO METRO
* 8 ARC VIEW LONG WALK PERCENT TO METRO
* 9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
* 10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT

```

## Appendix D: AEMS Fortran Control Files

---

```
* 11 ARC VIEW SHORT WALK PERCENT TO ALL OF TRANSIT
* 12 ARC VIEW LONG WALK PERCENT TO ALL OF TRANSIT
* 13 AREA TYPE
*   1=DC CORE
*   2=VA CORE
*   3=DC URBAN
*   4=MD URBAN
*   5=VA URBAN
*   6=MD OTHER
*   7=VA OTHER

* PARAMETERS
*=====
* AUTO OPERATING COSTS IN CENTS/mile
COMPUTE AUOP      >10
* AUTO OCCUPANCY FOR 3+ Reduced from 3.5 to 3.25 on 3/1/07 rm
COMPUTE OCC3     >3.25

* TERMINAL TIMES, USE i/j805 FOR HBW, HBS, AND HBO. USE i/j806 FOR NHB
* HBW/HBS/HBO
COMPUTE TERI     >i805
COMPUTE TERJ     >j805
* NHB
*COMPUTE TERI    >i806
*COMPUTE TERJ    >j806

* PARK COSTS, USE i/j801 802 803 804 FOR HBW, HBS, HBO, NHB RESPECTIVELY
* HBW
*COMPUTE PRKC    >j801/2.
* HBS
COMPUTE PRKC    >j802/2.
* HBO
* COMPUTE PRKC   >j803/2.
* NHB
* COMPUTE PRKC   >j804

* Percent of productions in long-walk area that are assumed to walk = 25% (i.e., 75% drive)
COMPUTE PCLM    >0.25
COMPUTE PCLT    >0.25
* PERCENT WALKS-METRO RAIL ONLY
COMPUTE PCMI    >(i807+PCLM*(i808-i807))/100.
COMPUTE PCMJ    >(j807+PCLM*(j808-j807))/100.
* PERCENT WALKS-PEAK
*COMPUTE PCTI    >(i809+PCLT*(i810-i809))/100.
*COMPUTE PCTJ    >(j809+PCLT*(j810-j809))/100.
* PERCENT WALKS-OFFPEAK
COMPUTE PCTI    >(i811+PCLT*(i812-i811))/100.
COMPUTE PCTJ    >(j811+PCLT*(j812-j811))/100.
COMPUTE PCMI    >MAX(PCMI,0)
COMPUTE PCMI    >MIN(PCMI,1)
COMPUTE PCMJ    >MAX(PCMJ,0)
COMPUTE PCMJ    >MIN(PCMJ,1)
COMPUTE PCTI    >MAX(PCTI,PCMI)
COMPUTE PCTI    >MIN(PCTI,1)
COMPUTE PCTJ    >MAX(PCTJ,PCMJ)
COMPUTE PCTJ    >MIN(PCTJ,1)
*
* DO TRIP SUBDIVISIONS
*
* HOME BASED ALTERNATIVES
COMPUTE TRP1    >m101
COMPUTE TRP2    >m102
COMPUTE TRP3    >m103
COMPUTE TRP4    >m104
* NON-HOME BASED
*COMPUTE TRP1    >0.25*m101
*COMPUTE TRP2    >0.25*m101
```

## Appendix D: AEMS Fortran Control Files

---

```
*COMPUTE TRP3      >0.25*m101
*COMPUTE TRP4      >0.25*m101
*
* BE SURE TO UPDATE THE IVTT COEFFICIENT IN FTA SECTION FOR EACH PURPOSE
*
*=====
*INITIALIZING ALL VARIABLES WITHIN IF STATEMENTS TO ZERO
COMPUTE DAIV      >0
COMPUTE DACS      >0
COMPUTE DATE      >0
COMPUTE S2IV      >0
COMPUTE S2CS      >0
COMPUTE S2TE      >0
COMPUTE S3IV      >0
COMPUTE S3CS      >0
COMPUTE S3TE      >0
COMPUTE WKIV      >0
COMPUTE WKOV      >0
COMPUTE WKXF      >0
COMPUTE WKCS      >0
COMPUTE WKXP      >0
COMPUTE WBIV      >0
COMPUTE WBOV      >0
COMPUTE WBXF      >0
COMPUTE WBCS      >0
COMPUTE WBXP      >0
COMPUTE WTIV      >0
COMPUTE WTOV      >0
COMPUTE WTXF      >0
COMPUTE WTCS      >0
COMPUTE WTXP      >0
COMPUTE WMIV      >0
COMPUTE WMOV      >0
COMPUTE WMXF      >0
COMPUTE WMCS      >0
COMPUTE WMXP      >0
COMPUTE PCIV      >0
COMPUTE PCAA      >0
COMPUTE PCOV      >0
COMPUTE PCXF      >0
COMPUTE PCCS      >0
COMPUTE PCXP      >0
COMPUTE PBIV      >0
COMPUTE PBAA      >0
COMPUTE PBOV      >0
COMPUTE PBXF      >0
COMPUTE PBCS      >0
COMPUTE PBXP      >0
COMPUTE PTIV      >0
COMPUTE PTAA      >0
COMPUTE PTOV      >0
COMPUTE PTXF      >0
COMPUTE PTCS      >0
COMPUTE PTXP      >0
COMPUTE PMIV      >0
COMPUTE PMAA      >0
COMPUTE PMOV      >0
COMPUTE PMXF      >0
COMPUTE PMCS      >0
COMPUTE PMXP      >0
COMPUTE KCIV      >0
COMPUTE KCAA      >0
COMPUTE KCOV      >0
COMPUTE KCXF      >0
COMPUTE KCCS      >0
COMPUTE KCXP      >0
COMPUTE KBIV      >0
COMPUTE KBAA      >0
```



## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE KBOV      >0
COMPUTE KBXF      >0
COMPUTE KBCS      >0
COMPUTE KBXP      >0
COMPUTE KTIV      >0
COMPUTE KTA A      >0
COMPUTE KTOV      >0
COMPUTE KTXF      >0
COMPUTE KTCS      >0
COMPUTE KTXP      >0
COMPUTE KMIV      >0
COMPUTE KMAA      >0
COMPUTE KMOV      >0
COMPUTE KMXF      >0
COMPUTE KMCS      >0
COMPUTE KMXP      >0

COMPUTE WCWK      >0
COMPUTE WBWK      >0
COMPUTE WTWK      >0
COMPUTE WMWK      >0
COMPUTE PCWK      >0
COMPUTE KCWK      >0
COMPUTE PBWK      >0
COMPUTE KBWK      >0
COMPUTE PTWK      >0
COMPUTE KTWK      >0
COMPUTE PMWK      >0
COMPUTE KMWK      >0

* SKIM VALUES, Divide distances by 10 to convert tenths of miles to whole miles
* DRIVE ALONE
COMPUTE           >IF (m201>0)
COMPUTE DAIV      >m201
COMPUTE DACS      >m202/10*AUOP+m203+PRKC
COMPUTE DATE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 2
COMPUTE           >IF (m204>0)
COMPUTE S2IV      >m204
COMPUTE S2CS      >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 3
COMPUTE           >IF (m207>0)
COMPUTE S3IV      >m207
COMPUTE S3CS      >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE      >TERI+TERJ
COMPUTE           >ENDIF

* Assign Intrazonal trips to Autos (mj11/04/05)
COMPUTE           >IF (P()=Q())
COMPUTE DAIV      >1
COMPUTE DACS      >m202/10*AUOP+m203+PRKC
COMPUTE DATE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 2
COMPUTE           >IF (P()=Q())
COMPUTE S2IV      >1
COMPUTE S2CS      >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 3
COMPUTE           >IF (P()=Q())
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE S3IV      >1
COMPUTE S3CS      >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE      >TERI+TERJ
COMPUTE           >ENDIF
```

\*End of Intrazonal trips

\* WALK COMMUTER RAIL

```
COMPUTE           >IF (m305>0)
COMPUTE WCIV      >m305/100.
COMPUTE WCOV      >(m303+m304)/100.
COMPUTE WCXF      >m312
COMPUTE WCCS      >m313
COMPUTE WCXP      >m314/100.
COMPUTE WCWK      >(m301+m302)/100.
COMPUTE           >ENDIF
```

\* WALK BUS

```
COMPUTE           >IF (m405>0)
COMPUTE WBIV      >m405/100.
COMPUTE WBOV      >(m403+m404)/100.
COMPUTE WBXF      >m412
COMPUTE WBXS      >m413
COMPUTE WBXP      >m414/100.
COMPUTE WBWK      >(m401+m402)/100.
COMPUTE           >ENDIF
```

\* WALK BUS/METRORAIL (TRANSIT)

```
COMPUTE           >IF (m605>0)
COMPUTE WTIV      >m605/100.
COMPUTE WTOV      >(m603+m604)/100.
COMPUTE WTXF      >m612
COMPUTE WTCS      >m613
COMPUTE WTXP      >m614/100.
COMPUTE WTWK      >(m601+m602)/100.
COMPUTE           >ENDIF
```

\* WALK METRORAIL

```
COMPUTE           >IF (m505>0)
COMPUTE WMIV      >m505/100.
COMPUTE WMOV      >(m503+m504)/100.
COMPUTE WMXF      >m512
COMPUTE WMCS      >m513
COMPUTE WMXP      >m514/100.
COMPUTE WMWK      >(m501+m502)/100.
COMPUTE           >ENDIF
```

\* PNR COMMUTER RAIL

```
COMPUTE           >IF (m319>0)
COMPUTE PCIV      >m319/100.
COMPUTE PCAA      >m329/100.
COMPUTE PCOV      >(m317+m318+m332)/100.
COMPUTE PCXF      >m326
COMPUTE PCCS      >m327+m331+m330/100*AUOP
COMPUTE PCXP      >m328/100.
COMPUTE PCWK      >(m315+m316)/100.
COMPUTE           >ENDIF
```

\* PNR BUS

```
COMPUTE           >IF (m419>0)
COMPUTE PBIV      >m419/100.
COMPUTE PBAA      >m429/100.
COMPUTE PBOV      >(m417+m418+m432)/100.
COMPUTE PBXF      >m426
COMPUTE PBXS      >m427+m431+m430/100*AUOP
COMPUTE PBXP      >m428/100.
COMPUTE PBWK      >(m415+m416)/100.
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE          >ENDIF

* PNR BUS/METRORAIL (TRANSIT)
COMPUTE          >IF (m619>0)
COMPUTE PTIV     >m619/100.
COMPUTE PTAA     >m629/100.
COMPUTE PTOV     >(m617+m618+m632)/100.
COMPUTE PTXF     >m626
COMPUTE PTCS     >m627+m631+m630/100*AUOP
COMPUTE PTXP     >m628/100.
COMPUTE PTWK     >(m615+m616)/100.
COMPUTE          >ENDIF

* PNR METRORAIL
COMPUTE          >IF (m519>0)
COMPUTE PMIV     >m519/100.
COMPUTE PMAA     >m529/100.
COMPUTE PMOV     >(m517+m518+m532)/100.
COMPUTE PMXF     >m526
COMPUTE PMCS     >m527+m531+m530/100*AUOP
COMPUTE PMXP     >m528/100.
COMPUTE PMWK     >(m515+m516)/100.
COMPUTE          >ENDIF

* KNR COMMUTER RAIL
COMPUTE          >IF (m319>0)
COMPUTE KCIV     >m319/100.
COMPUTE KCAA     >m329/100.
COMPUTE KCOV     >(m317+m318)/100.
COMPUTE KCXF     >m326
COMPUTE KCCS     >m327+m330/100*AUOP
COMPUTE KCXP     >m328/100.
COMPUTE KCWK     >(m315+m316)/100.
COMPUTE          >ENDIF

* KNR BUS
COMPUTE          >IF (m437>0)
COMPUTE KBIV     >m437/100.
COMPUTE KBAA     >m447/100.
COMPUTE KBOV     >(m435+m436)/100.
COMPUTE KBXF     >m444
COMPUTE KBCS     >m445+m448/100*AUOP
COMPUTE KBXP     >m446/100.
COMPUTE KBWK     >(m433+m434)/100.
COMPUTE          >ENDIF

* KNR BUS/METRORAIL (TRANSIT)
COMPUTE          >IF (m637>0)
COMPUTE KTIV     >m637/100.
COMPUTE KTAA     >m647/100.
COMPUTE KTOV     >(m635+m636)/100.
COMPUTE KTXF     >m644
COMPUTE KTCS     >m645+m648/100*AUOP
COMPUTE KTXP     >m646/100.
COMPUTE KTWK     >(m633+m634)/100.
COMPUTE          >ENDIF

* KNR METRORAIL
COMPUTE          >IF (m537>0)
COMPUTE KMIV     >m537/100.
COMPUTE KMAA     >m547/100.
COMPUTE KMOV     >(m535+m536)/100.
COMPUTE KMXF     >m544
COMPUTE KMCS     >m545+m548/100*AUOP
```

## Appendix D: AEMS Fortran Control Files

```

COMPUTE KMXF      >m546/100.
COMPUTE KMWK      >(m533+m534)/100.
COMPUTE           >ENDIF

```

```

*CONSTANTS BY CHOICE FOR EACH PURPOSE
*CHOICE          1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
PURP01 1INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 1INC 2    1>
PURP03 1INC 3    1>
PURP04 1INC 4    1>
PURP01 2INC 1    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP02 2INC 2    1> 2.000000 2.000000 2.000000 2.000000
PURP03 2INC 3    1>
PURP04 2INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 3INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 3INC 2    1>
PURP03 3INC 3    1>
PURP04 3INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 4INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 4INC 2    1>
PURP03 4INC 3    1>
PURP04 4INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 5INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 5INC 2    1>
PURP03 5INC 3    1>
PURP04 5INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 6INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 6INC 2    1>
PURP03 6INC 3    1>
PURP04 6INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 7INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 7INC 2    1>
PURP03 7INC 3    1>
PURP04 7INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 8INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 8INC 2    1>
PURP03 8INC 3    1>
PURP04 8INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP01 9INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP02 9INC 2    1>
PURP03 9INC 3    1>
PURP04 9INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0110INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0210INC 2    1>
PURP0310INC 3    1>
PURP0410INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0111INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0211INC 2    1>
PURP0311INC 3    1>
PURP0411INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0112INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0212INC 2    1>
PURP0312INC 3    1>
PURP0412INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0113INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0213INC 2    1>
PURP0313INC 3    1>
PURP0413INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0114INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0214INC 2    1>
PURP0314INC 3    1>
PURP0414INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0115INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0215INC 2    1>
PURP0315INC 3    1>
PURP0415INC 4    1> -2.000000 -2.000000 -2.000000 -2.000000
PURP0116INC 1    1> 2.000000 2.000000 2.000000 2.000000
PURP0216INC 2    1>

```

## Appendix D: AEMS Fortran Control Files

```

PURP0316INC 3 1>
PURP0416INC 4 1> -2.00000 -2.00000 -2.00000 -2.00000
PURP0117INC 1 1> 2.000000 2.000000 2.000000 2.000000
PURP0217INC 2 1>
PURP0317INC 3 1>
PURP0417INC 4 1>
PURP0118INC 1 1> -2.00000 -2.00000 -2.00000 -2.00000
PURP0218INC 2 1> 2.000000 2.000000 2.000000 2.000000
PURP0318INC 3 1>
PURP0418INC 4 1>
PURP0119INC 1 1> -2.00000 -2.00000 -2.00000 -2.00000
PURP0219INC 2 1> 2.000000 2.000000 2.000000 2.000000
PURP0319INC 3 1>
PURP0419INC 4 1> -2.00000 -2.00000 -2.00000 -2.00000
PURP0120INC 1 1> 2.000000 2.000000 2.000000 2.000000
PURP0220INC 2 1>
PURP0320INC 3 1>
PURP0420INC 4 1> -2.00000 -2.00000 -2.00000 -2.00000

```

```

TRIPIN01 >TRP1
TRIPIN02 >TRP2
TRIPIN03 >TRP3
TRIPIN04 >TRP4
TRIPIFACT01 >tfi1
TRIPIFACT02 >tfi2
TRIPIFACT03 >tfi3
TRIPIFACT04 >tfi4
COMPUTE tfi1 >1.0
COMPUTE tfi2 >1.0
COMPUTE tfi3 >1.0
COMPUTE tfi4 >1.0

```

```

*
*OUTPUT MATRICES AND OUTPUT FACTORS BY CHOICE FOR EACH PURPOSE
*CHOICE 1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
TRIPOUT01 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT01 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT02 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT02 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT03 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT03 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT04 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT04 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
**

```

```

**P AND A WALK PERCENTS BY CHOICE
*CHOICE 1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
WALK SEG CW 1 PCT 1>WSWM
WALK SEG CW 1 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 2 PCT 1>WSW1 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 2 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 3 PCT 1>WSW2 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 3 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 4 PCT 1>WSW3 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 4 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 5 PCT 1>WSM1 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 5 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 6 PCT 1>WSM2 Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 6 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG NT 7 PCT 1>WSNT Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG NT 7 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y
*SYNTAX OF COMMAND TO ADD A COMPONENT TO A SPECIFIC WALK SEGMENT IF DESIRED
*WALK SEG CW 1 COEF1> -0.04747 -0.04747 -0.04747 -0.04747 -0.04747
*WALK SEG CW 1 VAR 1> WTSS DTSS DISS WRSS DRSS DJSS
COMPUTE WSWM >PCMI*PCMJ
COMPUTE WSW1 >(PCTI-PCMI)*PCMJ
COMPUTE WSW2 >(PCTI-PCMI)*(PCTJ-PCMJ)
COMPUTE WSW3 >PCMI*(PCTJ-PCMJ)
COMPUTE WSM1 >(1-PCTI)*PCMJ

```

## Appendix D: AEMS Fortran Control Files

```

COMPUTE WSM2      >(1-PCTI)*(PCTJ-PCMJ)
COMPUTE WSN1     >1-WSWM-WSW1-WSW2-WSW3-WSM1-WSM2

*NEST DEFINITIONS BY CHOICE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
NEST 1,1=        1>Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 1,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 2,1=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 2,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 2,3=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 3,1=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 3,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 3,3=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 3,4=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 4,1=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 4,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 4,3=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 4,4=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 5,1=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 5,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 5,3=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 5,4=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 6,1=        1>Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 6,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 7,1=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
NEST 7,2=        1>          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y

IGRP DEFINITION  >i813
JGRP DEFINITION  >j813
* 1 DC CORE/URBAN-DC CORE
SEGMENT 1        > 1 1
SEGMENT 1        > 3 1
* 2 DC CORE/URBAN-VA CORE
SEGMENT 2        > 1 2
SEGMENT 2        > 3 2
* 3 DC CORE/URBAN-URBAN
SEGMENT 3        > 1 3
SEGMENT 3        > 3 3
SEGMENT 3        > 1 4
SEGMENT 3        > 3 4
SEGMENT 3        > 1 5
SEGMENT 3        > 3 5
* 4 DC CORE/URBAN-OTHER
SEGMENT 4        > 1 6
SEGMENT 4        > 3 6
SEGMENT 4        > 1 7
SEGMENT 4        > 3 7
* 5 MD URBAN-DC CORE
SEGMENT 5        > 4 1
* 6 MD URBAN-VA CORE
SEGMENT 6        > 4 2
* 7 MD URBAN-URBAN
SEGMENT 7        > 4 3
SEGMENT 7        > 4 4
SEGMENT 7        > 4 5
* 8 MD URBAN-OTHER
SEGMENT 8        > 4 6
SEGMENT 8        > 4 7
* 9 VA CORE/URBAN-DC CORE
SEGMENT 9        > 2 1
SEGMENT 9        > 5 1
*10 VA CORE/URBAN-VA CORE
SEGMENT 10       > 2 2
SEGMENT 10       > 5 2
*11 VA CORE/URBAN-URBAN
SEGMENT 11       > 2 3
SEGMENT 11       > 5 3
SEGMENT 11       > 2 4

```

## Appendix D: AEMS Fortran Control Files

```

SEGMENT 11      >  5  4
SEGMENT 11      >  2  5
SEGMENT 11      >  5  5
*12 VA CORE/URBAN-OTHER
SEGMENT 12      >  2  6
SEGMENT 12      >  5  6
SEGMENT 12      >  2  7
SEGMENT 12      >  5  7
*13 MD OTHER-DC CORE
SEGMENT 13      >  6  1
*14 MD OTHER-VA CORE
SEGMENT 14      >  6  2
*15 MD OTHER-URBAN
SEGMENT 15      >  6  3
SEGMENT 15      >  6  4
SEGMENT 15      >  6  5
*16 MD OTHER-OTHER
SEGMENT 16      >  6  6
SEGMENT 16      >  6  7
*17 VA OTHER-DC CORE
SEGMENT 17      >  7  1
*18 VA OTHER-VA CORE
SEGMENT 18      >  7  2
*19 VA OTHER-URBAN
SEGMENT 19      >  7  3
SEGMENT 19      >  7  4
SEGMENT 19      >  7  5
*20 VA OTHER-OTHER
SEGMENT 20      >  7  6
SEGMENT 20      >  7  7

* SEGMENT 1
NSTC 10 1GRND TOTAL>
NSTC 11 1AUTO      >  0.5  0.00000
NSTC 12 1TRANSIT  >  0.5 -1.80327
NSTC 20 1TOTAL TRN >
NSTC 21 1WALK ACC >  0.5  0.00000
NSTC 22 1PNR ACC  >  0.5 -1.74474
NSTC 23 1KNR ACC  >  0.5 -5.09216
NSTC 30 1WLK TRN
NSTC 31 1WLK CR   >  1.0  1.24196
NSTC 32 1WLK BUS  >  1.0  1.85744
NSTC 33 1WLK BU/MR >  1.0  1.90245
NSTC 34 1WLK METRO >  1.0  0.00000
NSTC 40 1PNR TRN
NSTC 41 1PNR CR   >  1.0 -2.65812
NSTC 42 1PNR BUS  >  1.0 -2.65812
NSTC 43 1PNR BU/MR >  1.0 -2.65812
NSTC 44 1PNR METRO >  1.0  0.00000
NSTC 50 1KNR TRN
NSTC 51 1KNR CR   >  1.0 -0.71535
NSTC 52 1KNR BUS  >  1.0 -0.71535
NSTC 53 1KNR BU/MR >  1.0 -0.71535
NSTC 54 1KNR METRO >  1.0  0.00000
NSTC 60 1AUTO
NSTC 61 1LOV      >  1.0  0.00000
NSTC 62 1HOV      >  0.5 -0.43617
NSTC 70 1HOV
NSTC 71 1HOV2     >  1.0  0.00000
NSTC 72 1HOV3+    >  1.0 -0.88412
* SEGMENT 2
NSTC 10 2GRND TOTAL>
NSTC 11 2AUTO      >  0.5  0.00000
NSTC 12 2TRANSIT  >  0.5 -2.41592
NSTC 20 2TOTAL TRN >
NSTC 21 2WALK ACC >  0.5  0.00000
NSTC 22 2PNR ACC  >  0.5 -2.31032
NSTC 23 2KNR ACC  >  0.5 -2.31032

```

## Appendix D: AEMS Fortran Control Files

```

NSTC 30 2WLK TRN
NSTC 31 2WLK CR > 1.0 -3.78181
NSTC 32 2WLK BUS > 1.0 -3.78181
NSTC 33 2WLK BU/MR > 1.0 -3.78181
NSTC 34 2WLK METRO > 1.0 0.00000
NSTC 40 2PNR TRN
NSTC 41 2PNR CR > 1.0 0.00001
NSTC 42 2PNR BUS > 1.0 0.00001
NSTC 43 2PNR BU/MR > 1.0 0.00001
NSTC 44 2PNR METRO > 1.0 0.00000
NSTC 50 2KNR TRN
NSTC 51 2KNR CR > 1.0 0.00001
NSTC 52 2KNR BUS > 1.0 0.00001
NSTC 53 2KNR BU/MR > 1.0 0.00001
NSTC 54 2KNR METRO > 1.0 0.00000
NSTC 60 2AUTO
NSTC 61 2LOV > 1.0 0.00000
NSTC 62 2HOV > 0.5 0.24928
NSTC 70 2HOV
NSTC 71 2HOV2 > 1.0 0.00000
NSTC 72 2HOV3+ > 1.0 -3.25465
* SEGMENT 3
NSTC 10 3GRND TOTAL>
NSTC 11 3AUTO > 0.5 0.00000
NSTC 12 3TRANSIT > 0.5 -1.15022
NSTC 20 3TOTAL TRN >
NSTC 21 3WALK ACC > 0.5 0.00000
NSTC 22 3PNR ACC > 0.5 -3.73394
NSTC 23 3KNR ACC > 0.5 -6.54862
NSTC 30 3WLK TRN
NSTC 31 3WLK CR > 1.0 -0.60068
NSTC 32 3WLK BUS > 1.0 -1.00465
NSTC 33 3WLK BU/MR > 1.0 -0.37273
NSTC 34 3WLK METRO > 1.0 0.00000
NSTC 40 3PNR TRN
NSTC 41 3PNR CR > 1.0 -3.80057
NSTC 42 3PNR BUS > 1.0 -3.80057
NSTC 43 3PNR BU/MR > 1.0 1.57080
NSTC 44 3PNR METRO > 1.0 0.00000
NSTC 50 3KNR TRN
NSTC 51 3KNR CR > 1.0 2.73961
NSTC 52 3KNR BUS > 1.0 3.91435
NSTC 53 3KNR BU/MR > 1.0 5.60339
NSTC 54 3KNR METRO > 1.0 0.00000
NSTC 60 3AUTO
NSTC 61 3LOV > 1.0 0.00000
NSTC 62 3HOV > 0.5 -0.72761
NSTC 70 3HOV
NSTC 71 3HOV2 > 1.0 0.00000
NSTC 72 3HOV3+ > 1.0 -0.70834
* SEGMENT 4
NSTC 10 4GRND TOTAL>
NSTC 11 4AUTO > 0.5 0.00000
NSTC 12 4TRANSIT > 0.5 -1.97278
NSTC 20 4TOTAL TRN >
NSTC 21 4WALK ACC > 0.5 0.00000
NSTC 22 4PNR ACC > 0.5 -2.22850
NSTC 23 4KNR ACC > 0.5 -2.22850
NSTC 30 4WLK TRN
NSTC 31 4WLK CR > 1.0 1.67310
NSTC 32 4WLK BUS > 1.0 1.77542
NSTC 33 4WLK BU/MR > 1.0 -1.59162
NSTC 34 4WLK METRO > 1.0 0.00000
NSTC 40 4PNR TRN
NSTC 41 4PNR CR > 1.0 0.00001
NSTC 42 4PNR BUS > 1.0 0.00001
NSTC 43 4PNR BU/MR > 1.0 0.00001
NSTC 44 4PNR METRO > 1.0 0.00000

```



## Appendix D: AEMS Fortran Control Files

---

```
NSTC 50 4KNR TRN
NSTC 51 4KNR CR > 1.0 0.00001
NSTC 52 4KNR BUS > 1.0 0.00001
NSTC 53 4KNR BU/MR > 1.0 0.00001
NSTC 54 4KNR METRO > 1.0 0.00000
NSTC 60 4AUTO
NSTC 61 4LOV > 1.0 0.00000
NSTC 62 4HOV > 0.5 -0.05598
NSTC 70 4HOV
NSTC 71 4HOV2 > 1.0 0.00000
NSTC 72 4HOV3+ > 1.0 -0.39536
* SEGMENT 5
NSTC 10 5GRND TOTAL>
NSTC 11 5AUTO > 0.5 0.00000
NSTC 12 5TRANSIT > 0.5 -1.30595
NSTC 20 5TOTAL TRN >
NSTC 21 5WALK ACC > 0.5 0.00000
NSTC 22 5PNR ACC > 0.5 -5.53080
NSTC 23 5KNR ACC > 0.5 -5.53080
NSTC 30 5WLK TRN
NSTC 31 5WLK CR > 1.0 0.17380
NSTC 32 5WLK BUS > 1.0 -1.64707
NSTC 33 5WLK BU/MR > 1.0 1.10993
NSTC 34 5WLK METRO > 1.0 0.00000
NSTC 40 5PNR TRN
NSTC 41 5PNR CR > 1.0 0.00001
NSTC 42 5PNR BUS > 1.0 0.00001
NSTC 43 5PNR BU/MR > 1.0 0.00001
NSTC 44 5PNR METRO > 1.0 0.00000
NSTC 50 5KNR TRN
NSTC 51 5KNR CR > 1.0 0.00001
NSTC 52 5KNR BUS > 1.0 0.00001
NSTC 53 5KNR BU/MR > 1.0 0.00001
NSTC 54 5KNR METRO > 1.0 0.00000
NSTC 60 5AUTO
NSTC 61 5LOV > 1.0 0.00000
NSTC 62 5HOV > 0.5 -3.60490
NSTC 70 5HOV
NSTC 71 5HOV2 > 1.0 0.00000
NSTC 72 5HOV3+ > 1.0 0.00001
* SEGMENT 6
NSTC 10 6GRND TOTAL>
NSTC 11 6AUTO > 0.5 0.00000
NSTC 12 6TRANSIT > 0.5 6.36265
NSTC 20 6TOTAL TRN >
NSTC 21 6WALK ACC > 0.5 0.00000
NSTC 22 6PNR ACC > 0.5 -9.17170
NSTC 23 6KNR ACC > 0.5 -9.17170
NSTC 30 6WLK TRN
NSTC 31 6WLK CR > 1.0 -11.81082
NSTC 32 6WLK BUS > 1.0 -11.81082
NSTC 33 6WLK BU/MR > 1.0 -11.81082
NSTC 34 6WLK METRO > 1.0 0.00000
NSTC 40 6PNR TRN
NSTC 41 6PNR CR > 1.0 0.00001
NSTC 42 6PNR BUS > 1.0 0.00001
NSTC 43 6PNR BU/MR > 1.0 0.00001
NSTC 44 6PNR METRO > 1.0 0.00000
NSTC 50 6KNR TRN
NSTC 51 6KNR CR > 1.0 0.00001
NSTC 52 6KNR BUS > 1.0 0.00001
NSTC 53 6KNR BU/MR > 1.0 0.00001
NSTC 54 6KNR METRO > 1.0 0.00000
NSTC 60 6AUTO
NSTC 61 6LOV > 1.0 0.00000
NSTC 62 6HOV > 0.5 0.00001
NSTC 70 6HOV
NSTC 71 6HOV2 > 1.0 0.00000
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 72 6HOV3+ > 1.0 0.00001
* SEGMENT 7
NSTC 10 7GRND TOTAL>
NSTC 11 7AUTO > 0.5 0.00000
NSTC 12 7TRANSIT > 0.5 -2.15330
NSTC 20 7TOTAL TRN >
NSTC 21 7WALK ACC > 0.5 0.00000
NSTC 22 7PNR ACC > 0.5 -4.48151
NSTC 23 7KNR ACC > 0.5 -4.48151
NSTC 30 7WLK TRN
NSTC 31 7WLK CR > 1.0 2.12350
NSTC 32 7WLK BUS > 1.0 2.30941
NSTC 33 7WLK BU/MR > 1.0 3.29707
NSTC 34 7WLK METRO > 1.0 0.00000
NSTC 40 7PNR TRN
NSTC 41 7PNR CR > 1.0 0.00001
NSTC 42 7PNR BUS > 1.0 0.00001
NSTC 43 7PNR BU/MR > 1.0 0.00001
NSTC 44 7PNR METRO > 1.0 0.00000
NSTC 50 7KNR TRN
NSTC 51 7KNR CR > 1.0 0.00001
NSTC 52 7KNR BUS > 1.0 0.00001
NSTC 53 7KNR BU/MR > 1.0 0.00001
NSTC 54 7KNR METRO > 1.0 0.00000
NSTC 60 7AUTO
NSTC 61 7LOV > 1.0 0.00000
NSTC 62 7HOV > 0.5 -0.55471
NSTC 70 7HOV
NSTC 71 7HOV2 > 1.0 0.00000
NSTC 72 7HOV3+ > 1.0 -0.24277
* SEGMENT 8
NSTC 10 8GRND TOTAL>
NSTC 11 8AUTO > 0.5 0.00000
NSTC 12 8TRANSIT > 0.5 -3.54985
NSTC 20 8TOTAL TRN >
NSTC 21 8WALK ACC > 0.5 0.00000
NSTC 22 8PNR ACC > 0.5 -1.59261
NSTC 23 8KNR ACC > 0.5 -1.99472
NSTC 30 8WLK TRN
NSTC 31 8WLK CR > 1.0 0.19071
NSTC 32 8WLK BUS > 1.0 2.56785
NSTC 33 8WLK BU/MR > 1.0 0.19071
NSTC 34 8WLK METRO > 1.0 0.00000
NSTC 40 8PNR TRN
NSTC 41 8PNR CR > 1.0 0.00001
NSTC 42 8PNR BUS > 1.0 0.00001
NSTC 43 8PNR BU/MR > 1.0 0.00001
NSTC 44 8PNR METRO > 1.0 0.00000
NSTC 50 8KNR TRN
NSTC 51 8KNR CR > 1.0 0.00001
NSTC 52 8KNR BUS > 1.0 9.08684
NSTC 53 8KNR BU/MR > 1.0 0.00001
NSTC 54 8KNR METRO > 1.0 0.00000
NSTC 60 8AUTO
NSTC 61 8LOV > 1.0 0.00000
NSTC 62 8HOV > 0.5 -0.41842
NSTC 70 8HOV
NSTC 71 8HOV2 > 1.0 0.00000
NSTC 72 8HOV3+ > 1.0 -1.08313
* SEGMENT 9
NSTC 10 9GRND TOTAL>
NSTC 11 9AUTO > 0.5 0.00000
NSTC 12 9TRANSIT > 0.5 0.08574
NSTC 20 9TOTAL TRN >
NSTC 21 9WALK ACC > 0.5 0.00000
NSTC 22 9PNR ACC > 0.5 -5.24596
NSTC 23 9KNR ACC > 0.5 -8.40773
NSTC 30 9WLK TRN

