National Capital Region Transportation Planning Board
Metropolitan Washington Council of Governments

User's Guide for the TPB Travel Forecasting Model, Version 2.3.36, on the 3,722-Zone Area System

Draft Report

November 18, 2011

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.

<table>
<tr><td rowspan="4">Title
User's Guide for the TPB Travel Forecasting Model, Version 2.3.36, on the 3,722-Zone Area System</td><td>Date</td><td>November 18, 2011</td></tr>
<tr><td>Number of pages</td><td></td></tr>
<tr><td>Publication no.</td><td></td></tr>
<tr><td>Availability</td><td>See below</td></tr>
<tr><td colspan="3">Agency
National Capital Region Transportation Planning Board (TPB).
Transportation planning at the regional level in the Washington area is coordinated by the National Capital Region Transportation Planning Board (TPB), the federally designated Metropolitan Planning Organization (MPO) for the region. The TPB is staffed by the Department of Transportation Planning (DTP) at the Metropolitan Washington Council of Governments (COG). COG is an independent, nonprofit association comprised of elected officials from 21 local governments, members of the Maryland and Virginia state legislatures, and members of the U.S. Congress.</td></tr>
<tr><td>Credits</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>Program Administration:</td><td colspan="2">Ronald Milone, Travel Forecasting Program Director</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td>Authors:</td><td colspan="2">Hamid Humeida, Maria Martchouk, Ronald Milone, Mark Moran, Meseret Seifu</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td colspan="3">Abstract: This report describes the application of a regional travel demand forecasting process, known as the TPB Version 2.3 Travel Model, which has been constructed for use in the Washington, D.C. region. This report describes build 36 of the Version 2.3 Travel Model, also known as Version 2.3.36. The Version 2.3 Travel Model is distinguished from prior TPB travel models in that it has been developed over a new 3,722 transportation analysis zone (TAZ) system, and it has been calibrated and validated with several sources of recently collected travel data, including the COG/TPB 2007/08 Household Travel Survey and several on-board transit surveys. The TPB Travel Forecasting Subcommittee provided oversight for the Version 2.3 Model development effort.</td></tr>
<tr><td colspan="3">Copies of this report can be found on the MWCOG Website:
www.mwcog.org

Metropolitan Washington Council of Governments
777 N. Capitol Street, N.E., Suite 300
Washington, D.C. 20002-4239
Tel. (202) 962-3200</td></tr>
</table>

Table of Contents

List of Figures

List of Tables

List of Equations

Appendices

A. Flowcharts
B. Batch files
C. Cube Voyager Scripts
D. AEMS Fortran Control Files

Chapter 1 Introduction

This report documents the application of the National Capital Region Transportation Planning Board (TPB) Version 2.3 Travel Model. This report is to be used in conjunction with the TPB Version 2.3 Travel Model calibration report, which is referenced at various points in this report. The Version 2.3 Travel Model became the adopted regional travel model for the for the Washington, D.C. area on November 16, 2011, when the TPB approved the Air Quality Conformity Determination of the 2011 Financially Constrained Long-Range Transportation Plan (CLRP), since that analysis was done using the new travel model.

1.1 Background

The TPB's previously adopted travel model was the Version 2.2 Travel Model, which was released on March 1, 2008.[1] The Version 2.2 travel model was developed on the 2,191-TAZ area system and most of its component models were estimated/calibrated with data from the COG/TPB 1994 Household Travel Survey. At the time when the Version 2.2 Travel Model was released, a parallel effort was also underway to combine a nested-logit mode choice (NL MC) model and revised truck models into the Version 2.2 framework. This development effort proved to be viable and resulted in a release of what was then called the "draft Version 2.3 travel model" in June of 2008.[2] The draft Version 2.3 model, like Version 2.2, was developed on the 2,191-TAZ area system.

The draft Version 2.3 model was not brought into production given that two related events were in motion during 2008. First, a new round of travel data collection was underway, including a major regional household travel survey (2007/2008 HTS) and both a bus on-board survey and a Metrorail passenger survey. Second, a new TAZ system was in development. The new zone system was envisioned to be developed over the same geographic area as the 2,191-TAZ system, but with smaller average zone sizes. TPB staff ultimately decided that the Version 2.3 Travel Model should not become the approved regional travel model until it incorporated the new zone system and the new data from the 2007/2008 Household Travel Survey (HTS) and the on-board transit surveys.

The last two to three years have been spent compiling and cleaning new survey data, preparing calibration files based on the new 3,722 TAZ system, and estimating/calibrating the models that make up the regional travel model. This report documents the TPB Version 2.3 Travel Model on the 3,722-TAZ area system that has been calibrated with the 2007/2008 HTS and other recent transit on-board surveys. A map of the modeled area can be found in the calibration report. Each minor change that occurs to the model, e.g., a change in a script or a batch file, is known as a new "build" of the travel model. This documentation is about build 36 of the Version 2.3 Travel Model, also known as Version 2.3.36. Note

---

[1] Ronald Milone et al., *TPB Travel Forecasting Model, Version 2.2: Specification, Validation, and User's Guide* (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, March 1, 2008).
[2] Ronald Milone et al., *TPB Travel Forecasting Model, Version 2.3: Specification, Validation, and User's Guide*, Draft report (Washington, D.C.: Metropolitan Washington Council of Governments, National Capital Region Transportation Planning Board, June 30, 2008).

that the most recent Air Quality Conformity (AQC) Determination done by TPB was done with Version 2.3.33. Below is a list of the changes that occurred between the two builds:

- Ver. 2.3.34: Updated six copy commands in the PP_Highway_Skims.bat batch file. Before this update, some versions of Microsoft Windows would return the following message: "The file cannot be copied onto itself".
- Ver. 2.3.35: Modified Highway_assignment.s to include market-specific volumes in speed feedback iteration #4.
- Ver. 2.3.36: Updated Highway_assignment.s so that it outputs separate text files with relative gap statistics. Removed the "40" from the name of the highway skims replacement batch file (e.g., HSR40_Highway_Skims.bat => HSR_Highway_Skims.bat), since this procedure is not specific to the year 2040 – it can be applied to other future years such as 2016, 2020, and 2030, all of which were modeled years in this year's AQC determination.

Note that the differences between Version 2.3.33 and Version 2.3.36 can be viewed as "cosmetic," since they do not affect the estimates of VMT or transit person trips coming out of the model.

1.2 Overview of the model

The major steps in the Version 2.3 Travel Model, including major inputs and outputs, can be found in Figure 1.

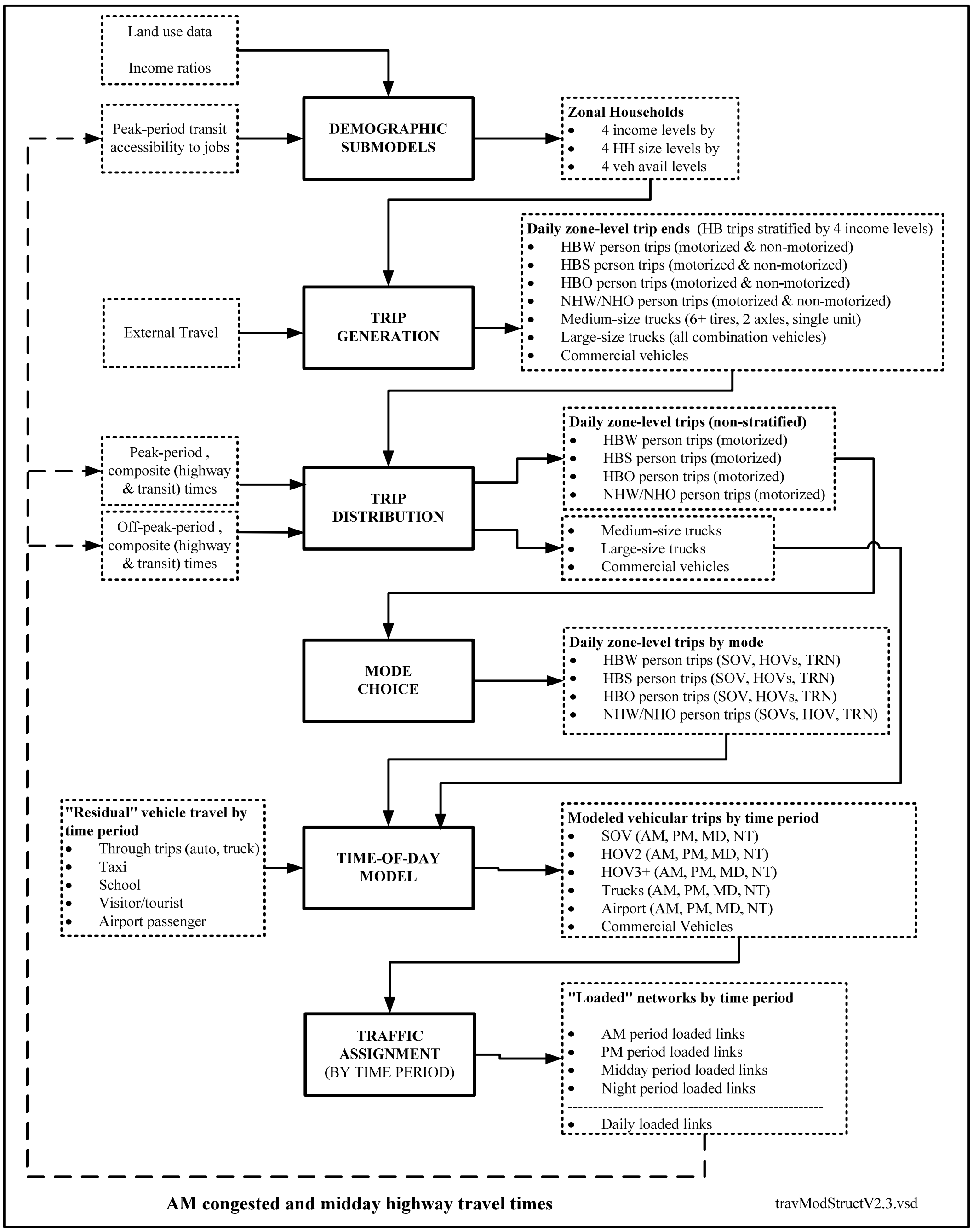


Figure 1 Major steps in the Version 2.3 Travel Model, including major inputs and outputs

This is a standard trip-based, regional travel model that uses a speed feedback (SFB) loop to ensure that the travel times and speeds coming out of traffic assignment are consistent with the travel times and speeds going into trip distribution and mode choice. Figure 2 also shows the major steps of the travel model, but with an emphasis on which steps occur within which SFB loop.

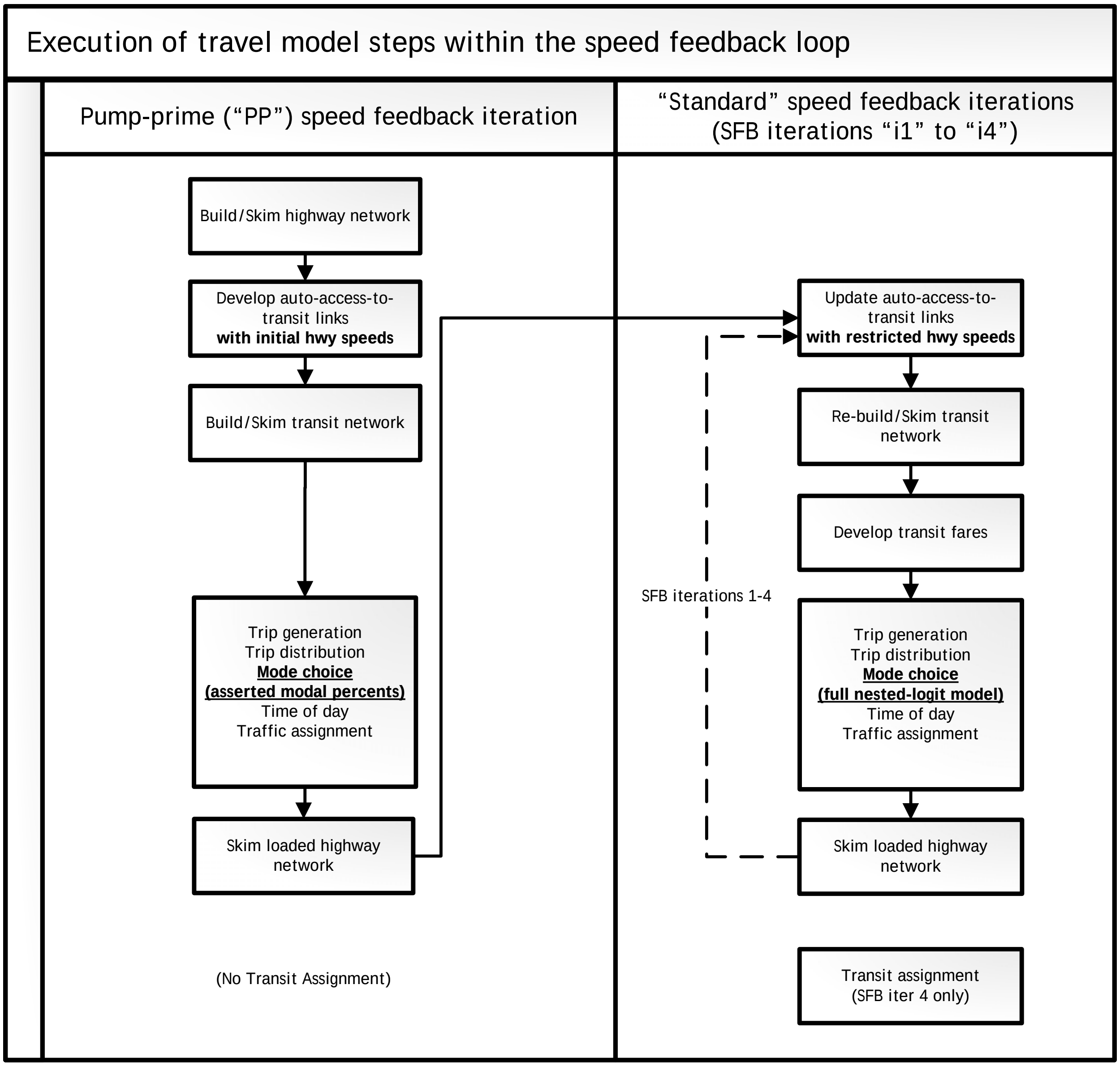


Figure 2 Application process of the Version 2.3 Travel Model

Ref: "I:\ateam\docum\FY11\Ver2.3\modelDoc_v3\02_userGuide\pumpPrime_vs_other_sfb_iter_v2.vsd"

Before the loop is begun, there is an initialization phase, known as the "pump-prime" iteration. In the pump prime iteration, a first pass of the travel model is performed using *initial* AM and off-peak highway speeds, and *initial* mode choice percentages (i.e., the mode choice model is not executed in the pump prime iteration). The "skimmed" highway times are used to develop drive-access-to-transit (zone-to-

PNR-lot) links as part of the transit network. After the transit network is built and skimmed, trip generation and trip distribution are executed. The resulting person trips are converted to vehicle trips on the basis of default zone-level mode choice and car occupancy percentages, and are assigned to the highway network.

The next series of "standard" SFB iterations (1 through 4) involve the execution of the complete travel model which includes: 1) a mode choice model execution and 2) the use of recycled traffic assignment-based speeds as input. The AM peak and off-peak restrained highway times are used to update the zone-to-PNR link speeds, and the transit network is re-built and skimmed. The highway and transit time skims are used as inputs to the mode choice model. The auto driver trips produced from the mode choice model are processed through the time-of-day model, which apportions the auto drivers among four time-of-day periods: the AM peak period (6 - 9 AM), the midday period (9 AM - 3 PM), the PM peak period (3 - 7 PM), and the night/early morning period (7 PM - 6 AM). The four time-of-day trip tables are subsequently loaded onto the highway network in separate traffic assignment procedures. The loaded-link volumes are successively averaged using the method of successive averages (MSA) to facilitate the convergence of the final link speeds. The averaging occurs individually for each of the four time-of-day periods at the link level, as follows:

- The "final" first iteration link volumes are equal to the "raw" assigned link volumes from the pump-prime iteration.
- The "final" second iteration link volume equals one half of the first iteration link volume plus one half of the second iteration assigned link volume.
- The "final" third iteration link volume equals 2/3 of the "final" second iteration link volume plus 1/3 of the third iteration assigned volume.
- The "final" fourth iteration volume is not averaged -- it is the direct assignment output.

In both the Version 2.2 and 2.3 travel models, a fixed number of speed-feedback (SFB) iterations are used. In the Version 2.2 model, it was six speed feedback iterations (in addition to the pump prime iteration). In the Version 2.3 model, we are using four speed feedback iterations (in addition to the pump prime iteration). Consequently, the final loaded-link highway network is now assign_output_i4.net, not i6hwy.net, as was the case with the Version 2.2 Travel Model.

TPB staff felt that it made sense to reduce the number of speed feedback iterations, given that the traffic assignment itself is now more converged. For example, the Version 2.2 model traffic assignment was achieving relative gaps between 10^{-4} (0.0001) and 10^{-2} (0.01), depending on the traffic assignment:

- about 10^{-2} (ca. 0.01 to 0.02) for the AM and PM non-HOV 3+ assignments
- about 10^{-3} (ca. 0.002) for the off-peak assignment
- about 10^{-4} (ca. 0.0001 to 0.0002) for the AM and PM HOV 3+ assignments

By contrast, the Version 2.3, there is a dual convergence/stopping criterion: attain a relative gap of 10^{-3} (0.001) or 300 user equilibrium iterations, whichever comes first. This is done for all six of the traffic assignments, as discussed in Chapter 8 of the calibration report and as shown in Table 1.

Table 1 Traffic assignment in the Version 2.3 Travel Model: six traffic assignments, with one, five, or six user classes

	# UE Iterations	Period	User classes
Assignment 1	Rel. gap= 10^{-3} OR 300 UE	AM	1. SOV 2. HOV 2 3. Trucks 4. Commercial Vehicles 5. Airport PAX
Assignment 2	Rel. gap= 10^{-3} OR 300 UE	AM	1. HOV 3+
Assignment 3	Rel. gap= 10^{-3} OR 300 UE	PM	1. SOV 2. HOV 2 3. Trucks 4. Commercial Vehicles 5. Airport PAX
Assignment 4	Rel. gap= 10^{-3} OR 300 UE	PM	1. HOV 3+
Assignment 5	Rel. gap= 10^{-3} OR 300 UE	Midday	1. SOV 2. HOV 2 3. HOV 3+ 4. Trucks 5. Commercial Vehicles 6. Airport PAX
Assignment 6	Rel. gap= 10^{-3} OR 300 UE	Night Time	1. SOV 2. HOV 2 3. HOV 3+ 4. Trucks 5. Commercial Vehicles 6. Airport PAX

Due to the higher level of traffic assignment convergence in Version 2.3 compared to Version 2.2, it was felt that four speed feedback iterations were sufficient.

1.3 Hardware and software requirements

The Version 2.3 Travel Model operates on only computers running a version of the Microsoft Windows operating system, such as Windows XP, Windows Vista, Windows 7, or Windows Server 2008. It is recommended that the computer have a central processing unit (CPU) with a speed of at least 3 GHz and have at least 3-4 GB of RAM and a hard disk drive with 500 GB of free space. Ample disk space is an important consideration as the Version 2.3 Travel Model generates over 1,600 files for a single scenario/year, requiring about 24 GB of space. Most standard desktop computers on the market today are equipped with a CPU that has multiple cores (e.g., a "quad core" computer). It is desirable to have a computer with four or more cores, since the Version 2.3 Travel Model is designed to allow one to run some steps in parallel on computers with multiple cores. This is known as "distributed processing," or DP, which is called Cube Cluster in Cube Voyager (described in more detail later in this chapter). TPB staff now performs the majority of model runs on servers that are accessed via Remote Desktop Connection, though some runs are still performed on desktop computers and laptop computers. For reference purposes, the computer specifications ("specs") of the two travel model servers used by the TPB staff models development group (bought in 2009 and 2011) are shown in Table 2 and Table 3.

Table 2 Specs of COG/TPB travel model server #3 (tms3), 2009

Attribute	Value
Processor Name	Intel Xeon W5580 CPU @ 3.20GHz
Number of processors in system	2
Active cores per processor	4
Total number of cores	8
Memory	4.0 GB
Hard drive	Network attached storage (NAS, O drive), 1.99 TB[3]
Operating system	Windows Server Standard, SP2, 32-bit

Table 3 Specs of COG/TPB travel model server #5 (tms5), 2011

Attribute	Value
Processor Name	Intel Xeon X5690 CPU @ 3.47 GHz
Number of processors in system	2
Active cores per processor	6
Total number of cores	12
Memory	12.0 GB
Hard drive	C (80 GB) and E (4.46 TGB, also mapped to X)
Operating system	Windows Server Enterprise, 2007, SP2, 32-bit

The Cube Voyager software[4] must be installed on the computer in order for the Version 2.3 model to function. TPB staff has determined that the Version 2.3 model operates under Cube Voyager versions 5.1.2 and 5.1.3. Unfortunately, the model will <u>not</u> run under Cube Voyager version 5.1.1. In addition to Cube Voyager, Cube Base, the graphical user interface of the Cube suite of software, is needed to edit and view binary highway and transit networks, and binary trip tables. It is recommended that one have

[3] At COG, the share "\\tms-nas\model_dev" is mapped to the O drive and the share "\\tms-nas\model_app" is mapped to the N drive. The names of these two shares follow the Universal Naming Convention or Uniform Naming Convention (UNC) format: of \\ComputerName\SharedFolder\Resource.
[4]Visit www.citilabs.com for more information on the software.

Cube Cluster, since it can be used to dramatically reduce model run times, though it is not strictly required to run the model.

The Version 2.3 Travel Model is run from the Windows command window/prompt (what used to be called the DOS prompt) using a series of batch files. One can use Windows Explorer to navigate to the "root" subdirectory/folder (see Figure 3) where a model run will be launched. In this case, it is useful to have a way to open a command window at the selected location in Windows Explorer. Some versions of Windows, such as Windows XP, have no built-in way to do this. In this case, it is useful to use the "Open Command Window Here" PowerToy (CmdHere.exe), which lets one open a command window from a Windows Explorer window with a "right-click" of the mouse. In other versions of Windows, however, such as Windows 7, this functionality is built into the operating system (one simply holds down the shift key while "right-clicking" the mouse). This is discussed in more detail in section 1.7 (Running the travel model) on page 1-20.

TPB staff currently includes two other software utilities which are used in the standard application batch files. First, TimeThis.exe is used to record the elapsed time of a specific command. The utility is used in the Version 2.3 process to time a model execution. The second utility, Tee.exe, is used to split standard output (normally sent to either the screen or a file) to both the screen and a file at the same time. This is part of the Windows 2000 Resource Kit.

TPB staff currently uses an ArcGIS-based process for developing the proportion of each TAZ that is within short- and long-walk distances to transit service coded in the regional transit network. This involves creating "walksheds" around transit stop nodes and then calculating the percent of each TAZ that lies within a short and long walk of transit. TPB staff has developed an automated GIS procedure, known as the "walkshed generator," which both creates the needed walksheds and then calculates the associated percent walk to transit values.[5] This procedure creates point buffers around transit stop nodes and then overlays these point buffers with TAZ boundaries. Using the walksheds, the procedure then calculates the six walk percent values needed by the model. Before running the "walkshed generator," it is useful to create a series of transit stop node shapefiles that are used by the walkshed generator.[6] The ArcGIS application is not currently distributed with the Version 2.3 application package, but it can be made available upon request. The application requires ArcGIS 9.3.1 software.

1.4 Model run times

Model run times have increased substantially in the Version 2.3 travel model (3,722 TAZ), compared to the Version 2.2 travel model (2,191 TAZ). For example, using a server bought in 2009 (such as tms3), a run of the Ver. 2.2 travel model would take 15-20 hours. By contrast, a run of the Version 2.3 travel model on the same computer required around 30 hours using Cube Cluster and 80-90 hours without

---

[5] Yew Yuan, *Walkshed Generator User Guide* (Washington, D.C.: National Capital Region Transportation Planning Board, August 25, 2010).

[6] Mark S. Moran, "Creating transit stop node shapefiles needed as inputs to the TPB procedure for generating transit walksheds and calculating the share of each zone within walking distance to transit," Memorandum, September 6, 2011.

Cube Cluster. On a new computer, such as tms5, the model run time is about 27 hours. The key reasons why the Version 2.3 Travel Model takes so much more model run time than the Version 2.2 Travel Model are explained below.

The number of zones (TAZ) has increased by a factor of 1.7 (2,191 in Ver. 2.2 and 3,722 in Ver. 2.3). This has resulted in an even larger increase in matrix sizes, which are now 2.89 times bigger than before (= $3{,}722^2/2{,}191^2$). The other factors causing longer run times are associated mainly with Version 2.3 refinements to the Version 2.2 traffic assignment process:

- The number of time-of-day periods went from three (AM, PM, and off peak) to four (AM, midday, PM, night/early morning)
- The number of user classes went from five to six (an explicit commercial-vehicle user class has been added);
- The number of traffic assignments has increased. The Version 2.2 travel model had originally used three traffic assignments, one for each time-of-day period (AM, PM, and off peak). Later versions of the Version 2.2 travel model split the peak assignments into two groups (HOV3+ and non-HOV3+, the so called "two step traffic assignment"), resulting in the five assignments shown in the left-hand column of Table 4. In the Version 2.3 travel model, the off-peak period has been further split into two parts: midday and night/early morning. So, the number of traffic assignments has increased from five in Version 2.2 to six in Version 2.3 (see the right-hand column of Table 4 or Table 1).
- Higher convergence thresholds
    - In the Version 2.2 model, all five traffic assignments were run with 60 user equilibrium (UE) iterations. This resulted in a range of relative gaps values, from a low value of 1.10×10^{-4} (0.0001) for the AM HOV3+ assignment to a high of 1.19×10^{-2} (0.0119) for the AM non-HOV3+ assignment.[7] [8]
    - In the Version 2.3 model, all six traffic assignments are run to either a relative gap of 0.001 (1×10^{-3}) or 300 user equilibrium iterations, whichever comes first. However, despite the higher levels of convergence, some recent sensitivity tests have indicated that some uses of the travel model, such a project planning studies, may require even higher levels of convergence, e.g., a relative gap of 10^{-4}, due, in part to the fact that the Version 2.3 highway networks have more road links than Version 2.2 highway networks.[9]

---

[7] From a model run representing year-2002 conditions from the air quality conformity determination of the 2009 CLRP/FY 2010-2015 TIP.

[8] The modeler can check the relative gap by consulting the highway assignment report file for the final speed feedback iteration (i.e., i4_Highway_Assignment.rpt). The variable is called RELGAP.

[9] Ronald Milone and Mark S. Moran, "TPB Version 2.3 travel model on the 3,722-TAZ area system: Status report and sensitivity tests" (presented at the July 22, 2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., July 22, 2011); Dusan Vuksan and Feng Xie, "TPB Version 2.3 travel model on the 3,722-TAZ area system: Corridor-level sensitivity tests" (presented at the September 23,

Table 4 Five traffic assignments in the Version 2.2 travel model becomes six traffic assignments in the Version 2.3 travel model

Version 2.2 model: Five assignments	Version 2.3 model: Six assignments
AM Non-HOV3+	AM Non-HOV3+
AM HOV3+	AM HOV3+
PM Non-HOV3+	PM Non-HOV3+
PM HOV3+	PM HOV3+
Off peak	Midday
	Night and early morning

1.5 Distributed processing

The use of Cube Cluster, the Citilabs' implementation of distributed processing, is another refinement of the Version 2.3 process. Distributed processing is essentially the use of multiple CPUs or cores or both, running in parallel, as a means of reducing the computation time. The distributed processing feature is designed to be used on computers with multiple CPUs, cores, or both.[10] Recent "builds" of the Version 2.3 Travel Model, such as Ver. 2.3.36, have been designed to make use of Cube Cluster, although users have the option to disable this feature, in cases where they do not own Cube Cluster or have older hardware.

Cube Cluster includes two types of distributed processing: intra-step distributed processing (IDP) and multi-step distributed processing (MDP). IDP can be applied to only two Voyager modules: MATRIX and HIGHWAY. MDP can be used for any Voyager module and even non-Voyager programs, such as Fortran programs. When distributed processing was added to the TPB travel model, it was first added to the highway assignment script (via intra-step distributed processing), since highway assignment accounted for about half the run time of the travel model. As of Version 2.3.27 and beyond, we have now added IDP to MATRIX procedures used by the following scripts:

- MFARE2.s
- Transit_Skims_[MR|CR|BM|AB].s

In order to use Cube Cluster, two items are needed: the Cube Cluster module and a computer capable of running distributed processing. The Cube Cluster module is available for purchase from Citilabs (www.citilabs.com). Your Cube license must also reflect the fact that you have Cube Cluster. In terms of computer requirements, a computer must either have multiple CPUs (processors), or one CPU with

2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., September 23, 2011).

[10] Computers with "hyperthreading" turned on may also benefit from distributed processing.

multiple cores, or a combination of these.[11] As noted earlier, for users who do not want to or cannot use Cube Cluster, this option can easily be disabled (as explained below).

1.6 Preparing for a model run

A structured application procedure has been established for applying the Version 2.3 model from a command-prompt window. The procedure involves:

- A series of pre-established batch files, which are used to call a series of Cube Voyager scripts (*.s) and Fortran programs (*.exe);
- A standardized subdirectory system, in which input files, output files, Cube Voyager scripts, and other files are rigidly organized; and
- The use of generically named input and output files, which are stored in designated locations in the subdirectory system.

An example subdirectory structure for applying the Version 2.3 model is shown in Figure 3. The "root" subdirectory appears at the top of the structure. The root subdirectory may exist anywhere on the computer hard drive and may be arbitrarily named by the analyst, but **it recommended that the name of the root subdirectory include information about both the travel model being used (e.g. Ver2.3.36)[12] and the modeling project being undertaken**. For example, an analyst performing model runs to support the Air Quality Conformity Determination of the 2011 Constrained Long-Range Transportation Plan (CLRP) might name the root subdirectory as follows:

`C:\modelRuns\fy12\`**`Ver2.3.36_aqc_2011clrp`**

Note that the root subdirectory need not be located directly off the root of the C drive (or D drive, etc.). In the example above, the root subdirectory is below the "fy12" subdirectory, which is below the "modelRuns" subdirectory. On the left side of Figure 3, there are five specially designated subdirectories under the root that are established:

- Fortran executables (\SOFTWARE)
- Control files that are required by some of the executables (\CONTROLS)
- Cube Voyager scripts (\SCRIPTS)
- General parameter files used by the scripts or executables (\SUPPORT)
- Summary scripts, which are optional (\SUMMARY)

The first four subdirectories are required, but the fifth subdirectory is optional. The SUPPORT subdirectory is reserved for parameter files that generally do not change by modeled scenario such as K-factors, F-factors, and the like. <u>The four required subdirectories must exist under the root, and must be</u>

[11] To the Windows operating system, a computer with one a dual-core CPU appears the same as a computer with two individual CPUs.

[12] The "36" in the version number (2.3.36) refers to the fact that is the 36th "build" or update of the Version 2.3 model.

named as shown. The optional subdirectory – summary – may carry any name. Furthermore, the files residing in these four required subdirectories should generally not be altered or renamed.