```

## Appendix D: AEMS Fortran Control Files

```

NSTC 31 9WLK CR > 1.0 -5.20419
NSTC 32 9WLK BUS > 1.0 -5.20419
NSTC 33 9WLK BU/MR > 1.0 -5.20419
NSTC 34 9WLK METRO > 1.0 0.00000
NSTC 40 9PNR TRN
NSTC 41 9PNR CR > 1.0 0.00001
NSTC 42 9PNR BUS > 1.0 0.00001
NSTC 43 9PNR BU/MR > 1.0 0.00001
NSTC 44 9PNR METRO > 1.0 0.00000
NSTC 50 9KNR TRN
NSTC 51 9KNR CR > 1.0 0.00001
NSTC 52 9KNR BUS > 1.0 0.00001
NSTC 53 9KNR BU/MR > 1.0 18.86512
NSTC 54 9KNR METRO > 1.0 0.00000
NSTC 60 9AUTO
NSTC 61 9LOV > 1.0 0.00000
NSTC 62 9HOV > 0.5 0.08149
NSTC 70 9HOV
NSTC 71 9HOV2 > 1.0 0.00000
NSTC 72 9HOV3+ > 1.0 -3.16162
* SEGMENT 10
NSTC 1010GRND TOTAL>
NSTC 1110AUTO > 0.5 0.00000
NSTC 1210TRANSIT > 0.5 -2.84014
NSTC 2010TOTAL TRN >
NSTC 2110WALK ACC > 0.5 0.00000
NSTC 2210PNR ACC > 0.5 -4.23791
NSTC 2310KNR ACC > 0.5 -4.23791
NSTC 3010WLK TRN
NSTC 3110WLK CR > 1.0 0.00001
NSTC 3210WLK BUS > 1.0 5.29676
NSTC 3310WLK BU/MR > 1.0 0.00001
NSTC 3410WLK METRO > 1.0 0.00000
NSTC 4010PNR TRN
NSTC 4110PNR CR > 1.0 0.00001
NSTC 4210PNR BUS > 1.0 0.00001
NSTC 4310PNR BU/MR > 1.0 0.00001
NSTC 4410PNR METRO > 1.0 0.00000
NSTC 5010KNR TRN
NSTC 5110KNR CR > 1.0 0.00001
NSTC 5210KNR BUS > 1.0 0.00001
NSTC 5310KNR BU/MR > 1.0 0.00001
NSTC 5410KNR METRO > 1.0 0.00000
NSTC 6010AUTO
NSTC 6110LOV > 1.0 0.00000
NSTC 6210HOV > 0.5 0.17498
NSTC 7010HOV
NSTC 7110HOV2 > 1.0 0.00000
NSTC 7210HOV3+ > 1.0 -0.65794
* SEGMENT 11
NSTC 1011GRND TOTAL>
NSTC 1111AUTO > 0.5 0.00000
NSTC 1211TRANSIT > 0.5 -0.95798
NSTC 2011TOTAL TRN >
NSTC 2111WALK ACC > 0.5 0.00000
NSTC 2211PNR ACC > 0.5 -5.98019
NSTC 2311KNR ACC > 0.5 -5.84700
NSTC 3011WLK TRN
NSTC 3111WLK CR > 1.0 -3.05688
NSTC 3211WLK BUS > 1.0 -4.45480
NSTC 3311WLK BU/MR > 1.0 -4.76612
NSTC 3411WLK METRO > 1.0 0.00000
NSTC 4011PNR TRN
NSTC 4111PNR CR > 1.0 -7.91940
NSTC 4211PNR BUS > 1.0 -7.91940
NSTC 4311PNR BU/MR > 1.0 -7.91940
NSTC 4411PNR METRO > 1.0 0.00000
NSTC 5011KNR TRN

```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5111KNR CR > 1.0 0.00001
NSTC 5211KNR BUS > 1.0 0.00001
NSTC 5311KNR BU/MR > 1.0 0.00001
NSTC 5411KNR METRO > 1.0 0.00000
NSTC 6011AUTO
NSTC 6111LOV > 1.0 0.00000
NSTC 6211HOV > 0.5 -0.42064
NSTC 7011HOV
NSTC 7111HOV2 > 1.0 0.00000
NSTC 7211HOV3+ > 1.0 -0.66973
* SEGMENT 12
NSTC 1012GRND TOTAL>
NSTC 1112AUTO > 0.5 0.00000
NSTC 1212TRANSIT > 0.5 -3.30571
NSTC 2012TOTAL TRN >
NSTC 2112WALK ACC > 0.5 0.00000
NSTC 2212PNR ACC > 0.5 -2.52477
NSTC 2312KNR ACC > 0.5 -2.52477
NSTC 3012WLK TRN
NSTC 3112WLK CR > 1.0 0.00001
NSTC 3212WLK BUS > 1.0 2.77467
NSTC 3312WLK BU/MR > 1.0 0.00001
NSTC 3412WLK METRO > 1.0 0.00000
NSTC 4012PNR TRN
NSTC 4112PNR CR > 1.0 0.00001
NSTC 4212PNR BUS > 1.0 0.00001
NSTC 4312PNR BU/MR > 1.0 0.00001
NSTC 4412PNR METRO > 1.0 0.00000
NSTC 5012KNR TRN
NSTC 5112KNR CR > 1.0 0.00001
NSTC 5212KNR BUS > 1.0 0.00001
NSTC 5312KNR BU/MR > 1.0 0.00001
NSTC 5412KNR METRO > 1.0 0.00000
NSTC 6012AUTO
NSTC 6112LOV > 1.0 0.00000
NSTC 6212HOV > 0.5 -0.54858
NSTC 7012HOV
NSTC 7112HOV2 > 1.0 0.00000
NSTC 7212HOV3+ > 1.0 -1.45975
* SEGMENT 13
NSTC 1013GRND TOTAL>
NSTC 1113AUTO > 0.5 0.00000
NSTC 1213TRANSIT > 0.5 -3.01056
NSTC 2013TOTAL TRN >
NSTC 2113WALK ACC > 0.5 0.00000
NSTC 2213PNR ACC > 0.5 0.07321
NSTC 2313KNR ACC > 0.5 -0.11357
NSTC 3013WLK TRN
NSTC 3113WLK CR > 1.0 0.00001
NSTC 3213WLK BUS > 1.0 4.84311
NSTC 3313WLK BU/MR > 1.0 4.54448
NSTC 3413WLK METRO > 1.0 0.00000
NSTC 4013PNR TRN
NSTC 4113PNR CR > 1.0 1.31320
NSTC 4213PNR BUS > 1.0 8.70887
NSTC 4313PNR BU/MR > 1.0 6.26240
NSTC 4413PNR METRO > 1.0 0.00000
NSTC 5013KNR TRN
NSTC 5113KNR CR > 1.0 1.03528
NSTC 5213KNR BUS > 1.0 1.03528
NSTC 5313KNR BU/MR > 1.0 5.83031
NSTC 5413KNR METRO > 1.0 0.00000
NSTC 6013AUTO
NSTC 6113LOV > 1.0 0.00000
NSTC 6213HOV > 0.5 -1.01997
NSTC 7013HOV
NSTC 7113HOV2 > 1.0 0.00000
NSTC 7213HOV3+ > 1.0 -3.24676
```

## Appendix D: AEMS Fortran Control Files

---

```
* SEGMENT 14
NSTC 1014GRND TOTAL>
NSTC 1114AUTO > 0.5 0.00000
NSTC 1214TRANSIT > 0.5 -2.36722
NSTC 2014TOTAL TRN >
NSTC 2114WALK ACC > 0.5 0.00000
NSTC 2214PNR ACC > 0.5 -3.95979
NSTC 2314KNR ACC > 0.5 -3.95979
NSTC 3014WLK TRN
NSTC 3114WLK CR > 1.0 0.00001
NSTC 3214WLK BUS > 1.0 0.00001
NSTC 3314WLK BU/MR > 1.0 6.64989
NSTC 3414WLK METRO > 1.0 0.00000
NSTC 4014PNR TRN
NSTC 4114PNR CR > 1.0 0.00001
NSTC 4214PNR BUS > 1.0 0.00001
NSTC 4314PNR BU/MR > 1.0 0.00001
NSTC 4414PNR METRO > 1.0 0.00000
NSTC 5014KNR TRN
NSTC 5114KNR CR > 1.0 0.00001
NSTC 5214KNR BUS > 1.0 0.00001
NSTC 5314KNR BU/MR > 1.0 0.00001
NSTC 5414KNR METRO > 1.0 0.00000
NSTC 6014AUTO
NSTC 6114LOV > 1.0 0.00000
NSTC 6214HOV > 0.5 1.56209
NSTC 7014HOV
NSTC 7114HOV2 > 1.0 0.00000
NSTC 7214HOV3+ > 1.0 -0.97900
* SEGMENT 15
NSTC 1015GRND TOTAL>
NSTC 1115AUTO > 0.5 0.00000
NSTC 1215TRANSIT > 0.5 -0.88046
NSTC 2015TOTAL TRN >
NSTC 2115WALK ACC > 0.5 0.00000
NSTC 2215PNR ACC > 0.5 -5.70853
NSTC 2315KNR ACC > 0.5 -7.82179
NSTC 3015WLK TRN
NSTC 3115WLK CR > 1.0 -0.94343
NSTC 3215WLK BUS > 1.0 -0.74881
NSTC 3315WLK BU/MR > 1.0 -3.07796
NSTC 3415WLK METRO > 1.0 0.00000
NSTC 4015PNR TRN
NSTC 4115PNR CR > 1.0 -3.19529
NSTC 4215PNR BUS > 1.0 -3.19529
NSTC 4315PNR BU/MR > 1.0 -3.19529
NSTC 4415PNR METRO > 1.0 0.00000
NSTC 5015KNR TRN
NSTC 5115KNR CR > 1.0 -0.42602
NSTC 5215KNR BUS > 1.0 -0.42602
NSTC 5315KNR BU/MR > 1.0 5.06489
NSTC 5415KNR METRO > 1.0 0.00000
NSTC 6015AUTO
NSTC 6115LOV > 1.0 0.00000
NSTC 6215HOV > 0.5 0.06359
NSTC 7015HOV
NSTC 7115HOV2 > 1.0 0.00000
NSTC 7215HOV3+ > 1.0 -0.45779
* SEGMENT 16
NSTC 1016GRND TOTAL>
NSTC 1116AUTO > 0.5 0.00000
NSTC 1216TRANSIT > 0.5 -3.08103
NSTC 2016TOTAL TRN >
NSTC 2116WALK ACC > 0.5 0.00000
NSTC 2216PNR ACC > 0.5 -13.18982
NSTC 2316KNR ACC > 0.5 -7.25362
NSTC 3016WLK TRN
NSTC 3116WLK CR > 1.0 5.37997
```

## Appendix D: AEMS Fortran Control Files

---

NSTC 3216WLK BUS >	1.0	5.63975
NSTC 3316WLK BU/MR >	1.0	1.00530
NSTC 3416WLK METRO >	1.0	0.00000
NSTC 4016PNR TRN		
NSTC 4116PNR CR >	1.0	0.00001
NSTC 4216PNR BUS >	1.0	12.70799
NSTC 4316PNR BU/MR >	1.0	0.00001
NSTC 4416PNR METRO >	1.0	0.00000
NSTC 5016KNR TRN		
NSTC 5116KNR CR >	1.0	0.00001
NSTC 5216KNR BUS >	1.0	7.40919
NSTC 5316KNR BU/MR >	1.0	0.00001
NSTC 5416KNR METRO >	1.0	0.00000
NSTC 6016AUTO		
NSTC 6116LOV >	1.0	0.00000
NSTC 6216HOV >	0.5	-0.20082
NSTC 7016HOV		
NSTC 7116HOV2 >	1.0	0.00000
NSTC 7216HOV3+ >	1.0	-0.19477
* SEGMENT 17		
NSTC 1017GRND TOTAL>		
NSTC 1117AUTO >	0.5	0.00000
NSTC 1217TRANSIT >	0.5	2.00248
NSTC 2017TOTAL TRN >		
NSTC 2117WALK ACC >	0.5	0.00000
NSTC 2217PNR ACC >	0.5	-10.21089
NSTC 2317KNR ACC >	0.5	-11.69422
NSTC 3017WLK TRN		
NSTC 3117WLK CR >	1.0	-10.13598
NSTC 3217WLK BUS >	1.0	-7.39427
NSTC 3317WLK BU/MR >	1.0	-10.13598
NSTC 3417WLK METRO >	1.0	0.00000
NSTC 4017PNR TRN		
NSTC 4117PNR CR >	1.0	2.43073
NSTC 4217PNR BUS >	1.0	2.43073
NSTC 4317PNR BU/MR >	1.0	11.73294
NSTC 4417PNR METRO >	1.0	0.00000
NSTC 5017KNR TRN		
NSTC 5117KNR CR >	1.0	0.00001
NSTC 5217KNR BUS >	1.0	0.00001
NSTC 5317KNR BU/MR >	1.0	0.00001
NSTC 5417KNR METRO >	1.0	0.00000
NSTC 6017AUTO		
NSTC 6117LOV >	1.0	0.00000
NSTC 6217HOV >	0.5	-0.31514
NSTC 7017HOV		
NSTC 7117HOV2 >	1.0	0.00000
NSTC 7217HOV3+ >	1.0	-0.09341
* SEGMENT 18		
NSTC 1018GRND TOTAL>		
NSTC 1118AUTO >	0.5	0.00000
NSTC 1218TRANSIT >	0.5	-13.69224
NSTC 2018TOTAL TRN >		
NSTC 2118WALK ACC >	0.5	0.00000
NSTC 2218PNR ACC >	0.5	-15.08574
NSTC 2318KNR ACC >	0.5	-15.16352
NSTC 3018WLK TRN		
NSTC 3118WLK CR >	1.0	0.00001
NSTC 3218WLK BUS >	1.0	26.85224
NSTC 3318WLK BU/MR >	1.0	0.00001
NSTC 3418WLK METRO >	1.0	0.00000
NSTC 4018PNR TRN		
NSTC 4118PNR CR >	1.0	0.00001
NSTC 4218PNR BUS >	1.0	17.09672
NSTC 4318PNR BU/MR >	1.0	0.00001
NSTC 4418PNR METRO >	1.0	0.00000
NSTC 5018KNR TRN		
NSTC 5118KNR CR >	1.0	0.00001

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5218KNR BUS > 1.0 0.00001
NSTC 5318KNR BU/MR > 1.0 0.00001
NSTC 5418KNR METRO > 1.0 0.00000
NSTC 6018AUTO
NSTC 6118LOV > 1.0 0.00000
NSTC 6218HOV > 0.5 -0.16396
NSTC 7018HOV
NSTC 7118HOV2 > 1.0 0.00000
NSTC 7218HOV3+ > 1.0 0.21786
* SEGMENT 19
NSTC 1019GRND TOTAL>
NSTC 1119AUTO > 0.5 0.00000
NSTC 1219TRANSIT > 0.5 -0.77879
NSTC 2019TOTAL TRN >
NSTC 2119WALK ACC > 0.5 0.00000
NSTC 2219PNR ACC > 0.5 -9.51379
NSTC 2319KNR ACC > 0.5 -10.40097
NSTC 3019WLK TRN
NSTC 3119WLK CR > 1.0 -4.33581
NSTC 3219WLK BUS > 1.0 -5.38760
NSTC 3319WLK BU/MR > 1.0 -3.99881
NSTC 3419WLK METRO > 1.0 0.00000
NSTC 4019PNR TRN
NSTC 4119PNR CR > 1.0 -5.48840
NSTC 4219PNR BUS > 1.0 -5.48840
NSTC 4319PNR BU/MR > 1.0 -5.48840
NSTC 4419PNR METRO > 1.0 0.00000
NSTC 5019KNR TRN
NSTC 5119KNR CR > 1.0 0.00001
NSTC 5219KNR BUS > 1.0 0.00001
NSTC 5319KNR BU/MR > 1.0 0.00001
NSTC 5419KNR METRO > 1.0 0.00000
NSTC 6019AUTO
NSTC 6119LOV > 1.0 0.00000
NSTC 6219HOV > 0.5 -0.31423
NSTC 7019HOV
NSTC 7119HOV2 > 1.0 0.00000
NSTC 7219HOV3+ > 1.0 -1.69031
* SEGMENT 20
NSTC 1020GRND TOTAL>
NSTC 1120AUTO > 0.5 0.00000
NSTC 1220TRANSIT > 0.5 -2.22791
NSTC 2020TOTAL TRN >
NSTC 2120WALK ACC > 0.5 0.00000
NSTC 2220PNR ACC > 0.5 -93.07792
NSTC 2320KNR ACC > 0.5 -97.36594
NSTC 3020WLK TRN
NSTC 3120WLK CR > 1.0 -0.68993
NSTC 3220WLK BUS > 1.0 0.97482
NSTC 3320WLK BU/MR > 1.0 -53.20142
NSTC 3420WLK METRO > 1.0 0.00000
NSTC 4020PNR TRN
NSTC 4120PNR CR > 1.0 0.00001
NSTC 4220PNR BUS > 1.0 20.94805
NSTC 4320PNR BU/MR > 1.0 0.00001
NSTC 4420PNR METRO > 1.0 0.00000
NSTC 5020KNR TRN
NSTC 5120KNR CR > 1.0 0.00001
NSTC 5220KNR BUS > 1.0 23.11707
NSTC 5320KNR BU/MR > 1.0 0.00001
NSTC 5420KNR METRO > 1.0 0.00000
NSTC 6020AUTO
NSTC 6120LOV > 1.0 0.00000
NSTC 6220HOV > 0.5 0.05322
NSTC 7020HOV
NSTC 7120HOV2 > 1.0 0.00000
NSTC 7220HOV3+ > 1.0 -0.12852
```

## Appendix D: AEMS Fortran Control Files

---

*DOWNTOWN=8		
*SELI	>	8
*UNION STATION=64		
*SELI	>	64
* =122		
*SELI	>	122
*BETHESDA=345		
*SELI	>	345
*SILVER SPRING=362		
*SELI	>	362
*N.SILVER SPRING=464		
*SELI	>	464
* =475		
*SELI	>	475
*SHADY GROVE RD=578		
*SELI	>	578
* =787		
*SELI	>	787
*ANDREWS AFB=829		
*SELI	>	829
*NEW CARROLTON=927		
*SELI	>	927
*BRISTOL=972		
*SELI	>	972
*FREDERICK=1043		
*SELI	>	1043
*JESSUP=1080		
*SELI	>	1080
*SCAGGSVILLE=1091		
*SELI	>	1091
*WALDORF=1216		
*SELI	>	1216
*PENTAGON=1231		
*SELI	>	1231
*ROSSLYN=1236		
*SELI	>	1236
*ALEXANDRIA=1337		
*SELI	>	1337
* =1455		
*SELI	>	1455
*SPRINGFIELD=1502		
*SELI	>	1502
* =1511		
*SELI	>	1511
*TYSONS CRNR=1537		
*SELI	>	1537



## Appendix D: AEMS Fortran Control Files

---

```
*FT BELVOIR=1554
*SELI      >    1554

*VIENNA=1619
*SELI      >    1619

*DULES AP=1698
*SELI      >    1698

*RESTON=1716
*SELI      >    1716

*LEESBURG=1842
*SELI      >    1842

*BRUNSWICK=1863
*SELI      >    1863

*DALE CITY=1942
*SELI      >    1942

*MANASSAS=1967
*SELI      >    1967

*SPOTSYLVANIA=2110
*SELI      >    2110

* =2055
*SELI      >    2055

*SELJ      >      8
*SELJ      >     63
*SELJ      >     64
*SELJ      >     77
*SELJ      >    100
*SELJ      >    344
*SELJ      >    345
*SELJ      >    362
*SELJ      >   1231
*SELJ      >   1236
*SELJ      >   1265
*SELJ      >   1337
*SELJ      >   1537
*SELI      >    523
*SELJ      >      9

TRACE      >      0
* OUTPUT % >
*PROCSEL   >
PRINT MS   >HBS_NL_MC.PRN
INPUT PRINT FILE >HBS_NL_MC.PRN
INPUT GOALS >HBS_NL_MC.GOL
INFILE 1   >hbs_income.ptt
INFILE 2   >hwyop.skm
INFILE 3   >TRNOP_CR.SKM
INFILE 4   >TRNOP_AB.SKM
INFILE 5   >TRNOP_MR.SKM
INFILE 6   >TRNOP_BM.SKM
ZINFILE 8  >ZONEV2.A2F
OUTFILE 9  >HBS_NL_MC.MTT

* FTA USER BENEFITS SPECIFICATIONS
*FTA RESULTS FILE >HBS_NL_MC.BEN
FTA TRANSIT COEFF >-0.02168
FTA AUTO COEFF >-0.02168
FTA PURPOSE NAME >HBO
FTA PERIOD NAME >ALLDAY
```

## Appendix D: AEMS Fortran Control Files

```

FTA ALTER. NAME      >CALIB
*CHOICE              1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR      WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR      KNR-BU/MR      PNR-MR      KNR-MR
FTA AUTO NEST        >              1              1
FTA MOTORIZED?       1>Y              Y              Y              Y              Y              Y              Y              Y              Y              Y              Y              Y              Y              Y
FTA TRANSIT?         1>

```

### 3 hbw\_nl\_mc.ctf

```

HBW AM NESTED LOGIT MC - #DATE: 9/17/2011 #VER: 21
CHOICE              1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR      WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR      KNR-BU/MR      PNR-MR      KNR-MR
*
*
*LOGIT COEFFICIENTS BY CHOICE FOR EACH SKIM (NO INPUT SKIM IS
*EQUIVALENT TO A CONSTANT)
*CHOICE              1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR      WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR      KNR-BU/MR      PNR-MR      KNR-MR
COEF01:IVTT         1>-0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128 -0.02128
SKIM01:IVTT         1>DAIV      S2IV      S3IV      WCIV      WBIV      WTIV      WMIV      PCIV      KCIV      PBIV      KBIV      PTIV      KTIV      PMIV      KMIV
COEF02:AUTO ACC     1>
SKIM02:AUTO ACC     1>
COEF03:TERM/OVTT    1>-0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320 -0.05320
SKIM03:TERM/OVTT    1>DATE      S2TE      S3TE      WCOV      WBOV      WTOV      WMOV      PCOV      KCOV      PBOV      KBOV      PTOV      KTOV      PMOV      KMOV
* LIMIT COEF 04 TO PURPOSE 1
COEF PURP04         >1
COEF04:COST INC1    1>-0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185 -0.00185
SKIM04:COST INC1    1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 05 TO PURPOSE 2
COEF PURP05         >2
COEF05:COST INC2    1>-0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093 -0.00093
SKIM05:COST INC2    1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 06 TO PURPOSE 3
COEF PURP06         >3
COEF06:COST INC3    1>-0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062 -0.00062
SKIM06:COST INC3    1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF PURP07         >4
* LIMIT COEF 07 TO PURPOSE 4
COEF07:COST INC4    1>-0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046 -0.00046
SKIM07:COST INC4    1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF08:TRN XFERS    1>
SKIM08:TRN XFERS    1>
COEF09:TRN BRDPEN  1>
SKIM09:TRN BRDPEN  1>
*WALK WEIGHT
COEF10:TRN WLKWT    1>
SKIM10:TRN WLKWT    1>
*SYNTAX TO LIMIT UTILITY ELEMENT TO A PARTICULAR WALK SEGMENT IN THIS EXAMPLE
* COEF 18 APPLIES ONLY TO WALK SEGMENT 1
*COEF WLKSEG18      >1
*
* ASSUMED MATRIX ORGANIZATION
* FILE 1 TRIP TABLE (SEPARATE FOR EACH PURPOSE)
* 1 INCOME 1 (HOME-BASED)/ALL NHB TRIPS
* 2 INCOME 2 (HOME-BASED)
* 3 INCOME 3 (HOME-BASED)
* 4 INCOME 4 (HOME-BASED)
*
* FILE 2 HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 SOV TIME (MIN)
* 2 SOV DIST (0.1 MILES)
* 3 SOV TOLL (2007 CENTS)
* 4 HOV2 TIME (MIN)
* 5 HOV2 DIST (0.1 MILES)
* 6 HOV2 TOLL (2007 CENTS)
* 7 HOV3+ TIME (MIN)
* 8 HOV3+ DIST (0.1 MILES)

```

## Appendix D: AEMS Fortran Control Files

```

* 9 HOV3+ TOLL (2007 CENTS)
*
* FILE 3=COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 4=BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 5=METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 6=BUS+METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)
* 2 WLK OTHER (.01 MIN) 16 PNR OTHER (.01 MIN) 34 KNR OTHER (.01 MIN)
* 3 WLK IWAIT (.01 MIN) 17 PNR IWAIT (.01 MIN) 35 KNR IWAIT (.01 MIN)
* 4 WLK XWAIT (.01 MIN) 18 PNR XWAIT (.01 MIN) 36 KNR XWAIT (.01 MIN)
* 5 WLK IVTT TOT(.01 MIN) 19 PNR IVTT TOT(.01 MIN) 37 KNR IVTT TOT(.01 MIN)
* 6 WLK IVTT CR (.01 MIN) 20 PNR IVTT CR (.01 MIN) 38 KNR IVTT CR (.01 MIN)
* 7 WLK IVTT XB (.01 MIN) 21 PNR IVTT XB (.01 MIN) 39 KNR IVTT XB (.01 MIN)
* 8 WLK IVTT MR (.01 MIN) 22 PNR IVTT MR (.01 MIN) 40 KNR IVTT MR (.01 MIN)
* 9 WLK IVTT NM (.01 MIN) 23 PNR IVTT NM (.01 MIN) 41 KNR IVTT NM (.01 MIN)
* 10 WLK IVTT NM2(.01 MIN) 24 PNR IVTT NM2(.01 MIN) 42 KNR IVTT NM2(.01 MIN)
* 11 WLK IVTT LB (.01 MIN) 25 PNR IVTT LB (.01 MIN) 43 KNR IVTT LB (.01 MIN)
* 12 WLK #XFERS (NUMBER ) 26 PNR #XFERS (NUMBER ) 44 KNR #XFERS (NUMBER )
* 13 WLK COST (.07CENTS) 27 PNR COST (.07CENTS) 45 KNR COST (.07CENTS)
* 14 WLK XPEN (.01 MIN) 28 PNR XPEN (.01 MIN) 46 KNR XPEN (.01 MIN)
* 29 PNR ACC TIME(.01 MIN) 47 KNR ACC TIME(.01 MIN)
* 30 PNR ACC DIST(.01 MIL) 48 KNR ACC DIST(.01 MIL)
* 31 PNR ACC COST(.07CENTS)
* 32 PNR STA TERM(.01 MIN)
*
* FILE 8=ZDATA
* 1 HBW PARK COST (2007 CENTS)
* 2 HBS PARK COST (2007 CENTS)
* 3 HBO PARK COST (2007 CENTS)
* 4 NHB PARK COST (2007 CENTS)
* 5 TERMINAL TIME (HOME BASED) (MINUTES)
* 6 TERMINAL TIME (NON HOME BASED) (MINUTES)
* 7 ARC VIEW SHORT WALK PERCENT TO METRO
* 8 ARC VIEW LONG WALK PERCENT TO METRO
* 9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
* 10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT
* 11 ARC VIEW SHORT WALK PERCENT TO ALL OF TRANSIT
* 12 ARC VIEW LONG WALK PERCENT TO ALL OF TRANSIT
* 13 AREA TYPE
* 1=DC CORE
* 2=VA CORE
* 3=DC URBAN
* 4=MD URBAN
* 5=VA URBAN
* 6=MD OTHER
* 7=VA OTHER

* PARAMETERS
*=====
* AUTO OPERATING COSTS IN CENTS/mile
COMPUTE AUOP >10
* AUTO OCCUPANCY FOR 3+
COMPUTE OCC3 >3.5

* TERMINAL TIMES, USE i/j805 FOR HBW, HBS, AND HBO. USE i/j806 FOR NHB
* HBW/HBS/HBO
COMPUTE TERI >i805
COMPUTE TERJ >j805
* NHB
*COMPUTE TERI >i806
*COMPUTE TERJ >j806

* PARK COSTS, USE i/j801 802 803 804 FOR HBW, HBS, HBO, NHB RESPECTIVELY
* HBW
COMPUTE PRKC >j801/2.
* HBS
* COMPUTE PRKC >j802/2.