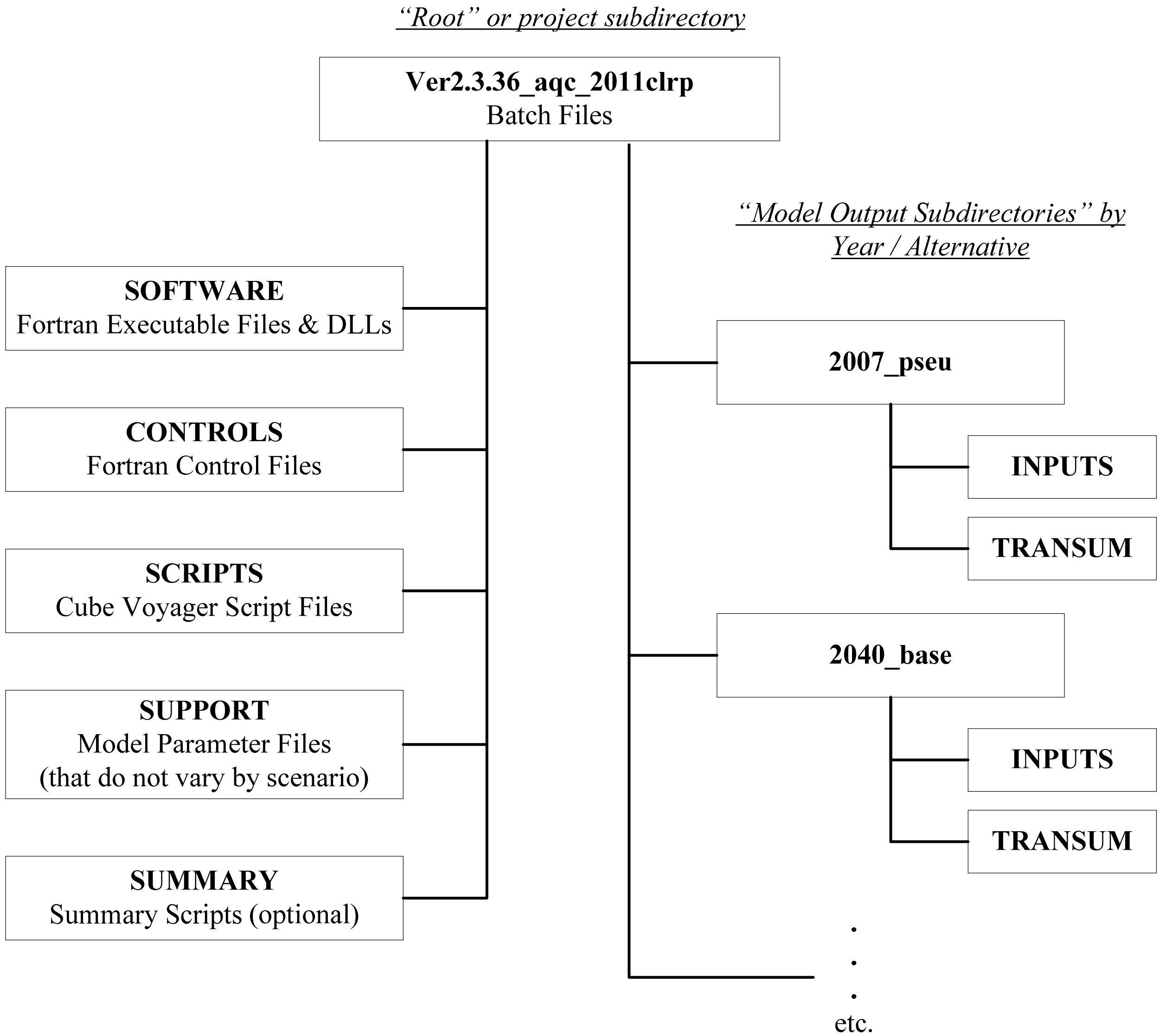


Figure 3 Subdirectory structure for executing the Version 2.3 travel model

Ref: "I:\ateam\docum\FY11\Ver2.3\modelDoc_v3\02_userGuide\directoryStruct_v23_model_v4.vsd"

The right side of Figure 3 shows two subdirectories, named "2007_pseu" and "2040_base." In the first case, the name means the year-2007 version of highway and transit networks, using the Pseudo Round 8.0 land use, which was used for the calibration of the travel model.[13] These two subdirectories are the

[13] Ronald Milone to DTP Technical Staff, "2007 Land Activity File Development for the Version 2.3 Model," Memorandum, June 7, 2010; Ronald Milone et al., *TPB Version 2.3 Travel Forecasting Model for the 3,722-Zone Area System: Calibration Report*, Draft report (Washington, D.C.: National Capital Region Transportation Planning Board, February 28, 2011).

output subdirectories (a.k.a. the scenario subdirectories). The user is free to choose any name for output subdirectories. A travel demand modeling project would typically have two or more scenarios or alternatives. Each alternative would get its own output subdirectory for scenario-specific outputs from the travel model. For example, a modeling project with four alternatives might have output scenarios with the following names:

- 2007_pseu
- 2030base
- 2030lowGrowth
- 2030highGrowth

Under each scenario-specific subdirectory that exists, there is an INPUTS subdirectory, which must be named "inputs" (names are case insensitive). The \inputs subdirectory is where one stores all necessary model inputs that area specific to a modeled scenarios. Note that some "inputs" that are common to all modeled scenarios are stored in the \support subdirectory (see the list of inputs later in this chapter). Input files in the \inputs folder are named generically (e.g., land use data is stored in a file named zone.dbf; network link data is stored in a file named link.dbf, etc.). The user may establish an unlimited number of output subdirectories, as long as each one contains one \inputs subdirectory. \inputs subdirectories cannot be shared among more than one alternative.

If one is running multiple scenarios, it is recommended that the analyst set up an electronic spreadsheet to keep track of metadata associated with each model run. The metadata of importance will vary from study to study, but might contain items such as:

- Run number/ID (a unique sequence number to quickly name a model run)
- Parent run number/ID (indicates the run number of the run that formed the basis for the current run). Useful in figuring which run was derived from which other runs.
- Subdirectory name (i.e., the name of the root folder/subdirectory)
- Key modeling assumption parameters, such as the network year, land use year, land use round (e.g., Round 7.2a), WMATA tariff number, etc.
- Key modeling output parameters, such as model run time, regional VMT, total transit, etc.

Pre-established "parent" and "child" batch files for executing the model reside in the root subdirectory. Typically "parent" batch files are edited to correspond to each modeled scenario, while "child" batch files remain unaltered. The parent batch files can be named as the user likes. The two main parent batch files are the "wrapper" batch file and the "run model steps" batch file (the latter file used to be called the "run all" batch file). Details about these two files can be found in section 1.7.1 on page 1-20 and section 1.7.2 on page 1-26. The child batch files are the ones that actually execute individual modeling steps, such as the trip generation step (e.g., Trip_Generation.bat) or the traffic assignment step (e.g., Highway_Assignment.bat). Child batch files generally call the Cube Voyager scripts and/or FORTRAN programs. The child batch files also assign names to report files that result from each model step. Listing files are typically assigned file extensions of RPT or TAB. The former refers to Cube Voyager report or listing files, while the latter refers to a subset tabulation of the report file containing only trip table totals or jurisdictional summaries. Parent batch files are used to string child batch files together so that the entire model execution can be initiated with a single command or batch file. The parent batch files also establish Windows environment variables that are used in the child batch files and Cube Voyager scripts, such as the iteration number, the model year, and the model description.

All of the input files located in the \inputs, \controls, and \support folders are listed in Table 5. It is the user's responsibility to make sure that the generically named files are appropriate for the modeled scenario and are in the prescribed format (described later). Additionally, all of the files shown in Table 5 must exist unless they are listed as optional. The advantage of using generic filenames is that the input and output filenames referenced in each Cube Voyager script and control file do not need to be tailored to match the different scenarios that are run. The disadvantage of using generic filenames is that, when moving or sharing files, two files with the same name could be quite different (e.g., zone.dbf for the year 2007 has the same name as zone.dbf for the year 2040). Thus, the metadata that describes the scenario name is stored in the name of the output subdirectory (e.g., "2007"), not in the filenames themselves. A list of the FORTRAN executables and the dynamic-link library (DLL) files residing in the \SOFTWARE subdirectory is shown in Table 6. There are fewer executables (only two) used by the Versions 2.3 model than have been used in previous TPB models, since several Fortran routines have been converted to Cube Voyager scripts.

A listing of child batch files is provided in Table 7. The table also indicates the programs and/or Cube Voyager scripts that are invoked and the purpose of each batch file. Given the iterative application process of the model, most of the batch files are called multiple times during a model run. The sequence of batch file applications, by iteration, is shown in Table 8. The table indicates that there are 47 batch steps called during a standard application of the model. Some of the batch files are called once, while others (e.g., trip_generation.bat) are called during the pump-prime and all four standard iterations. A parent batch file is used to string each of the 47 child batch files together during a typical model execution. The parent batch files, like child batch files, reside in the root subdirectory. Two parent batch files are typically prepared for each individual model run. The process for executing a model is addressed in the next section. The remaining chapters address the specific details of each modeling step.

Table 5 Input files, sorted by folder, needed to run the Version 2.3 travel model

No.	Approx Sequ. in Model	Folder	Filename	Description	File Type	Category	First used by script/Fortran program	Transit/ Non-Transit Mode(s)
1	79	controls	HBO_NL_MC.ctl	HBO nested-logit mode choice mode control file	Text	Model	AEMS.exe	
2	78	controls	HBS_NL_MC.ctl	HBS nested-logit mode choice mode control file	Text	Model	AEMS.exe	
3	77	controls	HBW_NL_MC.ctl	HBW nested-logit mode choice mode control file	Text	Model	AEMS.exe	
4	81	controls	NHO_NL_MC.ctl	NHO nested-logit mode choice mode control file	Text	Model	AEMS.exe	
5	80	controls	NHW_NL_MC.ctl	NHW nested-logit mode choice mode control file	Text	Model	AEMS.exe	
6	58	inputs	airpax.adr	Air Passenger Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
7	9	inputs	AM_Tfac.dbf	AM Toll Factors by Vehicle Type	DBF	Assumptions	Highway_skims.s	
8	68	inputs	areadef3722.prn	Input TAZ-Mode choice district equivalence	Text	Assumptions	prefarv23.s	
9	25	inputs	bus_pnrn.tb	Bus PNR lots	Text	Transit network	transit_skims_??.s	1,2,6-9
10	75	inputs	BUSFARAM.ASC	AM Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
11	76	inputs	BUSFAROP.ASC	OP Bus Fare matrix (Bus fares zones '1' to '21')	Text	Assumptions	mfare2.s	
12	31	inputs	com_bus.tb	Transfer link (walk) between commuter rail station and bus & LRT stop	Text	Transit network	transit_skims_??.s	12
13	18	inputs	com_link.tb	Commuter rail links	Text	Transit network	transit_skims_??.s	4
14	22	inputs	com_node.tb	Commuter rail stations	Text	Transit network	transit_skims_??.s	4
15	27	inputs	com_pnrn.tb	Commuter rail PNR lots	Text	Transit network	transit_skims_??.s	4
16	1	inputs	CPI_File.txt	Assumed rate of inflation, based on historical CPI	Text	Assumptions	Set_CPI.s	
17	37	inputs	Ext_PsAs.dbf	External Productions and Attractions	DBF	Observed data	trip_generation.s	
18	49	inputs	HBO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
19	48	inputs	HBS_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
20	47	inputs	HBW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
21	15	inputs	Jur.dbf	Equiv. between juris and river superdistricts: Disallows river crossings for PNR	DBF	Land use	Autoacc4.s	
22	34	inputs	Lbus_timFTRS.asc	Local Bus Time Degradation Factors	Text	Assumptions	transit_skims_??.s	
23	4	inputs	Link.dbf	Highway network links	DBF	Highway network	V2.3_Highway_Build.s	
24	32	inputs	lrt_bus.tb	Transfer link (walk) between LRT station and bus stop	Text	Transit network	transit_skims_??.s	12
25	19	inputs	lrt_link.tb	LRT links	Text	Transit network	transit_skims_??.s	5
26	23	inputs	lrt_node.tb	LRT stations/stops	Text	Transit network	transit_skims_??.s	5
27	28	inputs	lrt_pnrn.tb	LRT PNR lots	Text	Transit network	transit_skims_??.s	5
28	10	inputs	MD_Tfac.dbf	MD Toll Factors by Vehicle Type	DBF	Assumptions	Highway_skims.s	
29	30	inputs	met_bus.tb	Transfer link (walk) between Metrorail station and bus stop	Text	Transit network	transit_skims_??.s	12
30	17	inputs	met_link.tb	Metrorail links	Text	Transit network	transit_skims_??.s	3
31	21	inputs	met_node.tb	Metrorail stations	Text	Transit network	transit_skims_??.s	3
32	26	inputs	met_pnrn.tb	Metrorail PNR lots	Text	Transit network	transit_skims_??.s	3
33	70	inputs	metlnkm1.tb	Metrorail Links	Text	Transit network	metrorail_skims.s	
34	69	inputs	metnodm1.tb	Metrorail Nodes	Text	Transit network	metrorail_skims.s	
35	72	inputs	mfare1.a1	Metrorail Sta XYs scaled to 1/100ths of miles	Text	Transit network	mfare1.s	
36	73	inputs	mfare1_Sta_Disc.ASC	Metrorail Sta fare discount array in cents	Text	Assumptions	mfare1.s	
37	35	inputs	MODE1AM,…MODE10AM.tb	AM Transit Line Files	Text	Transit network	transit_skims_??.s	
38	36	inputs	MODE1OP,…MODE10OP.tb	OP Transit Line Files	Text	Transit network	transit_skims_??.s	
39	33	inputs	new_bus.tb	Transfer link (walk) between BRT/streetcar stop and bus stop	Text	Transit network	transit_skims_??.s	12
40	20	inputs	new_link.tb	BRT/streetcar links	Text	Transit network	transit_skims_??.s	10
41	24	inputs	new_node.tb	BRT/streetcar stations/stops	Text	Transit network	transit_skims_??.s	10
42	29	inputs	new_pnrn.tb	BRT/streetcar PNR lots	Text	Transit network	transit_skims_??.s	10
43	51	inputs	NHO_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	
44	50	inputs	NHW_NL_MC.MTT	Pre-existing mode choice model output	Binary	Assumptions	pp_auto_drivers.s	

No.	Approx Sequ. in Model	Folder	Filename	Description	File Type	Category	First used by script/Fortran program	Transit/ Non-Transit Mode(s)
45	2	inputs	Node.dbf	XY coordinates of nodes in highway network	DBF	Highway network	AreaType_File.s	
46	64	inputs	NT_Tfac.dbf	NT Toll Factors by Vehicle Type	DBF	Assumptions	highway_assignment.s	
47	16	inputs	Pen.dbf	List of TAZs considered to be in the "slugging" shed of the Pentagon	DBF	Assumptions	Autoacc4.s	
48	63	inputs	PM_Tfac.dbf	PM Toll Factors by Vehicle Type	DBF	Assumptions	highway_assignment.s	
49	57	inputs	schl.adr	School Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
50	14	inputs	StaAcc.dbf	Lookup table: Maximum drive-access-to-transit distances	DBF	Assumptions	Autoacc4.s	
51	13	inputs	station.dbf	Station file: Metrorail, commuter rail, LRT stations/PNR lots and bus PNR lots	DBF	Transit network	parker.s	
52	74	inputs	tariff.txt	WMATA tariff policy	Text	Assumptions	mfare1.s	
53	55	inputs	taxi.adr	Taxi Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
54	67	inputs	tazfrzn.asc	Fare Zone File	Text	Assumptions	prefarv23.s	
55	7	inputs	Toll_Esc.dbf	Toll escalation assumptions: Highway tolls & deflators	DBF	Assumptions	V2.3_Highway_Build.s	
56	71	inputs	trnpen.dat	Turn Penalty file	Text	Assumptions	metrorail_skims.s	
57	56	inputs	visi.adr	Visitor Auto Driver Trips	Binary	Assumptions	misc_time-of-day.s	
58	11	inputs	xtrawalk.dbf	Extra walk links that the analyst wishes to include	DBF	Transit network	walkacc.s	13
59	54	inputs	xxaut.vtt	Auto Driver Through Trips	Binary	Assumptions	misc_time-of-day.s	
60	53	inputs	XXCVT.vtt	Com/Mtk/Htk through Trips	Binary	Calculated data	misc_time-of-day.s	
61	3	inputs	Zone.dbf	Land use/land activity data at zonal level, 3722 TAZ	DBF	Land use	areaType_File.s	
62	11	inputs	Areawalk.txt	TAZ area within walking distance to transit (created in pct. walk)	Text	Transit Network	walkacc.s	
63	5	support	AMSPD.LKP	Initial lookup speeds used for highway links, AM period	Text	Highway network	V2.3_Highway_Build.s	
64	42	support	AttrRates.dbf	Trip Attractions	DBF	Calculated data	trip_generation.s	
65	60	support	cvdelta_3722.trp	Calibration matrix, or "delta table" for commercial vehicles	Binary	Assumptions	misc_time-of-day.s	
66	38	support	GIS_variables.dbf	Input Zonal GIS variable File	DBF	Calculated data	trip_generation.s	
67	43	support	HBINCRAT.dbf	HB Income Shares	DBF	Calculated data	trip_generation.s	
68	65	support	hwy_assign_capSpeedLookup.s	FT x AT Speed & Capacity lookup	Text	Highway network	highway_assignment.s	
69	66	support	hwy_assign_Conical_VDF.s	Volume Delay Functions file	Text	Highway network	highway_assignment.s	
70	6	support	MDSPD.LKP	Initial lookup speeds used for highway links, midday	Text	Highway network	V2.3_Highway_Build.s	
71	41	support	NMArates.dbf	Non-motorized Trip Attractions	DBF	Calculated data	trip_generation.s	
72	40	support	NMPrates.dbf	Non-motorized Trip Productions	DBF	Calculated data	trip_generation.s	
73	59	support	tkdelta_3722.trp	Calibration matrix, or "delta table" for med and hvy truck	Binary	Assumptions	misc_time-of-day.s	
74	52	support	todcomp_2008HTS.dbf	Time of day model/factors	Binary	Assumptions	time-of-day.s	
75	45	support	toll.inc	Equivalent minutes (min/'07$) by period & income level	Text	Assumptions	trip_distribution.s	
76	8	support	toll_minutes.txt	Toll minutes equivalence file by Vehicle Type	Text	Assumptions	Highway_Assignment.s	
77	44	support	Truck_Com_Trip_Rates.dbf	Truck and Commercial Vehicle Trip Rates	DBF	Calculated data	truck_com_trip_generation.s	
78	46	support	Version_23_FFtrs.dbf	F-factors	DBF	Calculated data	trip_distribution.s	
79	39	support	weighted_trip_rates.dbf	Trip Productions	DBF	Calculated data	trip_generation.s	

Ref: I:\ateam\docum\FY11\Ver2.3\modelDoc_v2\02_userGuide\V23_inputs_v8.xlsx

Table 6 Fortran executable files and dynamic-link library files required for running the Version 2.3 travel model

Executable Name	Ver	Date	Size (bytes)	Program Function	Requires a control file?
`AEMS.exe`		07/28/2009	195,900	Mode choice application program	yes
`cw3240.dll`		02/09/1998	827,392	Dynamic-link library file associated w/ AEMS.exe	no
`DFORMD.dll`		11/10/2004	425,984	Dynamic-link library file associated w/ AEMS.exe	no
`Tppdlibx.dll`		03/31/2011	274,432	Dynamic-link library file associated w/ AEMS.exe	no
`Tputlib.dll`		03/31/2011	1,069,568	Dynamic-link library file associated w/ AEMS.exe	no
`extrtab.exe`		11/23/2010	464,559	Extracts sections from Cube Voyager report files	no
`Linevol.exe`	1.0	12/03/2007	114,688	Consolidates multiple transit loaded link files (DBF) to one transit loaded link file (DBF)	yes
`Linesum.exe`	1.8	04/26/2011	167,936	Creates reports summarizing transit loaded link files	yes

Note that the date and size of the `Tppdlibx.dll` and `Tputlib.dll` files can vary with the version of Cube Voyager installed. The dates and sizes listed in the table are for Cube Voyager 5.1.3. Ideally, the versions of these two files found in the software folder (e.g., X:\modelRuns\fy12\Ver2.3.36\Software) should be the same as those found in the Citilabs folder (i.e., C:\Program Files\Citilabs\Cube for computers running 32-bit versions of Windows operating systems; or C:\Program Files (x86)\Citilabs\Cube for computers running 64-bit versions of Windows operating systems).

Table 7 "Child" batch files used for running the Version 2.3 travel model

Batch File	Scripts / Programs	Purpose
Set_Factors.bat	Set_Factors.s	Create K-factors and time penalties.
Set_CPI.bat	Set_CPI.s	Create highway and transit cost deflators.
PP_Highway_Build.bat	AreaType_File.s V2.3_highway_build.s	Build highway networks.
PP_Highway_Skims.bat	Highway_Skims.s, modnet.s Highway_Skims_mod.s joinskims.s	Create initial AM/ off-peak highway skims.
Transit_Skim_All_Modes.bat	PARKER.s WALKACC.s AUTOACC4.s Transit_skims_CR.s Transit_skims_MR.s Transit_skims_AB.s Transit_skims_BM.s transit_Accessibility.s	Create transit networks.
Transit_Fare.bat	prefarV23.s Metrorail_Skims.s Mfare1.s Mfare2.s Assemble_Skims_CR.s Assemble_Skims_MR.s Assemble_Skims_AB.s Assemble_Skims_BM.s	Create current iteration transit fares.
Trip_Generation.bat	Demo_models.s Trip_Generation.s Trip_Generation_Summary.s Truck_Com_Trip_Generation.s	Execute daily trip generation.
Trip_Distribution.bat	Prepare_Ext_Auto_Ends.s	Execute daily trip distribution.

Batch File	Scripts / Programs	Purpose
	Prepare_Ext_ComTrk_Ends.s Trip_Distribution.s	
Mode_Choice.bat	AEMS.EXE mc_NL_summary.s	Execute daily mode choice model (optionally execute mode choice model with the Transit Constraint (TC) and/or with HOV Skim Replacement (HSR)).
Auto_Driver.bat	MC_Auto_Drivers.s	Generate initial auto drivers after mode choice model.
PP_Auto_Drivers.bat	PP_Auto_Drivers.s	Generate initial auto drivers (without mode choice model).
Time-of-Day.bat	Time-of-Day.s Misc_Time-of-Day.s Prepare_Trip_Tables_for_ Assignment.s	Convert daily modeled trips to AM, PM, and Off-peak.
Highway_Assignment.bat	Highway_Assignment.s	Execute user equilibrium highway assignment for three time periods.
Highway_Skims.bat	Highway_Skims.s modnet.s Highway_Skims_mod.s joinskims.s	Create highway skims from assignment.
Transit_Assignment.bat	Combine_Tables_For_ TrAssign.s Transit_assignment_CR.s Transit_assignment_MR.s Transit_assignment_AB.s Transit_assignment_BM.s	Execute transit assignment for peak and off-peak periods.

Ref: I:\ateam\docum\FY11\Ver2.3\modelDoc\02_userGuide\V23_Flowchart_Table_v2.xlsx

Table 8 Sequence of the child batch files in the Version 2.3 travel model

		Initial (Pump Prime) Iteration	Standard Iterations			
Batch File	**Scripts / Programs**	**PP**	**1**	**2**	**3**	**4**
Set_CPI.bat	Set_CPI.s	1				
PP_Highway_Build.bat	V2.3_Highway_Build.s	2				
PP_Highway_Skims.bat	Highway_Skims.s modnet.s Highway_Skims_mod.s joinskims.s	3				
Transit_Skim_All_Modes.bat	prefarV23.s Highway_Unbuild.s PARKER.EXE WALKACC.EXE AUTOACC4.EXE transit_skims_CR.s transit_skims_MR.s transit_skims_AB.s transit_skims_BM.s	4	11	20	29	38
Transit_Fare.bat	Metrorail_Skims.s Mfare1.s Mfare2.s Assemble_Skims_CR.s Assemble_Skims_MR.s Assemble_Skims_AB.s Assemble_Skims_BM.s		12	21	30	39
Trip_Generation.bat	Demo_models.s Trip_Generation.s	5	13	22	31	40
Trip_Distribution.bat	Trip_Distribution.s	6	14	23	32	41
Mode_Choice.bat	AEMS.EXE mc_NL_summary.s		15	24	33	42
Auto_Driver.bat	MC_Auto_Drivers.s		16	25	34	43
PP_Auto_Drivers.bat	PP_Auto_Drivers.s	7				
Time-of-Day.bat	Time-of-Day.s Misc_Time-of-Day.s	8	17	26	35	44
Highway_Assignment.bat	Highway_Assignment.s	9	18	27	36	45
Highway_Skims.bat	Highway_Skims.s modnet.s Highway_Skims_mod.s joinskims.s	10	19	28	37	46
Transit_Assignment.bat	Combine_Tables_For_ TrAssign.s Transit_assignment_CR.s Transit_assignment_MR.s Transit_assignment_AB.s Transit_assignment_BM.s					47
TranSum.bat	LINEVOL.EXE LINESUM.EXE					48

Ref: I:\ateam\docum\FY11\Ver2.3\modelDoc_v2\02_userGuide\V23_Flowchart_Table_v4.xlsx

1.7 Running the travel model

To run the Version 2.3 Travel Model, the user must edit two batch files and then run one of the batch files, which, in turn, will call the other file. These two batch files are known as the parent batch files. The first parent batch file is called the "wrapper" batch file or the "run model" batch file (see Figure 4) and the second is called the "run model steps" (formerly "run all") batch file (see Figure 7 or Figure 8). In computer programming, the term "wrapper function" is used for a function whose main purpose is to call a second function. We are using this term in a similar vein, since the main purpose of our wrapper batch file is to call a second batch file (the "run model steps" batch file) and set up the running environment for the model run. Once the user has edited the two parent batch files with a text editor, the user launches the model run by launching the wrapper batch file within a command prompt window that is pointing to the root directory. For example, if the root directory is

```
C:\modelRuns\fy12\Ver2.3.36_aqc_2011clrp
```

Then the user would open a command prompt window at this location and type the name of the wrapper batch file an press Enter to execute it. One can open Windows Explorer and navigate to the root directory, and then select the root subdirectory. In Windows XP and Windows Server 2003 (if one has installed the Windows "Open Command Window Here" PowerToy -- CmdHere.exe), one right-clicks the root folder and selects "Open Command Window Here." In Windows 7 and Windows Server 2008, where this feature is built into the operating system, one selects the folder in the left pane, and then, with nothing selected in the right pane, one uses the mouse to shift-right-click in the right pane, selecting "Open Command Window Here."

1.7.1 Wrapper batch file

The first part of the "wrapper" batch file (Figure 4), lines 4-6, sets up some Windows environment variables (root, scenar, runbat), described below:

- Root: The name of the root folder where the model is to be run. The default value is a single period ("."), which refers to the current directory where the command is launched. If one wants, one can specify an absolute path (e.g., C:\modelRuns\fy12\Ver2.3.36_aqc_2011clrp) or a relative path (e.g., ..\fy12\Ver2.3.36_aqc_2011clrp). But, generally, it is easier and safer to use the default value of "." (the current working directory).
- Scenar: The name of the run scenario. This is the name of the output folder/subdirectory. Typical names are four-digit years (e.g., 2007, 2040), but the user is free to use any name, provided it matches the name of the output subdirectory (e.g., 2007, 2040_households_out, 2040_jobs_in). In this example, the name is "2007_pseu", which refers to the year-2007 network with the Pseudo Round 8.0 land use, which was used for the calibration of the travel model.
- Runbat: This is the name of the "run model steps" batch file, e.g., run_ModelSteps_Ver2.3.36_2007_pseu.bat

```
 1 :: E:\modelRuns\fy12\Ver2.3.36\run_Model_Ver2.3.36_2007_pseu.bat
 2 :: 2011-10-28 Fri 15:21:08
 3
 4 set root=.
 5 set scenar=2007_pseu
 6 set runbat=run_ModelSteps_Ver2.3.36_2007_pseu.bat
 7 :: Environment variables for (intrastep) distributed processing:
 8 ::     use IDP = t/f (for true or false)
 9 ::     Number of subnodes:  1-3 => 3 subnodes and one main node = 4 nodes in total
10 set useIdp=t
11 set subnode=1-3
12
13 :: This command will
14 ::   1) time the model run (using timethis.exe and the double quotes)
15 ::   2) redirect standard output and standard error to a file
16 ::   3) Use the tee command so that stderr & stdout are sent both to the file and the screen
17 ::
18
19 :: Start Cube Cluster, create N sub-nodes for processing (these will work with the main node already running)
20 Cluster.exe %root%\%scenar%\mwcog %subnode% start exit
21
22 timethis "%runbat%  %scenar%" 2>&1 | tee %root%\%scenar%\%scenar%_fulloutput.txt
23
24 :: Close the N sub-nodes
25 Cluster.exe %root%\%scenar%\mwcog %subnode% close exit
26
27 :: Open up the file containing the stderr and stdout
28 if exist %root%\%scenar%\%scenar%_fulloutput.txt start %root%\%scenar%\%scenar%_fulloutput.txt
29
30 :: Look four errors in the reports and output files
31 call searchForErrs.bat  %scenar%
32 :: Open up the file containing any errors found
33 if exist %root%\searchForErrs.txt      start %root%\searchForErrs.txt
34
35 :: Open up other report files
36 if exist %root%\%scenar%\i4_Highway_Assignment.rpt start %root%\%scenar%\i4_Highway_Assignment.rpt
37 if exist %root%\%scenar%\i4_mc_NL_summary.txt      start %root%\%scenar%\i4_mc_NL_summary.txt
38
39 :: Cleanup
40 set root=
41 set scenar=
42 set runbat=
43 set subnode=
```

Figure 4 Wrapper batch file (run_Model_Ver2.3.36_2007_pseu.bat), which is used to call the second parent batch file ("run model steps")

Ref: X:\modelRuns\fy12\Ver2.3.36\run_Model_Ver2.3.36_2007_pseu.bat

Note that comments in batch files are indicated by lines that start with either REM or a double colon ("::"). A single colon (":") before a word indicates a label, which is often the target of a GOTO statement.

Lines 10 and 11,

```
set useIdp=f
set subnode=1-3
```

are used to run the model with Cube Cluster.

When you use Cube Cluster, you distribute a computing task across multiple processors or cores. Each processor or core is referred to as a "node". There is generally a main process and one or more sub-processes (or sub-nodes). Although Cube Cluster can be applied to either multiple processors within a computer (e.g., a server) or to multiple processors/computers connected via a local area network, at

COG, we have applied Cube Cluster in only the first manner. If you want to use Cube Cluster with four nodes, you are using one main node and three sub-nodes. The command "`set subnode=1-3`" tells Cube Voyage that you want to use three sub-nodes (i.e., a total of four nodes).

Line 20 of the wrapper batch file is used to start Cube Cluster, create N sub-nodes for processing, which will work with the main node already running:

```
Cluster.exe %root%\%scenar%\mwcog %subnode% start exit
```

Similarly, line 25 closes the N sub-nodes, once they are no longer needed:

```
Cluster.exe %root%\%scenar%\mwcog %subnode% close exit
```

The Version 2.3.36 model is set up to run with Cube Cluster turned on and to run using four cores (i.e., one main node/core and three sub-nodes/cores). If the user wants to run without Cube Cluster, he/she must make changes in the wrapper batch file and in the highway_assignment.s script, which are described below:

Running without Cube Cluster: Changes to the wrapper batch file

Comment out the two commands on lines 10 and 11:

```
::set useIdp=t
::set subnode=1-3
```

Running without Cube Cluster: Changes to the highway_assignment.s script

The highway_assignment.s script has been modified to run Cube Cluster. The following modifications were made. **First**, one needs a global statement to turn Cube Cluster on or off:

```
distribute intrastep=t multistep=f
```

In our highway assignment script, we have the command:

```
distribute intrastep=%useIdp% multistep=f
```

The "%useIdp%" is an environment variable that is set in the wrapper batch file to be either true ("t") or false ("f"). We have not used any multi-step distributed processing (MDP), so this flag is always set to false. It is worth noting that, on the subject of this global statement, Citilabs documentation states, "If turned off, distributed processing will not be invoked even if there are DistributeINTRASTEP and DistributeMULTISTEP statements in the following script." [14] However, we found that the model did not run correctly, so, for users who want to disable Cube Cluster, we recommend that you manually comment out all of the lines in highway_assignment.s that deal with Cube Cluster.

The highway assignment script makes three calls to the HIGHWAY module of Cube Voyager (RUN PGM=HIGHWAY). The first one is for the non-HOV3+ traffic assignment (both AM and PM peak). The

[14] Citilabs, Inc., "Cube Voyager Reference Guide, Version 5.1.2" (Citilabs, Inc., October 22, 2010), 995.

second is for the HOV3+ traffic assignment (both AM and PM peak). And the last is for the off-peak period traffic assignment (both midday and night/early morning). **Second**, in each of these three calls, one must have the following command:

```
distributeIntrastep processId='mwcog', ProcessList=%subnode%
```

Again, the "%subnode" refers to an environment variable set in the wrapper batch file. As noted earlier, this command and others related to Cube Cluster are commented out in the current highway assignment script, so one would need to add the comment character (";") if one wants to run the model without Cube Cluster.