```

## Appendix D: AEMS Fortran Control Files

---

```
* HBO
* COMPUTE PRKC      >j803/2.
* NHB
* COMPUTE PRKC      >j804

* Percent of productions in long-walk area that are assumed to walk = 25% (i.e., 75% drive)
COMPUTE PCLM      >0.25
COMPUTE PCLT      >0.25
* PERCENT WALKS-METRO RAIL ONLY
COMPUTE PCMI      >(i807+PCLM*(i808-i807))/100.
COMPUTE PCMJ      >(j807+PCLM*(j808-j807))/100.
* PERCENT WALKS-PEAK
COMPUTE PCTI      >(i809+PCLT*(i810-i809))/100.
COMPUTE PCTJ      >(j809+PCLT*(j810-j809))/100.
* PERCENT WALKS-OFFPEAK
*COMPUTE PCTI      >(i811+PCLT*(i812-i811))/100.
*COMPUTE PCTJ      >(j811+PCLT*(j812-j811))/100.
COMPUTE PCMI      >MAX(PCMI,0)
COMPUTE PCMI      >MIN(PCMI,1)
COMPUTE PCMJ      >MAX(PCMJ,0)
COMPUTE PCMJ      >MIN(PCMJ,1)
COMPUTE PCTI      >MAX(PCTI,PCMI)
COMPUTE PCTI      >MIN(PCTI,1)
COMPUTE PCTJ      >MAX(PCTJ,PCMJ)
COMPUTE PCTJ      >MIN(PCTJ,1)
*
* DO TRIP SUBDIVISIONS
*
* HOME BASED ALTERNATIVES
COMPUTE TRP1      >m101
COMPUTE TRP2      >m102
COMPUTE TRP3      >m103
COMPUTE TRP4      >m104
* NON-HOME BASED
*COMPUTE TRP1      >0.25*m101
*COMPUTE TRP2      >0.25*m101
*COMPUTE TRP3      >0.25*m101
*COMPUTE TRP4      >0.25*m101
*
* BE SURE TO UPDATE THE IVTT COEFFICIENT IN FTA SECTION FOR EACH PURPOSE
*
*=====

*INITIALIZING ALL VARIABLES WITHIN IF STATEMENTS TO ZERO
COMPUTE DAIV      >0
COMPUTE DACS      >0
COMPUTE DATE      >0
COMPUTE S2IV      >0
COMPUTE S2CS      >0
COMPUTE S2TE      >0
COMPUTE S3IV      >0
COMPUTE S3CS      >0
COMPUTE S3TE      >0
COMPUTE WKIV      >0
COMPUTE WKOV      >0
COMPUTE WKXF      >0
COMPUTE WKCS      >0
COMPUTE WKXP      >0
COMPUTE WBIV      >0
COMPUTE WBOV      >0
COMPUTE WBXF      >0
COMPUTE WBCS      >0
COMPUTE WBXP      >0
COMPUTE WTIV      >0
COMPUTE WTOV      >0
COMPUTE WTXF      >0
COMPUTE WTCS      >0
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE WTXF >0
COMPUTE WMIV >0
COMPUTE WMOV >0
COMPUTE WMXF >0
COMPUTE WMCS >0
COMPUTE WMXP >0
COMPUTE PCIV >0
COMPUTE PCAA >0
COMPUTE PCOV >0
COMPUTE PCXF >0
COMPUTE PCCS >0
COMPUTE PCXP >0
COMPUTE PBIV >0
COMPUTE PBAA >0
COMPUTE PBOV >0
COMPUTE PBXF >0
COMPUTE PBCS >0
COMPUTE PBXP >0
COMPUTE PTIV >0
COMPUTE PTAA >0
COMPUTE PTOV >0
COMPUTE PTXF >0
COMPUTE PTCS >0
COMPUTE PTXP >0
COMPUTE PMIV >0
COMPUTE PMAA >0
COMPUTE PMOV >0
COMPUTE PMXF >0
COMPUTE PMCS >0
COMPUTE PMXP >0
COMPUTE KCIV >0
COMPUTE KCAA >0
COMPUTE KCOV >0
COMPUTE KCXF >0
COMPUTE KCCS >0
COMPUTE KCXP >0
COMPUTE KBIV >0
COMPUTE KBAA >0
COMPUTE KBOV >0
COMPUTE KBXF >0
COMPUTE KBCS >0
COMPUTE KBXP >0
COMPUTE KTIV >0
COMPUTE KTAA >0
COMPUTE KTOV >0
COMPUTE KTXF >0
COMPUTE KTCS >0
COMPUTE KTXP >0
COMPUTE KMIV >0
COMPUTE KMAA >0
COMPUTE KMOV >0
COMPUTE KMXF >0
COMPUTE KMCS >0
COMPUTE KMXF >0

COMPUTE WCWK >0
COMPUTE WBWK >0
COMPUTE WTWK >0
COMPUTE WMWK >0
COMPUTE PCWK >0
COMPUTE KCWK >0
COMPUTE PBWK >0
COMPUTE KBWK >0
COMPUTE PTWK >0
COMPUTE KTWK >0
COMPUTE PMWK >0
COMPUTE KMWK >0
```

## Appendix D: AEMS Fortran Control Files

---

```
* SKIM VALUES, Divide distances by 10 to convert tenths of miles to whole miles
* DRIVE ALONE
COMPUTE          >IF (m201>0)
COMPUTE DAIV     >m201
COMPUTE DACS     >m202/10*AUOP+m203+PRKC
COMPUTE DATE     >TERI+TERJ
COMPUTE          >ENDIF

* SHARED RIDE 2
COMPUTE          >IF (m204>0)
COMPUTE S2IV     >m204
COMPUTE S2CS     >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE     >TERI+TERJ
COMPUTE          >ENDIF

* SHARED RIDE 3
COMPUTE          >IF (m207>0)
COMPUTE S3IV     >m207
COMPUTE S3CS     >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE     >TERI+TERJ
COMPUTE          >ENDIF

* Assign Intrazonal trips to Autos (mj11/04/05)
COMPUTE          >IF (P()=Q())
COMPUTE DAIV     >1
COMPUTE DACS     >m202/10*AUOP+m203+PRKC
COMPUTE DATE     >TERI+TERJ
COMPUTE          >ENDIF

* SHARED RIDE 2
COMPUTE          >IF (P()=Q())
COMPUTE S2IV     >1
COMPUTE S2CS     >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE     >TERI+TERJ
COMPUTE          >ENDIF

* SHARED RIDE 3
COMPUTE          >IF (P()=Q())
COMPUTE S3IV     >1
COMPUTE S3CS     >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE     >TERI+TERJ
COMPUTE          >ENDIF

*End of Intrazonal trips

* WALK COMMUTER RAIL
COMPUTE          >IF (m305>0)
COMPUTE WCIV     >m305/100.
COMPUTE WCOV     >(m303+m304)/100.
COMPUTE WCXF     >m312
COMPUTE WCCS     >m313
COMPUTE WCXP     >m314/100.
COMPUTE WCWK     >(m301+m302)/100.
COMPUTE          >ENDIF

* WALK BUS
COMPUTE          >IF (m405>0)
COMPUTE WBIV     >m405/100.
COMPUTE WBOV     >(m403+m404)/100.
COMPUTE WBXF     >m412
COMPUTE WBXS     >m413
COMPUTE WBXP     >m414/100.
COMPUTE WBWK     >(m401+m402)/100.
COMPUTE          >ENDIF

* WALK BUS/METRORAIL (TRANSIT)
COMPUTE          >IF (m605>0)
COMPUTE WTIV     >m605/100.
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE WTOV      >(m603+m604)/100.
COMPUTE WTXF      >m612
COMPUTE WTCS      >m613
COMPUTE WTXP      >m614/100.
COMPUTE WTWK      >(m601+m602)/100.
COMPUTE           >ENDIF
```

```
* WALK METRORAIL
COMPUTE           >IF (m505>0)
COMPUTE WMIV      >m505/100.
COMPUTE WMOV      >(m503+m504)/100.
COMPUTE WMXF      >m512
COMPUTE WMCS      >m513
COMPUTE WMXP      >m514/100.
COMPUTE WMWK      >(m501+m502)/100.
COMPUTE           >ENDIF
```

```
* PNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE PCIV      >m319/100.
COMPUTE PCAA      >m329/100.
COMPUTE PCOV      >(m317+m318+m332)/100.
COMPUTE PCXF      >m326
COMPUTE PCCS      >m327+m331+m330/100*AUOP
COMPUTE PCXP      >m328/100.
COMPUTE PCWK      >(m315+m316)/100.
COMPUTE           >ENDIF
```

```
* PNR BUS
COMPUTE           >IF (m419>0)
COMPUTE PBIV      >m419/100.
COMPUTE PBAA      >m429/100.
COMPUTE PBOV      >(m417+m418+m432)/100.
COMPUTE PBXF      >m426
COMPUTE PBXS      >m427+m431+m430/100*AUOP
COMPUTE PBXP      >m428/100.
COMPUTE PBWK      >(m415+m416)/100.
COMPUTE           >ENDIF
```

```
* PNR BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m619>0)
COMPUTE PTIV      >m619/100.
COMPUTE PTAA      >m629/100.
COMPUTE PTOV      >(m617+m618+m632)/100.
COMPUTE PTXF      >m626
COMPUTE PTCS      >m627+m631+m630/100*AUOP
COMPUTE PTXP      >m628/100.
COMPUTE PTWK      >(m615+m616)/100.
COMPUTE           >ENDIF
```

```
* PNR METRORAIL
COMPUTE           >IF (m519>0)
COMPUTE PMIV      >m519/100.
COMPUTE PMAA      >m529/100.
COMPUTE PMOV      >(m517+m518+m532)/100.
COMPUTE PMXF      >m526
COMPUTE PMCS      >m527+m531+m530/100*AUOP
COMPUTE PMXP      >m528/100.
COMPUTE PMWK      >(m515+m516)/100.
COMPUTE           >ENDIF
```

```
* KNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE KCIV      >m319/100.
COMPUTE KCAA      >m329/100.
```

## Appendix D: AEMS Fortran Control Files

```

COMPUTE KCOV      >(m317+m318)/100.
COMPUTE KCXF      >m326
COMPUTE KCCS      >m327+m330/100*AUOP
COMPUTE KCXP      >m328/100.
COMPUTE KCWK      >(m315+m316)/100.
COMPUTE           >ENDIF

```

```

* KNR BUS
COMPUTE           >IF (m437>0)
COMPUTE KBIV      >m437/100.
COMPUTE KBAA      >m447/100.
COMPUTE KBOV      >(m435+m436)/100.
COMPUTE KBXF      >m444
COMPUTE KBXS      >m445+m448/100*AUOP
COMPUTE KBXP      >m446/100.
COMPUTE KBWK      >(m433+m434)/100.
COMPUTE           >ENDIF

```

```

* KNR BUS/METORAIL (TRANSIT)
COMPUTE           >IF (m637>0)
COMPUTE KTIV      >m637/100.
COMPUTE KTAA      >m647/100.
COMPUTE KTOV      >(m635+m636)/100.
COMPUTE KTXF      >m644
COMPUTE KTCS      >m645+m648/100*AUOP
COMPUTE KTXP      >m646/100.
COMPUTE KTWK      >(m633+m634)/100.
COMPUTE           >ENDIF

```

```

* KNR METORAIL
COMPUTE           >IF (m537>0)
COMPUTE KMIV      >m537/100.
COMPUTE KMAA      >m547/100.
COMPUTE KMOV      >(m535+m536)/100.
COMPUTE KMXF      >m544
COMPUTE KMCS      >m545+m548/100*AUOP
COMPUTE KMXP      >m546/100.
COMPUTE KMWK      >(m533+m534)/100.
COMPUTE           >ENDIF

```

\*CONSTANTS BY CHOICE FOR EACH PURPOSE

*CHOICE	1>DR	ALONE	SR2	SR3+	WK-CR	WK-BUS	WK-BU/MR	WK-MR	PNR-CR	KNR-CR	PNR-BUS	KNR-BUS	PNR-BU/MR	KNR-BU/MR	PNR-MR	KNR-MR
PURP01	1INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	1INC	2	1>													
PURP03	1INC	3	1>													
PURP04	1INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								
PURP01	2INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	2INC	2	1>													
PURP03	2INC	3	1>													
PURP04	2INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								
PURP01	3INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	3INC	2	1>													
PURP03	3INC	3	1>													
PURP04	3INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								
PURP01	4INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	4INC	2	1>													
PURP03	4INC	3	1>													
PURP04	4INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								
PURP01	5INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	5INC	2	1>													
PURP03	5INC	3	1>													
PURP04	5INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								
PURP01	6INC	1	1>		2.000000	2.000000	2.000000	2.000000								
PURP02	6INC	2	1>													
PURP03	6INC	3	1>													
PURP04	6INC	4	1>		-2.000000	-2.000000	-2.000000	-2.000000								



## Appendix D: AEMS Fortran Control Files

PURP01 7INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP02 7INC 2	1>				
PURP03 7INC 3	1>				
PURP04 7INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP01 8INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP02 8INC 2	1>				
PURP03 8INC 3	1>				
PURP04 8INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP01 9INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP02 9INC 2	1>				
PURP03 9INC 3	1>				
PURP04 9INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0110INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0210INC 2	1>				
PURP0310INC 3	1>				
PURP0410INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0111INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0211INC 2	1>				
PURP0311INC 3	1>				
PURP0411INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0112INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0212INC 2	1>				
PURP0312INC 3	1>				
PURP0412INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0113INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0213INC 2	1>				
PURP0313INC 3	1>				
PURP0413INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0114INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0214INC 2	1>				
PURP0314INC 3	1>				
PURP0414INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0115INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0215INC 2	1>				
PURP0315INC 3	1>				
PURP0415INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0116INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0216INC 2	1>				
PURP0316INC 3	1>				
PURP0416INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0117INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0217INC 2	1>				
PURP0317INC 3	1>				
PURP0417INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0118INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0218INC 2	1>				
PURP0318INC 3	1>				
PURP0418INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0119INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0219INC 2	1>				
PURP0319INC 3	1>				
PURP0419INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
PURP0120INC 1	1>	2.000000	2.000000	2.000000	2.000000
PURP0220INC 2	1>				
PURP0320INC 3	1>				
PURP0420INC 4	1>	-2.000000	-2.000000	-2.000000	-2.000000
TRIPIN01	>TRP1				
TRIPIN02	>TRP2				
TRIPIN03	>TRP3				
TRIPIN04	>TRP4				
TRIPIFACT01	>tfi1				
TRIPIFACT02	>tfi2				
TRIPIFACT03	>tfi3				
TRIPIFACT04	>tfi4				
COMPUTE tfi1	>1.0				
COMPUTE tfi2	>1.0				
COMPUTE tfi3	>1.0				

## Appendix D: AEMS Fortran Control Files

```

COMPUTE tfi4      >1.0

*
*OUTPUT MATRICES AND OUTPUT FACTORS BY CHOICE FOR EACH PURPOSE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
TRIPPOUT01      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT01      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPPOUT02      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT02      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPPOUT03      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT03      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPPOUT04      1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT04      1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
**
**P AND A WALK PERCENTS BY CHOICE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
WALK SEG CW 1 PCT 1>WSWM
WALK SEG CW 1 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 2 PCT 1>WSW1      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 2 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 3 PCT 1>WSW2      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 3 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 4 PCT 1>WSW3      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 4 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 5 PCT 1>WSM1      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 5 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 6 PCT 1>WSM2      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 6 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG NT 7 PCT 1>WSNT      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG NT 7 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
*SYNTAX OF COMMAND TO ADD A COMPONENT TO A SPECIFIC WALK SEGMENT IF DESIRED
*WALK SEG CW 1 COEF1>      -0.04747  -0.04747  -0.04747  -0.04747  -0.04747
*WALK SEG CW 1 VAR 1>      WTSS      DTSS      DISS      WRSS      DRSS      DJSS
COMPUTE WSWM      >PCMI*PCMJ
COMPUTE WSW1      >(PCTI-PCMI)*PCMJ
COMPUTE WSW2      >(PCTI-PCMI)*(PCTJ-PCMJ)
COMPUTE WSW3      >PCMI*(PCTJ-PCMJ)
COMPUTE WSM1      >(1-PCTI)*PCMJ
COMPUTE WSM2      >(1-PCTI)*(PCTJ-PCMJ)
COMPUTE WSNT      >1-WSWM-WSW1-WSW2-WSW3-WSM1-WSM2

*NEST DEFINITIONS BY CHOICE
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
NEST 1,1=      1>Y      Y
NEST 1,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,4=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,4=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,4=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 6,1=      1>Y
NEST 6,2=      1>      Y      Y
NEST 7,1=      1>      Y      Y
NEST 7,2=      1>      Y      Y

IGRP DEFINITION  >i813
JGRP DEFINITION  >j813
* 1 DC CORE/URBAN-DC CORE

```

## Appendix D: AEMS Fortran Control Files

---

```
SEGMENT 1      >  1  1
SEGMENT 1      >  3  1
* 2 DC CORE/URBAN-VA CORE
SEGMENT 2      >  1  2
SEGMENT 2      >  3  2
* 3 DC CORE/URBAN-URBAN
SEGMENT 3      >  1  3
SEGMENT 3      >  3  3
SEGMENT 3      >  1  4
SEGMENT 3      >  3  4
SEGMENT 3      >  1  5
SEGMENT 3      >  3  5
* 4 DC CORE/URBAN-OTHER
SEGMENT 4      >  1  6
SEGMENT 4      >  3  6
SEGMENT 4      >  1  7
SEGMENT 4      >  3  7
* 5 MD URBAN-DC CORE
SEGMENT 5      >  4  1
* 6 MD URBAN-VA CORE
SEGMENT 6      >  4  2
* 7 MD URBAN-URBAN
SEGMENT 7      >  4  3
SEGMENT 7      >  4  4
SEGMENT 7      >  4  5
* 8 MD URBAN-OTHER
SEGMENT 8      >  4  6
SEGMENT 8      >  4  7
* 9 VA CORE/URBAN-DC CORE
SEGMENT 9      >  2  1
SEGMENT 9      >  5  1
*10 VA CORE/URBAN-VA CORE
SEGMENT 10     >  2  2
SEGMENT 10     >  5  2
*11 VA CORE/URBAN-URBAN
SEGMENT 11     >  2  3
SEGMENT 11     >  5  3
SEGMENT 11     >  2  4
SEGMENT 11     >  5  4
SEGMENT 11     >  2  5
SEGMENT 11     >  5  5
*12 VA CORE/URBAN-OTHER
SEGMENT 12     >  2  6
SEGMENT 12     >  5  6
SEGMENT 12     >  2  7
SEGMENT 12     >  5  7
*13 MD OTHER-DC CORE
SEGMENT 13     >  6  1
*14 MD OTHER-VA CORE
SEGMENT 14     >  6  2
*15 MD OTHER-URBAN
SEGMENT 15     >  6  3
SEGMENT 15     >  6  4
SEGMENT 15     >  6  5
*16 MD OTHER-OTHER
SEGMENT 16     >  6  6
SEGMENT 16     >  6  7
*17 VA OTHER-DC CORE
SEGMENT 17     >  7  1
*18 VA OTHER-VA CORE
SEGMENT 18     >  7  2
*19 VA OTHER-URBAN
SEGMENT 19     >  7  3
SEGMENT 19     >  7  4
SEGMENT 19     >  7  5
*20 VA OTHER-OTHER
SEGMENT 20     >  7  6
SEGMENT 20     >  7  7
```

## Appendix D: AEMS Fortran Control Files

---

```
* SEGMENT 1
NSTC 10 1GRND TOTAL>
NSTC 11 1AUTO > 0.5 0.00000
NSTC 12 1TRANSIT > 0.5 3.72445
NSTC 20 1TOTAL TRN >
NSTC 21 1WALK ACC > 0.5 0.00000
NSTC 22 1PNR ACC > 0.5 -3.76433
NSTC 23 1KNR ACC > 0.5 -7.33524
NSTC 30 1WLK TRN
NSTC 31 1WLK CR > 1.0 -0.80725
NSTC 32 1WLK BUS > 1.0 -1.44958
NSTC 33 1WLK BU/MR > 1.0 -1.46039
NSTC 34 1WLK METRO > 1.0 0.00000
NSTC 40 1PNR TRN
NSTC 41 1PNR CR > 1.0 -0.39351
NSTC 42 1PNR BUS > 1.0 -2.45057
NSTC 43 1PNR BU/MR > 1.0 0.85057
NSTC 44 1PNR METRO > 1.0 0.00000
NSTC 50 1KNR TRN
NSTC 51 1KNR CR > 1.0 3.57299
NSTC 52 1KNR BUS > 1.0 1.26089
NSTC 53 1KNR BU/MR > 1.0 5.74345
NSTC 54 1KNR METRO > 1.0 0.00000
NSTC 60 1AUTO
NSTC 61 1LOV > 1.0 0.00000
NSTC 62 1HOV > 0.5 -1.29504
NSTC 70 1HOV
NSTC 71 1HOV2 > 1.0 0.00000
NSTC 72 1HOV3+ > 1.0 -1.55713
* SEGMENT 2
NSTC 10 2GRND TOTAL>
NSTC 11 2AUTO > 0.5 0.00000
NSTC 12 2TRANSIT > 0.5 4.41614
NSTC 20 2TOTAL TRN >
NSTC 21 2WALK ACC > 0.5 0.00000
NSTC 22 2PNR ACC > 0.5 -6.15269
NSTC 23 2KNR ACC > 0.5 -9.76278
NSTC 30 2WLK TRN
NSTC 31 2WLK CR > 1.0 -2.65644
NSTC 32 2WLK BUS > 1.0 -14.71756
NSTC 33 2WLK BU/MR > 1.0 -5.70638
NSTC 34 2WLK METRO > 1.0 0.00000
NSTC 40 2PNR TRN
NSTC 41 2PNR CR > 1.0 -0.73389
NSTC 42 2PNR BUS > 1.0 -0.73389
NSTC 43 2PNR BU/MR > 1.0 0.05000
NSTC 44 2PNR METRO > 1.0 0.00000
NSTC 50 2KNR TRN
NSTC 51 2KNR CR > 1.0 0.38242
NSTC 52 2KNR BUS > 1.0 0.38242
NSTC 53 2KNR BU/MR > 1.0 9.27713
NSTC 54 2KNR METRO > 1.0 0.00000
NSTC 60 2AUTO
NSTC 61 2LOV > 1.0 0.00000
NSTC 62 2HOV > 0.5 -1.77697
NSTC 70 2HOV
NSTC 71 2HOV2 > 1.0 0.00000
NSTC 72 2HOV3+ > 1.0 -0.97468
* SEGMENT 3
NSTC 10 3GRND TOTAL>
NSTC 11 3AUTO > 0.5 0.00000
NSTC 12 3TRANSIT > 0.5 6.67769
NSTC 20 3TOTAL TRN >
NSTC 21 3WALK ACC > 0.5 0.00000
NSTC 22 3PNR ACC > 0.5 -8.09017
NSTC 23 3KNR ACC > 0.5 -11.27367
NSTC 30 3WLK TRN
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 31 3WLK CR > 1.0 -5.64991
NSTC 32 3WLK BUS > 1.0 -9.07725
NSTC 33 3WLK BU/MR > 1.0 -8.59551
NSTC 34 3WLK METRO > 1.0 0.00000
NSTC 40 3PNR TRN
NSTC 41 3PNR CR > 1.0 -2.35310
NSTC 42 3PNR BUS > 1.0 -9.58041
NSTC 43 3PNR BU/MR > 1.0 -7.89452
NSTC 44 3PNR METRO > 1.0 0.00000
NSTC 50 3KNR TRN
NSTC 51 3KNR CR > 1.0 -0.11150
NSTC 52 3KNR BUS > 1.0 -3.90387
NSTC 53 3KNR BU/MR > 1.0 0.84566
NSTC 54 3KNR METRO > 1.0 0.00000
NSTC 60 3AUTO
NSTC 61 3LOV > 1.0 0.00000
NSTC 62 3HOV > 0.5 -1.45163
NSTC 70 3HOV
NSTC 71 3HOV2 > 1.0 0.00000
NSTC 72 3HOV3+ > 1.0 -1.23730
* SEGMENT 4
NSTC 10 4GRND TOTAL>
NSTC 11 4AUTO > 0.5 0.00000
NSTC 12 4TRANSIT > 0.5 6.39636
NSTC 20 4TOTAL TRN >
NSTC 21 4WALK ACC > 0.5 0.00000
NSTC 22 4PNR ACC > 0.5 -10.41608
NSTC 23 4KNR ACC > 0.5 -12.05800
NSTC 30 4WLK TRN
NSTC 31 4WLK CR > 1.0 -23.21476
NSTC 32 4WLK BUS > 1.0 -22.60831
NSTC 33 4WLK BU/MR > 1.0 -22.95296
NSTC 34 4WLK METRO > 1.0 0.00000
NSTC 40 4PNR TRN
NSTC 41 4PNR CR > 1.0 -0.12203
NSTC 42 4PNR BUS > 1.0 -7.87212
NSTC 43 4PNR BU/MR > 1.0 -6.32970
NSTC 44 4PNR METRO > 1.0 0.00000
NSTC 50 4KNR TRN
NSTC 51 4KNR CR > 1.0 1.27847
NSTC 52 4KNR BUS > 1.0 -1.79718
NSTC 53 4KNR BU/MR > 1.0 -3.84583
NSTC 54 4KNR METRO > 1.0 0.00000
NSTC 60 4AUTO
NSTC 61 4LOV > 1.0 0.00000
NSTC 62 4HOV > 0.5 -1.85795
NSTC 70 4HOV
NSTC 71 4HOV2 > 1.0 0.00000
NSTC 72 4HOV3+ > 1.0 -1.25793
* SEGMENT 5
NSTC 10 5GRND TOTAL>
NSTC 11 5AUTO > 0.5 0.00000
NSTC 12 5TRANSIT > 0.5 3.38848
NSTC 20 5TOTAL TRN >
NSTC 21 5WALK ACC > 0.5 0.00000
NSTC 22 5PNR ACC > 0.5 -6.69365
NSTC 23 5KNR ACC > 0.5 -8.68604
NSTC 30 5WLK TRN
NSTC 31 5WLK CR > 1.0 -3.88773
NSTC 32 5WLK BUS > 1.0 -10.33699
NSTC 33 5WLK BU/MR > 1.0 -9.34656
NSTC 34 5WLK METRO > 1.0 0.00000
NSTC 40 5PNR TRN
NSTC 41 5PNR CR > 1.0 -0.67674
NSTC 42 5PNR BUS > 1.0 -5.49833
NSTC 43 5PNR BU/MR > 1.0 0.80238
NSTC 44 5PNR METRO > 1.0 0.00000
NSTC 50 5KNR TRN
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 51 5KNR CR > 1.0 0.31162
NSTC 52 5KNR BUS > 1.0 0.98120
NSTC 53 5KNR BU/MR > 1.0 7.14475
NSTC 54 5KNR METRO > 1.0 0.00000
NSTC 60 5AUTO
NSTC 61 5LOV > 1.0 0.00000
NSTC 62 5HOV > 0.5 -1.53749
NSTC 70 5HOV
NSTC 71 5HOV2 > 1.0 0.00000
NSTC 72 5HOV3+ > 1.0 -1.78019
* SEGMENT 6
NSTC 10 6GRND TOTAL>
NSTC 11 6AUTO > 0.5 0.00000
NSTC 12 6TRANSIT > 0.5 2.26058
NSTC 20 6TOTAL TRN >
NSTC 21 6WALK ACC > 0.5 0.00000
NSTC 22 6PNR ACC > 0.5 -4.23119
NSTC 23 6KNR ACC > 0.5 -5.48867
NSTC 30 6WLK TRN
NSTC 31 6WLK CR > 1.0 -2.68777
NSTC 32 6WLK BUS > 1.0 -11.29239
NSTC 33 6WLK BU/MR > 1.0 -7.23534
NSTC 34 6WLK METRO > 1.0 0.00000
NSTC 40 6PNR TRN
NSTC 41 6PNR CR > 1.0 -0.87644
NSTC 42 6PNR BUS > 1.0 -0.87644
NSTC 43 6PNR BU/MR > 1.0 -0.25151
NSTC 44 6PNR METRO > 1.0 0.00000
NSTC 50 6KNR TRN
NSTC 51 6KNR CR > 1.0 -0.54440
NSTC 52 6KNR BUS > 1.0 -0.54440
NSTC 53 6KNR BU/MR > 1.0 -0.54440
NSTC 54 6KNR METRO > 1.0 0.00000
NSTC 60 6AUTO
NSTC 61 6LOV > 1.0 0.00000
NSTC 62 6HOV > 0.5 -1.47327
NSTC 70 6HOV
NSTC 71 6HOV2 > 1.0 0.00000
NSTC 72 6HOV3+ > 1.0 -2.55960
* SEGMENT 7
NSTC 10 7GRND TOTAL>
NSTC 11 7AUTO > 0.5 0.00000
NSTC 12 7TRANSIT > 0.5 2.17820
NSTC 20 7TOTAL TRN >
NSTC 21 7WALK ACC > 0.5 0.00000
NSTC 22 7PNR ACC > 0.5 -6.44780
NSTC 23 7KNR ACC > 0.5 -7.67687
NSTC 30 7WLK TRN
NSTC 31 7WLK CR > 1.0 -3.64739
NSTC 32 7WLK BUS > 1.0 -5.05571
NSTC 33 7WLK BU/MR > 1.0 -5.49456
NSTC 34 7WLK METRO > 1.0 0.00000
NSTC 40 7PNR TRN
NSTC 41 7PNR CR > 1.0 -1.30044
NSTC 42 7PNR BUS > 1.0 -4.34816
NSTC 43 7PNR BU/MR > 1.0 -1.66072
NSTC 44 7PNR METRO > 1.0 0.00000
NSTC 50 7KNR TRN
NSTC 51 7KNR CR > 1.0 -4.37215
NSTC 52 7KNR BUS > 1.0 -0.01143
NSTC 53 7KNR BU/MR > 1.0 2.83679
NSTC 54 7KNR METRO > 1.0 0.00000
NSTC 60 7AUTO
NSTC 61 7LOV > 1.0 0.00000
NSTC 62 7HOV > 0.5 -1.70324
NSTC 70 7HOV
NSTC 71 7HOV2 > 1.0 0.00000
NSTC 72 7HOV3+ > 1.0 -1.72701
```

## Appendix D: AEMS Fortran Control Files

---

```
* SEGMENT 8
NSTC 10 8GRND TOTAL>
NSTC 11 8AUTO > 0.5 0.00000
NSTC 12 8TRANSIT > 0.5 1.73906
NSTC 20 8TOTAL TRN >
NSTC 21 8WALK ACC > 0.5 0.00000
NSTC 22 8PNR ACC > 0.5 -5.88393
NSTC 23 8KNR ACC > 0.5 -8.39535
NSTC 30 8WLK TRN
NSTC 31 8WLK CR > 1.0 -7.98029
NSTC 32 8WLK BUS > 1.0 -6.94020
NSTC 33 8WLK BU/MR > 1.0 -7.93190
NSTC 34 8WLK METRO > 1.0 0.00000
NSTC 40 8PNR TRN
NSTC 41 8PNR CR > 1.0 -2.00162
NSTC 42 8PNR BUS > 1.0 -1.14146
NSTC 43 8PNR BU/MR > 1.0 -2.94853
NSTC 44 8PNR METRO > 1.0 0.00000
NSTC 50 8KNR TRN
NSTC 51 8KNR CR > 1.0 0.50461
NSTC 52 8KNR BUS > 1.0 4.30963
NSTC 53 8KNR BU/MR > 1.0 1.68178
NSTC 54 8KNR METRO > 1.0 0.00000
NSTC 60 8AUTO
NSTC 61 8LOV > 1.0 0.00000
NSTC 62 8HOV > 0.5 -2.12200
NSTC 70 8HOV
NSTC 71 8HOV2 > 1.0 0.00000
NSTC 72 8HOV3+ > 1.0 -1.07137
* SEGMENT 9
NSTC 10 9GRND TOTAL>
NSTC 11 9AUTO > 0.5 0.00000
NSTC 12 9TRANSIT > 0.5 7.03008
NSTC 20 9TOTAL TRN >
NSTC 21 9WALK ACC > 0.5 0.00000
NSTC 22 9PNR ACC > 0.5 -12.46855
NSTC 23 9KNR ACC > 0.5 -14.42780
NSTC 30 9WLK TRN
NSTC 31 9WLK CR > 1.0 -25.37241
NSTC 32 9WLK BUS > 1.0 -21.15433
NSTC 33 9WLK BU/MR > 1.0 -17.20596
NSTC 34 9WLK METRO > 1.0 0.00000
NSTC 40 9PNR TRN
NSTC 41 9PNR CR > 1.0 0.38872
NSTC 42 9PNR BUS > 1.0 0.66486
NSTC 43 9PNR BU/MR > 1.0 0.59496
NSTC 44 9PNR METRO > 1.0 0.00000
NSTC 50 9KNR TRN
NSTC 51 9KNR CR > 1.0 0.26627
NSTC 52 9KNR BUS > 1.0 0.26627
NSTC 53 9KNR BU/MR > 1.0 8.78342
NSTC 54 9KNR METRO > 1.0 0.00000
NSTC 60 9AUTO
NSTC 61 9LOV > 1.0 0.00000
NSTC 62 9HOV > 0.5 -1.46918
NSTC 70 9HOV
NSTC 71 9HOV2 > 1.0 0.00000
NSTC 72 9HOV3+ > 1.0 -1.94766
* SEGMENT 10
NSTC 1010GRND TOTAL>
NSTC 1110AUTO > 0.5 0.00000
NSTC 1210TRANSIT > 0.5 1.73132
NSTC 2010TOTAL TRN >
NSTC 2110WALK ACC > 0.5 0.00000
NSTC 2210PNR ACC > 0.5 -5.88064
NSTC 2310KNR ACC > 0.5 -8.47752
NSTC 3010WLK TRN
NSTC 3110WLK CR > 1.0 -3.13572
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 3210WLK BUS > 1.0 -5.72946
NSTC 3310WLK BU/MR > 1.0 -7.52165
NSTC 3410WLK METRO > 1.0 0.00000
NSTC 4010PNR TRN
NSTC 4110PNR CR > 1.0 -1.99023
NSTC 4210PNR BUS > 1.0 -0.69594
NSTC 4310PNR BU/MR > 1.0 -1.99023
NSTC 4410PNR METRO > 1.0 0.00000
NSTC 5010KNR TRN
NSTC 5110KNR CR > 1.0 -0.28971
NSTC 5210KNR BUS > 1.0 -0.28971
NSTC 5310KNR BU/MR > 1.0 -0.28971
NSTC 5410KNR METRO > 1.0 0.00000
NSTC 6010AUTO
NSTC 6110LOV > 1.0 0.00000
NSTC 6210HOV > 0.5 -1.79093
NSTC 7010HOV
NSTC 7110HOV2 > 1.0 0.00000
NSTC 7210HOV3+ > 1.0 -1.37094
* SEGMENT 11
NSTC 1011GRND TOTAL>
NSTC 1111AUTO > 0.5 0.00000
NSTC 1211TRANSIT > 0.5 5.35269
NSTC 2011TOTAL TRN >
NSTC 2111WALK ACC > 0.5 0.00000
NSTC 2211PNR ACC > 0.5 -12.58348
NSTC 2311KNR ACC > 0.5 -13.89833
NSTC 3011WLK TRN
NSTC 3111WLK CR > 1.0 -12.85594
NSTC 3211WLK BUS > 1.0 -17.43408
NSTC 3311WLK BU/MR > 1.0 -16.91948
NSTC 3411WLK METRO > 1.0 0.00000
NSTC 4011PNR TRN
NSTC 4111PNR CR > 1.0 -0.22059
NSTC 4211PNR BUS > 1.0 -1.40483
NSTC 4311PNR BU/MR > 1.0 0.25582
NSTC 4411PNR METRO > 1.0 0.00000
NSTC 5011KNR TRN
NSTC 5111KNR CR > 1.0 -0.55664
NSTC 5211KNR BUS > 1.0 -0.55664
NSTC 5311KNR BU/MR > 1.0 -0.48224
NSTC 5411KNR METRO > 1.0 0.00000
NSTC 6011AUTO
NSTC 6111LOV > 1.0 0.00000
NSTC 6211HOV > 0.5 -1.87907
NSTC 7011HOV
NSTC 7111HOV2 > 1.0 0.00000
NSTC 7211HOV3+ > 1.0 -1.52300
* SEGMENT 12
NSTC 1012GRND TOTAL>
NSTC 1112AUTO > 0.5 0.00000
NSTC 1212TRANSIT > 0.5 4.23525
NSTC 2012TOTAL TRN >
NSTC 2112WALK ACC > 0.5 0.00000
NSTC 2212PNR ACC > 0.5 -9.35569
NSTC 2312KNR ACC > 0.5 -11.70605
NSTC 3012WLK TRN
NSTC 3112WLK CR > 1.0 -16.14143
NSTC 3212WLK BUS > 1.0 -20.83291
NSTC 3312WLK BU/MR > 1.0 -19.81743
NSTC 3412WLK METRO > 1.0 0.00000
NSTC 4012PNR TRN
NSTC 4112PNR CR > 1.0 -9.10845
NSTC 4212PNR BUS > 1.0 -6.88424
NSTC 4312PNR BU/MR > 1.0 -9.10845
NSTC 4412PNR METRO > 1.0 0.00000
NSTC 5012KNR TRN
NSTC 5112KNR CR > 1.0 -2.15853

```



## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5212KNNR BUS > 1.0 -0.17748
NSTC 5312KNNR BU/MR > 1.0 -4.78017
NSTC 5412KNNR METRO > 1.0 0.00000
NSTC 6012AUTO
NSTC 6112LOV > 1.0 0.00000
NSTC 6212HOV > 0.5 -2.19769
NSTC 7012HOV
NSTC 7112HOV2 > 1.0 0.00000
NSTC 7212HOV3+ > 1.0 -1.01759
* SEGMENT 13
NSTC 1013GRND TOTAL>
NSTC 1113AUTO > 0.5 0.00000
NSTC 1213TRANSIT > 0.5 2.53517
NSTC 2013TOTAL TRN >
NSTC 2113WALK ACC > 0.5 0.00000
NSTC 2213PNR ACC > 0.5 -4.78568
NSTC 2313KNNR ACC > 0.5 -6.42225
NSTC 3013WLK TRN
NSTC 3113WLK CR > 1.0 -7.49375
NSTC 3213WLK BUS > 1.0 -8.22635
NSTC 3313WLK BU/MR > 1.0 -8.77999
NSTC 3413WLK METRO > 1.0 0.00000
NSTC 4013PNR TRN
NSTC 4113PNR CR > 1.0 -1.37189
NSTC 4213PNR BUS > 1.0 -6.56855
NSTC 4313PNR BU/MR > 1.0 -0.31971
NSTC 4413PNR METRO > 1.0 0.00000
NSTC 5013KNNR TRN
NSTC 5113KNNR CR > 1.0 -4.43232
NSTC 5213KNNR BUS > 1.0 -6.67781
NSTC 5313KNNR BU/MR > 1.0 -1.36864
NSTC 5413KNNR METRO > 1.0 0.00000
NSTC 6013AUTO
NSTC 6113LOV > 1.0 0.00000
NSTC 6213HOV > 0.5 -1.60180
NSTC 7013HOV
NSTC 7113HOV2 > 1.0 0.00000
NSTC 7213HOV3+ > 1.0 -1.32632
* SEGMENT 14
NSTC 1014GRND TOTAL>
NSTC 1114AUTO > 0.5 0.00000
NSTC 1214TRANSIT > 0.5 1.17306
NSTC 2014TOTAL TRN >
NSTC 2114WALK ACC > 0.5 0.00000
NSTC 2214PNR ACC > 0.5 -1.31363
NSTC 2314KNNR ACC > 0.5 -3.50697
NSTC 3014WLK TRN
NSTC 3114WLK CR > 1.0 -8.30086
NSTC 3214WLK BUS > 1.0 -4.27224
NSTC 3314WLK BU/MR > 1.0 -5.32487
NSTC 3414WLK METRO > 1.0 0.00000
NSTC 4014PNR TRN
NSTC 4114PNR CR > 1.0 -5.72124
NSTC 4214PNR BUS > 1.0 -1.17606
NSTC 4314PNR BU/MR > 1.0 -1.23010
NSTC 4414PNR METRO > 1.0 0.00000
NSTC 5014KNNR TRN
NSTC 5114KNNR CR > 1.0 -9.21450
NSTC 5214KNNR BUS > 1.0 -1.14640
NSTC 5314KNNR BU/MR > 1.0 -1.11396
NSTC 5414KNNR METRO > 1.0 0.00000
NSTC 6014AUTO
NSTC 6114LOV > 1.0 0.00000
NSTC 6214HOV > 0.5 -1.83504
NSTC 7014HOV
NSTC 7114HOV2 > 1.0 0.00000
NSTC 7214HOV3+ > 1.0 -1.32021
* SEGMENT 15
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 1015GRND TOTAL>
NSTC 1115AUTO > 0.5 0.00000
NSTC 1215TRANSIT > 0.5 2.06591
NSTC 2015TOTAL TRN >
NSTC 2115WALK ACC > 0.5 0.00000
NSTC 2215PNR ACC > 0.5 -4.78366
NSTC 2315KNR ACC > 0.5 -5.89947
NSTC 3015WLK TRN
NSTC 3115WLK CR > 1.0 -9.75610
NSTC 3215WLK BUS > 1.0 -6.22465
NSTC 3315WLK BU/MR > 1.0 -7.57288
NSTC 3415WLK METRO > 1.0 0.00000
NSTC 4015PNR TRN
NSTC 4115PNR CR > 1.0 -3.84892
NSTC 4215PNR BUS > 1.0 -1.70538
NSTC 4315PNR BU/MR > 1.0 -1.45540
NSTC 4415PNR METRO > 1.0 0.00000
NSTC 5015KNR TRN
NSTC 5115KNR CR > 1.0 -6.67092
NSTC 5215KNR BUS > 1.0 -2.35052
NSTC 5315KNR BU/MR > 1.0 -2.29305
NSTC 5415KNR METRO > 1.0 0.00000
NSTC 6015AUTO
NSTC 6115LOV > 1.0 0.00000
NSTC 6215HOV > 0.5 -2.00158
NSTC 7015HOV
NSTC 7115HOV2 > 1.0 0.00000
NSTC 7215HOV3+ > 1.0 -1.65818
* SEGMENT 16
NSTC 1016GRND TOTAL>
NSTC 1116AUTO > 0.5 0.00000
NSTC 1216TRANSIT > 0.5 0.00011
NSTC 2016TOTAL TRN >
NSTC 2116WALK ACC > 0.5 0.00000
NSTC 2216PNR ACC > 0.5 -3.64900
NSTC 2316KNR ACC > 0.5 -3.99940
NSTC 3016WLK TRN
NSTC 3116WLK CR > 1.0 -5.28939
NSTC 3216WLK BUS > 1.0 -1.50798
NSTC 3316WLK BU/MR > 1.0 -2.94853
NSTC 3416WLK METRO > 1.0 0.00000
NSTC 4016PNR TRN
NSTC 4116PNR CR > 1.0 -1.79539
NSTC 4216PNR BUS > 1.0 -0.86965
NSTC 4316PNR BU/MR > 1.0 -0.59093
NSTC 4416PNR METRO > 1.0 0.00000
NSTC 5016KNR TRN
NSTC 5116KNR CR > 1.0 -4.26674
NSTC 5216KNR BUS > 1.0 -1.39508
NSTC 5316KNR BU/MR > 1.0 -1.66796
NSTC 5416KNR METRO > 1.0 0.00000
NSTC 6016AUTO
NSTC 6116LOV > 1.0 0.00000
NSTC 6216HOV > 0.5 -2.24901
NSTC 7016HOV
NSTC 7116HOV2 > 1.0 0.00000
NSTC 7216HOV3+ > 1.0 -1.45489
* SEGMENT 17
NSTC 1017GRND TOTAL>
NSTC 1117AUTO > 0.5 0.00000
NSTC 1217TRANSIT > 0.5 3.51488
NSTC 2017TOTAL TRN >
NSTC 2117WALK ACC > 0.5 0.00000
NSTC 2217PNR ACC > 0.5 -7.86894
NSTC 2317KNR ACC > 0.5 -8.86193
NSTC 3017WLK TRN
NSTC 3117WLK CR > 1.0 -17.57389
NSTC 3217WLK BUS > 1.0 -13.92998
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 3317WLK BU/MR > 1.0 -12.83641
NSTC 3417WLK METRO > 1.0 0.00000
NSTC 4017PNR TRN
NSTC 4117PNR CR > 1.0 -3.30493
NSTC 4217PNR BUS > 1.0 -0.70056
NSTC 4317PNR BU/MR > 1.0 0.23622
NSTC 4417PNR METRO > 1.0 0.00000
NSTC 5017KNR TRN
NSTC 5117KNR CR > 1.0 -6.57274
NSTC 5217KNR BUS > 1.0 -2.98946
NSTC 5317KNR BU/MR > 1.0 -1.94384
NSTC 5417KNR METRO > 1.0 0.00000
NSTC 6017AUTO
NSTC 6117LOV > 1.0 0.00000
NSTC 6217HOV > 0.5 -2.01828
NSTC 7017HOV
NSTC 7117HOV2 > 1.0 0.00000
NSTC 7217HOV3+ > 1.0 -1.57923
* SEGMENT 18
NSTC 1018GRND TOTAL>
NSTC 1118AUTO > 0.5 0.00000
NSTC 1218TRANSIT > 0.5 2.36783
NSTC 2018TOTAL TRN >
NSTC 2118WALK ACC > 0.5 0.00000
NSTC 2218PNR ACC > 0.5 -4.98412
NSTC 2318KNR ACC > 0.5 -5.91930
NSTC 3018WLK TRN
NSTC 3118WLK CR > 1.0 -11.40125
NSTC 3218WLK BUS > 1.0 -6.94533
NSTC 3318WLK BU/MR > 1.0 -7.86833
NSTC 3418WLK METRO > 1.0 0.00000
NSTC 4018PNR TRN
NSTC 4118PNR CR > 1.0 -1.01417
NSTC 4218PNR BUS > 1.0 1.09391
NSTC 4318PNR BU/MR > 1.0 -0.19550
NSTC 4418PNR METRO > 1.0 0.00000
NSTC 5018KNR TRN
NSTC 5118KNR CR > 1.0 -4.77777
NSTC 5218KNR BUS > 1.0 -2.21824
NSTC 5318KNR BU/MR > 1.0 -2.43815
NSTC 5418KNR METRO > 1.0 0.00000
NSTC 6018AUTO
NSTC 6118LOV > 1.0 0.00000
NSTC 6218HOV > 0.5 -2.10544
NSTC 7018HOV
NSTC 7118HOV2 > 1.0 0.00000
NSTC 7218HOV3+ > 1.0 -1.74176
* SEGMENT 19
NSTC 1019GRND TOTAL>
NSTC 1119AUTO > 0.5 0.00000
NSTC 1219TRANSIT > 0.5 2.76083
NSTC 2019TOTAL TRN >
NSTC 2119WALK ACC > 0.5 0.00000
NSTC 2219PNR ACC > 0.5 -6.71782
NSTC 2319KNR ACC > 0.5 -7.34757
NSTC 3019WLK TRN
NSTC 3119WLK CR > 1.0 -15.09131
NSTC 3219WLK BUS > 1.0 -11.42943
NSTC 3319WLK BU/MR > 1.0 -10.74147
NSTC 3419WLK METRO > 1.0 0.00000
NSTC 4019PNR TRN
NSTC 4119PNR CR > 1.0 -1.95993
NSTC 4219PNR BUS > 1.0 -0.65682
NSTC 4319PNR BU/MR > 1.0 -0.49789
NSTC 4419PNR METRO > 1.0 0.00000
NSTC 5019KNR TRN
NSTC 5119KNR CR > 1.0 -5.90216
NSTC 5219KNR BUS > 1.0 -3.43406
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 5319KNR BU/MR > 1.0 -3.07321
NSTC 5419KNR METRO > 1.0 0.00000
NSTC 6019AUTO
NSTC 6119LOV > 1.0 0.00000
NSTC 6219HOV > 0.5 -2.49049
NSTC 7019HOV
NSTC 7119HOV2 > 1.0 0.00000
NSTC 7219HOV3+ > 1.0 -2.56594
* SEGMENT 20
NSTC 1020GRND TOTAL>
NSTC 1120AUTO > 0.5 0.00000
NSTC 1220TRANSIT > 0.5 1.65769
NSTC 2020TOTAL TRN >
NSTC 2120WALK ACC > 0.5 0.00000
NSTC 2220PNR ACC > 0.5 -7.72107
NSTC 2320KNR ACC > 0.5 -6.76308
NSTC 3020WLK TRN
NSTC 3120WLK CR > 1.0 -16.37276
NSTC 3220WLK BUS > 1.0 -9.94345
NSTC 3320WLK BU/MR > 1.0 -11.56452
NSTC 3420WLK METRO > 1.0 0.00000
NSTC 4020PNR TRN
NSTC 4120PNR CR > 1.0 -2.92185
NSTC 4220PNR BUS > 1.0 -6.22180
NSTC 4320PNR BU/MR > 1.0 -3.80359
NSTC 4420PNR METRO > 1.0 0.00000
NSTC 5020KNR TRN
NSTC 5120KNR CR > 1.0 -9.14702
NSTC 5220KNR BUS > 1.0 -6.00517
NSTC 5320KNR BU/MR > 1.0 -7.60890
NSTC 5420KNR METRO > 1.0 0.00000
NSTC 6020AUTO
NSTC 6120LOV > 1.0 0.00000
NSTC 6220HOV > 0.5 -2.39718
NSTC 7020HOV
NSTC 7120HOV2 > 1.0 0.00000
NSTC 7220HOV3+ > 1.0 -2.07527

*DOWNTOWN=8
*SELI > 8

*UNION STATION=64
*SELI > 64

* =122
*SELI > 122

*BETHESDA=345
*SELI > 345

*SILVER SPRING=362
*SELI > 362

*N.SILVER SPRING=464
*SELI > 464

* =475
*SELI > 475

*SHADY GROVE RD=578
*SELI > 578

* =787
*SELI > 787

*ANDREWS AFB=829
*SELI > 829

```

## Appendix D: AEMS Fortran Control Files

---

*NEW CARROLTON=927		
*SELI	>	927
*BRISTOL=972		
*SELI	>	972
*FREDERICK=1043		
*SELI	>	1043
*JESSUP=1080		
*SELI	>	1080
*SCAGGSVILLE=1091		
*SELI	>	1091
*WALDORF=1216		
*SELI	>	1216
*PENTAGON=1231		
*SELI	>	1231
*ROSSLYN=1236		
*SELI	>	1236
*ALEXANDRIA=1337		
*SELI	>	1337
* =1455		
*SELI	>	1455
*SPRINGFIELD=1502		
*SELI	>	1502
* =1511		
*SELI	>	1511
*TYSONS CRNR=1537		
*SELI	>	1537
*FT BELVOIR=1554		
*SELI	>	1554
*VIENNA=1619		
*SELI	>	1619
*DULES AP=1698		
*SELI	>	1698
*RESTON=1716		
*SELI	>	1716
*LEESBURG=1842		
*SELI	>	1842
*BRUNSWICK=1863		
*SELI	>	1863
*DALE CITY=1942		
*SELI	>	1942
*MANASSAS=1967		
*SELI	>	1967
*SPOTSYLVANIA=2110		
*SELI	>	2110
* =2055		
*SELI	>	2055

## Appendix D: AEMS Fortran Control Files

```

*SELJ      >      8
*SELJ      >     63
*SELJ      >     64
*SELJ      >     77
*SELJ      >    100
*SELJ      >    344
*SELJ      >    345
*SELJ      >    362
*SELJ      >   1231
*SELJ      >   1236
*SELJ      >   1265
*SELJ      >   1337
*SELJ      >   1537

*SELI      >   523
*SELJ      >     9

TRACE      >     0
* OUTPUT % >
*PROCSEL   >
PRINT MS   >HBW_NL_MC.PRN
INPUT PRINT FILE >HBW_NL_MC.PRN
INPUT GOALS >HBW_NL_MC.GOL
INFILE 1   >hbw_income.ptt
INFILE 2   >hwyam.skm
INFILE 3   >TRNAM_CR.SKM
INFILE 4   >TRNAM_AB.SKM
INFILE 5   >TRNAM_MR.SKM
INFILE 6   >TRNAM_BM.SKM
ZINFILE 8  >ZONEV2.A2F
OUTFILE 9  >HBW_NL_MC.MTT

* FTA USER BENEFITS SPECIFICATIONS
*FTA RESULTS FILE >HBW_NL_MC.BEN
FTA TRANSIT COEFF >-0.02128
FTA AUTO COEFF >-0.02128
FTA PURPOSE NAME >HBW
FTA PERIOD NAME >ALLDAY
FTA ALTER. NAME >CALIB
*CHOICE      1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
FTA AUTO NEST >      1      1
FTA MOTORIZED? 1>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
FTA TRANSIT?  1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y

```

## 4 lineSum\_MR\_access.ctf

```

## Access reports focus on riders who arrive or depart using transit access links
## i.e., the summary does not include transfers
TITLE      Metrorail Station Access Summmmary
DEFAULT_FILE_FORMAT DBASE

PEAK RIDERSHIP_FILE 1          PK_VOL.DBF
PEAK RIDERSHIP_FORMAT 1      DBASE
OFFPEAK RIDERSHIP_FILE_1     OP_VOL.DBF
OFFPEAK RIDERSHIP_FORMAT_1   DBASE

STOP_NAME_FILE      ..\inputs\station_names.dbf
STOP_NAME_FORMAT    DBASE

ACCESS_REPORT_TITLE 1          All
ACCESS_REPORT_STOPS 1          8001..8100, 8119..8140, 8145..8148, 8150..8154, 8160..8166, 8169..8182
##ACCESS_REPORT_MODES_1      11,12,14,15,16
ACCESS_REPORT_MODES 1          ALL
##ACCESS_REPORT_DETAILS_1    MODE

```

## Appendix D: AEMS Fortran Control Files

```
NEW_ACCESS_REPORT_FILE_1      MR_access.txt
NEW_ACCESS_REPORT_FORMAT_1    TAB_DELIMITED
```

### 5 lineSum\_MR\_line.ctf

```
## Line reports summarize boardings, alightings, and ridership for one or more line
TITLE                          Metrorail Line Summmary
DEFAULT_FILE_FORMAT            DBASE

PEAK_RIDERSHIP_FILE_1          PK_VOL.DBF
PEAK_RIDERSHIP_FORMAT_1       DBASE
OFFPEAK_RIDERSHIP_FILE_1      OP_VOL.DBF
OFFPEAK_RIDERSHIP_FORMAT_1    DBASE

STOP_NAME_FILE                 ..\inputs\station_names.dbf
STOP_NAME_FORMAT              DBASE

LINE_REPORT_TITLE_1           All
LINE_REPORT_LINES_1          All
LINE_REPORT_MODES_1          3
NEW_TOTAL_RIDERSHIP_FILE_1    MR_line.txt
NEW_TOTAL_RIDERSHIP_FORMAT_1 TAB_DELIMITED
```

### 6 LineSum\_Volume.ctf

```
TITLE                          Merge the Transit Volumes
DEFAULT_FILE_FORMAT            DBASE

PEAK_RIDERSHIP_FILE_1          ..\i4_DRABAMlink.dbf           //DRIVE ACCESS
PEAK_RIDERSHIP_FILE_2          ..\i4_DRBMAMlink.dbf
PEAK_RIDERSHIP_FILE_3          ..\i4_DRCRAMlink.dbf
PEAK_RIDERSHIP_FILE_4          ..\i4_DRMRAMlink.dbf
PEAK_RIDERSHIP_FILE_5          ..\i4_KRABAMlink.dbf           //KISS AND RIDE ACCESS
PEAK_RIDERSHIP_FILE_6          ..\i4_KRBMAMlink.dbf
PEAK_RIDERSHIP_FILE_7          ..\i4_KRMRAMlink.dbf
PEAK_RIDERSHIP_FILE_8          ..\i4_WKABAMlink.dbf           //WALK ACCESS
PEAK_RIDERSHIP_FILE_9          ..\i4_WKBMAMlink.dbf
PEAK_RIDERSHIP_FILE_10         ..\i4_WKCRAMlink.dbf
PEAK_RIDERSHIP_FILE_11         ..\i4_WKMRAMlink.dbf

OFFPEAK_RIDERSHIP_FILE_1       ..\i4_DRABOPlink.dbf           //DRIVE ACCESS
OFFPEAK_RIDERSHIP_FILE_2       ..\i4_DRMOPlink.dbf
OFFPEAK_RIDERSHIP_FILE_3       ..\i4_DRCROPlink.dbf
OFFPEAK_RIDERSHIP_FILE_4       ..\i4_DRMOPlink.dbf
OFFPEAK_RIDERSHIP_FILE_5       ..\i4_KRABOPlink.dbf           //KISS AND RIDE ACCESS
OFFPEAK_RIDERSHIP_FILE_6       ..\i4_KRMOPlink.dbf
OFFPEAK_RIDERSHIP_FILE_7       ..\i4_KRMOPlink.dbf
OFFPEAK_RIDERSHIP_FILE_8       ..\i4_WKABOPlink.dbf           //WALK ACCESS
OFFPEAK_RIDERSHIP_FILE_9       ..\i4_WKMOPlink.dbf
OFFPEAK_RIDERSHIP_FILE_10      ..\i4_WKCROPlink.dbf
OFFPEAK_RIDERSHIP_FILE_11      ..\i4_WKMROPlink.dbf

NEW_PEAK_RIDERSHIP_FILE        PK_VOL.dbf
NEW_PEAK_RIDERSHIP_FORMAT      DBASE
NEW_OFFPEAK_RIDERSHIP_FILE     OP_VOL.dbf
NEW_OFFPEAK_RIDERSHIP_FORMAT   DBASE
```

# 7 nho\_nl\_mc.ctf

```

NHO OP NESTED LOGIT MC - #DATE: 9/18/2011 #VER: 21
CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
*
*
*LOGIT COEFFICIENTS BY CHOICE FOR EACH SKIM (NO INPUT SKIM IS
*EQUIVALENT TO A CONSTANT)
*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
COEF01:IVTT      1>-0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860  -0.02860
SKIM01:IVTT      1>DAIV      S2IV      S3IV      WCIV      WBIV      WTIV      WMIV      PCIV      KCIV      PBIV      KBIV      PTIV      KTIV      PMIV      KMIV
COEF02:AUTO ACC  1>
SKIM02:AUTO ACC  1>
COEF03:TERM/OVTT 1>-0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150  -0.07150
SKIM03:TERM/OVTT 1>DATE      S2TE      S3TE      WCOV      WBOV      WTOV      WMOV      PCOV      KCOV      PBOV      KBOV      PTOV      KTOV      PMOV      KMOV
* LIMIT COEF 04 TO PURPOSE 1
COEF PURP04      >1
COEF04:COST INC1 1>-0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994
SKIM04:COST INC1 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 05 TO PURPOSE 2
COEF PURP05      >2
COEF05:COST INC2 1>-0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994
SKIM05:COST INC2 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
* LIMIT COEF 06 TO PURPOSE 3
COEF PURP06      >3
COEF06:COST INC3 1>-0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994
SKIM06:COST INC3 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF PURP07      >4
* LIMIT COEF 07 TO PURPOSE 4
COEF07:COST INC4 1>-0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994  -0.00994
SKIM07:COST INC4 1>DACS      S2CS      S3CS      WCCS      WBCS      WTCS      WMCS      PCCS      KCCS      PBCS      KBCS      PTCS      KTCS      PMCS      KMCS
COEF08:TRN XFERS 1>
SKIM08:TRN XFERS 1>
COEF09:TRN BRDPEN 1>
SKIM09:TRN BRDPEN 1>
*WALK WEIGHT
COEF10:TRN WLKWT 1>
SKIM10:TRN WLKWT 1>
*SYNTAX TO LIMIT UTILITY ELEMENT TO A PARTICULAR WALK SEGMENT IN THIS EXAMPLE
* COEF 18 APPLIES ONLY TO WALK SEGMENT 1
*COEF WLKSEG18   >1
*
* ASSUMED MATRIX ORGANIZATION
* FILE 1 TRIP TABLE (SEPARATE FOR EACH PURPOSE)
* 1 INCOME 1 (HOME-BASED)/ALL NHB TRIPS
* 2 INCOME 2 (HOME-BASED)
* 3 INCOME 3 (HOME-BASED)
* 4 INCOME 4 (HOME-BASED)
*
* FILE 2 HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 SOV TIME (MIN)
* 2 SOV DIST (0.1 MILES)
* 3 SOV TOLL (2007 CENTS)
* 4 HOV2 TIME (MIN)
* 5 HOV2 DIST (0.1 MILES)
* 6 HOV2 TOLL (2007 CENTS)
* 7 HOV3+ TIME (MIN)
* 8 HOV3+ DIST (0.1 MILES)
* 9 HOV3+ TOLL (2007 CENTS)
*
* FILE 3=COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 4=BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 5=METRO RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 6=BUS+METRO RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)

```



## Appendix D: AEMS Fortran Control Files