The **third** and final change made to the highway assignment script was the changing of any occurrence of "IF (i=1)" to "IF (i=FirstZone)". One of the requirements in using IDP is that the order of processing groups of zones must be independent, so it does not matter which zone groups are processed first. In addition to this requirement, there are a number of commands/options that will cause IDP to turn off automatically due to data storage, calculation or input/output requirements that would overtake any benefits that IDP would provide. One example is the "IF (I=1)" statement. Other examples can be found in the online help.[15]

Continuation of the description of the wrapper batch file

Line 22 of the wrapper batch file is the line that actually calls the second parent batch files (which then runs the travel model):

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

The utility "`timethis`" is used to time the length of time for the entire model run. `Timethis` will time the command contained in the double quotes ("`%runbat%  %scenar%`"). The command "`2>&1`" tells the command interpreter to redirect both the standard output stream and the standard error stream to a file (in this case, the file "%root%\%scenar%\%scenar%_fulloutput.txt", which will expand to something like "`C:\modelRuns\fy12\Ver2.3.36_aqc_2011clrp`\2007_pseu\2007_pseu_fulloutput.txt". The vertical bar (|) is known as a pipe and it passes the output of one command to another. In this case, the output of the command is passed to the "`tee`" utility, which splits the standard output stream to two places (in this case the computer monitor and the file `%root%\%scenar%\%scenar%_fulloutput.txt`.

"Standard output" is the information that is normally written to the screen as a model run is in progress. "Standard error" includes any error messages that may be generated during the running of a model. For example, if a batch file tries to delete a file that does not exist, it will generate the error message "File not found," and this message is sent to the screen, since standard error is sent to the screen by default. Standard output includes any non-error messages, such as "1 file copied." Without using a command to re-direct the standard output and standard error to a set of files, a model user could still launch a travel model run, but the information about the progress of the travel model run would be sent to the

[15] Ibid., 984.

monitor/computer screen only. By using the aforementioned command, the information about the travel model run goes to both the monitor and a separate text file. When the model run is finished, the user should open and review the "*_fulloutput.txt" file (see Figure 6) and the searchForErrs.txt file (see Figure 5) in a text editor to make sure that there were no abnormalities during the run. Without these two files, one would have to watch the computer monitor during the entire model run. In Figure 5, we can see that there were no errors during the running of the model. The output.txt file is typically over 3,000 lines long, so only an excerpt of it is shown in Figure 6.

```
********** Searching for errors and anomalies after a travel model run **********
Program name: searchForErrs.bat

***** Searching *fulloutput.txt

  *** Searching for cases where a file could not be found


***** Searching report files (*.rpt)
  *** Searching for evidence that TP+ (TPMAIN) is running, instead of Voyager (PILOT)
  *** Searching for evidence of LINKO nodes that do not have XY values
2007_pseu\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
2007_pseu\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
2007_pseu\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
2007_pseu\i1_TRANSIT_SKIMS_AB.RPT:W(693): The following LINKO nodes do not have XY values:
2007_pseu\i1_TRANSIT_SKIMS_BM.RPT:W(693): The following LINKO nodes do not have XY values:
```

Figure 5 An excerpt from the "search for errors" file that is created during a model run

Ref: X:\modelRuns\fy12\Ver2.3.36\searchForErrs.txt

```
E:\modelRuns\fy12\Ver2.3.36>set _year_=2007

E:\modelRuns\fy12\Ver2.3.36>set _alt_=Ver2.3.36_2007_pseu

E:\modelRuns\fy12\Ver2.3.36>cd 2007_pseu

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>set _HOV3PATH_=

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>cd..

E:\modelRuns\fy12\Ver2.3.36>rem ====== Pump Prime Iteration ==========================================

E:\modelRuns\fy12\Ver2.3.36>set _iter_=pp

E:\modelRuns\fy12\Ver2.3.36>set _prev_=pp

E:\modelRuns\fy12\Ver2.3.36>call Set_CPI.bat                2007_pseu

E:\modelRuns\fy12\Ver2.3.36>cd 2007_pseu

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>REM  CPI Establishment

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>if exist voya*.* del voya*.*

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>if exist set_CPI.rpt del set_CPI.rpt

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>start /w Voyager.exe  ..\scripts\set_CPI.s /start -Pvoya -S..\2007_pseu

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>if errorlevel 1 goto error

E:\modelRuns\fy12\Ver2.3.36\2007_pseu>if exist voya*.prn copy voya*.prn set_CPI.rpt /y
voya0001.PRN
        1 file(s) copied.

(etc.)
```

Figure 6 An excerpt from an example of the standard output file that is created during a model run

Ref: X:\modelRuns\fy12\Ver2.3.36\2007_pseu\2007_pseu_fulloutput.txt

1.7.2 "Run model steps" batch file

The second of the two parent batch files is called the "run model steps" batch file -- formerly referred to as the "run all" batch file. This is the batch file that actually runs all the steps of the travel model.

Figure 7 shows a "run model steps" batch file for a base year, such as 2007. Figure 8 shows a "run model steps" batch file for a future year, such as 2040. Notice that the future-year "run model steps" makes use of highway skim replacement (HSR_Highway_Skims.bat) and the transit constraint through the regional core (Mode_Choice_TC_V23.bat), both of which are discussed later in this report.

These two files begin with two Windows environment variables, "_year_" and "_alt_", which the user should update before launching a model run. The command "`set _HOV3PATH_=`" is used only if one is going to perform a model run with HOV/HOT lanes that requires the "double run of the travel model to address Northern Virginia HOV/HOT lane policy," also known as the "HOV 3+ skim substitution option." (see Figure 8. This is discussed in Chapter 8). In the "run model steps" batch file shown in Figure 7, this is not being done, so the command is left blank.

```
:: E:\modelRuns\fy12\Ver2.3.36\run_ModelSteps_Ver2.3.36_2007_pseu.bat
:: 2011-10-28 Fri 15:16:54

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2007
set _alt_=Ver2.3.36_2007_pseu

:: Location of substitute HOV 3+ skims/ null location for this year
:: This will eventually be handled in the mode choice batch file
cd %1
set _HOV3PATH_=
cd..

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

set _iter_=pp
set _prev_=pp

call Set_CPI.bat              %1
call PP_Highway_Build.bat     %1
call PP_Highway_Skims.bat     %1
call Transit_Skim_All_Modes.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.bat   %1
call Highway_Skims.bat        %1

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

set _iter_=i1
```

```
set _prev_=pp

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Highway_Skims.bat          %1

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

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i3
set _prev_=i2

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i4
set _prev_=i3

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
```

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

:: rem ====== Transit assignment ============================================

call Transit_Assignment.bat %1
call TranSum.bat %1

:: rem ====== End of batch file =============================================

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

Figure 7 "Run model steps" batch file for a base-year run (run_ModelSteps_Ver2.3.36_2007_pseu.bat), used to call the child batch files (spans multiple pages)

Ref: X:\modelRuns\fy12\Ver2.3.36\run_ModelSteps_Ver2.3.36_2007_pseu.bat

```
:: O:\model_dev\Ver2.3.27\run_ModelSteps_Ver2.3.27_2040_final.bat
:: 2011-06-30 Thu 14:15:45

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2040
set _alt_=Ver2.3.33_2040_final

:: Location of substitute HOV 3+ skims/null location for this year
:: This is referenced in HSR_Highway_Skims.bat

cd %1
set _HOV3PATH_=..\2040_base
cd..

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

set _iter_=pp
set _prev_=pp

call Set_CPI.bat                %1
call PP_Highway_Build.bat       %1
call PP_Highway_Skims.bat       %1
call Transit_Skim_All_Modes.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.bat     %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat      %1

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

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice_TC_V23.bat     %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
rem call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat      %1

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

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice_TC_V23.bat     %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
rem call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat      %1

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

set _iter_=i3
set _prev_=i2

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice_TC_V23.bat     %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
```

```
call Average_Link_Speeds.bat    %1
rem call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat      %1

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

set _iter_=i4
set _prev_=i3

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice_TC_V23.bat     %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
rem call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat      %1

:: rem ====== Transit assignment ============================================

call Transit_Assignment.bat %1
call TranSum.bat %1

:: rem ====== End of batch file =============================================

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

Figure 8 "Run model steps" batch file for a 2040_final run (run_ModelSteps_Ver2.3.33_2040_final.bat), used to call the child batch files (spans multiple pages)

Ref: N:\model_app\CGV2_3_Conformity2011CLRP_RR_SA\run_ModelSteps_Ver2.3.33_2040_final.bat

1.7.3 Summary Scripts

In addition to the model, the user is provided with a number of summary scripts, which may be helpful in analyzing the model output. These are listed in Table 9.

Table 9 Travel Model Summary Scripts

Summary script	Description	Folder
Compare_mode_choice.s	Compares **estimated** mode choice results to HTS **observed** mode choice data	summary
Compare_mode_choice_adj.s	Compares **estimated** mode choice results to adjusted HTS and transit surveys **observed** mode choice data	summary
COMPARE_NL_MC.S	Compares **estimated** mode choice results between two different model runs.	summary
Compare_Trip_Distribution.s	Compares estimated trip distribution to observed trip distribution from HTS	summary
View_From_Space_V2.3_3722TAZ.s	Creates global summary of demographic info, trips, and VMT	summary
Summarize_Est_Obs_Volume_Daily.s	Compares estimated daily volumes on select links to observed counts	assignment summary
Summarize_Est_Obs_Volume_Period.s	Compares estimated AM, MD, PM, and NT volumes on select links to observed counts	assignment summary
Summarize_2007_2040_Screenlines.s	Compares estimated screenlines volumes in 2007 and 2040	assignment summary
Plot_Volume_Differences.s	Plots volume differences between two input networks	assignment summary

1.8 Known issues related to running the model

1.8.1 Cube Cluster differences

When using Cube Cluster, the estimated VMT coming from the model can change slightly, depending on how many cores are used. At the regional level, these differences in estimated VMT are on the order of 1/100 of a percent. However, for some individual links, the difference can be greater than 20%.[16] For this reason, we recommend that all model runs for a project be done in the same manner, e.g., use Cube Cluster for all runs and use the same number of cores for each run.

1.8.2 Model run stops before finishing

We have experienced some cases where a model run will stop ("crash") for no apparent reason. Sometimes the exact same model run will complete successfully if run on a different computer. While

[16] Ronald Milone and Mark Moran, "TPB Version 2.3 travel model on the 3,722-TAZ area system: Status report" (presented at the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., May 20, 2011), 25–32.

we are still trying to determine the cause of these crashes, we do, however, have a pragmatic way for dealing with these events. Determine where the model run crashed. Re-launch the model run, but comment out all the steps in the "run model steps" that have completed successfully, so that the model runs only the step that crashed and the steps that follow it. This procedure will typically result in a normal model run, even though it requires the analyst to intervene midstream.

1.8.3 Difficulty reaching convergence in the highway assignment step for out-year model runs with tolls

We have noticed some cases where the traffic assignment algorithm (currently bi-conjugate Frank-Wolfe, though we have also tested regular Frank-Wolfe) will not be able to come to a sufficiently converged solution in cases where one sets the relative gap tolerance to a smaller than normal value (e.g., 10^{-4}) for a future-year scenario that has variably-priced facilities. We are still investigating this situation and have contacted Citilabs, which thinks the difficulty reaching convergence may be due to large toll values that dominate the link-cost function. This is generally not a problem for most model runs which use the standard relative gap tolerance of 10^{-3}.

Chapter 2 Set-Up Programs and Highway Network Building

2.1 Overview

The initial Cube Voyager steps of the Version 2.3 Travel Model include steps to establish model parameters, SET_CPI.S and Set_factors.S (executed within the SET_CPI.bat file), and the highway network building steps, AreaType_File.S, V2.3_HIGHWAY_BUILD.S (executed within the PP_Highway_Build.bat file). The modeling steps included in these two batch files are shown on pages A-1 and A-2 of the flowchart in Appendix A. Set_CPI.s is used to establish deflation factors that are used in subsequent toll-related and transit fare-related steps. Set_Factors.s is used to establish K-Factors used in trip distribution. The Area_Type.s step establishes zonal area type codes based on land activity densities. The resulting area type file is used in the highway building step, V2.3_Highway_Build.s. These steps are not implemented within the "looping" process of the travel model; they are executed only once, in the "pump prime" execution of the travel model. The principal inputs to above modeling steps are listed in Table 10 and detailed in Table 11 through Table 14. The principal outputs are listed in

Table 15, and detailed in Table 16 and Table 17.

Table 10 Inputs to the set-up and highway network building process

CPI schedule and parameter file	\Inputs\CPI_File.txt	Text
Zonal land use file	\Inputs\ZONE.DBF	DBF
Node coordinate file	\Inputs\NODE.DBF	DBF
Zonal area type override file	\Inputs\AT_override.txt	Text
Link file	\Inputs\LINK.DBF	DBF
Initial AM and midday hwy. speed lookup files	\Support\AMSPD.LKP, \Support\MDSPD.LKP	Text
Toll parameter file	\Inputs\Toll_Esc.dbf	DBF

Table 11 Land Use File Format Description (zone.dbf)

File Name	Variable Name	Description
Zone.dbf	TAZ	TAZ (1-3,722)
	HH	Households
	HHPOP	Household Population
	GQPOP	Group Quarters Population
	TOTPOP	Total Population
	TOTEMP	Total Employment
	INDEMP	Industrial Employment
	RETEMP	Retail Employment
	OFFEMP	Office Employment
	OTHEMP	Other Employment
	JURCODE	Jurisdictiojn Code (0-23)
	LANDAREA	Gross Land Area (in sq. miles)
	HHINCIDX	Ratio of zonal HH median income to regional median HH income in tenths (i.e. 10=0.1) per 2007 ACS
	ADISTTOX	Airline distance to the nearest external station in whole miles.
	TAZXCRD	TAZ X-Coordintes (NAD83-based in whole feet)
	TAZYCRD	TAZ Y-Coordintes (NAD83-based in whole feet)

Table 12 Node Coordinate File Format Description (node.dbf)

File Name	Variable Name	Description
Node.dbf	N	Highway Node Number
	X	X - Coordintes (NAD83-based in whole feet)
	Y	Y- Coordinates (NAD83-based in whole feet)

Table 13 Base Highway Link File Format Description (link.dbf)

File Name	Variable Name	Description
Link.dbf	A	A-Node
	B	B_Node
	DISTANCE	Link distance (in 1/100ths of miles)
	JUR	Jurisdiction Code (0-23) *0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef*
	SCREEN	Screenline Code
	FTYPE	Link Facility Type Code (0-6)
		0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp
	TOLL	Toll Value in current year dollars
	TOLLGRP	Toll Group Code
	AMLANE	AM Peak No. of Lanes
	AMLIMIT	AM Peak Limit Code (0-9)
	PMLANE	PM Peak No. of Lanes
	PMLIMIT	PM Peak Limit Code (0-9)
	OPLANE	Off-Peak No. of Lanes
	OPLIMIT	Off-Peak Limit Code (0-9)
	EDGEID	Geometry network link identifier
	LINKID	Logical network link identifier
	NETWORKYEARLink	Planning year of network link
	SHAPE_LENGTLink	Geometry length of network link (in feet)
	PROJECTID	Project identifier
	CODE	Unused (place marker to flag network edits)

Notes:

- *The mode choice model requires that all costs be in 2007 dollars.*
- *Limit Codes are 0,1 = General Use, 2 = HOV2,3+ only, 3 = HOV 3+ Only, 4 = Truck Prohibited, 5 = Non-Airport Vehicles Prohibited, 6-8 = (unused), 9 = 'Transit Only' link (links used to more accurately*

depict coded transit routes, but are below the grain of the zone system; these links are not included in the highway assignment process).

Table 14 Toll Parameter File (Toll_esc.dbf)

File Name	Variable Name	Description
Toll_Esc.dbf	Tollgrp	Toll group code, 1=existing tolled facility, 2=ICC, 3+= VA Hot segments
	Escfac	Deflation factor override
	Dstfac	Distance (per mile) based toll factor in present year cents/dollar (optional)
	AM_Tftr	AM period Toll factor
	PM_Tftr	PM period toll factor
	OP_Tftr	Off-peak period toll factor
	AT_MIn	Area Type minimum override (optional)
	AT_Max	Area Type maximum override (optional)
	TollType	*Toll Type (1=operating in calibration year, 2= operating after calibration year)*

Table 15 Outputs of the set-up and highway network building process

Highway, transit deflator files	Trn_Deflator.txt, Hwy_Deflator.txt	Text
Summary text file of Fare CPI assumptions used	MFARE2_CPI.txt	Text
Zone centroid co-ordinates	TAZ_XYs.dbf	DBF
1-mile floating land use	Floating_LU.dbf	DBF
Area type file	AreaType_File.dbf	DBF
Unloaded/built highway network file	ZONEHWY.NET	Binary
Summary text file of Fare CPI assumptions used	MFARE2_CPI.txt	Text
Zonal K-factors	HBWK.DAT, HBSK.DAT, HBOK.DAT, NHWK.DAT, NHOK.DAT	Binary

Table 16 Zonal Area Type File (AreaType_File.dbf)

File Name	Variable Name	Description
AreaType_File.dbf	TAZ	TAZ Number (1-3,722)
	POP_10	One-mile "floating" Population density
	EMP_10	One-mile "floating" Employment density
	AREA_10	One-mile "floating" Area
	POPDEN	One-mile "floating" Population density
	EMPDEN	One-mile "floating" Employment density
	POPCODE	Population density code (1-7)
	EMPCODE	Employment density code (1-7)
	ATYPE	Area Type (1-6)

Table 17 Unloaded binary highway network file (Zonehwy.net)

File Name	VariableName	Description
zonehwy.net	A	A Node
	B	B Node
	DISTANCE	Link Distance in miles (x.xx)
	SPDC	(Not used)
	CAPC	(Not used)
	JUR	Jurisdiction Code (0-23)
		0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef
	SCREEN	Screenline Code (1-38)
	FTYPE	Link Facility Type Code (0-6)
		0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp
	TOLL	Toll value in current year dollars
	TOLLGRP	Toll Group Code (1-9999)
	<Period>LANE	<Period> No. of Lanes
	<Period>LIMIT	<Period> Limit Code (0-9)
	EDGEID	Geometry network link identifier
	LINKID	Logical network link identifier
	NETWORKYEA	Planning year of network link
	SHAPE_LENG	Geometry length of network link (in feet)
	PROJECTID	Project identifier
	TAZ	TAZ (1-3,722)
	ATYPE	Area Type (1-6)
	SPDCLASS	Speed Class
	CAPCLASS	Capacity Class
	DEFLATIONFTR	Deflation factor for converting existing year costs to 2007 costs
	<Period>TOLL	<Period>Toll value in current year cents (if applicable)
	<Period>TOLL_VP	<Period>Toll of future, variably priced facility only
	<Period> HTIME	<Period> Highway Time (min)
Key		
<Period>=	AM	AM Peak Period (6:00-9:00 AM)
	MD	Mid Day (9:00 AM - 3:00 PM)
	PM	PM Peak Period (3:00 - 7:00 PM)
	NT	All remaining hours

2.2 Application Details

The SET_CPI.S script is used to produce deflation factor files (Trn_Deflator.txt and Hwy_Deflator.txt) which are, in turn, used in subsequently scripts to compute highway tolls and transit fares in constant year-2007 prices. This procedure has been established to ensure that cost deflation for highway tolls and transit fares is treated consistently and appropriately.

The SET_CPI.S script reads a preexisting text table look-up file (\INPUTS\CPI_File.txt) containing historical annualized CPI figures published by the U.S. Bureau of Labor Statistics beginning with the model calibration year, 2007. The CPI figures are based on the U.S. city average of all urban consumers (100 = 1982-84). An example listing of the file appears below.

```
;; ----------------------- Inflation Parameters  - per V2.3 Model
;; - RJ Milone - 3/18/2011- CPI data from BLS / All Urban Consumers (CPI-U) US City Avg. / 1982-84=100
;;   ftp://ftp.bls.gov/pub/special.requests/cpi/cpiai.txt

InflationFTR        = 1.0            ; Inflation Assumption (DEFAULT IS 1.0)
Defl_OverRide       = 0.0            ; Deflation Override   (DEFAULT IS 0.0)  If Non-zero it is used as deflator
                                     ; Used as deflator IF NON-ZERO
BaseCPIYear         = 2007           ; Base year of the CPI Table
CurrCPIYear         = 2010           ; Current year on  CPI table below (Most current year for which CPI data is available)
;
;===============================================================
; Establish historic CPI table and Deflation Factor          =
;===============================================================
;
      LOOKUP Name=CPI_Table,
             LOOKUP[1]   = 1,Result = 2,                    ;  CPI index (from US BLS)
             LOOKUP[2]   = 1,Result = 3,                    ;  Compounded Growth Rate From Base Year
             LOOKUP[3]   = 1,Result = 4,                    ;  Deflation Factor
             Interpolate = N, FAIL=0,0,0,list=Y,
          ;;
          ;;          (((YrCPI/BsCPI)^(1/n))-1.0)*100  BsCPI/YrCPI)
          ;;                 Annual_Avg.               Historic Deflation
          ;; YEAR    CPI     Growth_Rate               Factor
          ;; ----   -----   -----------------       -----------
        R=' 2007,   207.3,      0.000,              1.00000', ; <--- BaseCPIYear
          ' 2008,   215.3,      3.859,              0.96284', ;
          ' 2009,   214.5,      1.722,              0.96643', ;
          ' 2010,   218.1,      1.707,              0.95048'  ; <--- Curr(ent)CPI Year

; --- end of CPI File -----------
```

The script computes a cost deflation factor using the CPI table and the parameters *BaseCPIYear*, *CurrCPIYear, InflationFTR* (all specified in the above text file), and the *_Year_* environment variable specified in the "RUN_ModelSteps" batch file. These parameters are defined as:

BaseCPIYear = the base (or calibration) year of the travel model

CurrCPIYear = the most recent year for which historical CPI data exists (as reflected in the CPI table)

Year = the year of the modeled scenario (as defined in the Run_ModelSteps.bat file)

InflationFTR = Factor reflecting special CPI growth assumptions beyond CurrCPIYear that might be considered in scenario testing. For example, a value of 1.0 indicates future cost escalation is assumed to remain constant with the historical rate of inflation; a value of 2.0 would indicate that the future cost escalation is assumed to be twice the historical rate of inflation; a value of 0.5 would indicate that the future cost escalation is assumed to be one half of the historical rate of inflation, etc. The default value is 1.0.

Under default conditions, if the modeled year (*_Year_*) is less than or equal to *CurrCPIYear*, the CPI factor will equal CPI_{2007}/ $CPI_{\_Year\_}$ from values provided in the CPI table. If the modeled year (*_Year_*) is greater than *CurrCPIYear*, the CPI factor will equal (CPI_{2007}/ $CPI_{CurrCPIYear}$) table. The user may optionally invoke the *InflationFTR* parameter to arrive at a deflation factor that reflects something other than the "historical inflation rate" assumption. In addition to the output deflation factor files mentioned above, the script also writes a text file (Mfare2_CPI.txt) that lists the input and output parameter values used.

The Set_Factors.s script is used to generate a family of F-factors by modeled purpose, to be used subsequently by the trip distribution process. The F-factors are jurisdiction based and have been formulated during the calibration and validation phase of the model development process. Separate F-factor files are produced by purpose as Cube/Voyager binary matrix files (zone-to-zone).

It is useful to understand the basic elements of the highway and transit network that are reflected in the highway link input to Version 2.3 model (link.dbf). The highway elements are shown in Table 18.

Table 18 Version 2.3 Highway Network Elements

Highway Network Element	Description	Numbering	Notes
Centroids	Activity "center of gravity" in TAZs	1 -3722	3676-3722 allocated as extl. stations
			3723-5000 reserved for TAZ expansion
			Jurisdictional ranges are established
			Some TAZs are unused
PNR "Centroids"	PNR Lots locations used to develop restrained	5001-7999	5001-5999 Metrorail
	highway times between TAZs & PNR lots		6001-6999 Commuter rail
			7001-7999 LRT/BRT/New
Highway nodes	Highway intersections, junctions, or	20000 - 60000	Ranges established by Jurisdiction
	points of zonal/PNR access to highway system		
Centroid connectors	TAZ-to-highway access linkages	-	Ftype= 0
PNR connectors	PNR-to-highway access linkages	-	Ftype= 4
Highway links	major highway segments	-	Ftype=0/ TAZ centroid connector, 1/freewy, 2/maj. art., 3/min. art., 4/collector or PNR access link, 6/ramp, 9/bus-only)

The network includes two types of centroids: a zonal centroid which represents the geographic center of land activity within a TAZ and a PNR centroid (also known as a "dummy PNR centroid") which represents PNR lot locations at Metrorail and commuter rail stations. The PNR centroid represents a KNR drop-off point if no PNR lot exists at a given station. The two centroid types are assigned specific numbering ranges. TAZ centroids are numbered 1-3722 and PNR centroids are numbered 5001-7999. The numbering gap between the TAZ and PNR ranges, 3723-5000, are reserved for future TAZ assignments[17]. The two centroid types are employed so that highway Level-of-Service (LOS) matrices may be built not only between TAZs, but also between TAZs and PNR lots.

Highway nodes representing intersections or highway access points from TAZs or PNR lots are assigned a numbering range of 20000 to 60000.

[17] The existing Version 2.3 scripts, inputs, and support files would need to be modified if additional TAZs were added to the highway network.

Network links (i.e., centroid connectors and highway links) are assigned facility type ("Ftype") attributes ranging from 0 to 6.

The highway network building process (i.e., the process for creating a binary highway network file which is used in subsequent modeling steps) is undertaken with two scripts that are executed in sequence: AreaType_File.S and V2.3_Highway_Build.S. The AreaType_File.s script reads a preexisting zonal land activity file (Zone.dbf) and a highway node coordinate file (Node.dbf) and computes the area type code associated with each TAZ. Area type codes range from 1 to 6 and are based on population and employment density, as detailed in Table 19 below:

Table 19 Area type codes-Based on population and employment density

One-Mile "Floating" Population Density (Pop/Sq mi)	One- mile "Floating" Employment Density (Emp/Sq mi)						
	0-100	101-350	351-1,500	1,501-3,550	3,551-13,750	13,751-15,000	15,001+
0-750	6	6	5	3	3	3	2
751-1,500	6	5	5	3	3	3	2
1,501-3,500	6	5	5	3	3	2	2
3,501-6,000	6	4	4	3	2	2	1
6,001-10,000	4	4	4	2	2	2	1
10,000-15,000	4	4	4	2	2	2	1
15,001+	2	2	2	2	2	1	1

The AreaType_File.s script produces three files: TAZ_Xys.dbf (zonal coordinates), Floating_LU.dbf (a zonal file containing the area, population, and employment within one mile), and Areatype_file.dbf (a zonal file containing the associated area type, in accordance with the land activity file) which is used in the next script.

The V2.3_Highway_Build.S script reads the zonal area type file, along with a node file, a link attribute file, a zone file, and four parameter files. The parameter files include initial speed and capacity files (AMSpd.lkp, MDSpd.lkp) both arrayed by facility type and area type. The deflation file created by the SET_CPI.s script (Hwy_Deflator.txt) is read. Finally, a toll parameter file (Toll_esc.dbf) is also used by the script. The file contains a number of toll-related parameters that are indexed by a tolled facility code (tollgrp) which is included as a link attribute.

The highway building process consists of the following steps:

1) Each highway link is evaluated against all TAZ centroid to determine its nearest zone (i.e., the TAZ centroid nearest to the airline mid-point of the link a-node and bnode). The nearest zone is then saved to a temporary link file containing the A-node, B-node, and nearest TAZ.
2) The link file, zonal area type file, and link-TAZ (from step 1) are merged to enable the zonal area type of the nearest TAZ to be assigned to each link. The link file contains basic link attributes, including distance, facility code, period-specific (AM, PM, OP) lanes and limit codes, coded tolls, toll group codes, jurisdiction, and screenline codes.
3) Toll parameters are merged to each link on the basis of the tollgrp code
4) Speed and capacity classes are next defined as a two digit integer, where the first digit is the facility type and the second digit is the area type
5) Period-specific tolls (AM, PM, and OP) are computed. The general form of the toll computation is:

   <prd>Toll = (Toll + ($DstFac_t$ * Distance * <prd>$\_TFtr_t$)) * ($EscFac_t$ if > 0.0; Otherwise: Hdefl)

   <u>Where:</u>

<prd>Toll	= period-specific toll coded on link in constant year dollars (e.g., Amtoll)
Toll	= link-coded "Toll" link attribute value
$DstFac_t$	= distance factor (cents/mi) for toll group "t", as specified in Toll_Esc.dbf
Distance	= link-coded distance (x.xx mi)
<prd>$\_TFtr_t$	= period-specific factor for toll group "t" as specified in Toll_Esc.dbf
Hdefl	= Default highway deflation factor based on CPI assumptions (Set_CPI.s)
Esc_Fac_t	= Hwy. deflation factor *override* for toll group "t", as specified in Toll_Esc.dbf

6) A period specific toll type code (<prd>Toll_VP) is established to distinguish whether the tolled link existed during the model calibration year or the tolled link is a future, variably priced facility. This information is relevant to subsequent toll skimming.
7) Initial AM and OP speeds are assigned based on facility and area type codes
8) Midday (MD) and Night (NT) attributes are set to off-peak (OP)-related attributes defined above

The resulting binary network file resulting from the highway network building process is named Zonehwy.net. Variables that are included in the zonehwy.net file are described in Table 20.

Table 20 zonehwy.net file Variables description

VariableName	Description
A	A-Node
B	B-Node
DISTANCE	Link Distance in miles (x.xx)
SPDC	Not used
CAPC	Not used
JUR	Jurisdiction Code (0-23) *0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef*
SCREEN	Screenline Code (1-38)
FTYPE	Link Facility Type Code (0-6) *0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp*
TOLL	Toll Value in current year dollars
TOLLGRP	Toll Group Code (1-9999)
<Period> LANE	<Period> No. of Lanes
<Period>LIMIT	<Period> Limit Code (0-9)
EDGEID	Geometry network link identifier
LINKID	Logical network link identifier
NETWORKYEA	Planning year of network link
SHAPE_LENG	Geometry length of network link (in feet)
PROJECTID	Project identifier
TAZ	TAZ (1-3,722)
ATYPE	Area Type (1-6)
SPDCLASS	Speed Class
CAPCLASS	Capacity Class
DEFLATIONFTR	Factor for deflating current year tolls to constant year tolls
<Period>TOLL	<Period> Toll Value in current year dollars
<Period>TOLL_VP	<Period> Toll Value in current year dollars - Variably priced tolled facilities only
<Period> HTIME	<Period> Highway Time - based on initial highway lookup speeds
KEY	
<Period>= AM	AM Peak Period (6:00-9:00 AM)
MD	Midday (9:00 AM - 3:00 PM)
PM	PM Peak Period (3:00 - 7:00 PM)
NT	All remaining hours

Chapter 3 Highway Skim File Development

3.1 Overview

Highway skimming is the process of developing zone-to-zone travel times, distances and tolls over a congested travel network, usually a loaded network with congested travel speeds generated from a traffic assignment process. Highway skims in the Version 2.3 model are generated after each traffic assignment step. Highway skims are generated by time period (AM and Midday), and by highway mode (SOV, HOV 2-occupant, HOV 3+occupant). In addition, truck skims are generated for the midday period only. Mode-specific paths are very important in the Washington, D.C. region due to special operating restrictions, particularly during the AM peak period.

The TPB's highway skimming is done two times, once to develop zone-to-zone (3722 x 3722) skim matrices and then again to develop zone/PNR lot-to-zone/PNR lot (7999 x 7999) skim matrices. The latter set enables restrained highway speeds and distances and speeds to be calculated between zones and PNR lots thus allowing transit auto access links to be built. The entire highway skimming process is applied with the scripts named HIGHWAY_SKIMS.S, JOINSKIMS.S, MODNET.S, Highway_Skims_mod.S, and RemovePPSpeed.S. These are invoked with the PP_Highway_Skims.bat file in the initial or pump-prime iteration (see page A-3 of Appendix A) and the Highway_Skims.bat file (see page A-10) in the standard iterations. The RemovePPSpeed.s script is executed in the pump-prime iteration only. The principal inputs and outputs are shown in Table 21 and Table 22 , respectively.

Table 21 Inputs to the highway skim file development

Built highway network file	<ITER>HWY.NET	Binary
Toll minutes equivalent	support\toll_minutes.txt	Text
AM toll factors by vehicle type	Inputs\AM_Tfac.dbf	DBF
MD toll factors by vehicle type	Inputs\MD_Tfac.dbf	DBF

Note: <ITER> =PP, i1…i4 <Prd>= AM and MD

Table 22 Outputs of the highway skim file development

Total highway skims	SKIMTOT<ITER>.DAT	Binary
Truck skims	TRK<ITER>MD.SKM	Binary
SOV skims	SOV<ITER><Prd>.SKM	Binary
HOV2 skims	HOV2<ITER><Prd>.SKM	Binary
HOV3+ skims	HOV3<ITER><Prd>.SKM	Binary
SOV skims (used by mode choice model)	SOV<ITER><Prd>_MC.SKM	Binary
HOV2 skims (used by mode choice model)	HOV2<ITER><Prd>_MC.SKM	Binary
HOV3+ skims (used by mode choice model)	HOV3<ITER><Prd>_MC.SKM	Binary
AM highway skims	HWY<ITER>AM.SKM	Binary
Off peak highway skims	HWY<ITER>OP.SKM	Binary
Network with added station centroid connectors	<ITER>HWYMOD.NET	Binary
Walk access links	WalkAcc_Links.dbf	DBF
	SOVM<ITER><Prd>.SKM	Binary
	HOV2M<ITER><Prd>.SKM	Binary
	HOV3M<ITER><Prd>.SKM	Binary
Highway network with PP speeds removed	ZoneHWY.NET	Binary

Note: <ITER> =PP, i1…i4 <Prd>= AM and MD

3.2 Application Details

Highway skimming process is used to develop time, cost, and toll values between i/j pairs on a minimum impedance path. The skimming process reads a highway network input file with preexisting restrained speeds. The restrained speeds used in the pump prime (PP) iteration initially are table look-up values based on time period (AM, Off-peak), facility type, and area type. After the PP iteration is completed (i.e., after the PP traffic assignment process is completed), the highway skimming is accomplished using traffic assignment-based link speeds. The generalized impedance for which minimum time paths are developed is defined as:

Impedance_V = Restrained over-the-network time_V + Toll-related Time_V

or

Impedance_V = Restrained over-the-network time_V + (Toll Cost_v * Time Rate_v * Vehicle Factor_{VF})

<u>Where:</u>

Impedance_V	= Impedance for vehicle class "V" for developing i/j paths in skimming
Restrained Time_V	= Congested/restrained network travel time (min) for vehicle class "V"
Toll Cost_V	= Tolls (2007 dollars) paid by vehicle class "V", if a tolled facility was used

Time rate_{V} = Time valuation (min/2007 dollar) of toll costs for vehicle class "V"
Vehicle Factor_{VF} = Vehicle class factor for tolled facility "F"

Note: Vehicle classes are:
SOVs, HOV2-occs, HOV3+occs, Commercial Vehs., Trucks, and airport passenger vehicles

The assumed time rates are provided by vehicle class and time period in the input file *toll_minutes.txt* (see below). The values shown are derived from average household income levels and information from the 2007/08 HTS. The values should not be altered.

```
;
;     ================================================================
;     = Equivalent Toll Minutes by Time Prd & Vehicle Type           =
;     = in minutes per 2007 dollar - rm 1/7/11                       =
;     ================================================================
;
;    AM Peak          Midday           PM Peak          Night
;---------------  ---------------    --------------     --------------
  SVAMEQM = 2.5     SVMDEQM = 3.0    SVPMEQM = 3.0     SVNTEQM = 3.0 ; <--- SOVs
  H2AMEQM = 1.5     H2MDEQM = 4.0    H2PMEQM = 2.0     H2NTEQM = 4.0 ; <--- HOVs-2 Occ
  H3AMEQM = 1.0     H3MDEQM = 4.0    H3PMEQM = 1.0     H3NTEQM = 4.0 ; <--- HOVs-3+Occ
  CVAMEQM = 2.0     CVMDEQM = 2.0    CVPMEQM = 2.0     CVNTEQM = 2.0 ; <--- Comm Veh
  TKAMEQM = 2.0     TKMDEQM = 2.0    TKPMEQM = 2.0     TKNTEQM = 2.0 ; <--- Trucks
  APAMEQM = 2.0     APMDEQM = 2.0    APPMEQM = 2.0     APNTEQM = 2.0 ; <--- Apaxs
```

The vehicle factors are provided by time period in the inputs files *AM_Tfac.dbf* and *MD_Tfac.dbf*. An example of the *AM_Tfac.dbf* file is shown below. The file is available to allow for the ability to reflect a facility-specific toll policy differential between vehicle classes. The table below specifies the default assumption that tolls do not vary between vehicle classes, except for trucks which are assumed to pay 2.5 times the toll that an auto would pay.

TOLLGRP	AMSOVTFTR	AMHV2TFTR	AMHV3TFTR	AMCOMTFTR	AMTRKTFTR	AMAPXTFTR
1	1.0000	1.0000	1.0000	1.0000	2.5000	1.0000

The standard zone-to-zone highway skims are developed by the script *Highway_Skim.s*. The script produces skim files pertaining time periods (AM and midday) and mode/path type (SOV, HOV2, HOV3+, and truck. The truck skim file contains one table pertaining to travel time. The SOV and HOV skim files contain three tables: time (min), distance in implied tenths of miles, and tolls (2007 cents).

Based on a past analysis of Version 2.2 model forecasts, TPB staff found substantial costs associated with planned variably priced highway facilities (the Northern Virginia HOT lanes and the ICC) had caused counterintuitive mode choice model results. Essentially, the added person trips induced by the HOT lane's accessibility benefit tended to be allocated among non-SOV modes because of the substantial costs for paying SOV's to use the HOT lane costs. The result was not considered reasonable since the objective of the facility was to attract paying SOVs by selling a travel time benefit. Staff speculated the result may be attributed to the specification of the mode choice model: the "SOV-pay" alternative was not included in the choice set when the model was calibrated (indeed, no such facility has ever operated in the region). It was decided that that costs associated with future variably priced highway facilities should not be considered by the mode choice model in application. This approach has been adopted for the Version 2.3 application. Consequently, two sets of SOV and HOV skim files are created, one in which all toll facility costs are skimmed (e.g., SOV<iter>AM.SKM), and another set in which the toll skims reflect base-year toll facilities only (SOV<iter>AM_MC.skm). The former is used as an input to the trip distribution model and the latter is used as an input to mode choice.

The *Joinskims.s* script is use to merge the six skim files used by the mode choice model into two files, HWY<iter>AM.skm and HWY<iter>MD.skm, which are read directly into the mode choice model.

Modnet.s reads the built highway network file and creates another modified binary network that includes an expanded set of zone centroids, zone centroids (numbered 1 to 3722) and PNR lot centroids (numbered 5001 to 7999). The expanded network is named, <iter>HwyMod.net. Modnet.s also generates a list of highway links that are considered as "walk network links" in the development of walk (mode 13) links for the transit network.

The Highway_Skims_Mod.s script reads the expanded network and creates an expanded set of highway skims dimensioned 7999 by 7999, which includes highway skims between zone centroids as well as between zone and PNR lot pairs. The latter will be used subsequently to create auto access link attributes.

The RemovePPSpeed.s script is used to remove the "PP" iteration speed attributes from the highway network. This is necessary in the initial (PP) iteration, when table lookup speeds are to be replaced by traffic assignment speeds in the PP iteration.

Chapter 4 Auto Driver Trip Development

4.1 Overview

The "auto drivers" step is used to apportion auto person trip tables (3722 x 3722) generated by the mode choice model by occupant groups (1, 2, and 3+) into auto driver trips by occupant groups. This is done using basic matrix manipulations. Because the mode choice file output is inclusive of internal movements only, total person trips produced by the trip distribution step are also used as a basis for developing external auto driver trips by occupant groups. External trips produced by the trip distribution process consist of auto person travel only. External auto driver trips are developed using assumed auto occupancies and "off-the-shelf" occupant disaggregation curves.

The scripts used are PP_AUTO_DRIVERS.S (invoked by the PP_AutoDrivers.bat file) and MC_AUTO_DRIVERS.S (invoked by the Auto_Drivers.bat file).

The inputs to this step are shown in Table 23. The outputs are shown in Table 24. The outputs consist of five purpose-specific auto driver files, each containing three tables (one for each occupant group). The output file files contain both internal and external auto driver movements.

Table 23 Inputs to auto driver trip development

Pre-existing final iteration AEMS mode choice model output modal trip tables	???_NL_MC.MTT	Binary
Pump Prime iteration person trip tables	???_PP_.PTT	Binary
Current iteration AEMS mode choice model output modal trip tables	???_NL_MC.MTT	Binary

Note: ??? = HBW, HBS, HBO, NHW, and NHO <ITER> =PP, i1…i4

Table 24 Outputs of auto driver trip development

Auto drivers by trip purpose	???<ITER>.ADR	Binary

Note: ??? = HBW, HBS, HBO, NHW, and NHO <ITER> =PP, i1…i4,

Chapter 5 Pre-Transit Network Processing

5.1 Overview

Prior to transit network building, a series of Cube Voyager scripts are executed to generate special access links that are subsequently folded into the transit network, along with highway links, transit links, and transit lines. The scripts include WALKACC.S (used to develop zonal walk access links), PARKER.S (used to generate PNR lot-to-rail station links), and AUTOACC4.S (used to generate TAZ-to-station links or auto access links). The automated approach for generating these links has greatly streamlined the transit network coding process. All three of these programs were been originally developed as stand-alone Fortran programs developed by AECOM Consult, but were more recently converted to Cube Voyager scripts to facilitate the implementation of future enhancements.

The inputs used by the above programs are list in Table 25 (specific file descriptions are shown in Table 26 through Table 29. The output files are shown in Table 30.

Table 25 Inputs to pre-transit network processing

Zonal land use file	Zone.dbf	DBF
Station file	Station.dbf	DBF
Highway network link file	link.dbf	DBF
Highway node file	node.dbf	DBF
HBW zonal parking costs/terminal time file	HBWV2a1.dbf	DBF
Supplemental walk link file	xtrawalk.dbf	DBF
Sidewalk network links	WalkAcc_Links.dbf	DBF
TAZ area that is within walking distance from transit stops	Areawalk.txt	Text
Station mode-station type-max access dist. Lookup	StaAcc.dbf	DBF
Jurisdiction code- jurisdiction group lookup	Jur.dbf	DBF
List of zones connected to the Pentagon Metrorail station for the purpose of creating long-distance kiss-and-ride (KNR) links, which represent "slugging" or casual carpooling	Pen.dbf	DBF
TAZ XY co-ordinates	TAZ_xys.dbf	DBF
SOV AM/Off-peak highway time skims file	SOVMAM.SKM, SOVMMD.SKM	Binary

Input File Descriptions and Formats

Table 26 Transit Station File format Description (Station.dbf)

File Name	Variable Name	Description
Station.dbf	SEQNO	Seqence number
	MM	Mode code ('M','C,','B','L')
		(M=Metrorail, C=Commuter Rail, B= Bus, and L=Light Rail)
	NCT	Transit access distance codes
	STAPARK	Station Parking lot flag ('Y' or blank)
	STAUSE	Station Active flag ('Y' or blank)
	SNAME	Station Name
	STAC	Station centroid (5000 - 8000 series)
	STAZ	Nearest TAZ centroid (1 - 3722 series)
	STAT	Station node (9000 - 11999 series)
	STAP	Station PNR node (12000 - 14999 series)
	STAN1	Bus Node connector
	STAN2	Bus Node connector
	STAN3	Bus Node connector
	STAN4	Bus Node connector
	STAPCAP	Parking lot capacity
	STAX	Station X-Coordinate
	STAY	Station Y-Coordinate
	STAPKCOST	AM peak daily parking cost
	STAOPCOST	Off peak parking cost
	STAPKSHAD	AM Shadow parking cost
	STAOPSHAD	Off peak Shadow parking cost
	FIRSTYR	Station opening year

Table 27 Interpretation of transit access distance codes (NCT)

Acc Dist Code	Interpretation
1	End-of-the-line station (e.g., Shady Grove Metro)
2	Intermediate station (e.g., Rockville Metro)
3	PNR close to a CBD (e.g., Rhode Island Ave. Metro, Fort Totten)
0	Only KNR-access links generated (e.g., Braddock Road, National Airport, Clarendon)
9	Metrorail sta. in use, but no PNR/KNR access (e.g., Dupont Circle, Farragut North, Metro Ctr.)
8	Pentagon Metro Sta., allows for very long KNR links, to represent "slugging" (informal carpool)

Table 28 HBW zonal parking costs/terminal time file (HBWV2a1.dbf)

File Name	Variable Name	Description
HBWV2a1.dbf	TAZ	TAZ (1-3,722)
	PCTWKSH	Percent short walk to transit
	PCTWKLG	Percent long walk to transit
	AREA	in sq. mile

Table 29 Walk Access Links (WalkAcc_Lnks.dbf)

File Name	Variable Name	Description
WalkAcc_Links.dbf	A	A-Node
	B	B_Node
	DISTANCE	Link distance (in 1/100ths of miles)
	FTYPE	Link Facility Type Code (0-6)
		0/centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp
	TAZ	TAZ (1-3,722)

Table 30 Outputs of pre-transit network processing

Transit support files in inputs subdirectory	met_link.tb, com_link.tb, lrt_link.tb, new_link.tb, met_node.tb, com_node.tb, lrt_node.tb, new_node.tb, bus_pnrn.tb, met_pnrn.tb, com_pnrn.tb, lrt_pnrn.tb, new_pnrn.tb, met_bus.tb, com_bus.tb, lrt_bus.tb, new_bus.tb	Text
Transit network walk link files	sidewalk.asc walkacc.asc support.asc	Text
Percent of TAZ within short/long walk from transit	HBWV2A1.dbf	DBF
	NLWalkPCT.txt	Text
PNR lot to station transfer links	metampnr.tb, comampnr.tb, busampnr.tb, newampnr.tb, lrtampnr.tb, metoppnr.tb, comoppnr.tb, busoppnr.tb, newoppnr.tb, lrtoppnr.tb	Text
Transit access link files	mrpram.asc, mrprop.asc, mrkram.asc, mrkrop.asc, cram.asc, crop.asc, buspram.asc, busprop.asc, buskram.asc, buskrop.asc, lrtam.asc, lrtop.asc, newam.asc, newop.asc, lrtkram.asc, lrtkrop.asc, newkram.asc, newkrop.asc, autoall.asc	Text

5.2 Application Details

It is important to understand the various elements of the Version 2.3 transit network system. The elements are listed in Table 31. The network consists of highway links, transit stops, PNR lots, rail stations, rail links, and transit lines (modes 1-10). The transit network contains access links relating to zonal access connections including zone/transit stop walking links (mode 16), and zone/PNR auto links (mode 11). The network also includes other walk-related connections such as sidewalk links used in transferring (mode 13), rail station/bus stop connections (mode 12), and PNR lot/station connections (mode 15). The above scripts are used to develop all of these types of "support" links, with the exception of station/bus transfer links which are addressed as part of pre-network development.

The Mode Choice Model chapter of this report addresses the how access links are developed by the WALKACC.S, Parker.S, and the AutoAcc4.S programs.

Table 31 Overview of Version 2.3 Transit Network Elements

Transit Network Element	Description	Numbering	Modes/Notes
Bus stop nodes	Highway nodes that reflect bus stops	20000 - 60000	boarding/alighting locations
PNR lots	Point location representing PNR lot	11001-13999	11001-11999 Metrorail
			12001-12999 Commuter rail
			13001-13999 LRT/BRT/Streetcar
Station	Point location representing rail stop	8001-10999	8001-8999 Metrorail
			9001-9999 Commuter rail
			10001-10999 LRT/BRT/New
Rail links	Fixed guideway segments connecting	-	Mode 3= Metrorail
	stations (non-highway transit links)		Mode 4 = Commuter rail
			Mode 5 = light rail
			Mode 10= BRT, Streetcar
Walk access links	TAZ -transit stop bike/ pedestrian connections	-	Mode 16= TAZ-to-transit stop node
			Mode 13= sidewalk links
Auto access links	TAZ-PNR lot driving connections	-	Mode 11
PNR lot-to station links	Walk transfer links from PNR lot to Station	-	Mode=15
Station-to-bus transfer link	Walk transfer links between stations & bus stops		Mode=12
Transit line files	Bus, Rail transit line data		Modes 1-10
	(line characteristics, node sequence of route)		

Chapter 6 Transit Skim File Development

6.1 Overview

The transit skimming file process involves the development 22 sets of Level-of-Service (LOS) skims corresponding to two time period (AM and off-peak) by four submode groups (Bus only, Metrorail only, Bus-Metrorail combination, and commuter rail) by three access mode (walk, PNR, KNR)[18]. The transit network building and skimming scripts are named TRANSIT_SKIMS_CR.S, TRANSIT_SKIMS_MR.S, TRANSIT_SKIMS_AB.S, TRANSIT_SKIMS_BM.S. Additionally, transit accessibility summaries are need to support the vehicle ownership model. The Transit_Accessibility.S script is used for this purpose.

Table 32 Inputs to transit skim file development

Local bus future time degradation factors	Lbus_TimFTRS.ASC	Text
Transit line files	MODE1,MODE2AM,...MODE10AM.TB MODE1,MODE2OP,...MODE10OP.TB	Text
Transit path tracing selection criteria	PATHTRACE.S	Script block
Binary highway network	ZONEHWY.NET	Binary
Transit support files in inputs subdirectory	met_link.tb, com_link.tb, lrt_link.tb, new_link.tb, met_node.tb, com_node.tb, lrt_node.tb, new_node.tb, bus_pnrn.tb, met_pnrn.tb, com_pnrn.tb, lrt_pnrn.tb, new_pnrn.tb, met_bus.tb, com_bus.tb, lrt_bus.tb, new_bus.tb	Text
Transit network walk link files	sidewalk.asc walkacc.asc support.asc	Text
PNR lot to station transfer links	metampnr.tb, comampnr.tb, busampnr.tb, newampnr.tb, lrtampnr.tb, metoppnr.tb, comoppnr.tb, busoppnr.tb, newoppnr.tb, lrtoppnr.tb	Text
Transit access link files	mrpram.asc, mrprop.asc, mrkram.asc, mrkrop.asc, cram.asc, crop.asc, buspram.asc, busprop.asc, buskram.asc, buskrop.asc, lrtam.asc, lrtop.asc, newam.asc, newop.asc, lrtkram.asc, lrtkrop.asc, newkram.asc, newkrop.asc, autoall.asc	Text

[18] This should equal 24 (2x3x4) but KNR access to commuter rail mode is not considered by the mode choice model, and so the total number of required path sets equals 22.

Table 33 Outputs of transit skim file development

Commuter rail skim files	SUPLCR*<AA><Prd>*.ASC SUPNCR*<AA><Prd>*.DBF TRNLCR*<AA><Prd>*.DBF <ITER>_*<Prd><AA>*_CR.STA <ITER>_*<Prd><AA>*_CR.SKM <ITER>_*<Prd><AA>*_CR.TTT	Text DBF DBF Binary Binary Binary
Metrorail support skim files	SUPLMR*<AA><Prd>*.ASC SUPNMR*<AA><Prd>*.DBF TRNLMR*<AA><Prd>*.DBF <ITER>_*<Prd><AA>*_MR.STA <ITER>_*<Prd><AA>*_MR.SKM <ITER>_*<Prd><AA>*_MR.TTT	Text DBF DBF Binary Binary Binary
All Bus support skim files	SUPLAB*<AA><Prd>*.ASC SUPNAB*<AA><Prd>*.DBF TRNLAB*<AA><Prd>*.DBF <ITER>_*<Prd><AA>*_AB.STA <ITER>_*<Prd><AA>*_AB.SKM <ITER>_*<Prd><AA>*_AB.TTT	Text DBF DBF Binary Binary Binary
Bus/Metrorail support skim files	SUPLBM*<AA><Prd>*.ASC SUPNBM*<AA><Prd>*.DBF TRNLBM*<AA><Prd>*.DBF <ITER>_*<Prd><AA>*_BM.STA <ITER>_*<Prd><AA>*_BM.SKM <ITER>_*<Prd><AA>*_BM.TTT	Text DBF DBF Binary Binary Binary
Job accessibility by transit	<ITER>_<Prd><AA>_<Path>_JobAcc.dbf	DBF

Note: <Prd>= AM and OP <AA>= WK, DR, KR <ITER> =PP, i1…i4

The inputs out outputs to transit skimming are shown in Table 32 and Table 33, respectively. Transit accessibility outputs are listed on Table 34.

Table 34 Job accessibility by transit file format description

<ITER> =PP , i1, i2,...i4, <PP> = AM, OP, <AA> =WK, DR, <Path> = BM, MR		
File Name	Variable Name	Description
<ITER>_<PP>_<Acc>_<Path>_JOBACC.dbf	TAZ	TAZ (1-3,722)
	EMP35	Transit accessibility to jobs With in 35 minutes
	EMP40	Transit accessibility to jobs With in 40 minutes
	EMP45	Transit accessibility to jobs With in 45 minutes
	EMP50	Transit accessibility to jobs With in 50 minutes
	EMPTOT	Transit accessibility to total regional jobs

Chapter 7 Transit Fare Development

7.1 Overview

Zone-to-zone transit fares are developed for the 22 paths sets described in the transit skimming section. The fares are developed using the scripts named Prefarv23.s, Metrorail_Skims.S, Mfare1.s, and Mfare2.s. The inputs to the fare process are shown in Table 35 and the outputs are shown in Table 36. After the fare process is executed four scripts are used to combine transit skims and fares into consolidated submode files, Assemble_Skims_MR.s, Assemble_Skims_AB.s, Assemble_Skims_BM.s.

Table 35 Inputs to transit fare development

Zonal transit walk percent	Inputs\NLwalkPct.txt	Text
Zonal TAZ-to-bus fare zone equivalence	Inputs\TAZFRZN.ASC	Text
Zonal Area Type file	AreaType_File.dbf	DBF
Zonal land use file	zone.dbf	DBF
Zonal TAZ-Mode choice district equiv.	areadef3722.prn	Text
Metro Station Link File	METLNKM1.TB	Text
Metro Station XY File	METNODM1.TB	Text
Metrorail turn penalty file	Inputs\trnpen.dat	Text
MFARE1 A1 (Coordinate) File	MFARE1.A1	Text
Metrorail station discount file	Inputs\MFARE1_STA_DISC.ASC	Text
WMATA tariff parameters	Inputs\tarriff.txt	Text
Deflation factor file	Trn_deflator.txt	Text
	<ITER>_*<Prd><AA>*_CR.STA <ITER>_*<Prd><AA>*_CR.SKM <ITER>_*<Prd><AA>*_MR.STA <ITER>_*<Prd><AA>*_MR.SKM <ITER>_*<Prd><AA>*_AB.STA <ITER>_*<Prd><AA>*_AB.SKM <ITER>_*<Prd><AA>*_BM.STA <ITER>_*<Prd><AA>*_BM.SKM	Binary
	<ITER>_*<Prd>*_*<AA>*_CR.FAR <ITER>_*<Prd>*_*<AA>*_MR.FAR <ITER>_*<Prd>*_*<AA>*_AB.FAR <ITER>_*<Prd>*_*<AA>*_BM.FAR	
Peak / Off-Peak MFARE2 Bus Fare Matrix	Inputs\busfaram.asc Inputs\busfarop.asc	Text
Peak /Off-Peak MFARE2 A2 File	FARE_A2.ASC	Text

Note: <Prd>= AM and OP <AA>= WK, DR, KR <ITER> =PP, i1…i4

Table 36 Outputs of transit fare development

Summary of walkshed area and walkshed percentage	Prepare_MC_Zfile.txt	Text
Output Zone file for the NL mode choice model	ZONEV2.A2F	Text
A "complete" A2 file for the MFARE2.S	Fare_a2.asc	
Metrorail distance skims	RLDIST.SKM	Text
Metrorail station to station fares	AM_Metrorail_Fares.TXT OP_Metrorail_Fares.TXT	Text
Zonal fares	<ITER>_*<Prd>*_*<AA>*_CR.FAR <ITER>_*<Prd>*_*<AA>*_CR.FR5 <ITER>_*<Prd>*_*<AA>*_CR.TXT <ITER>_*<Prd>*_*<AA>*_MR.FAR <ITER>_*<Prd>*_*<AA>*_MR.FR5 <ITER>_*<Prd>*_*<AA>*_MR.TXT <ITER>_*<Prd>*_*<AA>*_AB.FAR <ITER>_*<Prd>*_*<AA>*_AB.FR5 <ITER>_*<Prd>*_*<AA>*_AB.TXT <ITER>_*<Prd>*_*<AA>*_BM.FAR <ITER>_*<Prd>*_*<AA>*_BM.FR5 <ITER>_*<Prd>*_*<AA>*_BM.TXT	
Combined time and fare commuter rail skims	<ITER>_TRNAM_CR.SKM <ITER>_TRNOP_CR.SKM	Binary
Combined time and fare Metrorail skims	<ITER>_TRNAM_MR.SKM <ITER>_TRNOP_MR.SKM	Binary
Combined time and fare all bus skims	<ITER>_TRNAM_AB.SKM <ITER>_TRNOP_AB.SKM	Binary
Combined time and fare bus/Metrorail skims	<ITER>_TRNAM_BM.SKM <ITER>_TRNOP_BM.SKM	Binary

Table 37 TAZ / Bus Fare Zone Equivalency File Format Description (TAZFRZN.ASC)

Columns	Format	Field Description
Zonal data (All lines in the file)		
1-8	I4	TAZ Number (1-3,675) and Metrorail Station No. (1-150)
9-16	I4	1st Bus fare zone 1 (currently numbered 1 to 21)
17-24	I4	2nd Bus fare zone 2 (currently numbered 1 to 21)
Station data (first 150 lines of the file only)		
41-48	I4	1st Bus Fare Zone (currently numbered 1 to 21)
49-56	I4	2nd Bus Fare Zone (currently numbered 1 to 21)
57-64	I8	Jurisdiction code
65-72	I8	P discount
73-80	I8	A discount

Chapter 8 Demographic Submodels

Demographic submodels are applied within the Trip_Generation.bat batch file using the Demo_Models.s Cube Voyager script (see page A-5 of Appendix A). This script applies the three demographic submodels that are run prior to trip generation: household size, household income, and vehicle availability (see Chapter 3 of the calibration report for more details). The inputs to the model are zonal land use data (zone.dbf), data about area types (areaType_File.dbf), and information about the accessibility to jobs via transit. The zone.dbf file contains zonal households, population, jurisdiction code, and income index, as well as the household size and household income submodels (in the form of lookup tables). The households in each TAZ are then allocated to a household size group (1, 2 ,3, or 4+) and an income group (<50K, 50K-100K, 100K-150K, or 150+K). Final adjustments are then done based on the area type of each zone (obtained from the areatype.dbf file) and the script accumulates jurisdictional and regional totals.

Next, the Demo_Models.s reads in the number of jobs accessible by Metrorail and Bus/Metrorail modes within 45 minutes for each TAZ. This information along with household size, household income, area type, and the DC dummy variable are used to allocate households to the four vehicle ownership categories (0, 1, 2, or 3+). The totals are once again calculated by jurisdiction.

Then, a file is produced, for each of the four income levels, which contains the number of households by household size and vehicle availability. These files are later used in trip generation. Lastly, the script accumulates the households by area type and prints out the following summaries located in the Demo_Models<ITER>.txt:

- Regional Households by Size and Income Summary
- Jurisdictional Households by Size
- Jurisdictional Households by Income
- Regional Households by Vehicles Available and Size Summary
- Regional Households by Vehicles Available and Income Summary
- Jurisdictional Households by Vehicles Available
- Estimated Households By Size Level by Area Type
- Estimated Households By Income Level by Area Type
- Estimated Households By Vehicle Availability Level by Area Type

Process inputs are outputs are shown in Table 38 and Table 39.

Table 38 Inputs to the Demographic Models

Zonal Land Use File	Inputs\zone.dbf	DBF
Zonal Area Type File	AreaType_File.dbf	DBF