```

* 2 WLK OTHER (.01 MIN) 16 PNR OTHER (.01 MIN) 34 KNR OTHER (.01 MIN)
* 3 WLK IWAIT (.01 MIN) 17 PNR IWAIT (.01 MIN) 35 KNR IWAIT (.01 MIN)
* 4 WLK XWAIT (.01 MIN) 18 PNR XWAIT (.01 MIN) 36 KNR XWAIT (.01 MIN)
* 5 WLK IVTT TOT(.01 MIN) 19 PNR IVTT TOT(.01 MIN) 37 KNR IVTT TOT(.01 MIN)
* 6 WLK IVTT CR (.01 MIN) 20 PNR IVTT CR (.01 MIN) 38 KNR IVTT CR (.01 MIN)
* 7 WLK IVTT XB (.01 MIN) 21 PNR IVTT XB (.01 MIN) 39 KNR IVTT XB (.01 MIN)
* 8 WLK IVTT MR (.01 MIN) 22 PNR IVTT MR (.01 MIN) 40 KNR IVTT MR (.01 MIN)
* 9 WLK IVTT NM (.01 MIN) 23 PNR IVTT NM (.01 MIN) 41 KNR IVTT NM (.01 MIN)
* 10 WLK IVTT NM2(.01 MIN) 24 PNR IVTT NM2(.01 MIN) 42 KNR IVTT NM2(.01 MIN)
* 11 WLK IVTT LB (.01 MIN) 25 PNR IVTT LB (.01 MIN) 43 KNR IVTT LB (.01 MIN)
* 12 WLK #XFERS (NUMBER ) 26 PNR #XFERS (NUMBER ) 44 KNR #XFERS (NUMBER )
* 13 WLK COST (.07CENTS) 27 PNR COST (.07CENTS) 45 KNR COST (.07CENTS)
* 14 WLK XPEN (.01 MIN) 28 PNR XPEN (.01 MIN) 46 KNR XPEN (.01 MIN)
* 29 PNR ACC TIME(.01 MIN) 47 KNR ACC TIME(.01 MIN)
* 30 PNR ACC DIST(.01 MIL) 48 KNR ACC DIST(.01 MIL)
* 31 PNR ACC COST(.07CENTS)
* 32 PNR STA TERM(.01 MIN)
*
* FILE 8=ZDATA
* 1 HBW PARK COST (2007 CENTS)
* 2 HBS PARK COST (2007 CENTS)
* 3 HBO PARK COST (2007 CENTS)
* 4 NHB PARK COST (2007 CENTS)
* 5 TERMINAL TIME (HOME BASED) (MINUTES)
* 6 TERMINAL TIME (NON HOME BASED) (MINUTES)
* 7 ARC VIEW SHORT WALK PERCENT TO METRO
* 8 ARC VIEW LONG WALK PERCENT TO METRO
* 9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
* 10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT
* 11 ARC VIEW SHORT WALK PERCENT TO ALL OP TRANSIT
* 12 ARC VIEW LONG WALK PERCENT TO ALL OP TRANSIT
* 13 AREA TYPE
* 1=DC CORE
* 2=VA CORE
* 3=DC URBAN
* 4=MD URBAN
* 5=VA URBAN
* 6=MD OTHER
* 7=VA OTHER
*
* PARAMETERS
*=====
* AUTO OPERATING COSTS IN CENTS/mile
COMPUTE AUOP >10
* AUTO OCCUPANCY FOR 3+ Reduced from 3.5 to 3.35 on 3/1/07 rm
COMPUTE OCC3 >3.35
*
* TERMINAL TIMES, USE i/j805 FOR HBW, HBS, AND HBO. USE i/j806 FOR NHB
* HBW/HBS/HBO
*COMPUTE TERI >i805
*COMPUTE TERJ >j805
* NHB
COMPUTE TERI >i806
COMPUTE TERJ >j806
*
* PARK COSTS, USE i/j801 802 803 804 FOR HBW, HBS, HBO, NHB RESPECTIVELY
* HBW
*COMPUTE PRKC >j801/2.
* HBS
* COMPUTE PRKC >j802/2.
* HBO
* COMPUTE PRKC >j803/2.
* NHB
COMPUTE PRKC >j804
*
* Percent of productions in long-walk area that are assumed to walk = 25% (i.e., 75% drive)
COMPUTE PCLM >0.25
COMPUTE PCLT >0.25

```

## Appendix D: AEMS Fortran Control Files

---

```
* PERCENT WALKS-METRO RAIL ONLY
COMPUTE PCMI >(i807+PCLM*(i808-i807))/100.
COMPUTE PCMJ >(j807+PCLM*(j808-j807))/100.
* PERCENT WALKS-PEAK
*COMPUTE PCTI >(i809+PCLT*(i810-i809))/100.
*COMPUTE PCTJ >(j809+PCLT*(j810-j809))/100.
* PERCENT WALKS-OFFPEAK
COMPUTE PCTI >(i811+PCLT*(i812-i811))/100.
COMPUTE PCTJ >(j811+PCLT*(j812-j811))/100.
COMPUTE PCMI >MAX(PCMI,0)
COMPUTE PCMI >MIN(PCMI,1)
COMPUTE PCMJ >MAX(PCMJ,0)
COMPUTE PCMJ >MIN(PCMJ,1)
COMPUTE PCTI >MAX(PCTI,PCMI)
COMPUTE PCTI >MIN(PCTI,1)
COMPUTE PCTJ >MAX(PCTJ,PCMJ)
COMPUTE PCTJ >MIN(PCTJ,1)
*
* DO TRIP SUBDIVISIONS
*
* HOME BASED ALTERNATIVES
*COMPUTE TRP1 >m101
*COMPUTE TRP2 >m102
*COMPUTE TRP3 >m103
*COMPUTE TRP4 >m104
* NON-HOME BASED
COMPUTE TRP1 >0.25*m101
COMPUTE TRP2 >0.25*m101
COMPUTE TRP3 >0.25*m101
COMPUTE TRP4 >0.25*m101
*
* BE SURE TO UPDATE THE IVTT COEFFICIENT IN FTA SECTION FOR EACH PURPOSE
*
*-----
*INITIALIZING ALL VARIABLES WITHIN IF STATEMENTS TO ZERO
COMPUTE DAIV >0
COMPUTE DACS >0
COMPUTE DATE >0
COMPUTE S2IV >0
COMPUTE S2CS >0
COMPUTE S2TE >0
COMPUTE S3IV >0
COMPUTE S3CS >0
COMPUTE S3TE >0
COMPUTE WKIV >0
COMPUTE WKOV >0
COMPUTE WKXF >0
COMPUTE WKCS >0
COMPUTE WKXP >0
COMPUTE WBIV >0
COMPUTE WBOV >0
COMPUTE WBXF >0
COMPUTE WBCS >0
COMPUTE WBXP >0
COMPUTE WTIV >0
COMPUTE WTOV >0
COMPUTE WTXF >0
COMPUTE WTCS >0
COMPUTE WTXP >0
COMPUTE WMIV >0
COMPUTE WMOV >0
COMPUTE WMXF >0
COMPUTE WMCS >0
COMPUTE WMXP >0
COMPUTE PCIV >0
COMPUTE PCAA >0
COMPUTE PCOV >0
COMPUTE PCXF >0
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE PCCS      >0
COMPUTE PCXP      >0
COMPUTE PBIV      >0
COMPUTE PBAA      >0
COMPUTE PBOV      >0
COMPUTE PBXF      >0
COMPUTE PBCS      >0
COMPUTE PBXP      >0
COMPUTE PTIV      >0
COMPUTE PTAA      >0
COMPUTE PTOV      >0
COMPUTE PTXF      >0
COMPUTE PTCS      >0
COMPUTE PTXP      >0
COMPUTE PMIV      >0
COMPUTE PMAA      >0
COMPUTE PMOV      >0
COMPUTE PMXF      >0
COMPUTE PMCS      >0
COMPUTE PMXP      >0
COMPUTE KCIV      >0
COMPUTE KCAA      >0
COMPUTE KCOV      >0
COMPUTE KCXF      >0
COMPUTE KCCS      >0
COMPUTE KCXP      >0
COMPUTE KBIV      >0
COMPUTE KBAA      >0
COMPUTE KBOV      >0
COMPUTE KBXF      >0
COMPUTE KBXS      >0
COMPUTE KBXP      >0
COMPUTE KTIV      >0
COMPUTE KTAA      >0
COMPUTE KTOV      >0
COMPUTE KTXF      >0
COMPUTE KTCS      >0
COMPUTE KTXP      >0
COMPUTE KMIV      >0
COMPUTE KMAA      >0
COMPUTE KMOV      >0
COMPUTE KMXF      >0
COMPUTE KMCS      >0
COMPUTE KMXP      >0

COMPUTE WCWK      >0
COMPUTE WBWK      >0
COMPUTE WTWK      >0
COMPUTE WMWK      >0
COMPUTE PCWK      >0
COMPUTE KCWK      >0
COMPUTE PBWK      >0
COMPUTE KBWK      >0
COMPUTE PTWK      >0
COMPUTE KTWK      >0
COMPUTE PMWK      >0
COMPUTE KMWK      >0

* SKIM VALUES, Divide distances by 10 to convert tenths of miles to whole miles
* DRIVE ALONE
COMPUTE           >IF (m201>0)
COMPUTE DAIV      >m201
COMPUTE DACS      >m202/10*AUOP+m203+PRKC
COMPUTE DATE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 2
COMPUTE           >IF (m204>0)
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE S2IV      >m204
COMPUTE S2CS      >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 3
COMPUTE           >IF(m207>0)
COMPUTE S3IV      >m207
COMPUTE S3CS      >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE      >TERI+TERJ
COMPUTE           >ENDIF

* Assign Intrazonal trips to Autos (mj11/04/05)
COMPUTE           >IF(P()==Q())
COMPUTE DAIV      >1
COMPUTE DACS      >m202/10*AUOP+m203+PRKC
COMPUTE DATE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 2
COMPUTE           >IF(P()==Q())
COMPUTE S2IV      >1
COMPUTE S2CS      >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE      >TERI+TERJ
COMPUTE           >ENDIF

* SHARED RIDE 3
COMPUTE           >IF(P()==Q())
COMPUTE S3IV      >1
COMPUTE S3CS      >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE      >TERI+TERJ
COMPUTE           >ENDIF

*End of Intrazonal trips

* WALK COMMUTER RAIL
COMPUTE           >IF(m305>0)
COMPUTE WCIV      >m305/100.
COMPUTE WCOV      >(m303+m304)/100.
COMPUTE WCXF      >m312
COMPUTE WCCS      >m313
COMPUTE WCXP      >m314/100.
COMPUTE WCWK      >(m301+m302)/100.
COMPUTE           >ENDIF

* WALK BUS
COMPUTE           >IF(m405>0)
COMPUTE WBIV      >m405/100.
COMPUTE WBOV      >(m403+m404)/100.
COMPUTE WBXF      >m412
COMPUTE WBCS      >m413
COMPUTE WBXP      >m414/100.
COMPUTE WBWK      >(m401+m402)/100.
COMPUTE           >ENDIF

* WALK BUS/METRORAIL (TRANSIT)
COMPUTE           >IF(m605>0)
COMPUTE WTIV      >m605/100.
COMPUTE WTOV      >(m603+m604)/100.
COMPUTE WTXF      >m612
COMPUTE WTCS      >m613
COMPUTE WTXP      >m614/100.
COMPUTE WTWK      >(m601+m602)/100.
COMPUTE           >ENDIF

* WALK METRORAIL
COMPUTE           >IF(m505>0)
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE WMIV      >m505/100.
COMPUTE WMOV      >(m503+m504)/100.
COMPUTE WMXF      >m512
COMPUTE WMCS      >m513
COMPUTE WMXP      >m514/100.
COMPUTE WMWK      >(m501+m502)/100.
COMPUTE           >ENDIF

* PNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE PCIV      >m319/100.
COMPUTE PCAA      >m329/100.
COMPUTE PCOV      >(m317+m318+m332)/100.
COMPUTE PCXF      >m326
COMPUTE PCCS      >m327+m331+m330/100*AUOP
COMPUTE PCXP      >m328/100.
COMPUTE PCWK      >(m315+m316)/100.
COMPUTE           >ENDIF

* PNR BUS
COMPUTE           >IF (m419>0)
COMPUTE PBIV      >m419/100.
COMPUTE PBAA      >m429/100.
COMPUTE PBOV      >(m417+m418+m432)/100.
COMPUTE PBXF      >m426
COMPUTE PBXS      >m427+m431+m430/100*AUOP
COMPUTE PBXP      >m428/100.
COMPUTE PBWK      >(m415+m416)/100.
COMPUTE           >ENDIF

* PNR BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m619>0)
COMPUTE PTIV      >m619/100.
COMPUTE PTAA      >m629/100.
COMPUTE PTOV      >(m617+m618+m632)/100.
COMPUTE PTXF      >m626
COMPUTE PTCS      >m627+m631+m630/100*AUOP
COMPUTE PTXP      >m628/100.
COMPUTE PTWK      >(m615+m616)/100.
COMPUTE           >ENDIF

* PNR METRORAIL
COMPUTE           >IF (m519>0)
COMPUTE PMIV      >m519/100.
COMPUTE PMAA      >m529/100.
COMPUTE PMOV      >(m517+m518+m532)/100.
COMPUTE PMXF      >m526
COMPUTE PMCS      >m527+m531+m530/100*AUOP
COMPUTE PMXP      >m528/100.
COMPUTE PMWK      >(m515+m516)/100.
COMPUTE           >ENDIF

* KNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE KCIV      >m319/100.
COMPUTE KCAA      >m329/100.
COMPUTE KCOV      >(m317+m318)/100.
COMPUTE KCXF      >m326
COMPUTE KCCS      >m327+m330/100*AUOP
COMPUTE KCXP      >m328/100.
COMPUTE KCWK      >(m315+m316)/100.
COMPUTE           >ENDIF

* KNR BUS
COMPUTE           >IF (m437>0)
```

## Appendix D: AEMS Fortran Control Files

```

COMPUTE KBIV      >m437/100.
COMPUTE KBAA      >m447/100.
COMPUTE KBOV      >(m435+m436)/100.
COMPUTE KBXF      >m444
COMPUTE KBCS      >m445+m448/100*AUOP
COMPUTE KBXP      >m446/100.
COMPUTE KBWK      >(m433+m434)/100.
COMPUTE           >ENDIF

```

```

* KNR BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m637>0)
COMPUTE KTIV      >m637/100.
COMPUTE KTA A     >m647/100.
COMPUTE KTOV      >(m635+m636)/100.
COMPUTE KTXF      >m644
COMPUTE KTCS      >m645+m648/100*AUOP
COMPUTE KTXP      >m646/100.
COMPUTE KTWK      >(m633+m634)/100.
COMPUTE           >ENDIF

```

```

* KNR METRORAIL
COMPUTE           >IF (m537>0)
COMPUTE KMIV      >m537/100.
COMPUTE KMAA      >m547/100.
COMPUTE KMOV      >(m535+m536)/100.
COMPUTE KMXF      >m544
COMPUTE KMCS      >m545+m548/100*AUOP
COMPUTE KMXP      >m546/100.
COMPUTE KMWK      >(m533+m534)/100.
COMPUTE           >ENDIF

```

\*CONSTANTS BY CHOICE FOR EACH PURPOSE

```

*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
PURP01 IINC 1    1>
PURP02 IINC 2    1>
PURP03 IINC 3    1>
PURP04 IINC 4    1>

```

```

TRIPIN01         >TRP1
TRIPIN02         >TRP2
TRIPIN03         >TRP3
TRIPIN04         >TRP4
TRIPFACT01       >tfi1
TRIPFACT02       >tfi2
TRIPFACT03       >tfi3
TRIPFACT04       >tfi4
COMPUTE tfi1     >1.0
COMPUTE tfi2     >1.0
COMPUTE tfi3     >1.0
COMPUTE tfi4     >1.0

```

\*  
\*OUTPUT MATRICES AND OUTPUT FACTORS BY CHOICE FOR EACH PURPOSE

```

*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
TRIPOUT01        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT01       1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT02        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT02       1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT03        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT03       1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
TRIPOUT04        1>m901      m902      m903      m904      m905      m906      m907      m908      m908      m909      m910      m911      m912      m913      m914
TRIPFACT04       1>1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00      1.00
**

```

\*\*P AND A WALK PERCENTS BY CHOICE

```

*CHOICE          1>DR ALONE  SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
WALK SEG CW 1 PCT 1>WSWM

```

## Appendix D: AEMS Fortran Control Files

```

WALK SEG CW 1 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 2 PCT 1>WSW1
WALK SEG CW 2 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 3 PCT 1>WSW2
WALK SEG CW 3 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODEL>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
*SYNTAX OF COMMAND TO ADD A COMPONENT TO A SPECIFIC WALK SEGMENT IF DESIRED
*WALK SEG CW 1 COEF1>      -0.04747  -0.04747  -0.04747  -0.04747  -0.04747  -0.04747
*WALK SEG CW 1 VAR 1>      WTSS      DTSS      DISS      WRSS      DRSS      DJSS
COMPUTE WSWM      >PCMI*PCMJ
COMPUTE WSW1      >(PCTI-PCMI)*PCMJ
COMPUTE WSW2      >(PCTI-PCMI)*(PCTJ-PCMJ)
COMPUTE WSW3      >PCMI*(PCTJ-PCMJ)
COMPUTE WSM1      >(1-PCTI)*PCMJ
COMPUTE WSM2      >(1-PCTI)*(PCTJ-PCMJ)
COMPUTE WSM2      >1-WSWM-WSW1-WSW2-WSW3-WSM1-WSM2

*NEST DEFINITIONS BY CHOICE
*CHOICE      1>DR ALONE      SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR      WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR      KNR-BU/MR      PNR-MR      KNR-MR
NEST 1,1=      1>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 1,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 2,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,1=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,2=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,3=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 3,4=      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,1      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,2      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,3      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 4,4      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,1      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,2      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,3      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 5,4      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 6,1      1>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 6,2      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 7,1      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
NEST 7,2      1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y

IGRP DEFINITION      >i813
JGRP DEFINITION      >j813
* 1 DC CORE/URBAN-DC CORE
SEGMENT 1      >      1      1
SEGMENT 1      >      3      1
* 2 DC CORE/URBAN-VA CORE
SEGMENT 2      >      1      2
SEGMENT 2      >      3      2
* 3 DC CORE/URBAN-URBAN
SEGMENT 3      >      1      3
SEGMENT 3      >      3      3
SEGMENT 3      >      1      4
SEGMENT 3      >      3      4
SEGMENT 3      >      1      5
SEGMENT 3      >      3      5
* 4 DC CORE/URBAN-OTHER
SEGMENT 4      >      1      6
SEGMENT 4      >      3      6
SEGMENT 4      >      1      7
SEGMENT 4      >      3      7

```

## Appendix D: AEMS Fortran Control Files

```

* 5 MD URBAN-DC CORE
SEGMENT 5 > 4 1
* 6 MD URBAN-VA CORE
SEGMENT 6 > 4 2
* 7 MD URBAN-URBAN
SEGMENT 7 > 4 3
SEGMENT 7 > 4 4
SEGMENT 7 > 4 5
* 8 MD URBAN-OTHER
SEGMENT 8 > 4 6
SEGMENT 8 > 4 7
* 9 VA CORE/URBAN-DC CORE
SEGMENT 9 > 2 1
SEGMENT 9 > 5 1
*10 VA CORE/URBAN-VA CORE
SEGMENT 10 > 2 2
SEGMENT 10 > 5 2
*11 VA CORE/URBAN-URBAN
SEGMENT 11 > 2 3
SEGMENT 11 > 5 3
SEGMENT 11 > 2 4
SEGMENT 11 > 5 4
SEGMENT 11 > 2 5
SEGMENT 11 > 5 5
*12 VA CORE/URBAN-OTHER
SEGMENT 12 > 2 6
SEGMENT 12 > 5 6
SEGMENT 12 > 2 7
SEGMENT 12 > 5 7
*13 MD OTHER-DC CORE
SEGMENT 13 > 6 1
*14 MD OTHER-VA CORE
SEGMENT 14 > 6 2
*15 MD OTHER-URBAN
SEGMENT 15 > 6 3
SEGMENT 15 > 6 4
SEGMENT 15 > 6 5
*16 MD OTHER-OTHER
SEGMENT 16 > 6 6
SEGMENT 16 > 6 7
*17 VA OTHER-DC CORE
SEGMENT 17 > 7 1
*18 VA OTHER-VA CORE
SEGMENT 18 > 7 2
*19 VA OTHER-URBAN
SEGMENT 19 > 7 3
SEGMENT 19 > 7 4
SEGMENT 19 > 7 5
*20 VA OTHER-OTHER
SEGMENT 20 > 7 6
SEGMENT 20 > 7 7

* SEGMENT 1
NSTC 10 IGRND TOTAL>
NSTC 11 1AUTO > 0.5 0.00000
NSTC 12 1TRANSIT > 0.5 -2.89068
NSTC 20 1TOTAL TRN >
NSTC 21 1WALK ACC > 0.5 0.00000
NSTC 22 1PNR ACC > 0.5 -0.84618
NSTC 23 1KNR ACC > 0.5 -3.05790
NSTC 30 1WLK TRN
NSTC 31 1WLK CR > 1.0 0.72167
NSTC 32 1WLK BUS > 1.0 0.64344
NSTC 33 1WLK BU/MR > 1.0 5.19110
NSTC 34 1WLK METRO > 1.0 0.00000
NSTC 40 1PNR TRN
NSTC 41 1PNR CR > 1.0 0.44068
NSTC 42 1PNR BUS > 1.0 0.63538

```



## Appendix D: AEMS Fortran Control Files