Transit Accessibility File (Metrorail only and Bus & Metrorail service)	<ITER>_AM_WK_MR_JOBACC.dbf <ITER>_AM_DR_MR_JOBACC.dbf <ITER>_AM_WK_BM_JOBACC.dbf <ITER>_AM_DR_BM_JOBACC.dbf	DBF

Note: <ITER> =PP, i1…i4

Table 39 Outputs of the Demographic Models

Zonal HHs of Income Level 1, Stratified by Size and Vehicle Avail.	HHI1_SV.txt	Text
Zonal HHs of Income Level 2, Stratified by Size and Vehicle Avail.	HHI2_SV.txt	Text
Zonal HHs of Income Level 3, Stratified by Size and Vehicle Avail.	HHI3_SV.txt	Text
Zonal HHs of Income Level 4, Stratified by Size and Vehicle Avail.	HHI4_SV.txt	Text
Interim Output: Zonal Households stratified by Income Level, household Size, and vehicle available (64 cross-classes)	Demo_Models_HHbyISV_<iter>.dbf	DBF

Chapter 9 Trip Generation

9.1 Control/Support File(s):

TRIP_GENERATION.S, TRIP_GENERATION_Summary.S, Truck_Com_Trip_Generation.s

9.2 Application Details:

Trip generation is executed within the Trip_Generation.bat batch file using three Cube Voyager scripts: Trip_Generation.s, Trip_Generation_Summary.s, and Truck_Com_Trip_Generation.s (as shown on page A-5 of Appendix A). The inputs to the Trip_Generation.bat batch file are shown in Table 40.

Table 40 Inputs to trip generation

Zonal Land Use File	zone.dbf	DBF
Zonal Area Type File	AreaType_File.dbf	DBF
Zonal Households stratified by Income Level, household Size, and vehicle available	Demo_Models_HHbyISV_<iter>.dbf	DBF
Zonal GIS variable File	GIS_variables.dbf	DBF
Trip production rates	weighted_trip_rates.dbf	DBF
External Production and Attraction File	Ext_PsAs.dbf	DBF
Non motorized trip production share model coefficients	NMPrates.dbf	DBF
Non motorized trips Attraction share model coefficients	NMArates.dbf	DBF
Trip attraction rates	AttrRates.dbf	DBF
HB income shares	HBINCRAT.dbf	DBF
Consolidated zonal land use file	TripGen_LUFile.dbf	Text
Truck and commercial vehicles trip rates	support\truck_com_trip_rates.dbf	DBF
Zonal access verification file	Skimtot<ITER>.dat	Text
	JurCore.dbf	DBF

The Trip_Generation.s script calculates zonal trip productions and attractions. The Trip_Generation_Summary.s summarizes the demographic information and the trip ends by jurisdiction. The Truck_Com_Trip_Generation.s produces trip ends for commercial vehicles and trucks.

The Trip_Generation.s script begins by reading the zonal land use (Zone.dbf); the area type file (AreaType_File.dbf); external trip productions and attractions (EXT_PsAs.dbf, described in Table 41); demographic model outputs (Demo_Models_HHbyISV_%_iter_%.dbf); zonal walkability factors (GIS_variables.dbf); trip production rates (weighted_trip_rates.dbf); non-motorized production model coefficients (NMPrates.dbf); non-motorized attraction model coefficients (NMArates.dbf); trip attraction model coefficients (AttrRates.dbf); and income shares for home-based trips (HBINCRAT.dbf).

Table 41 External Production and Attraction File (Ext_PsAs.dbf)

File Name	Variable Name	Description
Ext_PsAs.dbf	TAZ	TAZ Number (1-3,722)
	FACILITY	Facility name
	AAWT_CTL	Annua Average Weekday Traffic control total
	CNTFTR	<Not Used>
	AUTO_XI	Auto driver external-internal (X-I) trips
	AUTO_IX	Auto driver internal-external (I-X) trips
	AUTO_XX	Auto driver through trips (X-X)
	CV_XX	Commercial Vehicles through trips (X-X)
	HBW_XI	Home based Work (HBW) external-internal (X-I) trips
	HBS_XI	Home based Shop (HBS) external-internal (X-I) trips
	HBO_XI	Home based Other (HBO) external-internal (X-I) trips
	NHB_XI	Non-Home based (NHB) external-internal (X-I) trips
	CV_XI	Commercial Vehicles external-internal (X-I) trips
	HBW_IX	Home based Work (HBW) internal-external- (I-X) trips
	HBS_IX	Home based Shop (HBS) internal-external- (I-X) trips
	HBO_IX	Home based Other (HBO) internal-external- (I-X) trips
	NHB_IX	Non-Home based (NHB) internal-external- (I-X) trips
	CV_IX	Commercila Vehicles internal-external- (I-X) trips
	TRCK_XX	Truck through trips (X-X)
	TRCK_XI	Truck external-internal (X-I) trips
	TRCK_IX	Truck internal-external- (I-X) trips
	MTK_XI	Medium Trucks external-internal (X-I) trips
	HTK_XI	HeavyTrucks external-internal (X-I) trips

The one-mile floating density is then calculated for population and employment and a half-mile floating density is calculated for street blocks. These are saved in an intermediate file named TripGen_LUFile.dbf (Table 42). Then, the script calculates zonal trip productions based on demographic data and applies the non-motorized production model to the results. Motorized internal trips productions are then obtained by subtracting the non-motorized trips, internal-external trips (based on a set of curves described in the Section 4.3 of the Calibration Report) and applying a jurisdictional production adjustment factor (Appendix A of the Calibration Report). These are then disaggregated by income level and written out to the Trip_Gen_Productions_<ITER>.dbf file described in Table 43.

Table 42 Consolidated Zonal Land Use File

File Name	Variable Name	Description
TripGen_LUFile.dbf	TAZ	TAZ Number (1-3,722)
	HH	Number of house holds
	TOTPOP	Total Population
	TOTEMP	Total employment
	RETEMP	Retail employment
	NRETEMP	Non-retail employment
	OFFEMP	Office employment
	OTHEMP	Other employment
	INDEMP	Industrial employment
	HHPOP	House hold population
	GQPOP	Group quarter population
	LANDAREA	Land area (sq. mi.)
	POP_10	Number of population within one "floating" mile
	EMP_10	Number of employment within one "floating" mile
	AREA_10	Zonal Area within one "floating" mile
	POPDEN10	Population density within one "floating" mile
	EMPDEN10	Employment density within one "floating" mile
	ADISTTOX	Distance to the nearest external station
	BLOCKS05	Blocks within 0.5 mile "floating" blocks
	AREA05	Area within 0.5 mile "floating" blocks
	BLOCKDEN05	Block density within 0.5 mile "floating" blocks
	JURCODE	Jurisdiction code (0-23)
	ATYPE	Area Type (1-6)

Table 43 Zonal trip productions

File Name	Variable Name	Description
Trip_Gen_Productions_<iter>.dbf	TAZ	TAZ Number (1-3,722)
<ITER> =PP , i1, i2,...i4	HBW_MTR_PS	Home Based Work motorized person trips productions
	HBW_NMT_PS	Home Based Work non-motorized person trips productions
	HBW_ALL_PS	Home Based Work total person trips productions
	HBWMTRP_<iter>	Home Based Work motorized person trips productions by iterations (<iter>)
	HBS_MTR_PS	Home Based Shop motorized person trips productions
	HBS_NMT_PS	Home Based Shop non-motorized person trips productions
	HBS_ALL_PS	Home Based Shop total person trips productions
	HBSMTRP_<iter>	Home Based Shop motorized person trips productions by iterations (<iter>)
	HBO_MTR_PS	Home Based Other motorized person trips productions
	HBO_NMT_PS	Home Based Other non-motorized person trips productions
	HBO_ALL_PS	Home Based Other total person trips productions
	HBOMTRP_<iter>	Home Based Other motorized person trips productions by iterations (<iter>)
	NHW_MTR_PS	Non-Home-Work motorized person trips productions
	NHW_NMT_PS	Non-Home-Work non-motorized person trips productions
	NHW_ALL_PS	Non-Home-Work total person trips productions
	NHO_MTR_PS	Non-Home-Other motorized person trips productions
	NHO_NMT_PS	Non-Home-Other non-motorized person trips productions
	NHO_ALL_PS	Non-Home-Other total person trips productions

Next, the zonal trip attractions are calculated by applying the attraction trip models to the land use file. Non-motorized trip attractions are then determined and subtracted from the total trip attractions. Similar to productions, attractions are multiplied by an adjustment factor (Appendix A of the Calibration Report) and disaggregated by income level. The computed trip attractions are then written out to

Trip_Gen_Attractions_Comp_<ITER>.dbf file. Lastly, the scaling factors for attractions are calculated to account for the external trips and applied to the zonal trip attractions. The final trip attractions are saved in the Trip_Gen_Attractions_Final_<ITER>.dbf described in Table 44.

Table 44 Iteration-Specific Trip Generation Attractions file (Trip_Gen_Attractions_Final_<iter>.dbf)

File Name	Variable Name	Description
Trip_Gen_Attractions_Final_<iter>.dbf	TAZ	TAZ Number (1-3,722)
<ITER> =PP , i1, i2,...i4	HBW_MTR_AS	Home Based Work motorized person trips attractions
	HBW_NMT_AS	Home Based Work non-motorized person trips attractions
	HBW_ALL_AS	Home Based Work total person trips attractions
	HBWMTRA_<iter>	Home Based Work motorized person trips attractions by iterations (<iter>)
	HBS_MTR_AS	Home Based Shop motorized person trips attractions
	HBS_NMT_AS	Home Based Shop non-motorized person trips attractions
	HBS_ALL_AS	Home Based Shop total person trips attractions
	HBSMTRA_<iter>	Home Based Shop motorized person trips attractions by iterations (<iter>)
	HBO_MTR_AS	Home Based Other motorized person trips attractions
	HBO_NMT_AS	Home Based Other non-motorized person trips attractions
	HBO_ALL_AS	Home Based Other total person trips attractions
	HBOMTRA_<iter>	Home Based Other motorized person trips attractions by iterations (<iter>)
	NHW_MTR_AS	Non-Home-Work motorized person trips attractions
	NHW_NMT_AS	Non-Home-Work non-motorized person trips attractions
	NHW_ALL_AS	Non-Home-Work total person trips attractions
	NHO_MTR_AS	Non-Home-Other motorized person trips attractions
	NHO_NMT_AS	Non-Home-Other non-motorized person trips attractions
	NHO_ALL_AS	Non-Home-Other total person trips attractions

The Trip_Generation_Summary.s creates a summary text file, Trip_Generation_Summary_<ITER>.txt, which includes the following tables:

- Land Activity by Jurisdiction
- Land Activity by Area Type
- Motorized Trip Productions by Purpose and Jurisdiction
- Motorized Trip Productions per Household by Purpose and Jurisdiction
- Motorized Trip Productions by Purpose and Area Type
- Non-Motorized Trip Productions by Purpose and Jurisdiction
- Non-Motorized Trip Productions by Purpose and Area Type
- Home-Based Motorized Trip Productions by Purpose, Income, and Jurisdiction
- Home-Based Motorized Trip Productions by Purpose, Income, and Area Type
- Motorized Trip Attractions by Purpose and Jurisdiction
- Motorized Trip Attractions per Job by Purpose and Jurisdiction
- Motorized Trip Attractions by Purpose and Area Type
- Non-Motorized Trip Attractions by Purpose and Jurisdiction
- Non-Motorized Trip Attractions by Purpose and Area Type
- Home-Based Motorized Trip Attractions by Purpose, Income, and Jurisdiction
- Home-Based Motorized Trip Attractions by Purpose, Income, and Area Type

The Truck_Com_Trip_Generation.s script reads in the zonal land use file (Zone.dbf), the area type file (AreaType_File.dbf), external trip productions and attractions (EXT_PsAs.dbf), demographic model outputs (Demo_Models_HHbyISV_%_iter_%.dbf), truck and commercial trip model coefficients (truck_com_trip_rates.dbf), and the zonal access verification file (Skimtot<ITER>.dat). For the list of inputs, see Table 40. The script then uses the truck and commercial trip model coefficients and the land use data to calculate medium and heavy truck and commercial vehicle zonal trips. After an adjustment factor is applied, these are written out to a ComVeh_Truck_Ends_<ITER>.dbf file described in Table 45.

Table 45 Truck and commercial vehicles trip ends (ComVeh_Truck_Ends_<iter>.dbf)

File Name	Variable Name	Description
ComVeh_Truck_Ends_<ITER>.dbf	TAZ	TAZ Number (1-3,722)
	COMM_VEH	
	MED_TRUCK	
	HVY_TRUCK	
	ICOMM_VEH	
	IMED_TRUCK	
	IHVY_TRUCK	

The script also generates a summary text file- Truck_Com_Trip_Generation_<ITER>.txt, which includes the following tables:

- Regional Total Truck and Commercial Trip-Ends
- Truck and Commercial Vehicle Internal Trip Totals by Area Type
- Truck and Commercial Vehicle Internal Trip Totals by Jurisdiction

For the list of all output files, see Table 46.

Table 46 Outputs of trip generation

Zonal trip productions	Trip_Gen_Productions_<ITER>.dbf	DBF
Zonal Trip Attractions - Initial	Trip_Gen_Attractions_Comp_<ITER>.dbf	DBF
Zonal trip attractions –Scaled	Trip_Gen_Attractions_Final_<ITER>.dbf	DBF
Truck and commercial vehicles trip ends	Com_Veh_Truck_ends_<ITER>.dbf	DBF
Trip generation summary report	Trip_Generation_Summary_<ITER>.txt	Text

Chapter 10 Trip Distribution

10.1 Overview

The trip distribution process is invoked by the Trip_Distribution.bat file. The input and output files are listed in Table 47 and Table 48.

Table 47 Inputs to trip distribution

Trip productions	Trip_Gen_Productions_<iter>.dbf	DBF
Trip attractions	Trip_Gen_Attractions_Final_<iter>.dbf	DBF
Truck and commercial vehicles trip ends	ComVeh_Truck_Ends_<iter>.dbf	DBF
External productions and attractions	Ext_Trip_Gen_PsAs_<iter>.dbf	DBF
SOV Peak, Off-Peak Highway Skims	SOV<ITER>_AM.SKM, SOV<ITER>_MD.SKM	Binary
	<ITER>_am_wk_MR.ttt <ITER>_am_dr_MR.ttt <ITER>_op_wk_MR.ttt <ITER>_op_dr_MR.ttt	Binary
Zonal Area Type File	AreaType_File.dbf	DBF
Highway Terminal Time File	Ztermtm.asc	Text
F-Factor Files	Version_23_ffactors.dbf	DBF
K-Factor Files	HBWK.DAT, HBSK.DAT, HBOK.DAT, NHWK.DAT, NHOK.DAT	Binary
Income level Toll/Time Equivalent File	Toll.inc	Text

Note: <ITER> =PP, i1…i4

Table 48 Outputs of trip distribution

8 Trip Tables (HBW, HBS, HBO, NHW, NHO, Med Truck, Heavy Truck)	HBW_<ITER>.PTT, HBS_<ITER>.PTT, HBO_<ITER>.PTT, NHW_<ITER>.PTT, NHO_<ITER>.PTT, COM_<ITER>.PTT, MTK_<ITER>.PTT, HTK_<ITER>.PTT	Binary
Output matrices for mode choice model	<ITER>_hbw_NL.ptt, <ITER>_hbs_NL.ptt, <ITER>_hbo_NL.ptt, <ITER>_nhw_NL.ptt, <ITER>_nho_NL.ptt	Binary
Composite time impedances HBW income levels 1-4	hbwci1_4.dat, hbsci1_4.dat, hboci1_4.dat, nhwci1_4.dat, nhoci1_4.dat	Binary

The trip distribution process with two scripts used to summarize total trip-ends (Ps and As) by purpose, Prepare_Ext_Auto_Ends.s and Prepare_Ext_COMTRK_Ends.s. The total trip-ends are prepared as a trip pattern basis for distributing external trips which are treated separately in trip distribution. The trip distribution model is executed for all purposes using the script named Trip_distribution.s.

The trip distribution process uses different LOS impedances measures, depending on trip purpose. Work (HBW) trips are distributed using AM peak travel impedances while midday (MD) impedances are used for all remaining purposes.

The script first prepares zonal highway terminal times which are based on the zonal area type. The termional times range from 1 minute in the least developed areas to 5 minutes for highly developed areas. The terminal times are then added to the over-the-network highway travel time skims. Next, composite impedance tables are developed combining transit time and highway times, based on the following formulation:

$CT_i = 1.0 / [(1.0/(HT + TollT_i)) + (P_i/TT)]$

Where:
CT_i = Composite time for income level "i"
HT = Congested highway time (min) including terminal time
$TollT_i$ = Time equivalent (min) of toll(s) associated with the minimum time path for income "i"
P_i = Regional transit share of income "i" for the trip purpose
TT = Metrorail–related transit time (min) including in-vehicle & out-of-vehicle components

The basis of the $TollT_i$ term calculation is specified in Table 49. The table indicates the average time valuation, in minutes per 2007 dollar, that is assigned to a toll value by income level and trip type. The table indicates, for example, that a $1.00 toll equates to 8.7 minutes of travel time for a traveler in income level 1. More generally, the table indicates that travelers commuting to work are less sensitive to tolls than non-work-bound travelers because the time valuation of commuters is relatively high. The table also reflects the intuitive generalization that lower income travelers are more sensitive to tolls than the higher income travelers.

Work & Non-Work Time – Dollar Equivalents by Income Level

Table 49 Time Valuation (Minutes/2007$) by Purpose and Income Level

HH Income Quartile Range (1)	Assumed Mid-Point of HH Inc. Range	Hourly Rate per Worker (2)	2007 Time Valuation (Minutes per Dollar)	
			Work Trips (75% V.O.T.)	Non-work (50% V.O.T.)
$ 0 - $ 50,000	$25,000	$9.23	8.7	13.0
$ 50,000 - $ 100,000	$75,000	$27.70	2.9	4.3
$100,000 - $150,000	$125,000	$46.17	1.7	2.6
$150,000 +	$175,000	$64.64	1.2	1.9

Notes:
(1) Income groups based on 2007 ACS-based quartiles
(2) Hourly rate based on 1,920 annual hours/worker * 1.41 workers/HH = 2,707 hrs/HH
(3) Median 2007 annual HH income for the TPB modeled area is $84,280

Chapter 11 Mode Choice

11.1 Transit constraint through the regional core

The transit constraint through the regional core, or simply the "transit constraint," is a technical adjustment to the trip tables coming out of the mode choice process designed to reflect a WMATA policy assumption that, during peak periods, the Metrorail system may have insufficient capacity to handle all the demand traveling to and through the regional core. Typically, it is assumed that the Metrorail system will be able to handle all of the peak-period demand to and through the regional core in the near term, but, since demand is growing through time, the system might not be able to handle all the peak-period demand at some future time, depending on the amount of growth in demand and the number of rail cars available in a given year. The assumed year at which the Metrorail system will be at its peak capacity during the peak periods to and through the regional core is known as the "binding year." For years beyond the binding year, it is assumed that any growth in peak-period Metrorail demand to and through the regional core will be forced to switch to other travel modes (specifically, auto person trips). The transit constraint was initiated by WMATA in 2000 to address funding shortfalls restricting the expansion of the rail fleet.[19] WMATA policy sets the binding year, which is currently set at 2020. This means that, for any forecast year past 2020, the transit constraint is applied, i.e., forecasted peak-period Metrorail trips to and through the regional core are shifted to other travel modes (specifically, auto person trips). The regional core is defined as the set of Metrorail stations in the central employment area, i.e., the portion of the system bounded by Dupont Circle, Mt. Vernon Square, New York Avenue, Stadium Armory, Anacostia, National Airport, and Rosslyn stations. This area is also sometimes referred to by technical audiences as "Ring 0" and "Ring 1." Note that non-Metrorail-related transit trips and off-peak Metrorail trips are not affected by the transit constraint process.

The transit constraint is applied in the following way (assuming that 2020 is the binding year). Model runs representing the binding year and years prior to the binding year are conducted in the normal fashion, i.e., using the **mode_choice.bat** batch file (see page A-12 of Appendix A). Model runs representing any year following the binding year, e.g., 2030, are conducted using the **mode_choice_tc_v23.bat** batch file (see page A-12 of Appendix A), as follows:

Peak 2020 Metrorail trips to and through the core are estimated using a time-of-day model.
Peak 2030 Metrorail trips to and through the core are estimated using a time-of-day model.
Peak 2030 Metrorail trips to and through the core are adjusted (downward) to match 2020 ridership levels.
The "excess" 2030 Metrorail trips that cannot be accommodated are converted to auto person trips
The constraint process occurs for each speed feedback iteration ("i1" through "i4).

[19] Ronald Milone, "TPB Version 2.3 Travel Model on the 3,722-TAZ area system: Status report" (presented at the September 23, 2011 meeting of the Travel Forecasting Subcommittee of the Technical Committee of the National Capital Region Transportation Planning Board, held at the Metropolitan Washington Council of Governments, Washington, D.C., September 23, 2011).

Thus, the mode choice model is executed normally with the mode_choice.bat batch file, which invokes the following:

Mode choice model application program (AEMS.EXE);
Jurisdictional summary script (MC_NL_Summary.s);

By contrast, the mode choice model and transit constraint process are executed using the mode_choice_tc_v23.bat batch file, which invokes the following:

Mode choice model application program (AEMS.EXE);
Jurisdictional summary script (MC_NL_Summary.s);
Constraint adjustment script (mc_constraint_v23.s);

11.2 Control/Support Files

The nested-logit mode choice (NLMC) model is applied using a Fortran program called AEMS.[20] AEMS.EXE is the compiled version of the source code AEMS.FOR. In order to run, AEMS.EXE needs to have several DLL files. The model is run one for each of the five trip purposes, as shown on page A-12 of the flowchart in Appendix A. Each run of the mode choice model requires a "control file," so there are five in total: HBW_NL_MC.CTL, HBS_NL_MC.CTL, HBS_NL_MC.CTL, NHW_NL_MC.CTL, and NHO_NL_MC.CTL. After the five mode choice models run, there is a Cube Voyager script, MC_NL_Summary.s, which is used to create jurisdiction-to-jurisdiction tabulations of the trip tables output from the mode choice model. The inputs to the AEMS mode choice application program are shown in Table 50. The outputs are shown in Table 51.

Table 50 Inputs to the AEMS mode choice application program

Daily person trips, stratified by income group (1, 2, 3, 4), in production/attraction format (INFILE 1)	hbw_income.ptt, hbs_income.ptt, hbo_income.ptt, nhw_income.ptt, nho_income.ptt	Binary
Highway skims, nine tables – SOV, HOV2, HOV3+ for time, distance, and tolls on non-variably-priced facilities (INFILE 2)	hwyam.skm, hwyop.skm	Binary
Commuter rail transit skims (INFILE 3)	trnam_cr.skm, trnop_cr.skm	Binary
All bus transit skims (INFILE 4)	trnam_ab.skm, trnop_ab.skm	Binary
Metrorail transit skims (INFILE 5)	trnam_mr.skm, trnop_mr.skm	Binary
Bus/Metrorail transit skims (INFILE 6)	trnam_bm.skm, trnop_bm.skm	Binary
Zonal data (INFILE 8)	zonev2.a2f	Text

[20] AECOM Consult, Inc., *AECOM Consult Mode Choice Computation Programs, AEMS, Users Guide*, Draft report (Fairfax, Virginia: AECOM Consult, Inc., April 5, 2005).

Table 51 Outputs from the AEMS mode choice application program

Daily person trips, stratified by travel mode (14 tables): 1. DR ALONE 2. SR2 3. SR3+ 4. WK-CR 5. WK-BUS 6. WK-BU/MR 7. WK-MR 8. PNR-CR & KNR-CR 9. PNR-BUS 10. KNR-BUS 11. PNR-BU/MR 12. KNR-BU/MR 13. PNR-MR 14. KNR-MR	hbw_nl_mc.mtt, hbs_nl_mc.mtt, hbs_nl_mc.mtt, nhw_nl_mc.mtt, nho_nl_mc.mtt	Binary

11.3 Travel modes represented in the mode choice model