```
NSTC 43 1PNR BU/MR > 1.0 4.46631
NSTC 44 1PNR METRO > 1.0 0.00000
NSTC 50 1KNR TRN
NSTC 51 1KNR CR > 1.0 1.50131
NSTC 52 1KNR BUS > 1.0 5.96561
NSTC 53 1KNR BU/MR > 1.0 6.26754
NSTC 54 1KNR METRO > 1.0 0.00000
NSTC 60 1AUTO
NSTC 61 1LOV > 1.0 0.00000
NSTC 62 1HOV > 0.5 -4.48085
NSTC 70 1HOV
NSTC 71 1HOV2 > 1.0 0.00000
NSTC 72 1HOV3+ > 1.0 -4.30427
* SEGMENT 2
NSTC 10 2GRND TOTAL>
NSTC 11 2AUTO > 0.5 0.00000
NSTC 12 2TRANSIT > 0.5 14.80397
NSTC 20 2TOTAL TRN >
NSTC 21 2WALK ACC > 0.5 0.00000
NSTC 22 2PNR ACC > 0.5 -5.57714
NSTC 23 2KNR ACC > 0.5 -4.31947
NSTC 30 2WLK TRN
NSTC 31 2WLK CR > 1.0 -1.40295
NSTC 32 2WLK BUS > 1.0 -50.51343
NSTC 33 2WLK BU/MR > 1.0 -0.47818
NSTC 34 2WLK METRO > 1.0 0.00000
NSTC 40 2PNR TRN
NSTC 41 2PNR CR > 1.0 -2.51445
NSTC 42 2PNR BUS > 1.0 -2.51445
NSTC 43 2PNR BU/MR > 1.0 -2.51445
NSTC 44 2PNR METRO > 1.0 0.00000
NSTC 50 2KNR TRN
NSTC 51 2KNR CR > 1.0 -0.87117
NSTC 52 2KNR BUS > 1.0 -0.87117
NSTC 53 2KNR BU/MR > 1.0 -0.87117
NSTC 54 2KNR METRO > 1.0 0.00000
NSTC 60 2AUTO
NSTC 61 2LOV > 1.0 0.00000
NSTC 62 2HOV > 0.5 -2.49041
NSTC 70 2HOV
NSTC 71 2HOV2 > 1.0 0.00000
NSTC 72 2HOV3+ > 1.0 -6.00352
* SEGMENT 3
NSTC 10 3GRND TOTAL>
NSTC 11 3AUTO > 0.5 0.00000
NSTC 12 3TRANSIT > 0.5 -1.26930
NSTC 20 3TOTAL TRN >
NSTC 21 3WALK ACC > 0.5 0.00000
NSTC 22 3PNR ACC > 0.5 -5.23774
NSTC 23 3KNR ACC > 0.5 -4.24737
NSTC 30 3WLK TRN
NSTC 31 3WLK CR > 1.0 -0.12749
NSTC 32 3WLK BUS > 1.0 -0.34196
NSTC 33 3WLK BU/MR > 1.0 0.99865
NSTC 34 3WLK METRO > 1.0 0.00000
NSTC 40 3PNR TRN
NSTC 41 3PNR CR > 1.0 4.74896
NSTC 42 3PNR BUS > 1.0 4.04162
NSTC 43 3PNR BU/MR > 1.0 9.26987
NSTC 44 3PNR METRO > 1.0 0.00000
NSTC 50 3KNR TRN
NSTC 51 3KNR CR > 1.0 1.82164
NSTC 52 3KNR BUS > 1.0 3.65006
NSTC 53 3KNR BU/MR > 1.0 6.62155
NSTC 54 3KNR METRO > 1.0 0.00000
NSTC 60 3AUTO
NSTC 61 3LOV > 1.0 0.00000
NSTC 62 3HOV > 0.5 -2.69346
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 70 3HOV
NSTC 71 3HOV2 > 1.0 0.00000
NSTC 72 3HOV3+ > 1.0 -2.56739
* SEGMENT 4
NSTC 10 4GRND TOTAL>
NSTC 11 4AUTO > 0.5 0.00000
NSTC 12 4TRANSIT > 0.5 0.92721
NSTC 20 4TOTAL TRN >
NSTC 21 4WALK ACC > 0.5 0.00000
NSTC 22 4PNR ACC > 0.5 -5.24326
NSTC 23 4KNR ACC > 0.5 -5.14451
NSTC 30 4WLK TRN
NSTC 31 4WLK CR > 1.0 11.51204
NSTC 32 4WLK BUS > 1.0 -5.24510
NSTC 33 4WLK BU/MR > 1.0 -2.84670
NSTC 34 4WLK METRO > 1.0 0.00000
NSTC 40 4PNR TRN
NSTC 41 4PNR CR > 1.0 28.92420
NSTC 42 4PNR BUS > 1.0 -0.36191
NSTC 43 4PNR BU/MR > 1.0 -0.36191
NSTC 44 4PNR METRO > 1.0 0.00000
NSTC 50 4KNR TRN
NSTC 51 4KNR CR > 1.0 32.89755
NSTC 52 4KNR BUS > 1.0 4.37981
NSTC 53 4KNR BU/MR > 1.0 3.37564
NSTC 54 4KNR METRO > 1.0 0.00000
NSTC 60 4AUTO
NSTC 61 4LOV > 1.0 0.00000
NSTC 62 4HOV > 0.5 -2.28190
NSTC 70 4HOV
NSTC 71 4HOV2 > 1.0 0.00000
NSTC 72 4HOV3+ > 1.0 -3.47119
* SEGMENT 5
NSTC 10 5GRND TOTAL>
NSTC 11 5AUTO > 0.5 0.00000
NSTC 12 5TRANSIT > 0.5 -3.82517
NSTC 20 5TOTAL TRN >
NSTC 21 5WALK ACC > 0.5 0.00000
NSTC 22 5PNR ACC > 0.5 -3.03211
NSTC 23 5KNR ACC > 0.5 -3.98292
NSTC 30 5WLK TRN
NSTC 31 5WLK CR > 1.0 0.07440
NSTC 32 5WLK BUS > 1.0 0.83960
NSTC 33 5WLK BU/MR > 1.0 -0.04810
NSTC 34 5WLK METRO > 1.0 0.00000
NSTC 40 5PNR TRN
NSTC 41 5PNR CR > 1.0 0.96121
NSTC 42 5PNR BUS > 1.0 0.96121
NSTC 43 5PNR BU/MR > 1.0 9.38519
NSTC 44 5PNR METRO > 1.0 0.00000
NSTC 50 5KNR TRN
NSTC 51 5KNR CR > 1.0 -0.32124
NSTC 52 5KNR BUS > 1.0 -0.32124
NSTC 53 5KNR BU/MR > 1.0 -0.32124
NSTC 54 5KNR METRO > 1.0 0.00000
NSTC 60 5AUTO
NSTC 61 5LOV > 1.0 0.00000
NSTC 62 5HOV > 0.5 -6.26875
NSTC 70 5HOV
NSTC 71 5HOV2 > 1.0 0.00000
NSTC 72 5HOV3+ > 1.0 -6.82496
* SEGMENT 6
NSTC 10 6GRND TOTAL>
NSTC 11 6AUTO > 0.5 0.00000
NSTC 12 6TRANSIT > 0.5 7.14308
NSTC 20 6TOTAL TRN >
NSTC 21 6WALK ACC > 0.5 0.00000
NSTC 22 6PNR ACC > 0.5 -8.83064
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 23 6KNR ACC > 0.5 -8.83064
NSTC 30 6WLK TRN
NSTC 31 6WLK CR > 1.0 -6.07392
NSTC 32 6WLK BUS > 1.0 -6.07392
NSTC 33 6WLK BU/MR > 1.0 -6.07392
NSTC 34 6WLK METRO > 1.0 0.00000
NSTC 40 6PNR TRN
NSTC 41 6PNR CR > 1.0 0.00001
NSTC 42 6PNR BUS > 1.0 0.00001
NSTC 43 6PNR BU/MR > 1.0 0.00001
NSTC 44 6PNR METRO > 1.0 0.00000
NSTC 50 6KNR TRN
NSTC 51 6KNR CR > 1.0 0.00001
NSTC 52 6KNR BUS > 1.0 0.00001
NSTC 53 6KNR BU/MR > 1.0 0.00001
NSTC 54 6KNR METRO > 1.0 0.00000
NSTC 60 6AUTO
NSTC 61 6LOV > 1.0 0.00000
NSTC 62 6HOV > 0.5 0.00001
NSTC 70 6HOV
NSTC 71 6HOV2 > 1.0 0.00000
NSTC 72 6HOV3+ > 1.0 0.00001
* SEGMENT 7
NSTC 10 7GRND TOTAL>
NSTC 11 7AUTO > 0.5 0.00000
NSTC 12 7TRANSIT > 0.5 -2.60144
NSTC 20 7TOTAL TRN >
NSTC 21 7WALK ACC > 0.5 0.00000
NSTC 22 7PNR ACC > 0.5 -3.67073
NSTC 23 7KNR ACC > 0.5 -4.82536
NSTC 30 7WLK TRN
NSTC 31 7WLK CR > 1.0 2.06461
NSTC 32 7WLK BUS > 1.0 2.73223
NSTC 33 7WLK BU/MR > 1.0 4.35163
NSTC 34 7WLK METRO > 1.0 0.00000
NSTC 40 7PNR TRN
NSTC 41 7PNR CR > 1.0 0.00001
NSTC 42 7PNR BUS > 1.0 8.45233
NSTC 43 7PNR BU/MR > 1.0 0.00001
NSTC 44 7PNR METRO > 1.0 0.00000
NSTC 50 7KNR TRN
NSTC 51 7KNR CR > 1.0 1.84142
NSTC 52 7KNR BUS > 1.0 7.11052
NSTC 53 7KNR BU/MR > 1.0 1.84142
NSTC 54 7KNR METRO > 1.0 0.00000
NSTC 60 7AUTO
NSTC 61 7LOV > 1.0 0.00000
NSTC 62 7HOV > 0.5 -2.30306
NSTC 70 7HOV
NSTC 71 7HOV2 > 1.0 0.00000
NSTC 72 7HOV3+ > 1.0 -3.54959
* SEGMENT 8
NSTC 10 8GRND TOTAL>
NSTC 11 8AUTO > 0.5 0.00000
NSTC 12 8TRANSIT > 0.5 -2.62411
NSTC 20 8TOTAL TRN >
NSTC 21 8WALK ACC > 0.5 0.00000
NSTC 22 8PNR ACC > 0.5 -1.38606
NSTC 23 8KNR ACC > 0.5 -2.99286
NSTC 30 8WLK TRN
NSTC 31 8WLK CR > 1.0 14.07210
NSTC 32 8WLK BUS > 1.0 7.88250
NSTC 33 8WLK BU/MR > 1.0 7.60716
NSTC 34 8WLK METRO > 1.0 0.00000
NSTC 40 8PNR TRN
NSTC 41 8PNR CR > 1.0 0.00001
NSTC 42 8PNR BUS > 1.0 0.00001
NSTC 43 8PNR BU/MR > 1.0 0.00001
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 44 8PNR METRO > 1.0 0.00000
NSTC 50 8KNR TRN
NSTC 51 8KNR CR > 1.0 0.00001
NSTC 52 8KNR BUS > 1.0 10.95050
NSTC 53 8KNR BU/MR > 1.0 0.00001
NSTC 54 8KNR METRO > 1.0 0.00000
NSTC 60 8AUTO
NSTC 61 8LOV > 1.0 0.00000
NSTC 62 8HOV > 0.5 -1.87592
NSTC 70 8HOV
NSTC 71 8HOV2 > 1.0 0.00000
NSTC 72 8HOV3+ > 1.0 -3.24646
* SEGMENT 9
NSTC 10 9GRND TOTAL>
NSTC 11 9AUTO > 0.5 0.00000
NSTC 12 9TRANSIT > 0.5 5.33242
NSTC 20 9TOTAL TRN >
NSTC 21 9WALK ACC > 0.5 0.00000
NSTC 22 9PNR ACC > 0.5 -9.91438
NSTC 23 9KNR ACC > 0.5 -15.91141
NSTC 30 9WLK TRN
NSTC 31 9WLK CR > 1.0 -2.95276
NSTC 32 9WLK BUS > 1.0 -23.87797
NSTC 33 9WLK BU/MR > 1.0 -18.11138
NSTC 34 9WLK METRO > 1.0 0.00000
NSTC 40 9PNR TRN
NSTC 41 9PNR CR > 1.0 -0.96713
NSTC 42 9PNR BUS > 1.0 -0.96713
NSTC 43 9PNR BU/MR > 1.0 1.63569
NSTC 44 9PNR METRO > 1.0 0.00000
NSTC 50 9KNR TRN
NSTC 51 9KNR CR > 1.0 3.36069
NSTC 52 9KNR BUS > 1.0 3.36069
NSTC 53 9KNR BU/MR > 1.0 21.22636
NSTC 54 9KNR METRO > 1.0 0.00000
NSTC 60 9AUTO
NSTC 61 9LOV > 1.0 0.00000
NSTC 62 9HOV > 0.5 -5.05306
NSTC 70 9HOV
NSTC 71 9HOV2 > 1.0 0.00000
NSTC 72 9HOV3+ > 1.0 -6.75105
* SEGMENT 10
NSTC 1010GRND TOTAL>
NSTC 1110AUTO > 0.5 0.00000
NSTC 1210TRANSIT > 0.5 -2.41530
NSTC 2010TOTAL TRN >
NSTC 2110WALK ACC > 0.5 0.00000
NSTC 2210PNR ACC > 0.5 -3.08011
NSTC 2310KNR ACC > 0.5 -4.78478
NSTC 3010WLK TRN
NSTC 3110WLK CR > 1.0 -1.50889
NSTC 3210WLK BUS > 1.0 -5.99846
NSTC 3310WLK BU/MR > 1.0 -1.50889
NSTC 3410WLK METRO > 1.0 0.00000
NSTC 4010PNR TRN
NSTC 4110PNR CR > 1.0 0.00001
NSTC 4210PNR BUS > 1.0 0.00001
NSTC 4310PNR BU/MR > 1.0 0.00001
NSTC 4410PNR METRO > 1.0 0.00000
NSTC 5010KNR TRN
NSTC 5110KNR CR > 1.0 2.63356
NSTC 5210KNR BUS > 1.0 2.63356
NSTC 5310KNR BU/MR > 1.0 23.20613
NSTC 5410KNR METRO > 1.0 0.00000
NSTC 6010AUTO
NSTC 6110LOV > 1.0 0.00000
NSTC 6210HOV > 0.5 -3.79733
NSTC 7010HOV

```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 7110HOV2 > 1.0 0.00000
NSTC 7210HOV3+ > 1.0 -4.48043
* SEGMENT 11
NSTC 1011GRND TOTAL>
NSTC 1111AUTO > 0.5 0.00000
NSTC 1211TRANSIT > 0.5 -1.32035
NSTC 2011TOTAL TRN >
NSTC 2111WALK ACC > 0.5 0.00000
NSTC 2211PNR ACC > 0.5 -3.89313
NSTC 2311KNR ACC > 0.5 -8.06954
NSTC 3011WLK TRN
NSTC 3111WLK CR > 1.0 -2.20243
NSTC 3211WLK BUS > 1.0 -5.03132
NSTC 3311WLK BU/MR > 1.0 -1.72273
NSTC 3411WLK METRO > 1.0 0.00000
NSTC 4011PNR TRN
NSTC 4111PNR CR > 1.0 0.00001
NSTC 4211PNR BUS > 1.0 0.00001
NSTC 4311PNR BU/MR > 1.0 0.00001
NSTC 4411PNR METRO > 1.0 0.00000
NSTC 5011KNR TRN
NSTC 5111KNR CR > 1.0 5.91367
NSTC 5211KNR BUS > 1.0 5.91367
NSTC 5311KNR BU/MR > 1.0 14.23915
NSTC 5411KNR METRO > 1.0 0.00000
NSTC 6011AUTO
NSTC 6111LOV > 1.0 0.00000
NSTC 6211HOV > 0.5 -3.09923
NSTC 7011HOV
NSTC 7111HOV2 > 1.0 0.00000
NSTC 7211HOV3+ > 1.0 -3.38305
* SEGMENT 12
NSTC 1012GRND TOTAL>
NSTC 1112AUTO > 0.5 0.00000
NSTC 1212TRANSIT > 0.5 -1.27127
NSTC 2012TOTAL TRN >
NSTC 2112WALK ACC > 0.5 0.00000
NSTC 2212PNR ACC > 0.5 -3.54605
NSTC 2312KNR ACC > 0.5 -7.88962
NSTC 3012WLK TRN
NSTC 3112WLK CR > 1.0 2.92036
NSTC 3212WLK BUS > 1.0 -6.75103
NSTC 3312WLK BU/MR > 1.0 -6.64540
NSTC 3412WLK METRO > 1.0 0.00000
NSTC 4012PNR TRN
NSTC 4112PNR CR > 1.0 0.00001
NSTC 4212PNR BUS > 1.0 0.00001
NSTC 4312PNR BU/MR > 1.0 0.00001
NSTC 4412PNR METRO > 1.0 0.00000
NSTC 5012KNR TRN
NSTC 5112KNR CR > 1.0 0.00001
NSTC 5212KNR BUS > 1.0 10.01262
NSTC 5312KNR BU/MR > 1.0 0.00001
NSTC 5412KNR METRO > 1.0 0.00000
NSTC 6012AUTO
NSTC 6112LOV > 1.0 0.00000
NSTC 6212HOV > 0.5 -2.01527
NSTC 7012HOV
NSTC 7112HOV2 > 1.0 0.00000
NSTC 7212HOV3+ > 1.0 -2.05555
* SEGMENT 13
NSTC 1013GRND TOTAL>
NSTC 1113AUTO > 0.5 0.00000
NSTC 1213TRANSIT > 0.5 -3.74500
NSTC 2013TOTAL TRN >
NSTC 2113WALK ACC > 0.5 0.00000
NSTC 2213PNR ACC > 0.5 -1.34272
NSTC 2313KNR ACC > 0.5 -3.05848
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 3013WLK TRN
NSTC 3113WLK CR > 1.0 38.31333
NSTC 3213WLK BUS > 1.0 3.24213
NSTC 3313WLK BU/MR > 1.0 3.40520
NSTC 3413WLK METRO > 1.0 0.00000
NSTC 4013PNR TRN
NSTC 4113PNR CR > 1.0 11.89934
NSTC 4213PNR BUS > 1.0 2.55923
NSTC 4313PNR BU/MR > 1.0 9.66869
NSTC 4413PNR METRO > 1.0 0.00000
NSTC 5013KNR TRN
NSTC 5113KNR CR > 1.0 8.79815
NSTC 5213KNR BUS > 1.0 6.16253
NSTC 5313KNR BU/MR > 1.0 11.38694
NSTC 5413KNR METRO > 1.0 0.00000
NSTC 6013AUTO
NSTC 6113LOV > 1.0 0.00000
NSTC 6213HOV > 0.5 -4.57538
NSTC 7013HOV
NSTC 7113HOV2 > 1.0 0.00000
NSTC 7213HOV3+ > 1.0 -4.08303
* SEGMENT 14
NSTC 1014GRND TOTAL>
NSTC 1114AUTO > 0.5 0.00000
NSTC 1214TRANSIT > 0.5 7.45330
NSTC 2014TOTAL TRN >
NSTC 2114WALK ACC > 0.5 0.00000
NSTC 2214PNR ACC > 0.5 -4.22475
NSTC 2314KNR ACC > 0.5 -4.91582
NSTC 3014WLK TRN
NSTC 3114WLK CR > 1.0 -6.43934
NSTC 3214WLK BUS > 1.0 -6.43934
NSTC 3314WLK BU/MR > 1.0 -6.43934
NSTC 3414WLK METRO > 1.0 0.00000
NSTC 4014PNR TRN
NSTC 4114PNR CR > 1.0 18.24717
NSTC 4214PNR BUS > 1.0 14.90055
NSTC 4314PNR BU/MR > 1.0 55.66924
NSTC 4414PNR METRO > 1.0 0.00000
NSTC 5014KNR TRN
NSTC 5114KNR CR > 1.0 21.42469
NSTC 5214KNR BUS > 1.0 6.43581
NSTC 5314KNR BU/MR > 1.0 6.43581
NSTC 5414KNR METRO > 1.0 0.00000
NSTC 6014AUTO
NSTC 6114LOV > 1.0 0.00000
NSTC 6214HOV > 0.5 0.00001
NSTC 7014HOV
NSTC 7114HOV2 > 1.0 0.00000
NSTC 7214HOV3+ > 1.0 0.00001
* SEGMENT 15
NSTC 1015GRND TOTAL>
NSTC 1115AUTO > 0.5 0.00000
NSTC 1215TRANSIT > 0.5 -2.84274
NSTC 2015TOTAL TRN >
NSTC 2115WALK ACC > 0.5 0.00000
NSTC 2215PNR ACC > 0.5 -3.01395
NSTC 2315KNR ACC > 0.5 -3.23785
NSTC 3015WLK TRN
NSTC 3115WLK CR > 1.0 20.31630
NSTC 3215WLK BUS > 1.0 5.01207
NSTC 3315WLK BU/MR > 1.0 6.12937
NSTC 3415WLK METRO > 1.0 0.00000
NSTC 4015PNR TRN
NSTC 4115PNR CR > 1.0 9.37253
NSTC 4215PNR BUS > 1.0 2.23006
NSTC 4315PNR BU/MR > 1.0 12.81242
NSTC 4415PNR METRO > 1.0 0.00000
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 5015KNR TRN
NSTC 5115KNR CR > 1.0 6.11554
NSTC 5215KNR BUS > 1.0 6.01886
NSTC 5315KNR BU/MR > 1.0 13.42917
NSTC 5415KNR METRO > 1.0 0.00000
NSTC 6015AUTO
NSTC 6115LOV > 1.0 0.00000
NSTC 6215HOV > 0.5 -2.70601
NSTC 7015HOV
NSTC 7115HOV2 > 1.0 0.00000
NSTC 7215HOV3+ > 1.0 -2.75267
* SEGMENT 16
NSTC 1016GRND TOTAL>
NSTC 1116AUTO > 0.5 0.00000
NSTC 1216TRANSIT > 0.5 -3.11035
NSTC 2016TOTAL TRN >
NSTC 2116WALK ACC > 0.5 0.00000
NSTC 2216PNR ACC > 0.5 -1.01354
NSTC 2316KNR ACC > 0.5 -4.01929
NSTC 3016WLK TRN
NSTC 3116WLK CR > 1.0 14.35547
NSTC 3216WLK BUS > 1.0 9.20912
NSTC 3316WLK BU/MR > 1.0 3.96667
NSTC 3416WLK METRO > 1.0 0.00000
NSTC 4016PNR TRN
NSTC 4116PNR CR > 1.0 0.00001
NSTC 4216PNR BUS > 1.0 6.75744
NSTC 4316PNR BU/MR > 1.0 0.00001
NSTC 4416PNR METRO > 1.0 0.00000
NSTC 5016KNR TRN
NSTC 5116KNR CR > 1.0 0.00001
NSTC 5216KNR BUS > 1.0 8.82285
NSTC 5316KNR BU/MR > 1.0 0.00001
NSTC 5416KNR METRO > 1.0 0.00000
NSTC 6016AUTO
NSTC 6116LOV > 1.0 0.00000
NSTC 6216HOV > 0.5 -0.90927
NSTC 7016HOV
NSTC 7116HOV2 > 1.0 0.00000
NSTC 7216HOV3+ > 1.0 -1.31728
* SEGMENT 17
NSTC 1017GRND TOTAL>
NSTC 1117AUTO > 0.5 0.00000
NSTC 1217TRANSIT > 0.5 -0.62062
NSTC 2017TOTAL TRN >
NSTC 2117WALK ACC > 0.5 0.00000
NSTC 2217PNR ACC > 0.5 -4.15351
NSTC 2317KNR ACC > 0.5 -6.28041
NSTC 3017WLK TRN
NSTC 3117WLK CR > 1.0 -5.59293
NSTC 3217WLK BUS > 1.0 -5.59293
NSTC 3317WLK BU/MR > 1.0 -3.45907
NSTC 3417WLK METRO > 1.0 0.00000
NSTC 4017PNR TRN
NSTC 4117PNR CR > 1.0 -0.20558
NSTC 4217PNR BUS > 1.0 -0.20558
NSTC 4317PNR BU/MR > 1.0 5.18899
NSTC 4417PNR METRO > 1.0 0.00000
NSTC 5017KNR TRN
NSTC 5117KNR CR > 1.0 2.15306
NSTC 5217KNR BUS > 1.0 2.15306
NSTC 5317KNR BU/MR > 1.0 9.16471
NSTC 5417KNR METRO > 1.0 0.00000
NSTC 6017AUTO
NSTC 6117LOV > 1.0 0.00000
NSTC 6217HOV > 0.5 -6.60011
NSTC 7017HOV
NSTC 7117HOV2 > 1.0 0.00000
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 7217HOV3+ > 1.0 -7.94344
* SEGMENT 18
NSTC 1018GRND TOTAL>
NSTC 1118AUTO > 0.5 0.00000
NSTC 1218TRANSIT > 0.5 -2.80302
NSTC 2018TOTAL TRN >
NSTC 2118WALK ACC > 0.5 0.00000
NSTC 2218PNR ACC > 0.5 -3.08275
NSTC 2318KNR ACC > 0.5 -6.01764
NSTC 3018WLK TRN
NSTC 3118WLK CR > 1.0 0.00001
NSTC 3218WLK BUS > 1.0 1.65280
NSTC 3318WLK BU/MR > 1.0 5.30476
NSTC 3418WLK METRO > 1.0 0.00000
NSTC 4018PNR TRN
NSTC 4118PNR CR > 1.0 -2.83330
NSTC 4218PNR BUS > 1.0 -2.83330
NSTC 4318PNR BU/MR > 1.0 -2.83330
NSTC 4418PNR METRO > 1.0 0.00000
NSTC 5018KNR TRN
NSTC 5118KNR CR > 1.0 0.00001
NSTC 5218KNR BUS > 1.0 0.00001
NSTC 5318KNR BU/MR > 1.0 0.00001
NSTC 5418KNR METRO > 1.0 0.00000
NSTC 6018AUTO
NSTC 6118LOV > 1.0 0.00000
NSTC 6218HOV > 0.5 -2.39193
NSTC 7018HOV
NSTC 7118HOV2 > 1.0 0.00000
NSTC 7218HOV3+ > 1.0 -2.47964
* SEGMENT 19
NSTC 1019GRND TOTAL>
NSTC 1119AUTO > 0.5 0.00000
NSTC 1219TRANSIT > 0.5 -2.34424
NSTC 2019TOTAL TRN >
NSTC 2119WALK ACC > 0.5 0.00000
NSTC 2219PNR ACC > 0.5 -2.93519
NSTC 2319KNR ACC > 0.5 -3.71338
NSTC 3019WLK TRN
NSTC 3119WLK CR > 1.0 -0.74980
NSTC 3219WLK BUS > 1.0 -3.24727
NSTC 3319WLK BU/MR > 1.0 4.41295
NSTC 3419WLK METRO > 1.0 0.00000
NSTC 4019PNR TRN
NSTC 4119PNR CR > 1.0 -5.07238
NSTC 4219PNR BUS > 1.0 -5.07238
NSTC 4319PNR BU/MR > 1.0 -0.43380
NSTC 4419PNR METRO > 1.0 0.00000
NSTC 5019KNR TRN
NSTC 5119KNR CR > 1.0 -4.46187
NSTC 5219KNR BUS > 1.0 -4.46187
NSTC 5319KNR BU/MR > 1.0 -4.46187
NSTC 5419KNR METRO > 1.0 0.00000
NSTC 6019AUTO
NSTC 6119LOV > 1.0 0.00000
NSTC 6219HOV > 0.5 -3.55414
NSTC 7019HOV
NSTC 7119HOV2 > 1.0 0.00000
NSTC 7219HOV3+ > 1.0 -3.84660
* SEGMENT 20
NSTC 1020GRND TOTAL>
NSTC 1120AUTO > 0.5 0.00000
NSTC 1220TRANSIT > 0.5 -0.41089
NSTC 2020TOTAL TRN >
NSTC 2120WALK ACC > 0.5 0.00000
NSTC 2220PNR ACC > 0.5 -49.33698
NSTC 2320KNR ACC > 0.5 -34.42600
NSTC 3020WLK TRN
```



## Appendix D: AEMS Fortran Control Files

---

NSTC 3120WLK CR	>	1.0	-7.16790
NSTC 3220WLK BUS	>	1.0	-6.06277
NSTC 3320WLK BU/MR	>	1.0	-32.61037
NSTC 3420WLK METRO	>	1.0	0.00000
NSTC 4020PNR TRN			
NSTC 4120PNR CR	>	1.0	0.00001
NSTC 4220PNR BUS	>	1.0	21.73288
NSTC 4320PNR BU/MR	>	1.0	0.00001
NSTC 4420PNR METRO	>	1.0	0.00000
NSTC 5020KNR TRN			
NSTC 5120KNR CR	>	1.0	-12.65357
NSTC 5220KNR BUS	>	1.0	1.95969
NSTC 5320KNR BU/MR	>	1.0	-12.65357
NSTC 5420KNR METRO	>	1.0	0.00000
NSTC 6020AUTO			
NSTC 6120LOV	>	1.0	0.00000
NSTC 6220HOV	>	0.5	-0.82373
NSTC 7020HOV			
NSTC 7120HOV2	>	1.0	0.00000
NSTC 7220HOV3+	>	1.0	-1.28887

\*DOWNTOWN=8

*SELI	>	8
-------	---	---

\*UNION STATION=64

*SELI	>	64
-------	---	----

\* =122

*SELI	>	122
-------	---	-----

\*BETHESDA=345

*SELI	>	345
-------	---	-----

\*SILVER SPRING=362

*SELI	>	362
-------	---	-----

\*N.SILVER SPRING=464

*SELI	>	464
-------	---	-----

\* =475

*SELI	>	475
-------	---	-----

\*SHADY GROVE RD=578

*SELI	>	578
-------	---	-----

\* =787

*SELI	>	787
-------	---	-----

\*ANDREWS AFB=829

*SELI	>	829
-------	---	-----

\*NEW CARROLTON=927

*SELI	>	927
-------	---	-----

\*BRISTOL=972

*SELI	>	972
-------	---	-----

\*FREDERICK=1043

*SELI	>	1043
-------	---	------

\*JESSUP=1080

*SELI	>	1080
-------	---	------

\*SCAGGSVILLE=1091

*SELI	>	1091
-------	---	------

\*WALDORF=1216

*SELI	>	1216
-------	---	------

## Appendix D: AEMS Fortran Control Files

---

```
*PENTAGON=1231
*SELI      >    1231

*ROSSLYN=1236
*SELI      >    1236

*ALEXANDRIA=1337
*SELI      >    1337

* =1455
*SELI      >    1455

*SPRINGFIELD=1502
*SELI      >    1502

* =1511
*SELI      >    1511

*TYSONS CRNR=1537
*SELI      >    1537

*FT BELVOIR=1554
*SELI      >    1554

*VIENNA=1619
*SELI      >    1619

*DULES AP=1698
*SELI      >    1698

*RESTON=1716
*SELI      >    1716

*LEESBURG=1842
*SELI      >    1842

*BRUNSWICK=1863
*SELI      >    1863

*DALE CITY=1942
*SELI      >    1942

*MANASSAS=1967
*SELI      >    1967

*SPOTSYLVANIA=2110
*SELI      >    2110

* =2055
*SELI      >    2055

*SELJ      >      8
*SELJ      >     63
*SELJ      >     64
*SELJ      >     77
*SELJ      >    100
*SELJ      >    344
*SELJ      >    345
*SELJ      >    362
*SELJ      >   1231
*SELJ      >   1236
*SELJ      >   1265
*SELJ      >   1337
*SELJ      >   1537

*SELI      >   523
```

## Appendix D: AEMS Fortran Control Files

```

*SELJ          >      9

TRACE          >      0
* OUTPUT %    >
*PROCSEL      >
PRINT MS      >NHO_NL_MC.PRN
INPUT PRINT FILE >NHO_NL_MC.PRN
INPUT GOALS   >NHO_NL_MC.GOL
INFILE 1     >nho_income.ptt
INFILE 2     >hwyop.skm
INFILE 3     >TRNOP_CR.SKM
INFILE 4     >TRNOP_AB.SKM
INFILE 5     >TRNOP_MR.SKM
INFILE 6     >TRNOP_BM.SKM
ZINFILE 8    >ZONEV2.A2F
OUTFILE 9    >NHO_NL_MC.MTT

* FTA USER BENEFITS SPECIFICATIONS
*FTA RESULTS FILE >NHB_NL_MC.BEN
FTA TRANSIT COEFF >-0.02860
FTA AUTO COEFF    >-0.02860
FTA PURPOSE NAME  >NHB
FTA PERIOD NAME   >ALLDAY
FTA ALTER. NAME   >CALIB
*CHOICE          1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
FTA AUTO NEST    >      1      1
FTA MOTORIZED?  1>Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y
FTA TRANSIT?    1>      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y      Y

```

## 8 nhw\_nl\_mc.ctf

```

NHW OP NESTED LOGIT MC - #DATE: 9/18/2011 #VER: 21
CHOICE          1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
*
*LOGIT COEFFICIENTS BY CHOICE FOR EACH SKIM (NO INPUT SKIM IS
*EQUIVALENT TO A CONSTANT)
*CHOICE          1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
COEF01:IVTT     1>-0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860 -0.02860
SKIM01:IVTT     1>DAIV  S2IV  S3IV  WCIV  WBIV  WTIV  WMIV  PCIV  KCIV  PBIV  KBIV  PTIV  KTIV  PMIV  KMIV
COEF02:AUTO ACC 1>      -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290 -0.04290
SKIM02:AUTO ACC 1>      PCAA  KCAA  PBAA  KBAA  PTAA  KTAA  PMAA  KMAA
COEF03:TERM/OVTT 1>-0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150
SKIM03:TERM/OVTT 1>DATE  S2TE  S3TE  WCOV  WBOV  WTOV  WMOV  PCOV  KCOV  PBOV  KBOV  PTOV  KTOV  PMOV  KMOV
* LIMIT COEF 04 TO PURPOSE 1
COEF PURP04     >1
COEF04:COST INC1 1>-0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994
SKIM04:COST INC1 1>DACS  S2CS  S3CS  WCCS  WBCS  WTCS  WMCS  PCCS  KCCS  PBCS  KBCS  PTCS  KTCS  PMCS  KMCS
* LIMIT COEF 05 TO PURPOSE 2
COEF PURP05     >2
COEF05:COST INC2 1>-0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994
SKIM05:COST INC2 1>DACS  S2CS  S3CS  WCCS  WBCS  WTCS  WMCS  PCCS  KCCS  PBCS  KBCS  PTCS  KTCS  PMCS  KMCS
* LIMIT COEF 06 TO PURPOSE 3
COEF PURP06     >3
COEF06:COST INC3 1>-0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994
SKIM06:COST INC3 1>DACS  S2CS  S3CS  WCCS  WBCS  WTCS  WMCS  PCCS  KCCS  PBCS  KBCS  PTCS  KTCS  PMCS  KMCS
COEF PURP07     >4
* LIMIT COEF 07 TO PURPOSE 4
COEF07:COST INC4 1>-0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994 -0.00994
SKIM07:COST INC4 1>DACS  S2CS  S3CS  WCCS  WBCS  WTCS  WMCS  PCCS  KCCS  PBCS  KBCS  PTCS  KTCS  PMCS  KMCS
COEF08:TRN XFERS 1>      -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000 -0.00000
SKIM08:TRN XFERS 1>      WCXF  WBXF  WTXF  WMXF  PCXF  KCFX  PBXF  KBXF  PTXF  KTXF  PMXF  KMXF
COEF09:TRN BRDPEN 1>      -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150 -0.07150
SKIM09:TRN BRDPEN 1>      WCXP  WBXP  WTXP  WMXP  PCXP  KCXP  PBXP  KBXP  PTXP  KTXP  PMXP  KMXP

```

## Appendix D: AEMS Fortran Control Files

```

*WALK WEIGHT
COEF10:TRN WLKWT 1>          -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720 -0.05720
SKIM10:TRN WLKWT 1>          WCWK      WBWK      TWWK      WMWK      PCWK      KCWK      PBWK      KBWK      PTWK      KTWK      PMWK      KMWK

*SYNTAX TO LIMIT UTILITY ELEMENT TO A PARTICULAR WALK SEGMENT IN THIS EXAMPLE
*   COEF 18 APPLIES ONLY TO WALK SEGMENT 1
*COEF WLKSEG18      >1

* ASSUMED MATRIX ORGANIZATION
* FILE 1 TRIP TABLE (SEPARATE FOR EACH PURPOSE)
* 1 INCOME 1 (HOME-BASED)/ALL NHB TRIPS
* 2 INCOME 2 (HOME-BASED)
* 3 INCOME 3 (HOME-BASED)
* 4 INCOME 4 (HOME-BASED)
*
* FILE 2 HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 SOV  TIME (MIN)
* 2 SOV  DIST (0.1 MILES)
* 3 SOV  TOLL (2007 CENTS)
* 4 HOV2 TIME (MIN)
* 5 HOV2 DIST (0.1 MILES)
* 6 HOV2 TOLL (2007 CENTS)
* 7 HOV3+ TIME (MIN)
* 8 HOV3+ DIST (0.1 MILES)
* 9 HOV3+ TOLL (2007 CENTS)
*
* FILE 3=COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 4=BUS SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 5=METRO RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* FILE 6=BUS+METRO RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
* 1 WLK ACC/EGR (.01 MIN) 15 PNR ACC/EGR (.01 MIN) 33 KNR ACC/EGR (.01 MIN)
* 2 WLK OTHER (.01 MIN) 16 PNR OTHER (.01 MIN) 34 KNR OTHER (.01 MIN)
* 3 WLK IWAIT (.01 MIN) 17 PNR IWAIT (.01 MIN) 35 KNR IWAIT (.01 MIN)
* 4 WLK XWAIT (.01 MIN) 18 PNR XWAIT (.01 MIN) 36 KNR XWAIT (.01 MIN)
* 5 WLK IVTT TOT(.01 MIN) 19 PNR IVTT TOT(.01 MIN) 37 KNR IVTT TOT(.01 MIN)
* 6 WLK IVTT CR (.01 MIN) 20 PNR IVTT CR (.01 MIN) 38 KNR IVTT CR (.01 MIN)
* 7 WLK IVTT XB (.01 MIN) 21 PNR IVTT XB (.01 MIN) 39 KNR IVTT XB (.01 MIN)
* 8 WLK IVTT MR (.01 MIN) 22 PNR IVTT MR (.01 MIN) 40 KNR IVTT MR (.01 MIN)
* 9 WLK IVTT NM (.01 MIN) 23 PNR IVTT NM (.01 MIN) 41 KNR IVTT NM (.01 MIN)
* 10 WLK IVTT NM2(.01 MIN) 24 PNR IVTT NM2(.01 MIN) 42 KNR IVTT NM2(.01 MIN)
* 11 WLK IVTT LB (.01 MIN) 25 PNR IVTT LB (.01 MIN) 43 KNR IVTT LB (.01 MIN)
* 12 WLK #XFERS (NUMBER ) 26 PNR #XFERS (NUMBER ) 44 KNR #XFERS (NUMBER )
* 13 WLK COST (07CENTS) 27 PNR COST (07CENTS) 45 KNR COST (07CENTS)
* 14 WLK XPEN (.01 MIN) 28 PNR XPEN (.01 MIN) 46 KNR XPEN (.01 MIN)
*
* 29 PNR ACC TIME(.01 MIN) 47 KNR ACC TIME(.01 MIN)
* 30 PNR ACC DIST(.01 MIL) 48 KNR ACC DIST(.01 MIL)
*
* 31 PNR ACC COST(07CENTS)
* 32 PNR STA TERM(.01 MIN)
*
* FILE 8=ZDATA
* 1 HBW PARK COST (2007 CENTS)
* 2 HBS PARK COST (2007 CENTS)
* 3 HBO PARK COST (2007 CENTS)
* 4 NHB PARK COST (2007 CENTS)
* 5 TERMINAL TIME (HOME BASED) (MINUTES)
* 6 TERMINAL TIME (NON HOME BASED) (MINUTES)
* 7 ARC VIEW SHORT WALK PERCENT TO METRO
* 8 ARC VIEW LONG WALK PERCENT TO METRO
* 9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
* 10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT
* 11 ARC VIEW SHORT WALK PERCENT TO ALL OP TRANSIT
* 12 ARC VIEW LONG WALK PERCENT TO ALL OP TRANSIT
* 13 AREA TYPE
* 1=DC CORE
* 2=VA CORE
* 3=DC URBAN
* 4=MD URBAN
* 5=VA URBAN

```

## Appendix D: AEMS Fortran Control Files

---

```
* 6=MD OTHER
* 7=VA OTHER

* PARAMETERS
*=====
* AUTO OPERATING COSTS IN CENTS/mile
COMPUTE AUOP >10
* AUTO OCCUPANCY FOR 3+ Reduced from 3.5 to 3.35 on 3/1/07 rm
COMPUTE OCC3 >3.35

* TERMINAL TIMES, USE i/j805 FOR HBW, HBS, AND HBO. USE i/j806 FOR NHB
* HBW/HBS/HBO
*COMPUTE TERI >i805
*COMPUTE TERJ >j805
* NHB
COMPUTE TERI >i806
COMPUTE TERJ >j806

* PARK COSTS, USE i/j801 802 803 804 FOR HBW, HBS, HBO, NHB RESPECTIVELY
* HBW
*COMPUTE PRKC >j801/2.
* HBS
* COMPUTE PRKC >j802/2.
* HBO
* COMPUTE PRKC >j803/2.
* NHB
COMPUTE PRKC >j804

* Percent of productions in long-walk area that are assumed to walk = 25% (i.e., 75% drive)
COMPUTE PCLM >0.25
COMPUTE PCLT >0.25
* PERCENT WALKS-METRO RAIL ONLY
COMPUTE PCMI >(i807+PCLM*(i808-i807))/100.
COMPUTE PCMJ >(j807+PCLM*(j808-j807))/100.
* PERCENT WALKS-PEAK
*COMPUTE PCTI >(i809+PCLT*(i810-i809))/100.
*COMPUTE PCTJ >(j809+PCLT*(j810-j809))/100.
* PERCENT WALKS-OFFPEAK
COMPUTE PCTI >(i811+PCLT*(i812-i811))/100.
COMPUTE PCTJ >(j811+PCLT*(j812-j811))/100.
COMPUTE PCMI >MAX(PCMI,0)
COMPUTE PCMI >MIN(PCMI,1)
COMPUTE PCMJ >MAX(PCMJ,0)
COMPUTE PCMJ >MIN(PCMJ,1)
COMPUTE PCTI >MAX(PCTI,PCMI)
COMPUTE PCTI >MIN(PCTI,1)
COMPUTE PCTJ >MAX(PCTJ,PCMJ)
COMPUTE PCTJ >MIN(PCTJ,1)
*
* DO TRIP SUBDIVISIONS
*
* HOME BASED ALTERNATIVES
*COMPUTE TRP1 >m101
*COMPUTE TRP2 >m102
*COMPUTE TRP3 >m103
*COMPUTE TRP4 >m104
* NON-HOME BASED
COMPUTE TRP1 >0.25*m101
COMPUTE TRP2 >0.25*m101
COMPUTE TRP3 >0.25*m101
COMPUTE TRP4 >0.25*m101
*
* BE SURE TO UPDATE THE IVTT COEFFICIENT IN FTA SECTION FOR EACH PURPOSE
*
*=====
*INITIALIZING ALL VARIABLES WITHIN IF STATEMENTS TO ZERO
COMPUTE DAIV >0
COMPUTE DACS >0
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE DATE >0
COMPUTE S2IV >0
COMPUTE S2CS >0
COMPUTE S2TE >0
COMPUTE S3IV >0
COMPUTE S3CS >0
COMPUTE S3TE >0
COMPUTE WKIV >0
COMPUTE WKOV >0
COMPUTE WKXF >0
COMPUTE WKCS >0
COMPUTE WKXP >0
COMPUTE WBIV >0
COMPUTE WBOV >0
COMPUTE WBXF >0
COMPUTE WBSC >0
COMPUTE WBXP >0
COMPUTE WTIV >0
COMPUTE WTOV >0
COMPUTE WTXF >0
COMPUTE WTCS >0
COMPUTE WTXP >0
COMPUTE WMIV >0
COMPUTE WMOV >0
COMPUTE WMXF >0
COMPUTE WMCS >0
COMPUTE WMXP >0
COMPUTE PCIV >0
COMPUTE PCAA >0
COMPUTE PCOV >0
COMPUTE PCXF >0
COMPUTE PCCS >0
COMPUTE PCXP >0
COMPUTE PBIV >0
COMPUTE PBAA >0
COMPUTE PBOV >0
COMPUTE PBXF >0
COMPUTE PBCS >0
COMPUTE PBXP >0
COMPUTE PTIV >0
COMPUTE PTAA >0
COMPUTE PTOV >0
COMPUTE PTXF >0
COMPUTE PTCS >0
COMPUTE PTXP >0
COMPUTE PMIV >0
COMPUTE PMAA >0
COMPUTE PMOV >0
COMPUTE PMXF >0
COMPUTE PMCS >0
COMPUTE PMXP >0
COMPUTE KCIV >0
COMPUTE KCAA >0
COMPUTE KCOV >0
COMPUTE KCXF >0
COMPUTE KCCS >0
COMPUTE KCXP >0
COMPUTE KBIV >0
COMPUTE KBAA >0
COMPUTE KBOV >0
COMPUTE KBXF >0
COMPUTE KBCS >0
COMPUTE KBXP >0
COMPUTE KTIV >0
COMPUTE KTAA >0
COMPUTE KTOV >0
COMPUTE KTXF >0
COMPUTE KTCS >0
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE KTXP >0
COMPUTE KMIV >0
COMPUTE KMAA >0
COMPUTE KMOV >0
COMPUTE KMYF >0
COMPUTE KMCS >0
COMPUTE KMXF >0

COMPUTE WCWK >0
COMPUTE WBWK >0
COMPUTE WTWK >0
COMPUTE WMWK >0
COMPUTE PCWK >0
COMPUTE KCWK >0
COMPUTE PBWK >0
COMPUTE KBWK >0
COMPUTE PTWK >0
COMPUTE KTWK >0
COMPUTE PMWK >0
COMPUTE KMWK >0

* SKIM VALUES, Divide distances by 10 to convert tenths of miles to whole miles
* DRIVE ALONE
COMPUTE >IF(m201>0)
COMPUTE DAIV >m201
COMPUTE DACS >m202/10*AUOP+m203+PRKC
COMPUTE DATE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 2
COMPUTE >IF(m204>0)
COMPUTE S2IV >m204
COMPUTE S2CS >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 3
COMPUTE >IF(m207>0)
COMPUTE S3IV >m207
COMPUTE S3CS >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE >TERI+TERJ
COMPUTE >ENDIF

* Assign Intrazonal trips to Autos (mj11/04/05)
COMPUTE >IF(P()==Q())
COMPUTE DAIV >1
COMPUTE DACS >m202/10*AUOP+m203+PRKC
COMPUTE DATE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 2
COMPUTE >IF(P()==Q())
COMPUTE S2IV >1
COMPUTE S2CS >(m205/10*AUOP+m206+PRKC)/2.0
COMPUTE S2TE >TERI+TERJ
COMPUTE >ENDIF

* SHARED RIDE 3
COMPUTE >IF(P()==Q())
COMPUTE S3IV >1
COMPUTE S3CS >(m208/10*AUOP+m209+PRKC)/OCC3
COMPUTE S3TE >TERI+TERJ
COMPUTE >ENDIF

*End of Intrazonal trips

* WALK COMMUTER RAIL
COMPUTE >IF(m305>0)
```

## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE WCIV      >m305/100.
COMPUTE WCOV      >(m303+m304)/100.
COMPUTE WCXF      >m312
COMPUTE WCCS      >m313
COMPUTE WCXP      >m314/100.
COMPUTE WCWK      >(m301+m302)/100.
COMPUTE           >ENDIF

* WALK BUS
COMPUTE           >IF (m405>0)
COMPUTE WBIV      >m405/100.
COMPUTE WBOV      >(m403+m404)/100.
COMPUTE WBXF      >m412
COMPUTE WBCS      >m413
COMPUTE WBXP      >m414/100.
COMPUTE WBWK      >(m401+m402)/100.
COMPUTE           >ENDIF

* WALK BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m605>0)
COMPUTE WTIV      >m605/100.
COMPUTE WTOV      >(m603+m604)/100.
COMPUTE WTXF      >m612
COMPUTE WTCS      >m613
COMPUTE WTXP      >m614/100.
COMPUTE WTWK      >(m601+m602)/100.
COMPUTE           >ENDIF

* WALK METRORAIL
COMPUTE           >IF (m505>0)
COMPUTE WMIV      >m505/100.
COMPUTE WMOV      >(m503+m504)/100.
COMPUTE WMXF      >m512
COMPUTE WMCS      >m513
COMPUTE WMXP      >m514/100.
COMPUTE WMWK      >(m501+m502)/100.
COMPUTE           >ENDIF

* PNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE PCIV      >m319/100.
COMPUTE PCAA      >m329/100.
COMPUTE PCOV      >(m317+m318+m332)/100.
COMPUTE PCXF      >m326
COMPUTE PCCS      >m327+m331+m330/100*AUOP
COMPUTE PCXP      >m328/100.
COMPUTE PCWK      >(m315+m316)/100.
COMPUTE           >ENDIF

* PNR BUS
COMPUTE           >IF (m419>0)
COMPUTE PBIV      >m419/100.
COMPUTE PBAA      >m429/100.
COMPUTE PBOV      >(m417+m418+m432)/100.
COMPUTE PBXF      >m426
COMPUTE PBCS      >m427+m431+m430/100*AUOP
COMPUTE PBXP      >m428/100.
COMPUTE PBWK      >(m415+m416)/100.
COMPUTE           >ENDIF

* PNR BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m619>0)
COMPUTE PTIV      >m619/100.
COMPUTE PTAA      >m629/100.
COMPUTE PTOV      >(m617+m618+m632)/100.
COMPUTE PTXF      >m626
COMPUTE PTCS      >m627+m631+m630/100*AUOP
```



## Appendix D: AEMS Fortran Control Files

---

```
COMPUTE PTXP      >m628/100.
COMPUTE PTWK      >(m615+m616)/100.
COMPUTE           >ENDIF
```

```
* PNR METRORAIL
COMPUTE           >IF (m519>0)
COMPUTE PMIV      >m519/100.
COMPUTE PMAA      >m529/100.
COMPUTE PMOV      >(m517+m518+m532)/100.
COMPUTE PMXF      >m526
COMPUTE PMCS      >m527+m531+m530/100*AUOP
COMPUTE PMXP      >m528/100.
COMPUTE PMWK      >(m515+m516)/100.
COMPUTE           >ENDIF
```

```
* KNR COMMUTER RAIL
COMPUTE           >IF (m319>0)
COMPUTE KCIV      >m319/100.
COMPUTE KCAA      >m329/100.
COMPUTE KCOV      >(m317+m318)/100.
COMPUTE KCXF      >m326
COMPUTE KCCS      >m327+m330/100*AUOP
COMPUTE KCXP      >m328/100.
COMPUTE KCWK      >(m315+m316)/100.
COMPUTE           >ENDIF
```

```
* KNR BUS
COMPUTE           >IF (m437>0)
COMPUTE KBIV      >m437/100.
COMPUTE KBAA      >m447/100.
COMPUTE KBOV      >(m435+m436)/100.
COMPUTE KBXF      >m444
COMPUTE KBCS      >m445+m448/100*AUOP
COMPUTE KBXP      >m446/100.
COMPUTE KBWK      >(m433+m434)/100.
COMPUTE           >ENDIF
```

```
* KNR BUS/METRORAIL (TRANSIT)
COMPUTE           >IF (m637>0)
COMPUTE KTIV      >m637/100.
COMPUTE KTAA      >m647/100.
COMPUTE KTOV      >(m635+m636)/100.
COMPUTE KTXF      >m644
COMPUTE KTCS      >m645+m648/100*AUOP
COMPUTE KTXP      >m646/100.
COMPUTE KTWK      >(m633+m634)/100.
COMPUTE           >ENDIF
```

```
* KNR METRORAIL
COMPUTE           >IF (m537>0)
COMPUTE KMIV      >m537/100.
COMPUTE KMAA      >m547/100.
COMPUTE KMOV      >(m535+m536)/100.
COMPUTE KMXF      >m544
COMPUTE KMCS      >m545+m548/100*AUOP
COMPUTE KMXP      >m546/100.
COMPUTE KMWK      >(m533+m534)/100.
COMPUTE           >ENDIF
```

\*CONSTANTS BY CHOICE FOR EACH PURPOSE

```
*CHOICE          1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
PURP01 1INC 1    1>
PURP02 1INC 2    1>
```

## Appendix D: AEMS Fortran Control Files

```
PURP03 LINC 3 1>
PURP04 LINC 4 1>
```

```
TRIPIN01 >TRP1
TRIPIN02 >TRP2
TRIPIN03 >TRP3
TRIPIN04 >TRP4
TRIPFACT01 >tfi1
TRIPFACT02 >tfi2
TRIPFACT03 >tfi3
TRIPFACT04 >tfi4
COMPUTE tfi1 >1.0
COMPUTE tfi2 >1.0
COMPUTE tfi3 >1.0
COMPUTE tfi4 >1.0
```

```
*
*OUTPUT MATRICES AND OUTPUT FACTORS BY CHOICE FOR EACH PURPOSE
*CHOICE 1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
TRIPOUT01 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT01 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT02 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT02 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT03 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT03 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
TRIPOUT04 1>m901 m902 m903 m904 m905 m906 m907 m908 m908 m909 m910 m911 m912 m913 m914
TRIPFACT04 1>1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00
```

```
**
**P AND A WALK PERCENTS BY CHOICE
*CHOICE 1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
WALK SEG CW 1 PCT 1>WSWM
WALK SEG CW 1 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 2 PCT 1>WSW1
WALK SEG CW 2 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 3 PCT 1>WSW2
WALK SEG CW 3 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODEL>Y Y Y Y Y Y Y Y Y Y Y Y Y Y
*SYNTAX OF COMMAND TO ADD A COMPONENT TO A SPECIFIC WALK SEGMENT IF DESIRED
*WALK SEG CW 1 COEF1> -0.04747 -0.04747 -0.04747 -0.04747 -0.04747 -0.04747
*WALK SEG CW 1 VAR 1> WTSS DTSS DISS WRSS DRSS DJSS
COMPUTE WSWM >PCMI*PCMJ
COMPUTE WSW1 >(PCTI-PCMI)*PCMJ
COMPUTE WSW2 >(PCTI-PCMI)*(PCTJ-PCMJ)
COMPUTE WSW3 >PCMI*(PCTJ-PCMJ)
COMPUTE WSM1 >(1-PCTI)*PCMJ
COMPUTE WSM2 >(1-PCTI)*(PCTJ-PCMJ)
COMPUTE WSNT >1-WSWM-WSW1-WSW2-WSW3-WSM1-WSM2
```

```
*NEST DEFINITIONS BY CHOICE
*CHOICE 1>DR ALONE SR2 SR3+ WK-CR WK-BUS WK-BU/MR WK-MR PNR-CR KNR-CR PNR-BUS KNR-BUS PNR-BU/MR KNR-BU/MR PNR-MR KNR-MR
NEST 1,1= 1>Y Y
NEST 1,2= 1> Y Y Y Y Y Y Y Y
NEST 2,1= 1> Y Y Y Y Y Y Y Y
NEST 2,2= 1> Y Y Y Y Y Y Y Y
NEST 2,3= 1> Y Y Y Y Y Y Y Y
NEST 3,1= 1> Y Y Y Y Y Y Y Y
NEST 3,2= 1> Y Y Y Y Y Y Y Y
NEST 3,3= 1> Y Y Y Y Y Y Y Y
NEST 3,4= 1> Y Y Y Y Y Y Y Y
NEST 4,1 1> Y Y Y Y Y Y Y Y
NEST 4,2 1> Y Y Y Y Y Y Y Y
```

## Appendix D: AEMS Fortran Control Files

```

NEST 4,3      1>
NEST 4,4      1>
NEST 5,1      1>
NEST 5,2      1>
NEST 5,3      1>
NEST 5,4      1>
NEST 6,1      1>Y
NEST 6,2      1>      Y      Y
NEST 7,1      1>      Y
NEST 7,2      1>      Y

```

```

IGRP DEFINITION >i813
JGRP DEFINITION >j813
* 1 DC CORE/URBAN-DC CORE
SEGMENT 1      > 1 1
SEGMENT 1      > 3 1
* 2 DC CORE/URBAN-VA CORE
SEGMENT 2      > 1 2
SEGMENT 2      > 3 2
* 3 DC CORE/URBAN-URBAN
SEGMENT 3      > 1 3
SEGMENT 3      > 3 3
SEGMENT 3      > 1 4
SEGMENT 3      > 3 4
SEGMENT 3      > 1 5
SEGMENT 3      > 3 5
* 4 DC CORE/URBAN-OTHER
SEGMENT 4      > 1 6
SEGMENT 4      > 3 6
SEGMENT 4      > 1 7
SEGMENT 4      > 3 7
* 5 MD URBAN-DC CORE
SEGMENT 5      > 4 1
* 6 MD URBAN-VA CORE
SEGMENT 6      > 4 2
* 7 MD URBAN-URBAN
SEGMENT 7      > 4 3
SEGMENT 7      > 4 4
SEGMENT 7      > 4 5
* 8 MD URBAN-OTHER
SEGMENT 8      > 4 6
SEGMENT 8      > 4 7
* 9 VA CORE/URBAN-DC CORE
SEGMENT 9      > 2 1
SEGMENT 9      > 5 1
*10 VA CORE/URBAN-VA CORE
SEGMENT 10     > 2 2
SEGMENT 10     > 5 2
*11 VA CORE/URBAN-URBAN
SEGMENT 11     > 2 3
SEGMENT 11     > 5 3
SEGMENT 11     > 2 4
SEGMENT 11     > 5 4
SEGMENT 11     > 2 5
SEGMENT 11     > 5 5
*12 VA CORE/URBAN-OTHER
SEGMENT 12     > 2 6
SEGMENT 12     > 5 6
SEGMENT 12     > 2 7
SEGMENT 12     > 5 7
*13 MD OTHER-DC CORE
SEGMENT 13     > 6 1
*14 MD OTHER-VA CORE
SEGMENT 14     > 6 2
*15 MD OTHER-URBAN
SEGMENT 15     > 6 3
SEGMENT 15     > 6 4
SEGMENT 15     > 6 5

```

## Appendix D: AEMS Fortran Control Files

```

*16 MD OTHER-OTHER
SEGMENT 16 > 6 6
SEGMENT 16 > 6 7
*17 VA OTHER-DC CORE
SEGMENT 17 > 7 1
*18 VA OTHER-VA CORE
SEGMENT 18 > 7 2
*19 VA OTHER-URBAN
SEGMENT 19 > 7 3
SEGMENT 19 > 7 4
SEGMENT 19 > 7 5
*20 VA OTHER-OTHER
SEGMENT 20 > 7 6
SEGMENT 20 > 7 7

* SEGMENT 1
NSTC 10 1GRND TOTAL>
NSTC 11 1AUTO > 0.5 0.00000
NSTC 12 1TRANSIT > 0.5 -1.70935
NSTC 20 1TOTAL TRN >
NSTC 21 1WALK ACC > 0.5 0.00000
NSTC 22 1PNR ACC > 0.5 -1.95080
NSTC 23 1KNR ACC > 0.5 -4.17573
NSTC 30 1WLK TRN
NSTC 31 1WLK CR > 1.0 0.07631
NSTC 32 1WLK BUS > 1.0 -1.00068
NSTC 33 1WLK BU/MR > 1.0 4.39187
NSTC 34 1WLK METRO > 1.0 0.00000
NSTC 40 1PNR TRN
NSTC 41 1PNR CR > 1.0 0.17657
NSTC 42 1PNR BUS > 1.0 -0.43576
NSTC 43 1PNR BU/MR > 1.0 1.97829
NSTC 44 1PNR METRO > 1.0 0.00000
NSTC 50 1KNR TRN
NSTC 51 1KNR CR > 1.0 1.50904
NSTC 52 1KNR BUS > 1.0 3.59658
NSTC 53 1KNR BU/MR > 1.0 10.56658
NSTC 54 1KNR METRO > 1.0 0.00000
NSTC 60 1AUTO
NSTC 61 1LOV > 1.0 0.00000
NSTC 62 1HOV > 0.5 -4.87523
NSTC 70 1HOV
NSTC 71 1HOV2 > 1.0 0.00000
NSTC 72 1HOV3+ > 1.0 -4.43140
* SEGMENT 2
NSTC 10 2GRND TOTAL>
NSTC 11 2AUTO > 0.5 0.00000
NSTC 12 2TRANSIT > 0.5 -0.37515
NSTC 20 2TOTAL TRN >
NSTC 21 2WALK ACC > 0.5 0.00000
NSTC 22 2PNR ACC > 0.5 -0.54989
NSTC 23 2KNR ACC > 0.5 -1.55599
NSTC 30 2WLK TRN
NSTC 31 2WLK CR > 1.0 -0.12074
NSTC 32 2WLK BUS > 1.0 -0.12074
NSTC 33 2WLK BU/MR > 1.0 8.43549
NSTC 34 2WLK METRO > 1.0 0.00000
NSTC 40 2PNR TRN
NSTC 41 2PNR CR > 1.0 -0.26654
NSTC 42 2PNR BUS > 1.0 -0.26654
NSTC 43 2PNR BU/MR > 1.0 -0.29728
NSTC 44 2PNR METRO > 1.0 0.00000
NSTC 50 2KNR TRN
NSTC 51 2KNR CR > 1.0 -0.05759
NSTC 52 2KNR BUS > 1.0 -0.05759
NSTC 53 2KNR BU/MR > 1.0 -0.05759
NSTC 54 2KNR METRO > 1.0 0.00000
NSTC 60 2AUTO

```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 61 2LOV > 1.0 0.00000
NSTC 62 2HOV > 0.5 -3.13133
NSTC 70 2HOV
NSTC 71 2HOV2 > 1.0 0.00000
NSTC 72 2HOV3+ > 1.0 -1.53219
* SEGMENT 3
NSTC 10 3GRND TOTAL>
NSTC 11 3AUTO > 0.5 0.00000
NSTC 12 3TRANSIT > 0.5 -0.26424
NSTC 20 3TOTAL TRN >
NSTC 21 3WALK ACC > 0.5 0.00000
NSTC 22 3PNR ACC > 0.5 -2.29300
NSTC 23 3KNR ACC > 0.5 -4.87604
NSTC 30 3WLK TRN
NSTC 31 3WLK CR > 1.0 -0.29105
NSTC 32 3WLK BUS > 1.0 -0.86362
NSTC 33 3WLK BU/MR > 1.0 1.81912
NSTC 34 3WLK METRO > 1.0 0.00000
NSTC 40 3PNR TRN
NSTC 41 3PNR CR > 1.0 -0.83550
NSTC 42 3PNR BUS > 1.0 -6.44560
NSTC 43 3PNR BU/MR > 1.0 -2.44024
NSTC 44 3PNR METRO > 1.0 0.00000
NSTC 50 3KNR TRN
NSTC 51 3KNR CR > 1.0 1.83538
NSTC 52 3KNR BUS > 1.0 4.41171
NSTC 53 3KNR BU/MR > 1.0 7.43249
NSTC 54 3KNR METRO > 1.0 0.00000
NSTC 60 3AUTO
NSTC 61 3LOV > 1.0 0.00000
NSTC 62 3HOV > 0.5 -4.06190
NSTC 70 3HOV
NSTC 71 3HOV2 > 1.0 0.00000
NSTC 72 3HOV3+ > 1.0 -1.32971
* SEGMENT 4
NSTC 10 4GRND TOTAL>
NSTC 11 4AUTO > 0.5 0.00000
NSTC 12 4TRANSIT > 0.5 3.05913
NSTC 20 4TOTAL TRN >
NSTC 21 4WALK ACC > 0.5 0.00000
NSTC 22 4PNR ACC > 0.5 -1.64253
NSTC 23 4KNR ACC > 0.5 -6.24953
NSTC 30 4WLK TRN
NSTC 31 4WLK CR > 1.0 -5.42670
NSTC 32 4WLK BUS > 1.0 -14.41525
NSTC 33 4WLK BU/MR > 1.0 -10.92970
NSTC 34 4WLK METRO > 1.0 0.00000
NSTC 40 4PNR TRN
NSTC 41 4PNR CR > 1.0 -2.39215
NSTC 42 4PNR BUS > 1.0 -7.28798
NSTC 43 4PNR BU/MR > 1.0 -8.58062
NSTC 44 4PNR METRO > 1.0 0.00000
NSTC 50 4KNR TRN
NSTC 51 4KNR CR > 1.0 -0.93498
NSTC 52 4KNR BUS > 1.0 -0.16700
NSTC 53 4KNR BU/MR > 1.0 -4.34372
NSTC 54 4KNR METRO > 1.0 0.00000
NSTC 60 4AUTO
NSTC 61 4LOV > 1.0 0.00000
NSTC 62 4HOV > 0.5 -4.01537
NSTC 70 4HOV
NSTC 71 4HOV2 > 1.0 0.00000
NSTC 72 4HOV3+ > 1.0 -4.77121
* SEGMENT 5
NSTC 10 5GRND TOTAL>
NSTC 11 5AUTO > 0.5 0.00000
NSTC 12 5TRANSIT > 0.5 -2.18216
NSTC 20 5TOTAL TRN >
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 21 5WALK ACC > 0.5 0.00000
NSTC 22 5PNR ACC > 0.5 -2.53696
NSTC 23 5KNR ACC > 0.5 -4.48187
NSTC 30 5WLK TRN
NSTC 31 5WLK CR > 1.0 0.64051
NSTC 32 5WLK BUS > 1.0 1.29512
NSTC 33 5WLK BU/MR > 1.0 4.41115
NSTC 34 5WLK METRO > 1.0 0.00000
NSTC 40 5PNR TRN
NSTC 41 5PNR CR > 1.0 0.44877
NSTC 42 5PNR BUS > 1.0 0.44877
NSTC 43 5PNR BU/MR > 1.0 8.03600
NSTC 44 5PNR METRO > 1.0 0.00000
NSTC 50 5KNR TRN
NSTC 51 5KNR CR > 1.0 -0.16331
NSTC 52 5KNR BUS > 1.0 -0.16331
NSTC 53 5KNR BU/MR > 1.0 -0.16331
NSTC 54 5KNR METRO > 1.0 0.00000
NSTC 60 5AUTO
NSTC 61 5LOV > 1.0 0.00000
NSTC 62 5HOV > 0.5 -7.76821
NSTC 70 5HOV
NSTC 71 5HOV2 > 1.0 0.00000
NSTC 72 5HOV3+ > 1.0 0.37501
* SEGMENT 6
NSTC 10 6GRND TOTAL>
NSTC 11 6AUTO > 0.5 0.00000
NSTC 12 6TRANSIT > 0.5 6.14489
NSTC 20 6TOTAL TRN >
NSTC 21 6WALK ACC > 0.5 0.00000
NSTC 22 6PNR ACC > 0.5 -6.67670
NSTC 23 6KNR ACC > 0.5 -6.67670
NSTC 30 6WLK TRN
NSTC 31 6WLK CR > 1.0 0.24521
NSTC 32 6WLK BUS > 1.0 0.24521
NSTC 33 6WLK BU/MR > 1.0 10.99828
NSTC 34 6WLK METRO > 1.0 0.00000
NSTC 40 6PNR TRN
NSTC 41 6PNR CR > 1.0 0.00001
NSTC 42 6PNR BUS > 1.0 0.00001
NSTC 43 6PNR BU/MR > 1.0 0.00001
NSTC 44 6PNR METRO > 1.0 0.00000
NSTC 50 6KNR TRN
NSTC 51 6KNR CR > 1.0 0.00001
NSTC 52 6KNR BUS > 1.0 0.00001
NSTC 53 6KNR BU/MR > 1.0 0.00001
NSTC 54 6KNR METRO > 1.0 0.00000
NSTC 60 6AUTO
NSTC 61 6LOV > 1.0 0.00000
NSTC 62 6HOV > 0.5 0.00001
NSTC 70 6HOV
NSTC 71 6HOV2 > 1.0 0.00000
NSTC 72 6HOV3+ > 1.0 0.00001
* SEGMENT 7
NSTC 10 7GRND TOTAL>
NSTC 11 7AUTO > 0.5 0.00000
NSTC 12 7TRANSIT > 0.5 -1.90319
NSTC 20 7TOTAL TRN >
NSTC 21 7WALK ACC > 0.5 0.00000
NSTC 22 7PNR ACC > 0.5 -4.93686
NSTC 23 7KNR ACC > 0.5 -7.04432
NSTC 30 7WLK TRN
NSTC 31 7WLK CR > 1.0 1.23529
NSTC 32 7WLK BUS > 1.0 1.42469
NSTC 33 7WLK BU/MR > 1.0 7.10337
NSTC 34 7WLK METRO > 1.0 0.00000
NSTC 40 7PNR TRN
NSTC 41 7PNR CR > 1.0 2.80853
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 42 7PNR BUS > 1.0 -0.84836
NSTC 43 7PNR BU/MR > 1.0 9.86164
NSTC 44 7PNR METRO > 1.0 0.00000
NSTC 50 7KNR TRN
NSTC 51 7KNR CR > 1.0 6.82967
NSTC 52 7KNR BUS > 1.0 11.15244
NSTC 53 7KNR BU/MR > 1.0 16.54663
NSTC 54 7KNR METRO > 1.0 0.00000
NSTC 60 7AUTO
NSTC 61 7LOV > 1.0 0.00000
NSTC 62 7HOV > 0.5 -4.51873
NSTC 70 7HOV
NSTC 71 7HOV2 > 1.0 0.00000
NSTC 72 7HOV3+ > 1.0 -1.92480
* SEGMENT 8
NSTC 10 8GRND TOTAL>
NSTC 11 8AUTO > 0.5 0.00000
NSTC 12 8TRANSIT > 0.5 -1.14784
NSTC 20 8TOTAL TRN >
NSTC 21 8WALK ACC > 0.5 0.00000
NSTC 22 8PNR ACC > 0.5 -2.92089
NSTC 23 8KNR ACC > 0.5 -3.45931
NSTC 30 8WLK TRN
NSTC 31 8WLK CR > 1.0 3.91013
NSTC 32 8WLK BUS > 1.0 5.35921
NSTC 33 8WLK BU/MR > 1.0 4.30759
NSTC 34 8WLK METRO > 1.0 0.00000
NSTC 40 8PNR TRN
NSTC 41 8PNR CR > 1.0 -2.16154
NSTC 42 8PNR BUS > 1.0 -2.16154
NSTC 43 8PNR BU/MR > 1.0 -2.16154
NSTC 44 8PNR METRO > 1.0 0.00000
NSTC 50 8KNR TRN
NSTC 51 8KNR CR > 1.0 -1.99874
NSTC 52 8KNR BUS > 1.0 -1.99874
NSTC 53 8KNR BU/MR > 1.0 -1.99874
NSTC 54 8KNR METRO > 1.0 0.00000
NSTC 60 8AUTO
NSTC 61 8LOV > 1.0 0.00000
NSTC 62 8HOV > 0.5 -3.76444
NSTC 70 8HOV
NSTC 71 8HOV2 > 1.0 0.00000
NSTC 72 8HOV3+ > 1.0 -1.84497
* SEGMENT 9
NSTC 10 9GRND TOTAL>
NSTC 11 9AUTO > 0.5 0.00000
NSTC 12 9TRANSIT > 0.5 -1.94524
NSTC 20 9TOTAL TRN >
NSTC 21 9WALK ACC > 0.5 0.00000
NSTC 22 9PNR ACC > 0.5 -1.63823
NSTC 23 9KNR ACC > 0.5 -3.73340
NSTC 30 9WLK TRN
NSTC 31 9WLK CR > 1.0 -0.57665
NSTC 32 9WLK BUS > 1.0 -0.57665
NSTC 33 9WLK BU/MR > 1.0 5.73521
NSTC 34 9WLK METRO > 1.0 0.00000
NSTC 40 9PNR TRN
NSTC 41 9PNR CR > 1.0 1.87278
NSTC 42 9PNR BUS > 1.0 1.87278
NSTC 43 9PNR BU/MR > 1.0 9.48920
NSTC 44 9PNR METRO > 1.0 0.00000
NSTC 50 9KNR TRN
NSTC 51 9KNR CR > 1.0 0.22032
NSTC 52 9KNR BUS > 1.0 0.22032
NSTC 53 9KNR BU/MR > 1.0 5.74717
NSTC 54 9KNR METRO > 1.0 0.00000
NSTC 60 9AUTO
NSTC 61 9LOV > 1.0 0.00000