The mode choice model in the Version 2.3 Travel Model includes 15 choices as shown in Figure 9 and described in Chapter 6 of the Calibration Report. The model includes three auto modes (drive alone, shared ride 2 person, and shared ride 3+ person) and four transit modes (commuter rail, all bus, all Metrorail, and combined bus/Metrorail). The four transit modes are stratified by three modes of access to transit (park and ride [PNR]; kiss and ride [KNR or "ride to transit"]; and walk). As discussed in the calibration report, the definition of high-occupancy vehicle (HOV) trips has changed, compared to the definition that was used in the Version 2.2 Travel Model. Previously, HOV trips coming out of the mode choice model referred to *only those that use HOV facilities for a substantial portion of their trip*. Similarly, in previous models, the definition of low-occupancy vehicle (LOV) included both drive-alone and carpools (provided the carpools did not use a preferential HOV facility). By contrast, in the Version 2.3 NLMC model, the term LOV refers to only the drive-alone trips. Similarly, HOV refers to all shared-ride 2 (2-person carpools) and shared-ride 3 (3+ person carpools), irrespective of whether they use an HOV facility or not.

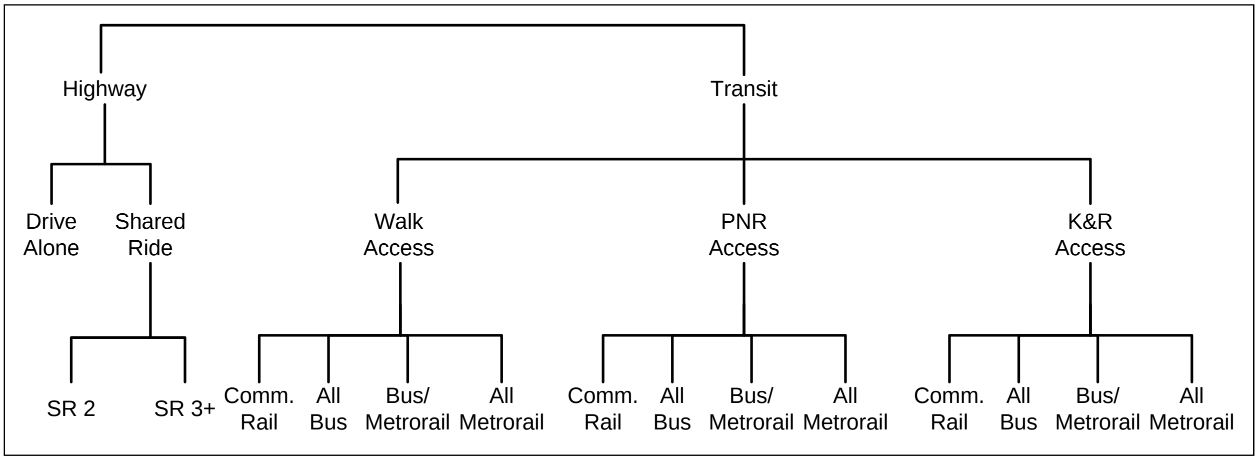


Figure 9 Nesting structure of the nested-logit mode choice model in the Version 2.3 travel model

Ref: "I:\ateam\nest_log\NestedChoice_Struct4.vsd"

11.3.1 Treatment of LRT, BRT, and streetcar

Note that the nesting structure of the TPB Version 2.3 NLMC model does not include branches for specialized transit modes, such as light-rail transit (LRT), bus rapid transit (BRT), and streetcar. From this, one might conclude that the mode choice model is not designed to deal with these special transit modes. In fact, the model is designed to deal with these special transit modes. This section of the report discusses how these modes are treated in both the mode choice model and the transit path skimming process that feeds the mode choice model. This is the scheme that was developed by AECOM in 2004-2005 and has been retained by TPB staff. One of the underlying assumptions is that "premium" transit modes (e.g., Metrorail, commuter rail, LRT, BRT, and streetcar) will typically travel faster than buses, since they have one or more of these characteristics:

A dedicated right-of-way, at least for part, if not all, of the route
Traffic signal priority
Superior acceleration/deceleration (compared to buses)

Network representation: LRT, BRT, and streetcar

In terms of network representation, LRT is typically coded as "mode 5." BRT and streetcar are coded as "mode 10," referred to in some parts of the model as the "new" mode. The thought is that LRT will travel mainly on its own grade-separated right-of-way (ROW), where it does not have to interact with road traffic. By contrast, it is assumed that streetcar will travel mostly in mixed traffic, i.e., it will share its at-grade right-of-way with road traffic. It is believed that AECOM chose to include BRT with streetcar, since although BRT will often include some grade-separated rights-of-way for the trunk-line portion of the route, the beginning and ending of the BRT route are likely to be in mixed traffic, making it more similar to the streetcar.

In cases where a travel demand modeler is coding a new transit line representing a "premium" transit mode,[21] the modeler must add "transit-only" links to the transit network to represent the new service, since the line requires a dedicated ROW which is not part of the highway network. In the past, one would have added these transit-only links to the rail link file (rail_link.bse). However, with the advent of TPB staff using an ESRI geodatabase to manage the highway and transit networks, the rail_link.bse file no longer exists. For a modeler working at COG, one should add transit-only links directly into the highway/transit network geodatabase. For a modeler working external to COG (who will not have access to the COGTools ArcGIS add-in for managing the geodatabase), one should modify the text *.tb files that are output from the create_support_files.s Cube Voyager script.[22]

The "station file" (station.dbf) contains information about transit stations in the modeled area. More formally, the station file contains information about Metrorail stations, commuter rail stations, light rail stations, bus rapid transit stations/stops, streetcar stations/stops, express-bus bus stops, and park-and-ride (PNR) lots that serve these stations/stops. One must add Mode 5 and Mode 10 station nodes to the station file using a mode code of "L" for LRT/Mode 5 and "N" for New/BRT/streetcar/Mode 10. Mode 5 and 10 stations do not require a station centroid number,[23] though the designated node number ranges in recent network documentation has designated the following range (7000-7999).[24] In cases where Mode 10 routes share a street segment with local bus and where one would like the Mode 10 routes to be "combined" in terms of TP+/Voyager computing effective headways, the station node numbers of the Mode 10 routes should be the same as those used for the bus stops on the link(s) of interest.

Transit path building and skimming, mode choice, and transit assignment: LRT, BRT, and streetcar
In transit path building and skimming, mode choice, and transit assignment, the following two rules apply:[25]

Mode 5 is treated like Metrorail (Mode 3)
Mode 10 is treated like local bus (Mode 1)

Fares: LRT, BRT, and streetcar
Fares for Mode 5 and Mode 10 are computed like those for local bus (Mode 1).

11.3.2 Other issues relating to travel modes

Table 52 list the ten transit modes that are handled by the Version 2.3 mode choice model and lists the mode code used in the station file (station.dbf), which is an input to the parker.s script that is part of the transit_skim_all_modes.bat batch file (see Chapter 6). Note that the consolidated station file does not

[21] Such as Metrorail (Mode 3), commuter rail (Mode 4), LRT (Mode 5), and BRT/streetcar (Mode 10).
[22] Meseret Seifu to Files, "Creating a station file," Memorandum, July 11, 2011.
[23] Manish Jain to Ronald Milone and Mark Moran, "MWCOG network coding guide for Nested Logit Model (First draft: September 20, 2007; Updated February 2008 and Oct. 2010)," Memorandum, October 2010, 6.
[24] Robert Snead et al., *FY-2010 Network Documentation: Highway and Transit Network Development: DRAFT* (Washington, D.C.: National Capital Region Transportation Planning Board, June 30, 2010), 86.
[25] Jain to Milone and Mark Moran, "MWCOG network coding guide for Nested Logit Model (First draft: September 20, 2007; Updated February 2008 and Oct. 2010)," 10.

include bus stops, except for bus stops that have their own PNR lot (generally express bus service). Transit routes are represented in Cube Voyager's TRNBUILD module using the LINE command, which is usually placed in a *.LIN file or, using COG/TPB convention, in a MODE*.TB file (a "mode" file).

Table 52 Transit sub-modes represented in the Version 2.3 travel model

Mode #	Transit sub-mode	Mode code in station file
1	Local Metrobus	(not represented in the sta. file)
2	Express Metrobus	B
3	Metrorail	M
4	Commuter rail	C
5	Light rail transit (LRT)	L
6	Other local bus in the WMATA service area	(not represented in the sta. file)
7	Other express bus in the WMATA service area	B
8	Other local bus beyond the WMATA service area	(not represented in the sta. file)
9	Other express bus beyond the WMATA service area	B
10	Bus rapid transit (BRT) and street car	N (for "New" mode)

In addition, there are five non-transit modes that are used to access transit and make transfers to, from, and between transit services. These are detailed in Table 53.

Table 53 Transit Access and Transfer Links

Mode #	Link Type
11	Drive access, for both PNR and KNR (from the zone centroid to a transit stop node)
12	Walk transfer link (between transit services or to/from transit station)
13	Sidewalk link
14	Unused
15	Walk transfer link between PNR lot and transit station
16	Walk access (from the zone centroid to a transit stop node)

All the modes described in Table 52 and Table 53 can be used in the path-building process (see Chapter 6). If no prohibitions are imposed, path building assumes that transfers between all modes are possible. For example, a person could theoretically access Metrorail by driving (mode 11) to the station, use Metrorail (mode 3), and egress Metrorail by driving (mode 11) as well. When trips are in production-attraction format, as is the case for transit path-building and mode choice, a person cannot egress from a station and take a car. To prevent the foregoing behavior in the model, some limitations with regard to transfers need to be imposed. These are described in Table 54. The mode interchanges where transfers are prohibited are denoted by "Y".

Table 54 Transfer Prohibitions (No Transfer or NOX)

From	To Mode															
Mode	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
2	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
3	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
4	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
5	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
6	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
7	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
8	n	n	n	n	n	n	n	n	n	n	Y	n	n	n	Y	n
9	n	n	n	n	n	n	n	n	n	n	Y	n	n	n	Y	n
10	n	n	n	n	n	n	n	Y	Y	n	Y	n	n	n	Y	n
11	n	n	n	n	n	n	n	n	n	n	Y	Y	n	Y	n	n
12	n	n	n	n	n	n	n	n	n	n	Y	Y	n	n	Y	n
13	n	n	n	n	n	n	n	n	n	n	Y	n	n	n	Y	n
14	n	n	n	n	n	n	n	n	n	n	Y	n	n	n	Y	n
15	n	n	n	n	n	n	n	n	n	n	Y	Y	Y	Y	Y	Y
16	n	n	n	n	n	n	n	n	n	n	Y	n	n	n	Y	Y

11.4 Transit access coding

In addition to the expanded set of transit submodes in the mode choice model, the Version 2.3 model includes new transit access coding enhancements which cover five areas:

1. The station file;
2. Sidewalk links and zonal walk links;
3. Zonal auto-access links;
4. Station transfer links; and
5. Zonal percent-walk-to-transit calculations.

11.4.1 Station file

The station file is a dBase file (station.dbf) that contains information about Metrorail stations, commuter rail stations, light rail stations, bus rapid transit stations/stops, street car stations/stops, express-bus bus stops, and park-and-ride lots that serve these stations/stops. Each station file is associated with one scenario, with the most typical scenarios being the "modeled year" (e.g., 2007, 2020, 2040). This file contains information such as:

The mode code, a single-letter code indicating Metrorail (M), commuter rail (C), etc.
A flag indicating whether the station is active in the given year/scenario (Y/N)
A flag indicating whether the station PNR lot is active (Y/N)
Station name

Six new columns/variables were added to the station file that were not present in earlier versions of the regional travel model (e.g., Ver. 2.2 and before). Only the first four of these six variables are currently used:

1. Access distance code (NCT)
2. Parking capacity
3. Peak-period parking cost
4. Off-peak-period parking cost
5. Peak-period shadow price (not used)
6. Off-peak-period shadow price (not used)

The full list of variables in the station file is described in Table 55, with the new variables highlighted with a yellow background.

Table 55 Station File Variables

Name	Type	Field Description
SEQNO	N	Sequence Number
MM	C	Mode Code (M=Metrorail, C=Commuter rail, B=Bus, L=Light rail, N= BRT/streetcar)
NCT	N	Access distance code (1, 2, 3, 0, 9, 8)
STAPARK	C	Parking Available? (Y/N)
STAUSE	C	Station Active? (Y/N)
SNAME	C	Station Name/PNR lot name
STAC	N	Network Centroid (5001-7999)
STAZ	N	TAZ location of Station/PNR lot (1-3722)
STAT	N	Rail Station Node (8000-8999, 9000-9999, 10000-10999)
STAP	N	Parking lot node
STAN1	N	1st Bus Node
STAN2	N	2nd Bus Node
STAN3	N	3rd Bus Node
STAN4	N	4th Bus Node
STAPCAP	N	Parking capacity (number of spaces at the PNR lot)
STAX	N	X coordinate of Station/PNR lot (MD State Plane, NAD83, feet)
STAY	N	Y coordinate of Station/PNR lot (MD State Plane, NAD83, feet)
STAPKCOST	N	Peak period parking cost (daily cost, cents)
STAOPCOST	N	Off-peak parking cost (hourly cost, cents)
STAPKSHAD	N	Peak-period shadow price (currently not used)
STAOPSHAD	N	Off-peak-period shadow price (currently not used)
FIRSTYR	N	Year of Station/PNR lot Opening (unused by scripts, but used as metadata)
STA_CEND	N	Project ID (Metadata)
	C	Scenario name, or left blank (Metadata)
	C	Comments, if any, regarding the file, since file cannot accept comment lines preceding the data lines

Source: (Jain, 2008)

Table 56 shows the mode codes that are used in the station file. "Station centroids" are used to build minimum-impedance paths to all Metrorail and commuter rail stations. In the table below, even though modes 5 and 10 are shown as having a range of numbers designated for station centroids, only Metrorail and commuter rail actually require station centroids..[26]

Table 56 Mode codes used in the consolidated station file/database (station.dbf)

Mode	Mode Code	Station Centroid Range	Station Node Range
Metrorail (Mode 3)	M	5000-5999	8000-8999
Commuter rail (Mode 4)	C	6000-6999	9000-9999
Light rail transit (Mode 5), Bus rapid transit/streetcar (Mode 10)	L, N	7000-7999*	10000-10999
Bus (Modes 1, 2, 6-9)	B	Not used	Not used

*Station centroids are not required for Light rail transit, bus rapid transit/streetcar

11.4.2 Sidewalk links and zonal walk links

In the Version 2.2 travel model and earlier models, there was a walk network (sidewalk network), used for transferring from one transit line to another, in downtown DC and downtown Silver Spring, Maryland. In the Version 2.3 travel model, there is a sidewalk network in almost the entire modeled area. The regional sidewalk network is generated automatically using a script walkacc.s (see p. A-4 of the flowchart in Appendix A). Walkacc.s creates a sidewalk network by converting all suitable highway links into sidewalk links (Mode 13). Examples of highway links that are not converted into sidewalk links include freeways, parkways, and ramps (Facility Type = 1, 5, or 6). In order to limit the size of the sidewalk network to links that are likely used for walking, walkacc.s eliminates sidewalk links from zones where the "percent walk to transit" is zero. There is also a way to supply the program with a list of sidewalk links to be manually added or subtracted to the automated list of sidewalk links. For example, one can manually add a sidewalk link for Memorial Bridge, and one can manually remove sidewalk links that should not exist due to a physical barrier. See Jain (2010)[27] for more details.

Walkacc.s also generates zonal walk-access-to-transit links (Mode 16 links). It automatically sweeps each TAZ, generating walk-access links from the zone centroid to all highway network nodes within a maximum walk distance (See Equation 1).

Equation 1 Maximum walk distance formula, used for generating walk-access-to-transit links

$$(\text{maximum walk distance}) = \sqrt{(\text{zonal area})} * 0.75$$

So, for a small, downtown zone with an area of 0.1 square miles, the program would calculate a maximum walk distance of 0.237 miles and connect all highway network stop nodes that lie within that distance from the zone centroid. There is an absolute maximum of 1.0 mile, which would be obtained

[26] Ibid., 6.
[27] Manish Jain to Ronald Milone and Mark Moran, "MWCOG network coding guide for Nested Logit Model (First draft: September 20, 2007; Updated February 2008 and Oct. 2010)," Memorandum, October 2010, 7.

for zones with a size of 1.78 square miles or greater. The actual calculated (straight-line) distance and computed walk time are stored on each link. No walk-access links are generated for zones with a zero percent walk to transit. Figure 10 shows zonal walk access links and sidewalk links in downtown Washington, D.C., near Farragut Square (TAZ 37, which is in the center of the figure). The thickest gray lines are the TAZ boundaries, which are not part of the actual highway or transit network, but are shown for reference. The lines emanating from each TAZ centroid (dark-blue, when the figure is viewed in color) are the zonal walk access links (Mode 16). The rectilinear (green) lines over many, but not all roads, are the sidewalk links (Mode 13). Mode 13 and 16 links can be shown in Cube Base by adding the four files "support link" files associated with walk-access to transit

- supl??wkam.asc for AM: suplABWKAM.asc, suplBMWKAM.asc, suplCRWKAM.asc, suplMRWKAM.asc
- supl??wkop.asc for off peak: suplABWKOP.asc, suplBMWKOP.asc, suplCRWKOP.asc, suplMRWKOP.asc

If prompted to give a coordinate file for 8,000-series nodes (Metrorail), use the following "support node" file: supnmrwkam.dbf. If prompted to give a coordinate file for 9,000-series nodes (commuter rail), use the following "support node" file: supncrwkam.dbf.

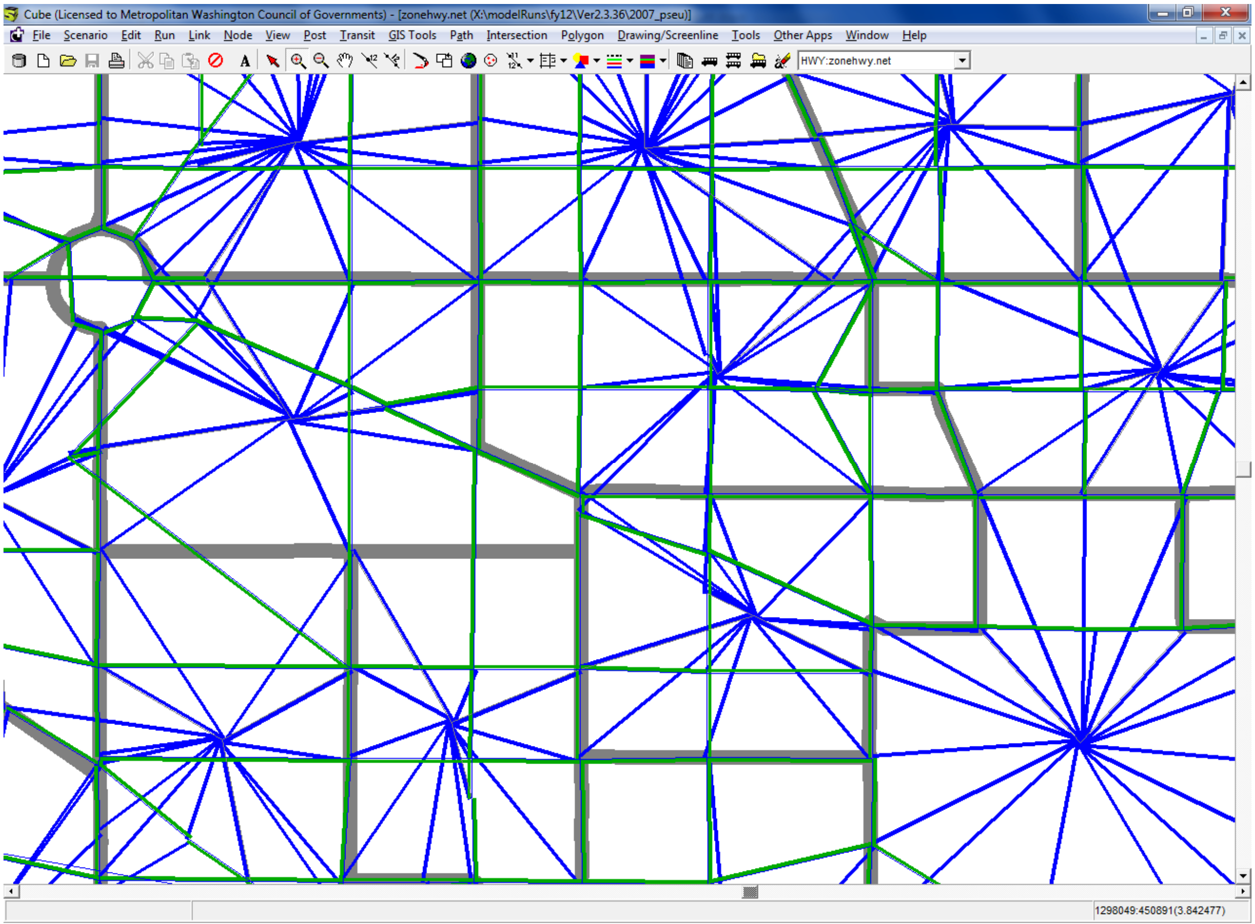


Figure 10 Zonal walk access links and sidewalk links in downtown DC near Farragut Square (Ver. 2.3 NL MC model)

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

11.4.3 Zonal auto-access links

The Cube Voyager script autoacc4.s, originally created by AECOM Consult as a Fortran program and later transferred to Voyager script by TPB staff, is used to generate auto-access-to-transit links. Zonal auto access links are generated by transit mode (Metrorail, commuter rail, light rail, BRT, streetcar, and bus) for both the peak ("AM") and off-peak ("mid-day") time periods. Auto access links (Mode 11) are a function of multiple criteria:

- Orientation toward downtown (defined as TAZ 35, centered at 20th and F Streets, NW)
- A backtracking penalty and a prohibition of crossing the Potomac River (except for trips from Loudoun County to MARC commuter rail);
- A maximum link distance, which is a function of station type (e.g., terminal vs. non-terminal) and transit mode;
- Manually specified overrides; and
- Distances based on the highway skims from the highway network that includes dummy centroids representing Metrorail and commuter rail stations.

The "access distance code," known as "NCT" in the autoacc4.s script, is a newly added variable in the station file that controls the number, extent, and directionality of PNR/KNR access links generated for each parking lot (in the case of PNR) or each station (in the case of KNR). Table 57 and Table 58 describe the meaning of each of the six access distance codes for Metrorail and commuter rail, respectively.

Table 57 Metrorail access distance codes and their meaning for the autoacc4.s program

Acc Dist Code	Interpretation
1	End-of-the-line station (e.g., Shady Grove Metro)
2	Intermediate station (e.g., Rockville Metro)
3	PNR close to a CBD (e.g., Rhode Island Ave. Metro, Fort Totten)
0	Only KNR-access links generated (e.g., Braddock Road, National Airport, Clarendon)
9	Metrorail sta. in use, but no PNR/KNR access (e.g., Dupont Circle, Farragut North, Metro Ctr.)
8	Pentagon Metro Sta., allows for very long KNR links, to represent "slugging" (informal carpool)

Table 58 Commuter rail access distance codes and their meaning for the autoacc4.s program

Acc Dist Code	Interpretation
0	Commuter rail station shares the parking lot with a Metrorail station or commuter rail station parking lot has fewer than 500 parking spaces
1	End-of-the-line station
2	Commuter rail station parking lot has more than 500 parking spaces unless it is an end-of-the-line station

The access distance code, along with the transit mode, determines the maximum link distance for the drive-access-to-transit links generated by autoacc4.s for the TPB nested-logit mode choice model. The maximum link distances for PNR are shown in Table 59. Although not shown in the table, the maximum allowed link distance for KNR links is 3 miles. It is also important to note that the KNR links are generated to Metrorail stations, light rail stations, streetcar stops, and bus stops with parking lots, but not commuter rail stations.

Table 59 Maximum link distances for drive-access-to-transit links: Ver. 2.3 NL MC model

Mode	Access Dist. Code	Maximum Connect. Length (miles)
Metrorail station PNR	1	15
Metrorail station PNR	2	5
Metrorail station PNR	3	3
Metrorail station PNR	0	3
Commuter rail station PNR	1	15
Commuter rail station PNR	2	10
Commuter rail station PNR	0	5
Bus PNR	1	5
Bus PNR	0	3
BRT/Street car PNR	1	5
BRT/Street car PNR	0	3
LRT PNR	1	5
LRT PNR	0	3

Ref: I:\ateam\meetings_conf\transitModelingGroup\2007-11-07\maxDistForAutoAccConnect.xls

Figure 11 shows kiss-and-ride (KNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia.

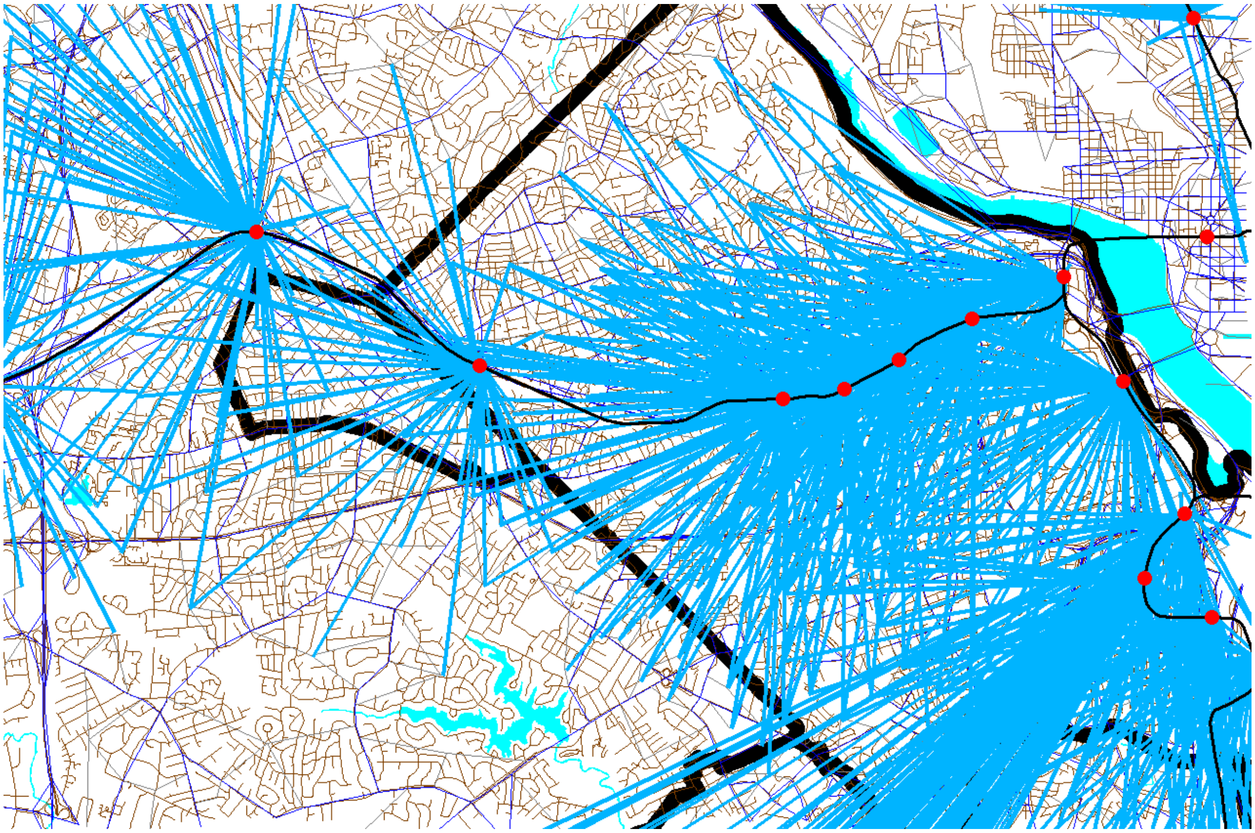


Figure 11 Kiss-and-ride (KNR) auto access links to Metrorail stations in Northern Virginia

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

Figure 12 shows park-and-ride (PNR) auto-access-to-transit links for the AM period associated with Metrorail stations in Northern Virginia. Notice that the Orange Line stations from Clarendon to Rosslyn do not have PNR-access links, since they do not have PNR lots. By contrast, these stations do have KNR-access links, since these stations can have KNR access. The Pentagon Metrorail station is another example of a station where the model does not allow travelers to have PNR access, but they may have KRN access. Notice that the KNR-access links and PNR-access links are not shaped like a circular "starburst," but are somewhat flattened, due to the backtracking penalty. This was done to mimic the behavior of travelers who tend not to want to backtrack when driving to park at or be dropped off at a Metrorail or commuter rail station.

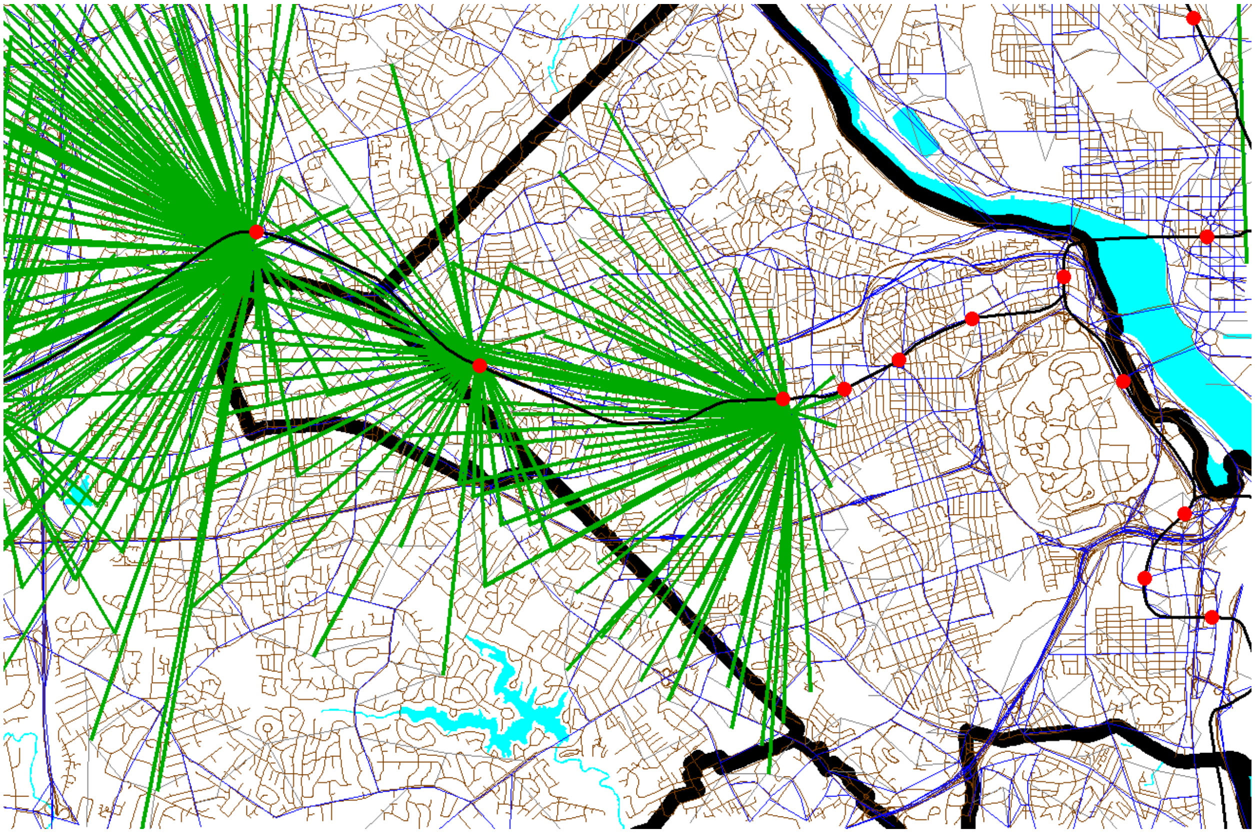


Figure 12 Park-and-ride (PNR) auto access links to Metrorail stations in Northern Virginia

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

11.4.4 Station transfer links

Station transfer links are walk links connecting:

Stations and sidewalks (Mode 12)
Stations and bus service (Mode 12)
Stations and PNR lots (Mode 15)

These links are generated automatically from data in the station file. For PNR-station transfer links, the walk time is a function of parking capacity and cost.[28] The Mode 15 links are generated by the script parker.s (see page A-4 of the flowchart in Appendix A). The following files contain the mode 15 links:

```
busampnr.tb
busoppnr.tb
comampnr.tb
comoppnr.tb
lrtampnr.tb
```

[28] Ibid., 6.

```
lrtoppnr.tb
metampnr.tb
metoppnr.tb
newampnr.tb
newoppnr.tb
```

The mode 12 links are developed manually using the TPBMAN geodatabase. These links can be found in the following files (see page A-4 of the flowchart in Appendix A):

```
Com_Bus.tb
LRT_bus.tb
Met_Bus.tb
NEW_bus.tb
```

11.4.5 Zonal percent walk to transit calculations

The zonal percent walk is the percent of a zone's area that lies within walking distance to transit service (i.e., a transit stop node, such as a bus stop or rail station). A short walk is defined as one that is less than or equal to 0.5 miles and a long walk is defined as one that is less than or equal to one mile. The following walk designations are used:

Short walk to Metrorail (<= 0.5 miles);
Long walk to Metrorail (>0.5 and <= 1.0 miles);
Short walk to AM transit;
Long walk to AM transit;
Short walk to off-peak transit;
Long walk to off-peak transit.

These walk-to-transit areas are sometimes called transit walksheds. TPB staff has developed an automated GIS procedure, known as the "walkshed generator," which both creates the needed walksheds and then calculates the associated percent walk to transit values.[29] This procedure creates point buffers around transit stop nodes and then overlays these point buffers with TAZ boundaries. Using the walksheds, the procedure then calculates the six walk percent values needed by the model. For example, Figure 13 shows an example of the short- and long-walk buffers generated around Metrorail stations. Before running the "walkshed generator," it is useful to create a series of transit stop node shapefiles that are used by the walkshed generator.[30]

[29] Yew Yuan, *Walkshed Generator User Guide* (Washington, D.C.: National Capital Region Transportation Planning Board, August 25, 2010).
[30] Mark S. Moran, "Creating transit stop node shapefiles needed as inputs to the TPB procedure for generating transit walksheds and calculating the share of each zone within walking distance to transit," Memorandum, September 6, 2011.

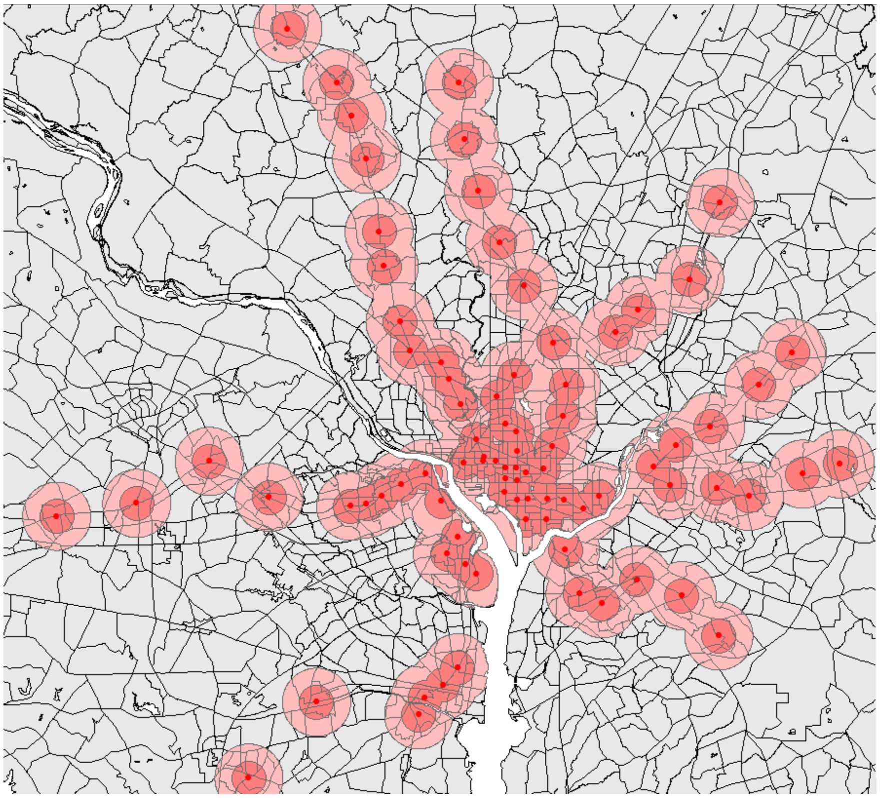


Figure 13 Short- and long-walk buffers around Metrorail stations

11.5 Transit path-building procedures

Given the segmentation in the model, 24 separate transit paths can be enumerated between each production zone and attraction zone:

Three modes of access to transit

1. Walk
2. Park and ride (PNR driver)
3. Ride to transit/KNR (drop-off/pick-up, or ride with a PNR driver)

Four transit modes/combinations

4. Commuter rail (alone and in combination with bus and/or Metrorail)
5. Bus-Metrorail (bus and Metrorail used in combination)
6. All bus (buses only)
7. All Metrorail (Metrorail only)

Two time-of-day periods

8. Peak (represented by transit service in the AM peak hour)
9. Off-peak (represented by transit service in the five-hour midday period)

However, at present, PNR and KNR to commuter rail are combined as a single path, since, for commuter rail, the PNR- and KNR-access links are identical. Consequently, the number of transit paths built between each production/attraction zone pair is 22. Table 60 summarizes the paths and available transit sub modes in each path. Again, in this figure, "drive to commuter rail" and "KNR to commuter rail" are combined into one category.

Run times for transit routes are controlled by the RUNTIME keyword (TRNBUILD) and output bus IVT skims are adjusted to reflect the general level of road congestion using a factor table (LBUS_TimFtrs.ASC). As stated previously, path weights are consistent with the weights used in the mode choice model:

- Drive access time: Equal to 1.5 times the in-vehicle time
- Walk access time: Equal to 2.0 times the in-vehicle time
- Other out-of-vehicle time: Equal to 2.5 times the in-vehicle time

Headway combination between two or more transit routes is allowed to occur provided 1) the routes share the same transit mode code and 2) the difference between the run time and the minimum run time is less than a designated number of minutes (5 minutes for AM and 10 minutes for off peak). A maximum path time is set at 360 weighted minutes. There is no weighting of in-vehicle time by transit sub-modes (i.e., all transit modes have an IVT weight of 1.0). The maximum initial wait time for all ten transit modes is set at 60 perceived minutes. The minimum transfer wait time is 4.0 minutes for bus (Modes 1, 2, 6, 7, 8), 0 minutes for Metrorail (Mode 3), 4.0 minutes for commuter rail (Mode 4), 0 minutes for LRT (Mode 5), 10.0 minutes for express bus (Mode 9), and 4.0 minutes for Mode 10 (streetcar and/or BRT).

Table 60 Path-specific parameters used in transit path building

		Transit Submodes			
Path	**Path Parameter**	**Comm Rail**	**Express Bus**	**Local Bus**	**Metrorail**
Walk-to-Commuter Rail	Modes Available	X		X	X
	Weight	1.0		1.0	1.0
	Path Testing	must appear		can appear	can appear
Walk-to-Bus/Metrorail	Modes Available		X	X	X
	Weight		1.0	1.0	1.0
	Path Testing		either must appear		must appear
Walk-to-Bus	Modes Available		X	X	
	Weight		1.0	1.0	
	Path Testing		either must appear		
Walk-to-Metrorail	Modes Available				X
	Weight				1.0
	Path Testing				must appear
Drive-to-Commuter Rail	Modes Available	X		X	X
	Weight	1.0		1.0	1.0
	Auto access links to	CRsta. w/ parking		no	no
	Path Testing	must appear		can appear	can appear
K&R-to-Commuter Rail	Modes Available	X		X	X
	Weight	1.0		1.0	1.0
	Auto access links to	CRsta. w/ parking		no	no
	Path Testing	must appear		can appear	can appear
Drive-to-Bus/Metrorail	Modes Available		X	X	X
	Weight		1.0	1.0	1.0
	Auto access links to		all Bus park-ride lots		MRsta. w/ parking
	Path Testing		either must appear		must appear
K&R-to-Bus/Metrorail	Modes Available		X	X	X
	Weight		1.0	1.0	1.0
	Auto access links to		all Bus park-ride lots		all MRsta.
	Path Testing		either must appear		must appear
Drive-to-Bus	Modes Available		X	X	
	Weight		1.0	1.0	
	Auto access links to		all Bus park-ride lots		MRsta. w/ parking
	Path Testing		either must appear		
K&R-to-Bus	Modes Available		X	X	
	Weight		1.0	1.0	
	Auto access links to		all Bus park-ride lots		all MRsta.
	Path Testing		either must appear		
Drive-to-Metrorail	Modes Available				X
	Weight				1.0
	Auto access links				MRsta. w/ parking
	Path Testing				must appear
K&R-to-Metrorail	Modes Available				X
	Weight				1.0
	Auto access links				all MRsta.
	Path Testing				must appear

Source: AECOM Consult, Inc.[31]

[31] AECOM Consult, Inc., *Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User's Guide* (AECOM Consult, Inc., March 2005).

11.6 Treatment of Parking Costs and Terminal Times

11.6.1 Parking costs

In applying the Version 2.3 model, prior to the execution of the mode choice model, a Voyager script (prefarv23.s) is used to generate zonal files containing zonal parking costs and highway terminal times (the time to park and "un-park" a vehicle). The files are, in turn, read into the mode choice model upon execution. The Version 2.3 model includes a new parking cost model estimated based on the 2007/2008 HTS.[32] HBW trip purpose utilizes the daily parking rate, while all other purposes use the hourly parking rate. Thus, two separate parking cost models were estimated, one for daily rates and one of hourly rates. For the daily rates model, the observed data indicated that it is rare for a traveler to incur parking costs in area types 4 and above, thus the model was estimated only for area types 1-3. A daily parking cost was estimated to be:

Equation 2 Daily Parking Cost for Area Types 1-3

$$\text{Parking cost} = 2.1724 * \ln(\text{floating employment density}) - 15.533$$

The resulting parking costs are also shown in Figure 14.

[32] Mary Martchouk to Mark S. Moran, "Developing a Parking Cost Model for Automobile Modes in the Version 2.3 Travel Model," Memorandum, June 14, 2010.

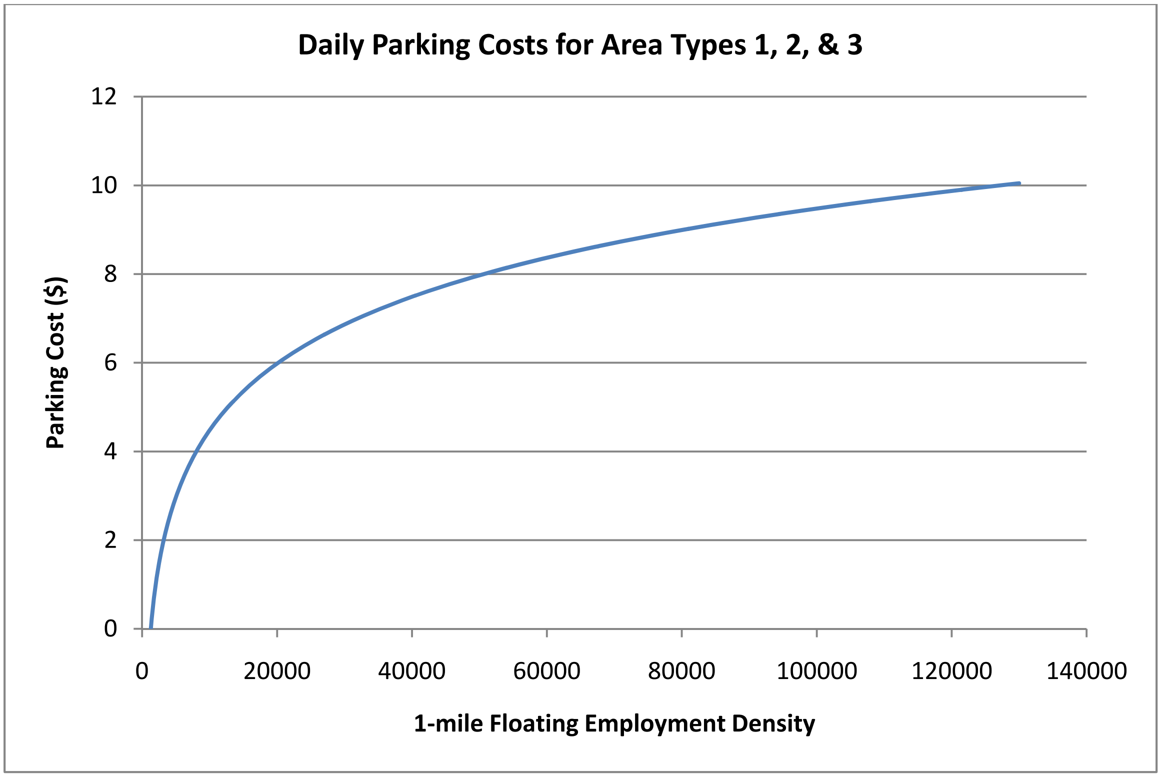


Figure 14 Daily Parking Cost Model for the Version 2.3 Model

Ref: I:\ateam\docum\FY11\Ver2.3\modelDoc_v3\02_userGuide\parking_scatterplots.xlsx

For hourly rates, there was insufficient data to estimate a reliable model. Thus, a decision was made to assign a flat rate based on the prevalent metered rates for each area type. For area type 1, the most prevalent metered rate of parking is $2.00 and thus that value was selected. For area type 2, the average hourly parking cost is anticipated to be $1.00. For area type 3, the value of $0.25 per hour was selected. For area types 4 and higher, no parking cost was anticipated.

11.6.2 Highway terminal time assumptions

Highway terminal time is typically associated with the average time spent parking or "un-parking" an automobile. The current mode choice model application program considers highway terminal time only at the attraction end. Highway time is calculated as a function of employment density, as shown in Table 61.

Table 61 Highway terminal time as a function of employment density

Employment density range (Emp/Sq. Mi.)	Highway terminal time (minutes)
0 - 4,617	1
4,618 - 6,631	2
6,632 - 11,562	4
11,563 - 32,985	6
32,986 +	8

11.7 Auto Operating Costs

The auto operating cost in the mode choice model relate to out-of-pocket expenditures directly associated with the requirements of an automobile trip, including fuel, oil, maintenance, tire wear, etc. (auto ownership costs including insurance, registration fees, etc. are not included). The mode choice model expresses operating costs as a per-mile rate (2007 cents) that is specified as a parameter in the nested-logit mode choice model control files. We are currently using 10 cents per mile and this rate is not varied over time (i.e., the auto operating cost for 2002 and 2030 are both assumed to be 10 cents per mile).

Chapter 12 Time-of-Day Processing

12.1 Overview

The time of day process is applied to convert daily vehicles among the four modeled time periods, prior to being assigned to the network. The process is applied with the Time-of-Day.s and Misc_Time-of-Day.s scripts. The Prepare_Trip_Tables_for_Assignment.s script is used to combine the various trips by time period into combined trip tables for the traffic assignment process. The input and output files are listed in . and Table 62.

Daily Auto Driver Trips, by Occupancy Levels	HBW<ITER>.ADR, HBS<ITER>.ADR, HBO<ITER>.ADR, NHW<ITER>.ADR, NHO<ITER>.ADR	Binary
Daily Miscellaneous and Truck Trips (From the \Inputs subdirectory)	VISI.ADR, TAXI.ADR, SCHL.ADR, AIRPAX.ADR, XXCVT.VTT, XXAUT.VTT,	Binary
Truck and commercial vehicle trip tables	MTK<ITER>.PTT, HTK<ITER>.PTT, COM<ITER>.PTT	Binary
Adjustment or 'delta' trip tables used for commercial and truck models	CVDelta_3722.trp TKDelta_3722.trp	Binary
Time of Day Percent File by Purpose, Mode, and Direction	todcomp_2008HTS.dbf	DBF

Note: <ITER> =PP, i1…i4

Table 62 Outputs of time-of-day process

Trip Tables by Time Period	AM<ITER>.ADR, MD<ITER>.ADR, PM<ITER>.ADR, NT<ITER>.ADR,	Binary
Miscellaneous Time-of-Day Files	MISCAM<ITER>.TT, MISCMD<ITER>.TT, MISCPM<ITER>.TT, MISCNT<ITER>.TT	Binary
Total Vehicle Trip by 6 Markets T1 – SOVs, Commercial vehicles T2 – HOV- 2 occ. vehicles T3 – HOV- 3+occ. Vehicles T4- Medium trucks T5- Airport passenger vehicles T6-Heavy trucks	<ITER>AM.VTT, <ITER>MD.VTT, <ITER>PM.VTT, <ITER>NT.VTT	Binary

Chapter 13 Traffic Assignment

Table 63 Inputs to traffic assignment process

Volume delay parameters and freeflow speed assumptions	support\hwy_assign_Conical_VDF.s support\hwy_assign_capSpeedLookup.s	Script block
Total Vehicle Trip by 6 Markets	<ITER>AM.VTT, <ITER>MD.VTT, <ITER>PM.VTT, <ITER>NT.VTT	Binary
Toll minutes equivalence file	support\toll_minutes.txt	Text
AM Toll Factors by Vehicle Type	Inputs\AM_Tfac.dbf	DBF
Midday Toll Factors by Vehicle Type	Inputs\MD_Tfac.dbf	DBF
PM Toll Factors by Vehicle Type	Inputs\PM_Tfac.dbf	DBF
Night Toll Factors by Vehicle Type	Inputs\NT_Tfac.dbf	DBF
Network files	ZONEHWY.NET, <ITER>HWY.NET	Binary

Note: <ITER> =PP, i1…i4

Table 64 Outputs of traffic assignment process

Loaded Links Files by Time Period	<ITER>AMLLNK.ASC, ITER>MDLLNK.ASC, <ITER>PMLLNK.ASC, <ITER>NTLLNK.ASC	Text
Loaded Highway Network	Assign_output_<ITER>.net	Binary

Note: <ITER> =PP, i1…i4, PP= AM & OP

Table 65 Final Iteration Loaded Highway Network (Assign_output_i4.net) Variables Description

Variable Name	Description
A	A-Node
B	B-Node
DISTANCE	Link Distance in miles (x.xx)
SPDC	(Not used)
CAPC	(Not used)
JUR	Jurisdiction Code (0-23) *0/dc, 1/mtg, 2/pg, 3/alr/, 4/alx,5, ffx, 6/ldn, 7/ pw, 8/(unused), 9/ frd, 10/how, 11/aa, 12/chs, 13/(unused), 14/car, 15/cal, 16/stm, 17/ kg, 18/fbg, 19/stf, 20/spts, 21/fau, 22/clk, 23/jef*
SCREEN	Screenline Code (1-38)
FTYPE	Link Facility Type Code (0-6)
	0/Centroids, 1/Freeways, 2/Major Art., 3/Minor Art, 4/ Collector, 5/ Expressway, 6/ Ramp
TOLL	Toll Value in current year dollars
TOLLGRP	Toll Group Code (1-9999)
<Period> LANE	<Period> No. of Lanes
<Period>LIMIT	<Period> Limit Code (0-9)
EDGEID	Geometry network link identifier
LINKID	Logical network link identifier
NETWORKYEA	Planning year of network link
SHAPE_LENG	Geometry length of network link (in feet)
PROJECTID	Project identifier
TAZ	Nearest TAZ centroid to midpoint of link (1-3,722)
ATYPE	Area Type (1-6)
SPDCLASS	Speed Class
CAPCLASS	Capacity Class
DEFLATIONFTR	Factor for deflating current year tolls to constant year tolls
<Period>TOLL	<Period> Toll Value in current year dollars - all tolled facilities
<Period>TOLL_VP	<Period> Toll Value in current year dollars - Variably priced tolled facilities only
<Period> HTIME	<Period> Highway Time - based on initial highway lookup speeds
I4<Period>SOV	Iteration 4 <Period> assigned SOV Volume
I4<Period>HV2	Iteration 4 <Period> assigned HOV2 Volume
I4<Period>HV3	Iteration 4 <Period> assigned HOV3 Volume
I4<Period>CV	Iteration 4 <Period> assigned Commercial Vehicle Volume
I4<Period>TRK	Iteration 4 <Period> assigned Truck Volume
I4<Period>APX	Iteration 4 <Period> assigned Airport Passenger Volume
I4<Period> VOL	Iteration 4 <Period> assigned Volume
I4<Period>VMT	Iteration 4 <Period> Vehicle Miles Travelled (VMT)
I4<Period>FFSPD	Iteration 4 <Period> free flow speed (mph)
<Period>HRLKCAP	<Period> hourly link capacity
<Period>HRLNCAP	<Period> hourly lane capacity
I4<Period>VC	Iteration 4 <Period> Volume Capacity ratio
I4<Period>VDF	Iteration 4 <Period> Volume Delay function
I4<Period>SPD	Iteration 4 <Period> Speed (mph)
I424VOL	Iteration 4 Daily (24 hour) Volume Capacity ratio

KEY	
<Period>= AM	AM Peak Period (6:00-9:00 AM)
MD	Mid Day (9:00 AM - 3:00 PM)
PM	PM Peak Period (3:00 - 7:00 PM)
NT	All remaining hours

Table 66 Time Valuation (Minutes/2007$ by Vehicle Type and Time Period

	Equivalent Minutes per Dollar			
Mode	AM Peak	Midday	PM Peak	Night
SOV	2.5	3.0	3.0	3.0
HOV 2-occupant auto	1.5	4.0	2.0	4.0
HOV 3+occupant auto	1.0	4.0	1.0	4.0
Light duty commercial vehicle	2.0	2.0	2.0	2.0
Truck	2.0	2.0	2.0	2.0
Auto serving airport passenger	2.0	2.0	2.0	2.0

(Time_Valuation_V2.3.xls)

Chapter 14 Transit Assignment

14.1 Transit assignment process

Transit assignment is a new feature of the Version 2.3 Travel Model that was not part of the Version 2.2 Travel Model. Transit assignment is where transit trips are loaded on to the transit network. Although highway and transit assignment have some similarities, it is useful to point out some of the differences between these two assignment procedures. First, whereas highway assignment is done with trip tables in origin/destination (O/D) format, transit assignment is done with trip tables in production/attraction (P/A) format. Second, whereas highway assignment is capacity constrained, transit assignment is not. Lastly, whereas highway assignment is done in each of the five speed feedback loops (i.e., pump prime, i1, i2, i3, and i4), transit assignment is conducted only at the conclusion of the i4 speed feedback loop (See Figure 2). Procedures for transit assignment are shown on pages A-14 through A-16 in the flowchart in Appendix A. The transit assignment is run using the transit_assignment.bat batch file, which is called from the "run model steps" batch file. The transit assignment process is run in the standard scenario/output folder (e.g., X:\modelRuns\fy12\Ver2.3.36\2007_pseu). This batch file in turn calls five Cube Voyager scripts, which are described in more detail below. The inputs to the transit_assignment.bat batch file are shown in Table 67 and the outputs are shown in Table 68.

Table 67 Inputs to transit assignment process

Trip tables segmented by mode (coming from the mode choice model)	Inputs\<ITER>_HBW_NL_MC.MTT Inputs\<ITER>_HBS_NL_MC.MTT Inputs\<ITER>_HBO_NL_MC.MTT Inputs\<ITER>_NHB_NL_MC.MTT	Binary
Local bus future time degradation factors	Inputs\LBus_TimFTRS.ASC	Text
Highway network	Zonehwy.net	Binary
AM peak transit lines	Inputs\MODE1AM…MODE10AM.TB	Binary
Off peak transit lines	Inputs\MODE1OP…MODE10OP.TB	Binary
Transit network files	met_node.tb, met_bus.tb, met_link.tb, com_node.tb, com_bus.tb, com_link.tb, lrt_node.tb, lrt_bus.tb, lrt_link.tb new_node.tb, new_bus.tb, new_link.tb met_pnrn.tb, com_pnrn.tb, bus_pnrn.tb, lrt_pnrn.tb, new_pnrn.tb met[AM|OP]pnr.tb, com[AM|OP]pnr.tb, bus[AM|OP]pnr.tb, lrt[AM|OP]pnr.tb, new[AM|OP]pnr.tb mrpr[AM|OP].asc, cr[AM|OP].asc, buspr[AM|OP].asc, lrt [AM|OP].asc, new[AM|OP].asc mrkr[AM|OP].asc, buskr[AM|OP].asc, lrtkr[AM|OP].asc, newkr[AM|OP].asc met_bus.tb, com_bus.tb, lrt_bus.tb, new_bus.tb	Text
Transit network walk links	walkacc.asc, sidewalk.asc	Text

Note: <ITER> =PP, i1…i4

Table 68 Outputs of transit assignment process

Combined transit trip file	<ITER>_<Prd>MS.TRP	Binary
Transit assignment node file	<ITER>_*<AA><??><Prd>*node.dbf	DBF
Transit assignment Link file	<ITER>_*<AA><??><Prd>*link.dbf	DBF
Support links	Supl<??>*<AA><Prd>*.asc	Text

Note: <ITER> =PP, i1…i4, <AA>= WK, DR, KR ??= CR, MR, AB, BM, Prd=AM, OP

The transit assignment is done for two time-of-day periods: the peak period and the off peak period. The peak period is represented by the three-hour AM peak period. The off-peak period is represented by the five-hour midday period. Thus, when calculating peak-period travel times on transit ("skims") the AM peak period is used to represent the level of service in both the AM and PM peak period. Similarly, when calculating the average headway and average run time for each transit route, these calculations are done for the peak period (represented by the AM peak) and the off-peak period (represented by the midday period). It is assumed that the majority of HBW trips will occur in the peak periods and that the majority of non-work trips will occur in the off-peak periods. Consequently, prior to the actual transit assignment, the five trip tables coming out of mode choice are combined into two tables: one for the peak period and one for the off-peak period. The peak-period trip table ("AM") contains only one trip table (HBW). By contrast, the off-peak period trip table ("OP") contains the trip tables from the other four trip purposes (HBS, HBO, NHW, NHO) as shown in Table 69.

Table 69 Mapping/concatenation of trip tables by trip purposes into peak and off-peak period trip tables prior to transit assignment

Before combining trip tables	After combining trip tables
i4_HBW_NL_MC.MTT	i4_AMMS.TRP
i4_HBS_NL_MC.MTT i4_HBO_NL_MC.MTT i4_NHw_NL_MC.MTT i4_NHO_NL_MC.MTT	i4_OPMS.TRP

This is mapping/concatenation of trip tables done with the Cube Voyager script Combine_Tables_For_TrAssign.s script. There are 11 tables on the *.TRP files, not 12, since, for commuter rail, KNR and PNR are combined:

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

There are four transit assignment scripts, one for each transit submode (commuter rail, Metrorail, all bus, and bus/Metrorail):

```
transit_assignment_CR.s
transit_assignment_MR.s
transit_assignment_AB.s
transit_assignment_BM.s
```

14.1.1 Inputs to the transit assignment

As can be seen on page A-14 of Appendix A, the specific list of inputs for transit assignment varies for each of the four transit submodes.

14.1.2 Outputs of the transit assignment

The output of the four transit assignment scripts are a series of transit link files and transit node files in dBase (DBF) format. These files are generated in Cube Voyager's TRNBUILD module using the LINKO and NODEO keywords. The transit node files (NODEO) simply contain the node number and its X and Y coordinates, as shown in Figure 15.

	A	B	C
1	N	X	Y
2	1	1298543	446898
3	2	1298807	445281
4	3	1297889	443318
5	4	1296811	441898
6	5	1303089	442174
7	6	1301409	443113
8	7	1299596	445914
9	8	1301916	446878
10	9	1302004	445336
11	10	1302622	443982
12	11	1303826	443797
13	12	1305207	444137
14	13	1303781	445659
15	14	1304865	446730

Figure 15 Excerpt from one of the transit node DBF files output from transit assignment (i4_WKMRAMnode.dbf)

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\i4_WKMRAMnode.dbf"

Transit link files (LINKO) files include the following attributes:[33]

- A: A-node of link
- B: B-node of link
- TIME: A-B time (hundredths of minutes)
- MODE: Mode of link (1-255)
- COLOR: User designated drawing color
- STOP_A: 1 = A is a stop node
- STOP_B: 1 = B is a stop node
- DIST: A-B distance
- NAME: Name of line on this link
- FREQ: Service frequency (min)
- PLOT: Always = 0

The following additional attributes are included due to transit assignment:

- SEQ: Link sequence in the line
- OWNER: Line owner (first ten characters)
- AB_VOL: Volume
- AB_BRDA: Number of trip boardings at A
- AB_XITA: Number of exits at A

[33] Citilabs, Inc., "Cube Voyager Reference Guide, Version 5.1.2," 916.

- AB_BRDB: Number of boardings at B
- AB_XITB: Number of exits at B
- (last 5 variables are also repeated for B-A direction)

Some examples of the LINKO attribute values can be found in Figure 16 through Figure 18. For example, Figure 16 shows a portion of the AM walk-access to Metrorail LINKO file (i4_WKMRAMlink.dbf) that has mode-16 links (walk access to transit). Similarly, Figure 17 shows a portion of the AM walk-access to Metrorail LINKO file (i4_WKMRAMlink.dbf) that has mode-3 links (Metrorail line segments). Lastly, Figure 18 shows a portion of the AM walk-access to Metrorail LINKO file (i4_WKMRAMlink.dbf) that has mode-12 links (walk transfer links).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	A	B	TIME	MODE	FREQ	PLOT	COLOR	STOP_A	STOP_B	DIST	NAME	SEQ	OWNER	AB_VOL	AB_BRDA	AB_XITA	AB_BRDB	AB_XITB	BA_VOL	BA_BRDA	BA_XITA	BA_BRDB	BA_XITB
2	1	20263	280	16	0.00	0	6	0	0	14	*16	0		0	0	0	0	0	0	0	0	0	0
3	1	20266	200	16	0.00	0	6	0	0	10	*16	0		0	0	0	0	0	0	0	0	0	0
4	1	20269	180	16	0.00	0	6	0	0	9	*16	0		0	0	0	0	0	1344	0	0	0	0
5	1	20341	300	16	0.00	0	6	0	0	15	*16	0		0	0	0	0	0	0	0	0	0	0
6	1	20344	240	16	0.00	0	6	0	0	12	*16	0		0	0	0	0	0	0	0	0	0	0
7	1	20346	300	16	0.00	0	6	0	0	15	*16	0		0	0	0	0	0	0	0	0	0	0
8	1	20442	60	16	0.00	0	6	0	0	3	*16	0		0	0	0	0	0	0	0	0	0	0

Figure 16 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-16 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\i4_WKMRAMlink.dbf"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	A	B	TIME	MODE	FREQ	PLOT	COLOR	STOP_A	STOP_B	DIST	NAME	SEQ	OWNER	AB_VOL	AB_BRDA	AB_XITA	AB_BRDB	AB_XITB	BA_VOL	BA_BRDA	BA_XITA	BA_BRDB	BA_XITB
2	8001	8002	406	3	6.00	0	0	1	1	261	WMREDA	1	WMATA;SHAD	601	601	0	813	28	138	0	138	9	568
3	8002	8001	406	3	6.00	0	0	1	1	261	WMREDA-	26	WMATA;SHAD	138	9	568	0	138	601	813	28	601	0
4	8002	8003	329	3	6.00	0	0	1	1	213	WMREDA	2	WMATA;SHAD	1385	813	28	970	93	697	9	568	54	994
5	8003	8002	329	3	6.00	0	0	1	1	213	WMREDA-	25	WMATA;SHAD	697	54	994	9	568	1385	970	93	813	28
6	8003	8004	308	3	6.00	0	0	1	1	109	WMREDA	3	WMATA;SHAD	2263	970	93	593	89	1637	54	994	54	940
7	8004	8003	308	3	6.00	0	0	1	1	109	WMREDA-	24	WMATA;SHAD	1637	54	940	54	994	2263	593	89	970	93
8	8004	8005	203	3	6.00	0	0	1	1	135	WMREDA	4	WMATA;SHAD	2767	593	89	814	8	2523	54	940	143	47
9	8005	8004	203	3	6.00	0	0	1	1	135	WMREDA-	23	WMATA;SHAD	2523	143	47	54	940	2767	814	8	593	89
10	8005	8006	305	3	6.00	0	0	1	1	219	WMREDA	5	WMATA;SHAD	3573	814	8	344	213	2427	143	47	50	825
11	8005	8006	306	3	6.00	0	0	1	1	219	WMREDB	1	WMATA;GROS	801	801	0	337	63	44	0	44	1	697
12	8006	8005	305	3	6.00	0	0	1	1	219	WMREDA-	22	WMATA;SHAD	2427	50	825	143	47	3573	344	213	814	8
13	8006	8005	306	3	6.00	0	0	1	1	219	WMREDB-	19	WMATA;GROS	44	1	697	0	44	801	337	63	801	0
14	8006	8007	201	3	6.00	0	0	1	1	102	WMREDA	6	WMATA;SHAD	3704	344	213	1362	383	3202	50	825	266	1886
15	8006	8007	201	3	6.00	0	0	1	1	102	WMREDB	2	WMATA;GROS	1074	337	63	1329	117	740	1	697	62	1656
16	8007	8006	201	3	6.00	0	0	1	1	102	WMREDA-	21	WMATA;SHAD	3202	266	1886	50	825	3704	1362	383	344	213
17	8007	8006	201	3	6.00	0	0	1	1	102	WMREDB-	18	WMATA;GROS	740	62	1656	1	697	1074	1329	117	337	63
18	8007	8008	308	3	6.00	0	0	1	1	170	WMREDA	7	WMATA;SHAD	4683	1362	383	1739	169	4822	266	1886	401	845
19	8007	8008	309	3	6.00	0	0	1	1	170	WMREDB	3	WMATA;GROS	2286	1329	117	1706	97	2334	62	1656	254	771
20	8008	8007	308	3	6.00	0	0	1	1	170	WMREDA-	20	WMATA;SHAD	4822	401	845	266	1886	4683	1739	169	1362	383
21	8008	8007	309	3	6.00	0	0	1	1	170	WMREDB-	17	WMATA;GROS	2334	254	771	62	1656	2286	1706	97	1329	117

Figure 17 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-3 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\i4_WKMRAMlink.dbf"

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	A	B	TIME	MODE	FREQ	PLOT	COLOR	STOP_A	STOP_B	DIST	NAME	SEQ	OWNER	AB_VOL	AB_BRDA	AB_XITA	AB_BRDB	AB_XITB	BA_VOL	BA_BRDA	BA_XITA	BA_BRDB	BA_XITB
256	8001	22395	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	138	0	0	0	0	601	0	0	0	0
257	8002	9005	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	0	0	0	0	0	0	0	0	0	0
258	8002	22351	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	160	0	0	0	0	236	0	0	0	0
259	8002	22370	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	436	0	0	0	0	586	0	0	0	0
260	8003	22344	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	542	0	0	0	0	776	0	0	0	0
261	8003	22672	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	545	0	0	0	0	248	0	0	0	0
262	8004	22332	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	830	0	0	0	0	613	0	0	0	0
263	8004	22670	20	12	0.00	0	2	0	0	1	*12	0	WMATA;SHAD	199	0	0	0	0	34	0	0	0	0
264	8005	22327	20	12	0.00	0	2	0	0	1	*12	0	WMATA;GROS	99	0	0	0	0	1757	0	0	0	0

Figure 18 Excerpt from one of the transit link DBF files output from transit assignment (i4_WKMRAMlink.dbf) showing mode-12 links

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\i4_WKMRAMlink.dbf"

14.2 Transit assignment summary process

The purpose of the transit assignment *summary* process is to summarize the output from the transit assignment process. The process is run with the transum.bat batch file, which, like the transit_assignment.bat batch file, is called from the "run model steps" batch file (see page A-15 of Appendix A). Whereas the transit assignment process is run in the scenario/output folder (e.g., X:\modelRuns\fy12\Ver2.3.36\2007_pseu), the transit assignment summary process is run in the **transum** folder (e.g., X:\modelRuns\fy12\Ver2.3.36\2007_pseu**transum**), which is a subfolder of the scenario/output folder. An excerpt from the transum.bat batch file is shown in Figure 19 and the complete batch file can be found in Appendix B. TranSum.bat calls the two Fortran programs written by AECOM (LINEVOL and LINESUM), using the following control files, which are stored in the "controls" directory:

```
pk_vol.ctl
op_vol.ctl
linesumMR.ctl
linesumMB.ctl
linesumOB.ctl
access.ctl
accessPkHbw.ctl
linksum.ctl
total.ctl
```

Figure 19 An excerpt of tranSum.bat transit summary batch file

```
:: TranSum.bat

CD %1\Transum

REM Step: Transit Volume Summary

REM Consolidate Peak and Off-Peak Volumes
..\..\software\LINEVOL ..\..\controls\pk_vol.ctl
..\..\software\LINEVOL ..\..\controls\op_vol.ctl
REM        if errorlevel 1 goto error

REM Metro Rail Line Summary
..\..\software\LINESUM ..\..\controls\linesumMR.ctl
REM        if errorlevel 1 goto error

REM Metro Bus Line Summary
..\..\software\LINESUM ..\..\controls\linesumMB.ctl
REM        if errorlevel 1 goto error

REM Other Bus Line Summary
..\..\software\LINESUM ..\..\controls\linesumOB.ctl
REM        if errorlevel 1 goto error

REM Metro Station Access
..\..\software\LINESUM ..\..\controls\access.ctl
..\..\software\LINESUM ..\..\controls\accessPkHbw.ctl

(etc.)
```

Ref: X:\modelRuns\fy12\Ver2.3.36\TranSum.bat

The summary process is governed by two Fortran summary programs written by AECOM: LINEVOL and LINESUM.[34] [35]

14.2.1 LINEVOL Fortran program

LINEVOL consolidates the 11 TRNBUILD transit line volume DBF files into one DBF file for each time-of-day period: (AM and midday/off-peak). The LINEVOL program is run twice – once for the peak period (using the pk_vol.ctl control file) and once for the off-peak/midday period (using the op_vol.ctl control file). For example, in pk_vol.ctl, a series of dBase files are collected into one dBase file (pk_vol.dbf), as shown in Figure 20.

Figure 20 Pk_vol.ctl control file for LINEVOL