```

## Appendix D: AEMS Fortran Control Files

---

```

NSTC 62 9HOV      >    0.5  -4.95306
NSTC 70 9HOV
NSTC 71 9HOV2    >    1.0   0.00000
NSTC 72 9HOV3+  >    1.0  -2.87423
* SEGMENT 10
NSTC 1010GRND TOTAL>
NSTC 1110AUTO    >    0.5   0.00000
NSTC 1210TRANSIT >    0.5  -0.98629
NSTC 2010TOTAL TRN >
NSTC 2110WALK ACC >    0.5   0.00000
NSTC 2210PNR ACC >    0.5  -1.92854
NSTC 2310KNR ACC >    0.5  -9.29652
NSTC 3010WLK TRN
NSTC 3110WLK CR  >    1.0  -0.21324
NSTC 3210WLK BUS >    1.0  -2.04747
NSTC 3310WLK BU/MR >    1.0   1.38113
NSTC 3410WLK METRO >    1.0   0.00000
NSTC 4010PNR TRN
NSTC 4110PNR CR  >    1.0  -1.90108
NSTC 4210PNR BUS >    1.0  -1.90108
NSTC 4310PNR BU/MR >    1.0   8.60131
NSTC 4410PNR METRO >    1.0   0.00000
NSTC 5010KNR TRN
NSTC 5110KNR CR  >    1.0   1.33407
NSTC 5210KNR BUS >    1.0  12.03628
NSTC 5310KNR BU/MR >    1.0   1.33407
NSTC 5410KNR METRO >    1.0   0.00000
NSTC 6010AUTO
NSTC 6110LOV    >    1.0   0.00000
NSTC 6210HOV    >    0.5  -3.80087
NSTC 7010HOV
NSTC 7110HOV2   >    1.0   0.00000
NSTC 7210HOV3+ >    1.0  -2.33367
* SEGMENT 11
NSTC 1011GRND TOTAL>
NSTC 1111AUTO    >    0.5   0.00000
NSTC 1211TRANSIT >    0.5  -1.70990
NSTC 2011TOTAL TRN >
NSTC 2111WALK ACC >    0.5   0.00000
NSTC 2211PNR ACC >    0.5  -3.88749
NSTC 2311KNR ACC >    0.5  -8.39229
NSTC 3011WLK TRN
NSTC 3111WLK CR  >    1.0   0.12014
NSTC 3211WLK BUS >    1.0  -0.86151
NSTC 3311WLK BU/MR >    1.0   4.22610
NSTC 3411WLK METRO >    1.0   0.00000
NSTC 4011PNR TRN
NSTC 4111PNR CR  >    1.0  -0.49626
NSTC 4211PNR BUS >    1.0  -0.49626
NSTC 4311PNR BU/MR >    1.0   9.05560
NSTC 4411PNR METRO >    1.0   0.00000
NSTC 5011KNR TRN
NSTC 5111KNR CR  >    1.0   0.99252
NSTC 5211KNR BUS >    1.0  10.47425
NSTC 5311KNR BU/MR >    1.0   0.99252
NSTC 5411KNR METRO >    1.0   0.00000
NSTC 6011AUTO
NSTC 6111LOV    >    1.0   0.00000
NSTC 6211HOV    >    0.5  -4.38214
NSTC 7011HOV
NSTC 7111HOV2   >    1.0   0.00000
NSTC 7211HOV3+ >    1.0  -3.73961
* SEGMENT 12
NSTC 1012GRND TOTAL>
NSTC 1112AUTO    >    0.5   0.00000
NSTC 1212TRANSIT >    0.5   0.67121
NSTC 2012TOTAL TRN >
NSTC 2112WALK ACC >    0.5   0.00000

```



## Appendix D: AEMS Fortran Control Files

---

```
NSTC 2212PNR ACC > 0.5 -1.14949
NSTC 2312KNR ACC > 0.5 -8.71034
NSTC 3012WLK TRN
NSTC 3112WLK CR > 1.0 -5.96552
NSTC 3212WLK BUS > 1.0 -10.73534
NSTC 3312WLK BU/MR > 1.0 -7.73078
NSTC 3412WLK METRO > 1.0 0.00000
NSTC 4012PNR TRN
NSTC 4112PNR CR > 1.0 -3.28325
NSTC 4212PNR BUS > 1.0 -7.12253
NSTC 4312PNR BU/MR > 1.0 -5.60399
NSTC 4412PNR METRO > 1.0 0.00000
NSTC 5012KNR TRN
NSTC 5112KNR CR > 1.0 4.58043
NSTC 5212KNR BUS > 1.0 9.58185
NSTC 5312KNR BU/MR > 1.0 2.96969
NSTC 5412KNR METRO > 1.0 0.00000
NSTC 6012AUTO
NSTC 6112LOV > 1.0 0.00000
NSTC 6212HOV > 0.5 -4.25589
NSTC 7012HOV
NSTC 7112HOV2 > 1.0 0.00000
NSTC 7212HOV3+ > 1.0 -3.72821
* SEGMENT 13
NSTC 1013GRND TOTAL>
NSTC 1113AUTO > 0.5 0.00000
NSTC 1213TRANSIT > 0.5 -1.77008
NSTC 2013TOTAL TRN >
NSTC 2113WALK ACC > 0.5 0.00000
NSTC 2213PNR ACC > 0.5 -2.94858
NSTC 2313KNR ACC > 0.5 -4.56381
NSTC 3013WLK TRN
NSTC 3113WLK CR > 1.0 0.09843
NSTC 3213WLK BUS > 1.0 -0.98380
NSTC 3313WLK BU/MR > 1.0 1.27962
NSTC 3413WLK METRO > 1.0 0.00000
NSTC 4013PNR TRN
NSTC 4113PNR CR > 1.0 1.03617
NSTC 4213PNR BUS > 1.0 1.03617
NSTC 4313PNR BU/MR > 1.0 13.98204
NSTC 4413PNR METRO > 1.0 0.00000
NSTC 5013KNR TRN
NSTC 5113KNR CR > 1.0 0.53175
NSTC 5213KNR BUS > 1.0 4.75813
NSTC 5313KNR BU/MR > 1.0 7.72848
NSTC 5413KNR METRO > 1.0 0.00000
NSTC 6013AUTO
NSTC 6113LOV > 1.0 0.00000
NSTC 6213HOV > 0.5 -7.20588
NSTC 7013HOV
NSTC 7113HOV2 > 1.0 0.00000
NSTC 7213HOV3+ > 1.0 -5.89808
* SEGMENT 14
NSTC 1014GRND TOTAL>
NSTC 1114AUTO > 0.5 0.00000
NSTC 1214TRANSIT > 0.5 -1.52373
NSTC 2014TOTAL TRN >
NSTC 2114WALK ACC > 0.5 0.00000
NSTC 2214PNR ACC > 0.5 -1.21762
NSTC 2314KNR ACC > 0.5 -2.64082
NSTC 3014WLK TRN
NSTC 3114WLK CR > 1.0 2.29298
NSTC 3214WLK BUS > 1.0 2.29298
NSTC 3314WLK BU/MR > 1.0 5.71890
NSTC 3414WLK METRO > 1.0 0.00000
NSTC 4014PNR TRN
NSTC 4114PNR CR > 1.0 -0.11400
NSTC 4214PNR BUS > 1.0 -0.11400
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 4314PNR BU/MR > 1.0 -0.11400
NSTC 4414PNR METRO > 1.0 0.00000
NSTC 5014KNR TRN
NSTC 5114KNR CR > 1.0 0.95707
NSTC 5214KNR BUS > 1.0 0.95707
NSTC 5314KNR BU/MR > 1.0 8.68830
NSTC 5414KNR METRO > 1.0 0.00000
NSTC 6014AUTO
NSTC 6114LOV > 1.0 0.00000
NSTC 6214HOV > 0.5 -4.42074
NSTC 7014HOV
NSTC 7114HOV2 > 1.0 0.00000
NSTC 7214HOV3+ > 1.0 -8.01996
* SEGMENT 15
NSTC 1015GRND TOTAL>
NSTC 1115AUTO > 0.5 0.00000
NSTC 1215TRANSIT > 0.5 -2.28383
NSTC 2015TOTAL TRN >
NSTC 2115WALK ACC > 0.5 0.00000
NSTC 2215PNR ACC > 0.5 -2.14848
NSTC 2315KNR ACC > 0.5 -3.45211
NSTC 3015WLK TRN
NSTC 3115WLK CR > 1.0 3.28167
NSTC 3215WLK BUS > 1.0 4.17529
NSTC 3315WLK BU/MR > 1.0 5.66237
NSTC 3415WLK METRO > 1.0 0.00000
NSTC 4015PNR TRN
NSTC 4115PNR CR > 1.0 1.45732
NSTC 4215PNR BUS > 1.0 2.32449
NSTC 4315PNR BU/MR > 1.0 11.18615
NSTC 4415PNR METRO > 1.0 0.00000
NSTC 5015KNR TRN
NSTC 5115KNR CR > 1.0 3.71563
NSTC 5215KNR BUS > 1.0 7.88022
NSTC 5315KNR BU/MR > 1.0 9.58647
NSTC 5415KNR METRO > 1.0 0.00000
NSTC 6015AUTO
NSTC 6115LOV > 1.0 0.00000
NSTC 6215HOV > 0.5 -5.43607
NSTC 7015HOV
NSTC 7115HOV2 > 1.0 0.00000
NSTC 7215HOV3+ > 1.0 -5.07390
* SEGMENT 16
NSTC 1016GRND TOTAL>
NSTC 1116AUTO > 0.5 0.00000
NSTC 1216TRANSIT > 0.5 -1.60322
NSTC 2016TOTAL TRN >
NSTC 2116WALK ACC > 0.5 0.00000
NSTC 2216PNR ACC > 0.5 -4.81326
NSTC 2316KNR ACC > 0.5 -6.19274
NSTC 3016WLK TRN
NSTC 3116WLK CR > 1.0 5.17100
NSTC 3216WLK BUS > 1.0 6.04527
NSTC 3316WLK BU/MR > 1.0 2.97596
NSTC 3416WLK METRO > 1.0 0.00000
NSTC 4016PNR TRN
NSTC 4116PNR CR > 1.0 -0.58069
NSTC 4216PNR BUS > 1.0 -1.47766
NSTC 4316PNR BU/MR > 1.0 0.48045
NSTC 4416PNR METRO > 1.0 0.00000
NSTC 5016KNR TRN
NSTC 5116KNR CR > 1.0 9.98358
NSTC 5216KNR BUS > 1.0 13.44178
NSTC 5316KNR BU/MR > 1.0 11.70992
NSTC 5416KNR METRO > 1.0 0.00000
NSTC 6016AUTO
NSTC 6116LOV > 1.0 0.00000
NSTC 6216HOV > 0.5 -2.78160
```

## Appendix D: AEMS Fortran Control Files

---

```
NSTC 7016HOV
NSTC 7116HOV2 > 1.0 0.00000
NSTC 7216HOV3+ > 1.0 -2.10252
* SEGMENT 17
NSTC 1017GRND TOTAL>
NSTC 1117AUTO > 0.5 0.00000
NSTC 1217TRANSIT > 0.5 -0.41434
NSTC 2017TOTAL TRN >
NSTC 2117WALK ACC > 0.5 0.00000
NSTC 2217PNR ACC > 0.5 -4.33384
NSTC 2317KNR ACC > 0.5 -5.62499
NSTC 3017WLK TRN
NSTC 3117WLK CR > 1.0 -3.33849
NSTC 3217WLK BUS > 1.0 -7.63137
NSTC 3317WLK BU/MR > 1.0 -4.26420
NSTC 3417WLK METRO > 1.0 0.00000
NSTC 4017PNR TRN
NSTC 4117PNR CR > 1.0 0.56158
NSTC 4217PNR BUS > 1.0 -4.97916
NSTC 4317PNR BU/MR > 1.0 6.91078
NSTC 4417PNR METRO > 1.0 0.00000
NSTC 5017KNR TRN
NSTC 5117KNR CR > 1.0 -1.41271
NSTC 5217KNR BUS > 1.0 -1.41271
NSTC 5317KNR BU/MR > 1.0 3.35818
NSTC 5417KNR METRO > 1.0 0.00000
NSTC 6017AUTO
NSTC 6117LOV > 1.0 0.00000
NSTC 6217HOV > 0.5 -7.97328
NSTC 7017HOV
NSTC 7117HOV2 > 1.0 0.00000
NSTC 7217HOV3+ > 1.0 -5.53235
* SEGMENT 18
NSTC 1018GRND TOTAL>
NSTC 1118AUTO > 0.5 0.00000
NSTC 1218TRANSIT > 0.5 -2.96534
NSTC 2018TOTAL TRN >
NSTC 2118WALK ACC > 0.5 0.00000
NSTC 2218PNR ACC > 0.5 -1.62945
NSTC 2318KNR ACC > 0.5 -2.90087
NSTC 3018WLK TRN
NSTC 3118WLK CR > 1.0 -1.59937
NSTC 3218WLK BUS > 1.0 -3.84058
NSTC 3318WLK BU/MR > 1.0 -0.85520
NSTC 3418WLK METRO > 1.0 0.00000
NSTC 4018PNR TRN
NSTC 4118PNR CR > 1.0 -0.05531
NSTC 4218PNR BUS > 1.0 -6.52413
NSTC 4318PNR BU/MR > 1.0 8.18028
NSTC 4418PNR METRO > 1.0 0.00000
NSTC 5018KNR TRN
NSTC 5118KNR CR > 1.0 -2.29458
NSTC 5218KNR BUS > 1.0 -2.29458
NSTC 5318KNR BU/MR > 1.0 6.44597
NSTC 5418KNR METRO > 1.0 0.00000
NSTC 6018AUTO
NSTC 6118LOV > 1.0 0.00000
NSTC 6218HOV > 0.5 -7.80731
NSTC 7018HOV
NSTC 7118HOV2 > 1.0 0.00000
NSTC 7218HOV3+ > 1.0 -8.75547
* SEGMENT 19
NSTC 1019GRND TOTAL>
NSTC 1119AUTO > 0.5 0.00000
NSTC 1219TRANSIT > 0.5 -0.17488
NSTC 2019TOTAL TRN >
NSTC 2119WALK ACC > 0.5 0.00000
NSTC 2219PNR ACC > 0.5 -7.33506
```

## Appendix D: AEMS Fortran Control Files

```

NSTC 2319KNR ACC > 0.5 -8.07075
NSTC 3019WLK TRN
NSTC 3119WLK CR > 1.0 -6.09488
NSTC 3219WLK BUS > 1.0 -10.09793
NSTC 3319WLK BU/MR > 1.0 -5.63421
NSTC 3419WLK METRO > 1.0 0.00000
NSTC 4019PNR TRN
NSTC 4119PNR CR > 1.0 -1.72798
NSTC 4219PNR BUS > 1.0 -1.72798
NSTC 4319PNR BU/MR > 1.0 7.81577
NSTC 4419PNR METRO > 1.0 0.00000
NSTC 5019KNR TRN
NSTC 5119KNR CR > 1.0 -1.26088
NSTC 5219KNR BUS > 1.0 -1.26088
NSTC 5319KNR BU/MR > 1.0 5.15590
NSTC 5419KNR METRO > 1.0 0.00000
NSTC 6019AUTO
NSTC 6119LOV > 1.0 0.00000
NSTC 6219HOV > 0.5 -5.53103
NSTC 7019HOV
NSTC 7119HOV2 > 1.0 0.00000
NSTC 7219HOV3+ > 1.0 -5.64451
* SEGMENT 20
NSTC 1020GRND TOTAL>
NSTC 1120AUTO > 0.5 0.00000
NSTC 1220TRANSIT > 0.5 0.91611
NSTC 2020TOTAL TRN >
NSTC 2120WALK ACC > 0.5 0.00000
NSTC 2220PNR ACC > 0.5 -13.52438
NSTC 2320KNR ACC > 0.5 -13.09938
NSTC 3020WLK TRN
NSTC 3120WLK CR > 1.0 -8.51553
NSTC 3220WLK BUS > 1.0 -9.12224
NSTC 3320WLK BU/MR > 1.0 -4.57420
NSTC 3420WLK METRO > 1.0 0.00000
NSTC 4020PNR TRN
NSTC 4120PNR CR > 1.0 0.00001
NSTC 4220PNR BUS > 1.0 2.95019
NSTC 4320PNR BU/MR > 1.0 -0.86366
NSTC 4420PNR METRO > 1.0 0.00000
NSTC 5020KNR TRN
NSTC 5120KNR CR > 1.0 0.00001
NSTC 5220KNR BUS > 1.0 5.13893
NSTC 5320KNR BU/MR > 1.0 0.63254
NSTC 5420KNR METRO > 1.0 0.00000
NSTC 6020AUTO
NSTC 6120LOV > 1.0 0.00000
NSTC 6220HOV > 0.5 -2.72978
NSTC 7020HOV
NSTC 7120HOV2 > 1.0 0.00000
NSTC 7220HOV3+ > 1.0 -2.36884

*DOWNTOWN=8
*SELI > 8

*UNION STATION=64
*SELI > 64

* =122
*SELI > 122

*BETHESDA=345
*SELI > 345

*SILVER SPRING=362
*SELI > 362

*N.SILVER SPRING=464

```

## Appendix D: AEMS Fortran Control Files

---

*SELI	>	464
* =475		
*SELI	>	475
*SHADY GROVE RD=578		
*SELI	>	578
* =787		
*SELI	>	787
*ANDREWS AFB=829		
*SELI	>	829
*NEW CARROLTON=927		
*SELI	>	927
*BRISTOL=972		
*SELI	>	972
*FREDERICK=1043		
*SELI	>	1043
*JESSUP=1080		
*SELI	>	1080
*SCAGGSVILLE=1091		
*SELI	>	1091
*WALDORF=1216		
*SELI	>	1216
*PENTAGON=1231		
*SELI	>	1231
*ROSSLYN=1236		
*SELI	>	1236
*ALEXANDRIA=1337		
*SELI	>	1337
* =1455		
*SELI	>	1455
*SPRINGFIELD=1502		
*SELI	>	1502
* =1511		
*SELI	>	1511
*TYSONS CRNR=1537		
*SELI	>	1537
*FT BELVOIR=1554		
*SELI	>	1554
*VIENNA=1619		
*SELI	>	1619
*DULES AP=1698		
*SELI	>	1698
*RESTON=1716		
*SELI	>	1716
*LEESBURG=1842		
*SELI	>	1842

## Appendix D: AEMS Fortran Control Files

```

*BRUNSWICK=1863
*SELI      >      1863

*DALE CITY=1942
*SELI      >      1942

*MANASSAS=1967
*SELI      >      1967

*SPOTSYLVANIA=2110
*SELI      >      2110

* =2055
*SELI      >      2055

*SELJ      >        8
*SELJ      >       63
*SELJ      >       64
*SELJ      >       77
*SELJ      >      100
*SELJ      >      344
*SELJ      >      345
*SELJ      >      362
*SELJ      >     1231
*SELJ      >     1236
*SELJ      >     1265
*SELJ      >     1337
*SELJ      >     1537

*SELI      >     523
*SELJ      >        9

TRACE      >        0
* OUTPUT % >
*PROCSEL   >
PRINT MS   >NHW_NL_MC.PRN
INPUT PRINT FILE >NHW_NL_MC.PRN
INPUT GOALS >NHW_NL_MC.GOL
INFILE 1   >nhw_income.ptt
INFILE 2   >hwyOp.skm
INFILE 3   >TRNOP_CR.SKM
INFILE 4   >TRNOP_AB.SKM
INFILE 5   >TRNOP_MR.SKM
INFILE 6   >TRNOP_BM.SKM
ZINFILE 8  >ZONEV2.A2F
OUTFILE 9  >NHW_NL_MC.MTT

* FTA USER BENEFITS SPECIFICATIONS
*FTA RESULTS FILE >NHB_NL_MC.BEN
FTA TRANSIT COEFF >-0.02860
FTA AUTO COEFF    >-0.02860
FTA PURPOSE NAME  >NHB
FTA PERIOD NAME   >ALLDAY
FTA ALTER. NAME   >CALIB
*CHOICE           1>DR ALONE SR2      SR3+      WK-CR      WK-BUS      WK-BU/MR  WK-MR      PNR-CR      KNR-CR      PNR-BUS      KNR-BUS      PNR-BU/MR  KNR-BU/MR  PNR-MR      KNR-MR
FTA AUTO NEST     >          1          1
FTA MOTORIZED?    1>Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y          Y
FTA TRANSIT?      1>

```

## 9 station\_names.dbf

ID	STATION	ID	STATION	ID	STATION	ID	STATION	ID	STATION	ID	STATION
8001	Shady Grove	8051	Braddock Road	9015	Harpers Ferry	22539	Burtonsville Crossi	34684	Fairfax City	47030	La Plata PNR
8002	Rockville	8052	National Airport	9016	Duffields	22542	US 29 /ICC	34719	WorldGate	47053	Waldorf
8003	Twinbrook	8053	Crystal City	9017	Monocacy/I-270	22549	White Oak Park-n-Ri	34738	Poplar Tree	47062	Smallwood Drive
8004	White Flint	8054	Pentagon City	9018	Frederick	22551	Tech Road	34952	Roberts Parkway	47110	Mattawoman Beantown
8005	Grosvenor	8055	Pentagon	9020	Riverdale	22554	Fairland	34986	Springfield Methodi	49007	Dunkirk Park
8006	Medical Center	8056	Arlington Cemetery	9021	College Park	22562	MD 650/ICC	35132	Tyson/Westpark Trn	49015	Lord Calvert Md4/MD
8007	Bethesda	8057	Vienna	9023	Laurel	22564	Colesville	35248	VA 7900/Gambrill	49058	Sunderland
8008	Friendship Heights	8058	Dunn Loring	9024	Savage	22595	Wheaton Plaza	35270	U. Meth. Church/VA2	49080	Prince Frederick PN
8009	Tenleytown	8059	West Falls Church	9025	Jessup	22696	Damascus Park-n-Rid	35313	Backlick Rd. North	49510	Lexington Park PNR
8010	Van Ness-UDC	8060	East Falls Church	9026	Greenbelt	22722	I-270 / MD 117 PNR	35654	Herndon/Monroe St.	49548	Mechanicsville PNR
8011	Cleveland Park	8061	Ballston	9027	Dorsey	22728	I-270/MD 124 PNR/Q. Orchard	35657	Reston East/Wiehle	49549	MD 5/MD 235 PNR
8012	Woodley Park-Zoo	8062	Virginia Square	9028	Muirkirk	22742	Urbana PNR	38011	Cascades/Sterling	49556	Charlotte Hall
8013	Dupont Circle	8063	Clarendon	9029	Laurel Race Track	22832	Kingsview/Clopper R	38018	Sully Station	51096	Rt 630 / I-95
8014	Farragut North	8064	Court House	9030	New Carrollton	26050	PG Plaza	38078	Leesburg/Dulles Gwa	51111	Rt 17 / Falmouth
8015	Metro Center	8065	Rosslyn	9031	Seabrook	26130	Capital Plaza	38109	US 50 at Stone Ridg	51115	Rt 610 / Rt684
8016	Gallery Place	8066	Foggy Bottom-GWU	9032	Bowie State	26277	Laurel	38137	Leesburg (Catoctin	52003	Rt 208 Courthouse Ro
8017	Judiciary Square	8067	Farragut West	9033	Odenton	26312	Montpelier Rec Prk	38160	VA625/VA28	52008	Rt 3 Germanna Hwy
8018	Union Station	8068	McPherson Square	9034	BWI Airport	26323	Greenbelt Armory	38215	Purcellville		
8019	Rhode Island Ave	8069	Federal Triangle	9057	L'Enfant Plaza	26385	Bowie	38310	Dulles Town Center		
8020	Brookland-CUA	8070	Smithsonian	9058	Crystal City	26388	Belair Plaza	38312	VA Route 772/GWay		
8021	Fort Totten	8071	Federal Center SW	9060	Alexandria	26400	Eastover Shopping C	38313	Western Regional		
8022	Takoma	8072	Capitol South	9061	Backlick Road	26413	MD 197/US 50	40009	NVCC		
8023	Silver Spring	8073	Eastern Market	9062	Rolling Road	26419	US 301/MD 214	40014	Lindendale PNR		
8024	Forest Glen	8074	Potomac Avenue	9063	Burke Center	26499	I-95/I-495 PNR	40036	Hechinger		
8025	Wheaton	8075	Stadium Armory	9065	Manassas Park	26535	Penn Mar Shop Ctr	40043	Lake Ridge		
8026	Glenmont	8076	Minnesota Avenue	9066	Manassas City	26652	US 301/MD5	40044	Horner Road		
8027	Greenbelt	8077	Deanwood	9067	Broad Run/Airport	26672	MD 4/MD 202	40047	Potomac Mills Mall		
8028	College Park	8078	Cheverly	9077	Franconia-Spgfld	26766	Oxon Hill	40048	Dale City/Minnievil		
8029	PG Plaza	8079	Landover	9078	Lorton	26777	Padgettes Corner	40055	Hillendale		
8030	West Hyattsville	8080	New Carrollton	9079	Woodbridge	26792	ABC Drive-In	40120	I-95/VA 123		
8031	Georgia Ave	8081	Benning Road	9080	Rippon	26803	Fort Washington	40124	I-95 VA 234/US 1		
8032	Columbia Heights	8082	Capitol Heights	9081	Quantico	26831	Accokeek/Livingston Rd	40128	Dumfries Road/Moncl		
8033	U-Street-Cardozo	8083	Addison Road	9082	Brooke	26840	PG Equestr Ctr (U	40150	Haymarket		
8034	Shaw-Howard Univ	8085	Morgan Boulevard	9083	Leeland Road	27208	Clinton	40161	Gainesville		
8035	Mt Vernon Square	8086	Largo Town Center	9084	Fredericksburg	30072	Four Mile/VEPCO	40172	Gordon Blvd/Rte 123		
8036	Archives	8098	New York Ave NE.	20811	Carter Barron	34035	Shirley Plaza	40277	PRTC Transit Ctr.		
8037	L'Enfant Plaza	9001	Union Station	21130	Soldiers Home	34123	Backlick Rd. South	40418	VA 234 Bypass / I-6		
8038	Waterfront	9002	Silver Spring	21365	South Capitol Stree	34230	Springfield Plaza	42033	Francis Scott Key M		
8039	Navy Yard	9003	Kensington	22128	Germantown Transit	34254	S Run Dist Prk /P	42050	Urbana Road		
8040	Anacostia	9004	Garrett Park	22167	Forcey Memorial	34257	Rolling Valley	44030	Scaggsville West		
8041	Congress Heights	9005	Rockville	22298	Milestone Md118/Md3	34259	Sydenstricker / Hoo	44040	Ellicot City		
8042	Southern Avenue	9006	Washington Grove	22319	Montgomery Mall	34341	Centreville Sq./VA2	44058	MD108 (Clarksville)		
8043	Naylor Road	9007	Gaithersburg	22333	Mid-Pike Plz Korvet	34344	Centreville	44129	Snowden River Pkwy		
8044	Suitland	9008	Metropolitan Grove	22430	Lake Forest Mall	34524	McLean Bible Church	44132	Broken Land Pkwy		
8045	Branch Avenue	9009	Germantown	22442	Village Mall	34564	City of Reston	45549	South Laurel/MD197		
8046	Van Dorn Street	9010	Boyd's	22463	Comus Road	34618	Nottoway Park	45555	Lower Pindell Rd Md		
8047	Franconia-Springfie	9011	Barnesville	22493	Norbeck Road	34621	Reston Transit Ctr.	45558	Bristol		
8048	Huntington	9012	Dickerson	22495	Olney TrnCtr	34644	Reston South	45569	Navy Station		
8049	Eisenhower Avenue	9013	Point of Rocks	22507	Briggs Chaney	34654	Fair Oaks Mall	45585	Harry Truman		
8050	King Street	9014	Brunswick	22526	Greencastle Rd.	34680	Stringfellow Rd. &	47021	Smallwood		