```
Merge the Transit Volumes
PK_VOL.DBF                                //---- output file ----
..\i4_DRABAMlink.dbf   //--- drive-access
..\i4_DRBMAMlink.dbf
..\i4_DRCRAMlink.dbf
..\i4_DRMRAMlink.dbf
..\i4_KRABAMlink.dbf   //--- kiss-and-ride access
..\i4_KRBMAMlink.dbf
..\i4_KRMRAMlink.dbf
..\i4_WKABAMlink.dbf   //--- walk-access
..\i4_WKBMAMlink.dbf
..\i4_WKCRAMlink.dbf
..\i4_WKMRAMlink.dbf
```

Ref: X:\modelRuns\fy12\Ver2.3.36\Controls\pk_vol.ctl

The output from these two runs of LINEVOL is two transit loaded link files: pk_vol.dbf and op_vol.dbf. Either of these transit loaded link files can be brought into Cube Base as the transit layer, as is shown in Figure 21 through Figure 25.

[34] Manish Jain, "LINEVOL, LINESUM User's Guide – DRAFT", December 2006.
[35] Manish Jain to Ronald Milone and Mark S. Moran, "Transmittal of Transum setup and User's Guide," Memorandum, December 1, 2006.

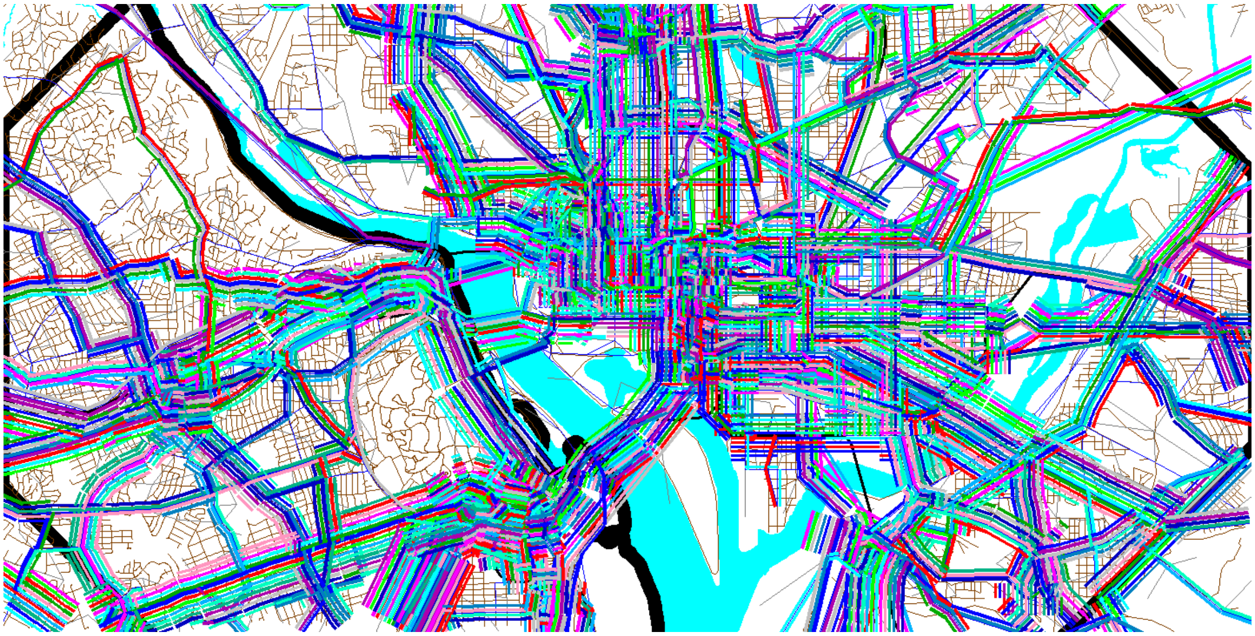


Figure 21 Using the pk_vol.dbf file in Cube Base as the transit layer: All transit routes turned on, but non-transit links (modes 11-16) turned off

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

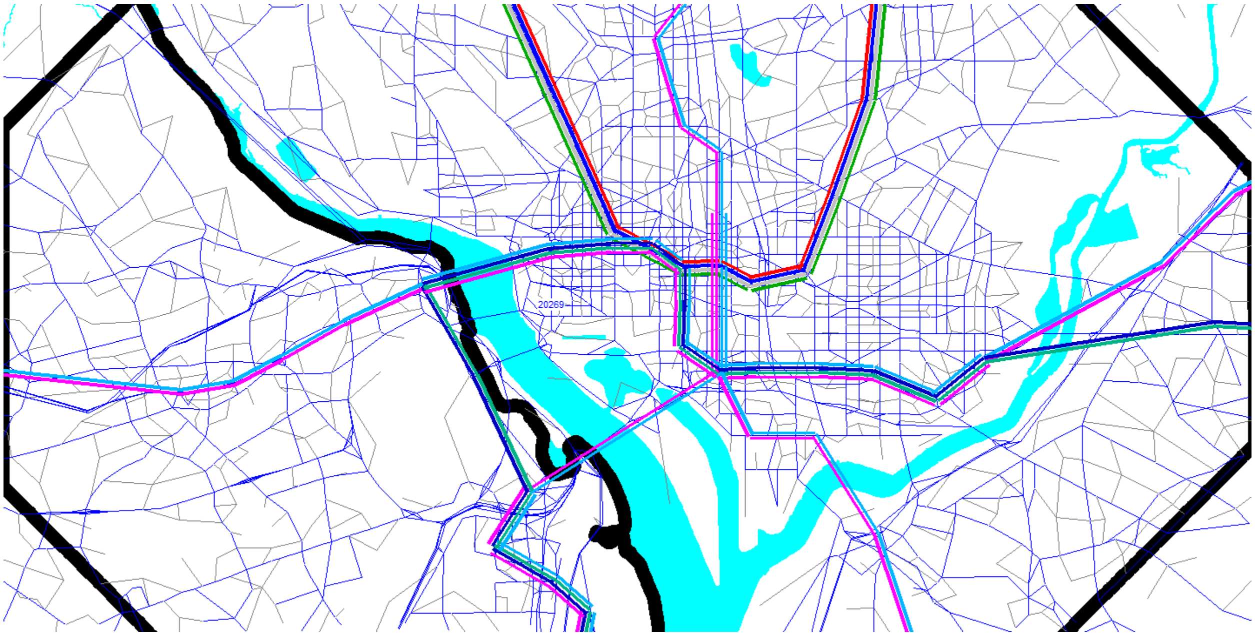


Figure 22 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

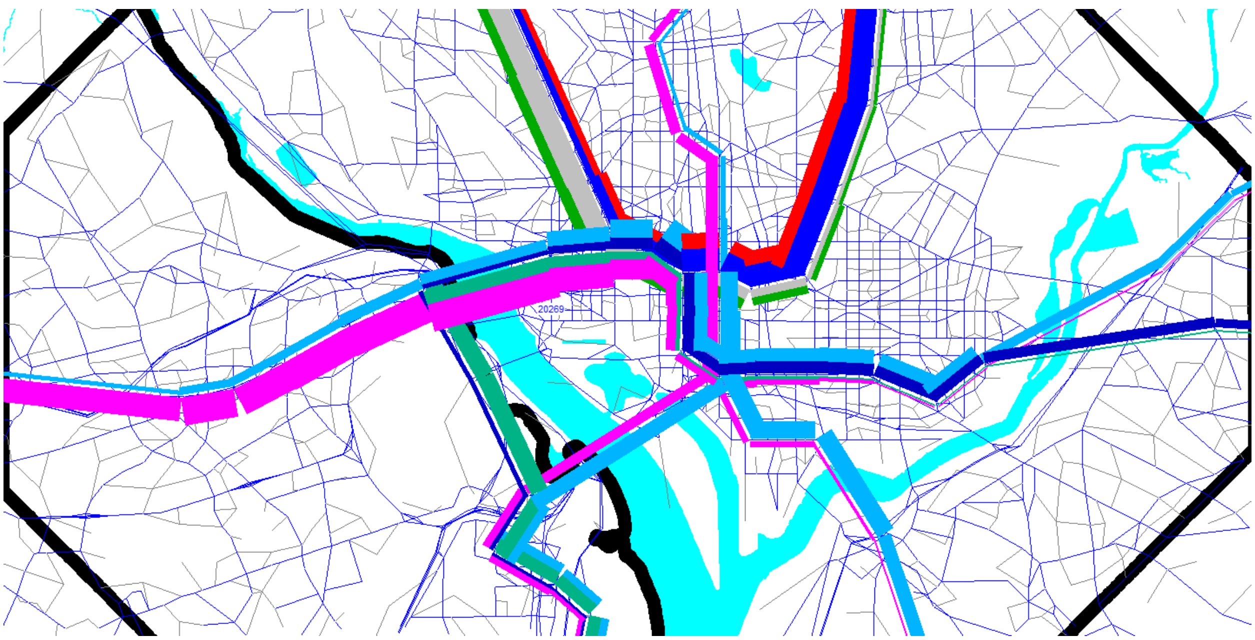


Figure 23 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab_vol): Arlington and DC

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

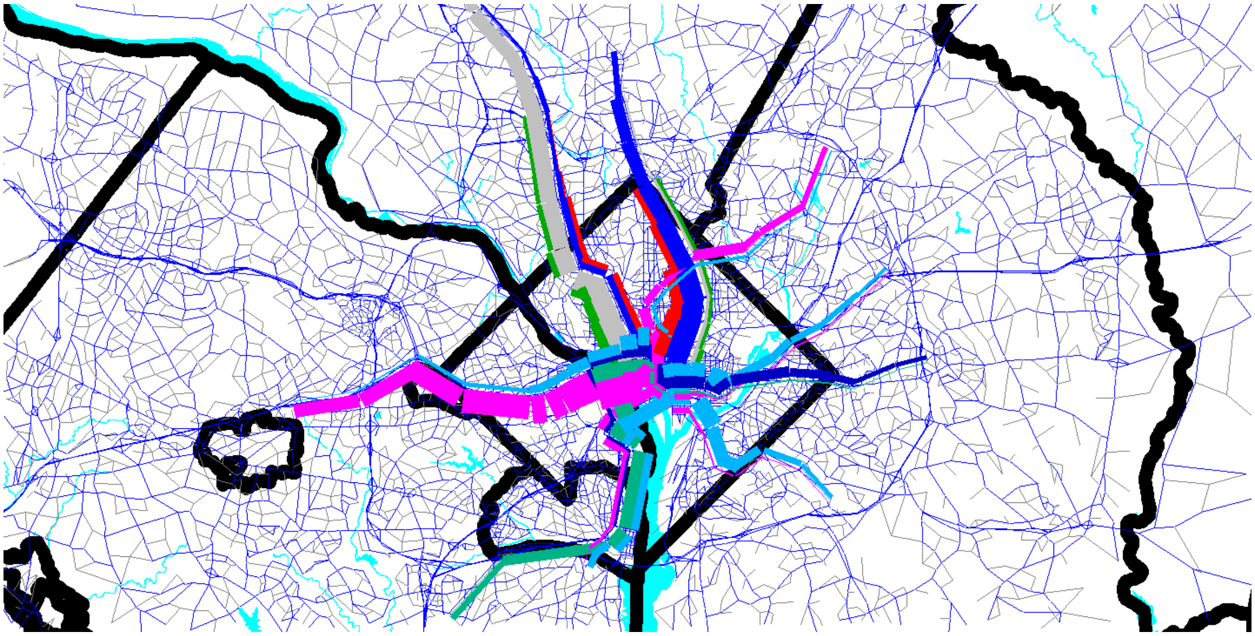


Figure 24 Using the pk_vol.dbf file in Cube Base as the transit layer: Only mode-3 (Metrorail) links turned on; using multi-bandwidth to represent transit loads (ab_vol): Metrorail system

Ref: "X:\modelRuns\fy12\Ver2.3.36\2007_pseu\zonehwy.net"

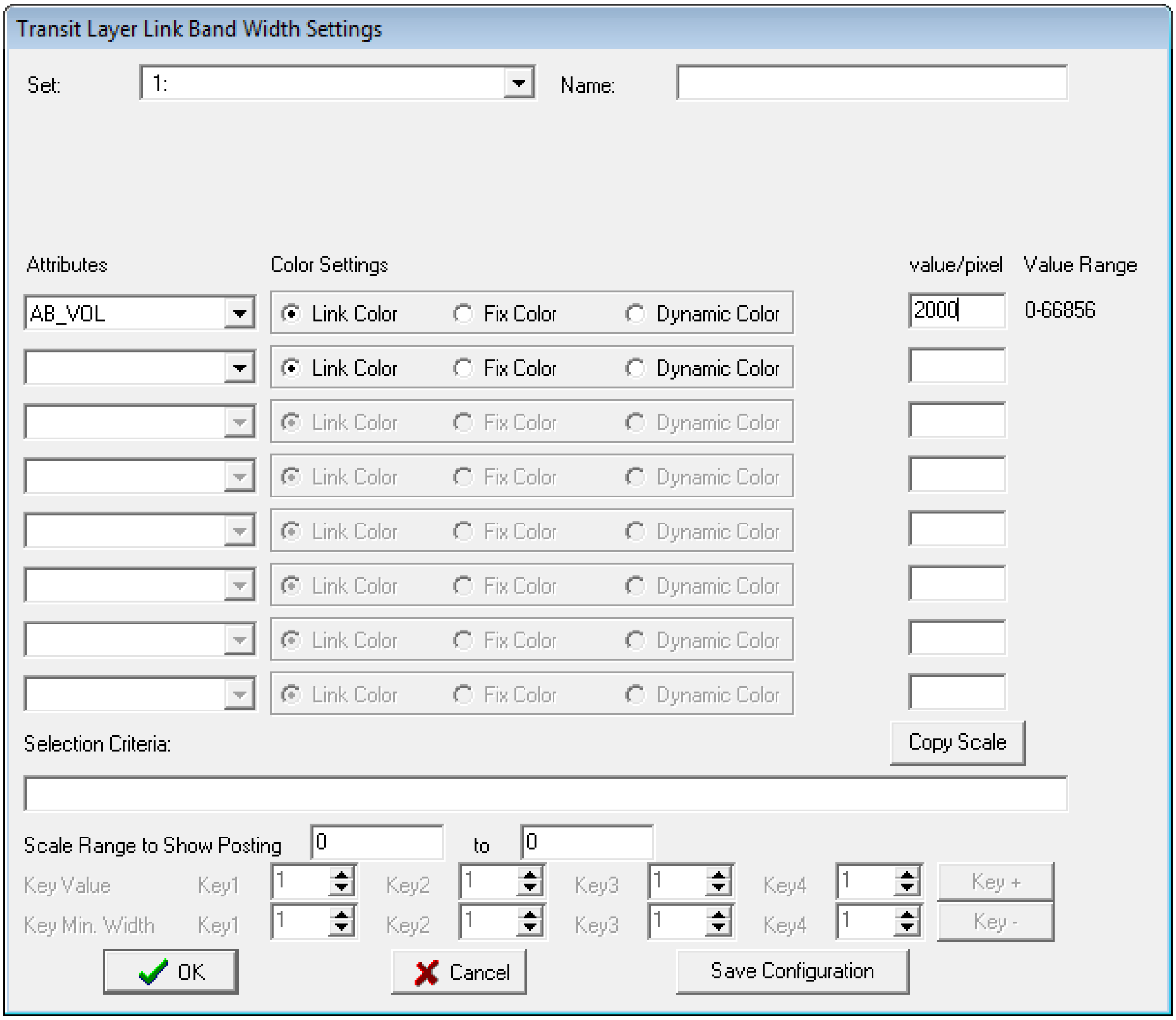


Figure 25 Using the multi-bandwidth option in Cube Base to show transit volumes on the Metrorail system.

14.2.2 LINESUM Fortran program

The LINESUM Fortran program summarizes transit line volume data stored in a TRNBUILD loaded link DBF file. It can be used to create the following summaries:

- Boarding/alighting information
- Station access information
- Link-based summaries (i.e., between stations).
- Transit route/line summaries

For example, the following command in transum.bat uses LINESUM to create a Metrorail line summary:

```
..\..\software\LINESUM ..\..\controls\linesumMR.ctl
```

The control file can be seen in Figure 26.

Figure 26 Metrorail line summary control file: linesumMR.ctl

```
Transit Summary Reports
PK_VOL.DBF                   //---- peak transit network ----
OP_VOL.DBF                   //---- offpeak transit network ----
NODE_NAME=STANAME.prn

LINE WMBLUA, NAME="BLUE LINE;FRANCONIA/SPRINGFIELD STA;LARGO TWN CTR STA"
LINE WMGRNA, NAME="GREEN LINE;GREENBELT STATION;BRANCH AVE STATION"
LINE WMORNA, NAME="ORANGE LINE;VIENNA STATION;NEW CARROLLTON STATION"
LINE WMREDA, NAME="RED LINE;SHADY GROVE STATION;GLENMONT STATION"
LINE WMREDB, NAME="RED LINE;GROSVENOR STATION;SILVER SPRING STATION"
LINE WMYELA, NAME="YELLOW LINE;MT VERNON SQ STATION;HUNTINGTON STATION"
LINE WMBLUA, WMGRNA, WMORNA, WMREDA, WMREDB, WMYELA, NAME="ALL METRORAIL"
```

Ref: X:\modelRuns\fy12\Ver2.3.36\Controls\linesumMR.ctl

Lines 2 and 3 of linesumMR.ctl are the consolidated dBase files that were obtained from running LINEVOL.EXE. Line 4 (NODE_NAME=STANAME.prn) contains the name of an optional file, which gives each station number an alphanumeric name. This equivalency file should be in the transum folder under the scenario/output folder (e.g., X:\modelRuns\fy12\Ver2.3.36\2007_pseu\transum\staname.prn). This file includes Metrorail stations, commuter rail station, and other "named" nodes, such as:

Metrorail	Commuter rail	Other named nodes
8001 Shady Grove 8002 Rockville 8003 Twinbrook 8004 White Flint 8005 Grosvenor 8006 Medical Center 8007 Bethesda	9001 Union Station 9002 Silver Spring 9003 Kensington 9004 Garrett Park 9005 Rockville 9006 Washington Grove 9007 Gaithersburg	45558 Bristol 44132 Broken Land Pkwy 22539 Burtonsville Crossi 26130 Capital Plaza 20811 Carter Barron 49556 Charlotte Hall 27208 Clinton

One of the typical summaries that is produced is a summary of HBW Metrorail trip ends (boardings) by Metrorail segments (station groups), e.g., year-2007 HBW Metrorail boardings and alightings. This is produced by the Fortran program LINESUM (`..\..\software\LINESUM ..\..\controls\accessPkHbw.ctl`) and the control file AccessPkHbw.ctl, which can be seen in Figure 27.

Figure 27 AccessPkHbw.ctl: Station access summary control file

```
Station Access Summary
PK_VOL.DBF                   //---- peak transit network ----

ACCESS
      8001, 8002, 8003, 8004, 8005, 8006, 8007, 8008, 8009, 8010, 8011, 8012, 8013,
      8014, 8015, 8016, 8017, 8018, 8019, 8020, 8021, 8022, 8023, 8024, 8025, 8026,
      8027, 8028, 8029, 8030, 8031, 8032, 8033, 8034, 8035, 8036, 8037, 8038, 8039,
      8040, 8041, 8042, 8043, 8044, 8045, 8046, 8047, 8048, 8049, 8050, 8051, 8052,
      8053, 8054, 8055, 8056, 8057, 8058, 8059, 8060, 8061, 8062, 8063, 8064, 8065,
      8066, 8067, 8068, 8069, 8070, 8071, 8072, 8073, 8074, 8075, 8076, 8077, 8078,
```

```
     8079, 8080, 8081, 8082, 8083, 8085, 8086, 8098

MODES=11-16, DETAIL=YES, NAME="Station Access", FILE=BOARDINGPKHBW.ASC
```

Ref: X:\modelRuns\fy12\Ver2.3.36\Controls\AccessPkHbw.ctl

We are using the "ACCESS" keyword, which produces a summary of the number of riders accessing each station, as shown in Figure 28.

Figure 28 An excerpt of the station access summary (BOARDINGPKHBW.ASC) created by AccessPkHbw.ctl

```
  8001   21349    1805
  8002    8793    2521
  8003    2481    3141
  8004    2413    2964
  8005    8131     265
  8006    3248    7099
  8007    9324   12472
  8008    9402    5640
  8009    6997    3488
  8010    4265    2410
  8011    4477     924
  8012    6985    2380
  8013    8306   23593
  8014    2113   30607
  8015    1109   18656
  8016     870    9228
  8017     648   19440
  8018    9623   20961
  8019    8571    4109
  8020    3217    2851
```

Ref: X:\modelRuns\fy12\Ver2.3.36\2007_pseu\transum\BOARDINGPKHBW.ASC

Another example of using the "ACCESS" keyword can be seen in Figure 29. By contrast, if we use the "LINE" keyword, the program produces boarding and alighting information, as shown in Figure 30.

Figure 29 An excerpt from a report file from the LINESUM program: Using the ACCESS keyword to summarize the number of riders accessing a station

```
       ********************************************
       |                                          |
       |          LINESUM - Version 1.8           |
       |        Copyright (c) 2011 by AECOM       |
       |         Sat Oct 29 18:28:41 2011         |
       |                                          |
       ********************************************

       Control File = ..\..\controls\accessPkHbw.ctl
       Station Access Summary

       Peak Line File = e:\modelruns\fy12\ver2.3.36\2007_pseu\transum\pk_vol.dbf

       Off-Peak Line File = NULL
       Station Access Summary
       Sat Oct 29 18:28:42 2011  LINESUM  page 2

   Title: Station Access
   Modes: 11-16

   Stop     Mode   Node   Board  Alight

   8001      11     507      11       0
             11     508      22       0
             11     509      87       0
             11     511     150       0
             11     512      67       0
             11     513      45       0
             11     514      59       0
             11     515      34       0
             11     516      20       0
             11     517      13       0
             11     518      18       0
             11     521       5       0
             11     522      46       0
             11     523      73       0
             11     524      32       0
             11     525      19       0
             11     526      59       0
             11     528      43       0
             11     530      38       0
             11     534       1       0
             11     716       1       0
             11     718       3       0
             11     719       4       0
             11     720      28       0
             11     721      21       0
```

```
              11     723       2       0
              11     724      52       0
              11     725       3       0
              11     727      55       0
              11     731      80       0
              11     732       3       0
              11     734      11       0
              11     735      19       0
              11     736      47       0
              11     738       1       0
              11     740      21       0
              11     743      91       0
              11     744       3       0
              11     745      33       0
              11     747     108       0
              11     748      53       0
              11     752      26       0
              12   22395    8427    1805
              15   11001   11415       0
           Total      44   21349    1805
(etc.)
```

Ref: X:\modelRuns\fy12\Ver2.3.36\2007_pseu\transum\accessPkHbw.prn

Figure 30 An excerpt from a report file from the LINESUM program: Using the LINE keyword produces boarding and alighting information

```
          ********************************************
          |                                          |
          |          LINESUM - Version 1.8           |
          |        Copyright (c) 2011 by AECOM       |
          |         Sat Oct 29 18:27:53 2011         |
          |                                          |
          ********************************************
          Control File = ..\..\controls\linesumMR.ctl

          Transit Summary Reports

          Peak Line File = e:\modelruns\fy12\ver2.3.36\2007_pseu\transum\pk_vol.dbf
          Off-Peak Line File = e:\modelruns\fy12\ver2.3.36\2007_pseu\transum\op_vol.dbf
          Transit Summary Reports
          Sat Oct 29 18:27:53 2011  LINESUM  page 2

     Title: BLUE LINE;FRANCONIA/SPRINGFIELD STA;LARGO TWN CTR STA
     Lines: WMBLUA

                            -------------- A->B Direction (Read Down) -------------  ------------- B->A Direction (Read Up) ----------------  ------Total------
            Dist   Time   -------Peak------  -----Off-Peak----  ------Daily------  -------Peak------  -----Off-Peak----  ------Daily------  ------Daily------
     Stop  (miles) (min)     On   Off  Ride     On   Off  Ride     On   Off  Ride     On   Off  Ride     On   Off  Ride     On   Off  Ride     On   Off  Ride

Franconia-    3.49   6.14  12713     0 12713   1993     0  1993  14706     0 14706      0   818   818      0   825   825      0  1643  1643  14706  1643 16349
Van Dorn S    3.86   4.95   8395   136 20972   2157    63  4088  10552   199 25060    118   599  1301     31   823  1616    149  1422  2917  10701  1621 27977
King Stree    0.68   2.02   3625  1205 23390    760   590  4256   4385  1795 27646    190  1558  2668    492   704  1829    682  2262  4497   5067  4057 32143
Braddock R    2.80   4.46   3492   617 26262   1182   445  4994   4674  1062 31256    141  1156  3683    183   689  2335    324  1845  6018   4998  2907 37274
National A    0.49   2.59      5    32 26234     35     4  5023     40    36 31257     24    69  3727     26    24  2334     50    93  6061     90   129 37318
Crystal Ci    0.76   2.02   2629  1295 27569   1458   259  6224   4087  1554 33793     82  2578  6226    166  1394  3563    248  3972  9789   4335  5526 43582
Pentagon C    0.61   0.99   2659   733 29499   1431   358  7294   4090  1091 36793    190  1558  7592    287  1406  4683    477  2964 12275   4567  4055 49068
Pentagon      1.24   2.91  12950  3815 38635   2847   580  9560  15797  4395 48195   1327  2458  8723    369  1949  6260   1696  4407 14983  17493  8802 63178
Arlington     0.99   2.08     12     0 38647     85     0  9646     97     0 48293     46     0  8677     56     0  6204    102     0 14881    199     0 63174
Rosslyn       1.35   3.12   3260 12114 29788   1670  3691  7626   4930 15805 37414   6381  6655  8952   3321  3051  5933   9702  9706 14885  14632 25511 52299
Foggy Bott    0.57   2.08    848  6870 23768   1643  2089  7179   2491  8959 30947    808  8686 16829   1115  3341  8159   1923 12027 24988   4414 20986 55935
Farragut W    0.38   0.97    528  9232 15066    990  1789  6381   1518 11021 21447    659  7091 23260    910  1849  9097   1569  8940 32357   3087 19961 53804
McPherson     0.46   1.09   1425  7483  9007   1017  2002  5397   2442  9485 14404   1704 11073 32630   1075  2306 10329   2779 13379 42959   5221 22864 57363
Metro Cent    0.29   0.91   6987  6076  9920   1601  2199  4797   8588  8275 14717  16736  7042 22936   3365  2455  9420  20101  9497 32356  28689 17772 47073
Federal Tr    0.41   2.10      0  2147  7775    142   454  4485    142  2601 12260    329   520 23128    371   211  9259    700   731 32387    842  3332 44647
Smithsonia    0.59   2.28     20  3210  4583    178  1063  3601    198  4273  8184    154  1227 24199    724   543  9078    878  1770 33277   1076  6043 41461
L'Enfant P    0.33   1.94   4476   799  8259   2081  1293  4389   6557  2092 12648   6466  6627 24361   2479  2811  9411   8945  9438 33772  15502 11530 46420
Federal Ce    0.57   1.91     37  2871  5422    130  1300  3219    167  4171  8641    445  1121 25037    919   531  9024   1364  1652 34061   1531  5823 42702
Capitol So    0.50   1.94     31  3448  2008    116  1569  1764    147  5017  3772    406  1238 25866   1030   647  8639   1436  1885 34505   1583  6902 38277
Eastern Ma    0.63   1.97     72   583  1496    111   598  1277    183  1181  2773   1866   447 24448   1255   255  7642   3121   702 32090   3304  1883 34863
Potomac Av    0.66   0.97    104   302  1298     96   357  1015    200   659  2313   2725   178 21900   1444   190  6388   4169   368 28288   4369  1027 30601
Stadium Ar    2.69   3.09    332   387  1242    176   369   822    508   756  2064   2861  1219 20257   1367   327  5350   4228  1546 25607   4736  2302 27671
Benning Ro    1.42   2.83    152   235  1159     84   307   600    236   542  1759   4137   134 16256   1981   143  3511   6118   277 19767   6354   819 21526
Capitol He    0.97   2.88     55   200  1010     34   214   418     89   414  1428   4657    35 11633   1832    35  1715   6489    70 13348   6578   484 14776
```

```
Addison Ro    1.77   3.04     86   339   761     25   107   337    111   446  1098   2767    59  8923    582    14  1149   3349    73 10072   3460   519 11170
Summerfiel    1.23   2.72     50   325   485     16   133   221     66   458   706   3917    22  5028    420    12   738   4337    34  5766   4403   492  6472
Largo Town                         485                221                706         5028                 738                5766                5766   706

     Total   29.74  64.00        64943             22058              87001              64164              26538              90702             177703

     Max      3.86   6.14  12950 12114 38647   2847  3691  9646  15797 15805 48293  16736 11073 32630   3365  3341 10329  20101 13379 42959  28689 25511 63178

     Passenger Miles                 441656             107283             548939             302473             115845             418318             967257
     Passenger Hours                  15711               4091              19802              12495               4848              17343              37145
     Average Trip Length (miles)        6.8                4.9               11.7                4.7                4.4                9.1               20.7
     Average Trip Length (minutes)     14.5               11.1               25.6               11.7               11.0               22.6               48.3
         Transit Summary Reports
         Sat Oct 29 18:27:54 2011  LINESUM  page 3

(etc.)

         Sat Oct 29 18:27:55 2011  Processing Complete.
```

Ref: X:\modelRuns\fy12\Ver2.3.36\2007_pseu\transum\linesumMR.prn

Appendix A. Flowcharts

Ref: V2.3_flowChart_2011-11-01.vsd

Flowchart Numbers associated with Flowchart Steps

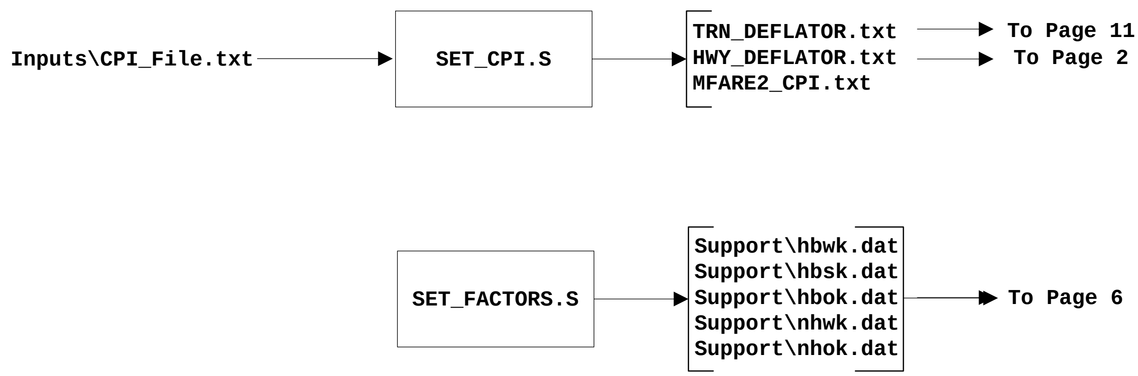
This Appendix contains detailed data processing flow charts showing the relationship of input and output files to the processing steps comprising the Version 2.3 travel model. The flowcharts are arranged on the basis of the 14 batch files used in the model application. Many of the batch files are reused during the application of the model. The table below describes the sequence of each batch file used by iteration. The flowcharts are numbered in accordance with the numbering system (1-14), shown in the table below.

	Initial (Pump Prime) Iteration				
		Standard Iterations			
Batch File	PP	1	2	3	4
Set_CPI.bat	1				
PP_Highway_Build.bat	2				
PP_Highway_Skims.bat	3				
Transit_Skim_All_Modes.bat		4			
Transit_Fare.bat		11			
Trip_Generation.bat		5			
Trip_Distribution.bat		6			
Mode_Choice.bat /Mode_Choice_TC_V23.bat		12			
Auto_Driver.bat		13			
PP_Auto_Drivers.bat	7				
Time-of-Day.bat		8			
Highway_Assignment.bat		9			
Highway_Skims.bat		10			
Transit_Assignment.bat		14			

Ref: V23_Flowchart_Table_V2.3.36.xls

TITLE:	**Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)**
COMPANY: COG/TPB	**CREATOR:** RM/MS
DATE: 2011-11	PG: 1 OF 16
FILENAME: V2.3_flowChart_2011-11-01.vsd	

Set_CPI.bat:
Develop CPI and K-factors



Report Files Generated by Set_CPI.bat:
Set_CPI.rpt
Set_Factors.rpt

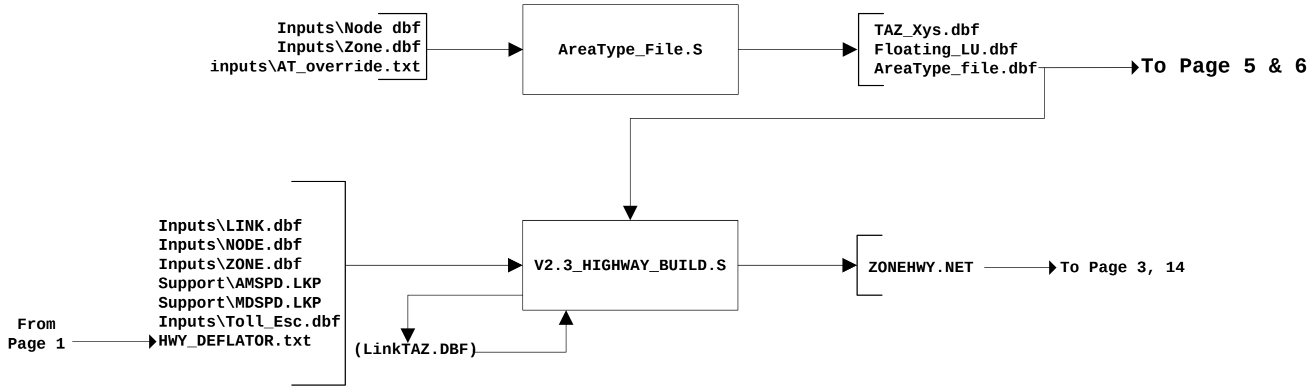
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR:** RM/MS

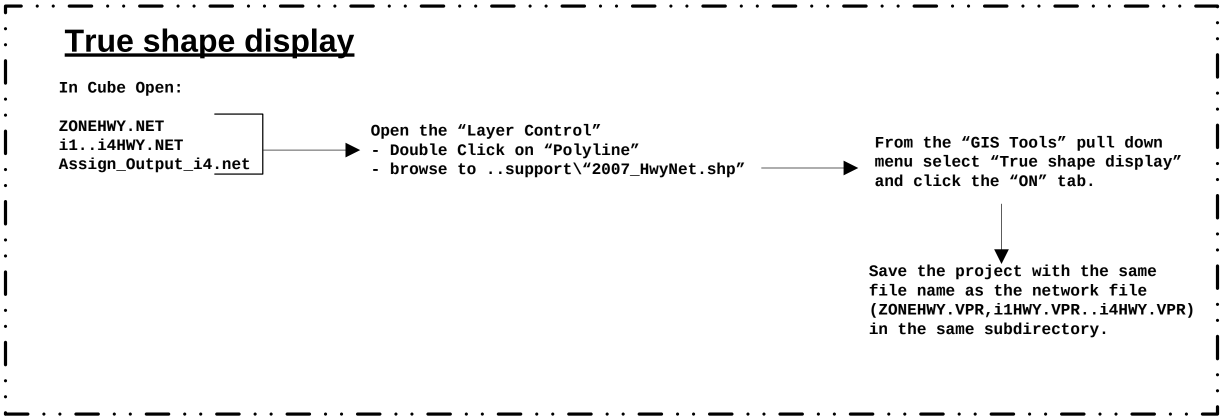
DATE: 2011-11 PG: 2 OF 16

FILENAME: V2.3_flowChart_2011-11-01.vsd

PP_Highway_Build.bat:
Highway Network Preparation



Optional



Report Files Generated by PP_Highway_Build.bat:
AreaType_File.rpt
V2.3_highway_build.rpt

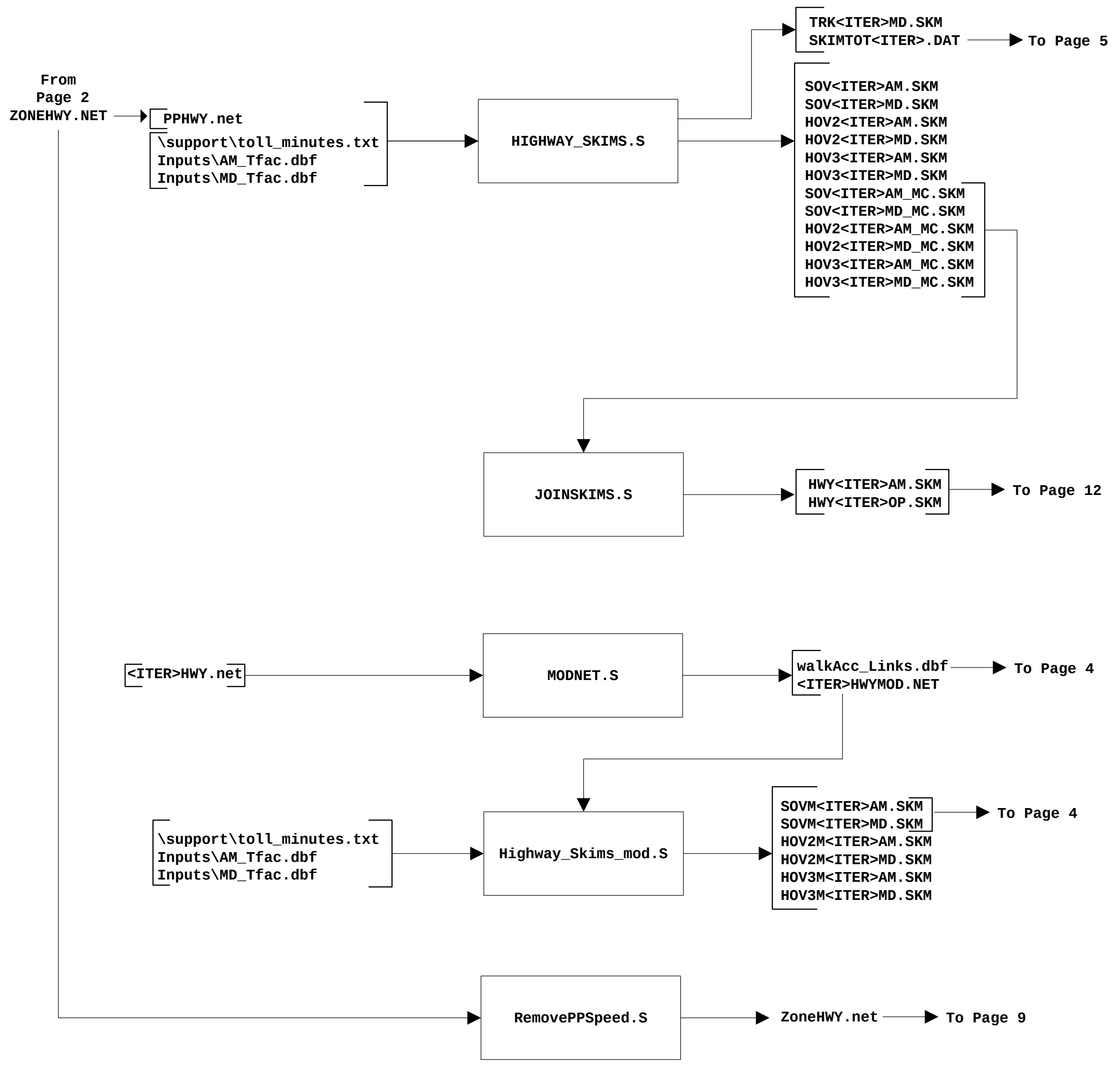
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR:** RM/MS

DATE: 2011-11 PG: 3 OF 16

FILENAME: V2.3_flowChart_2011-11-01.vsd

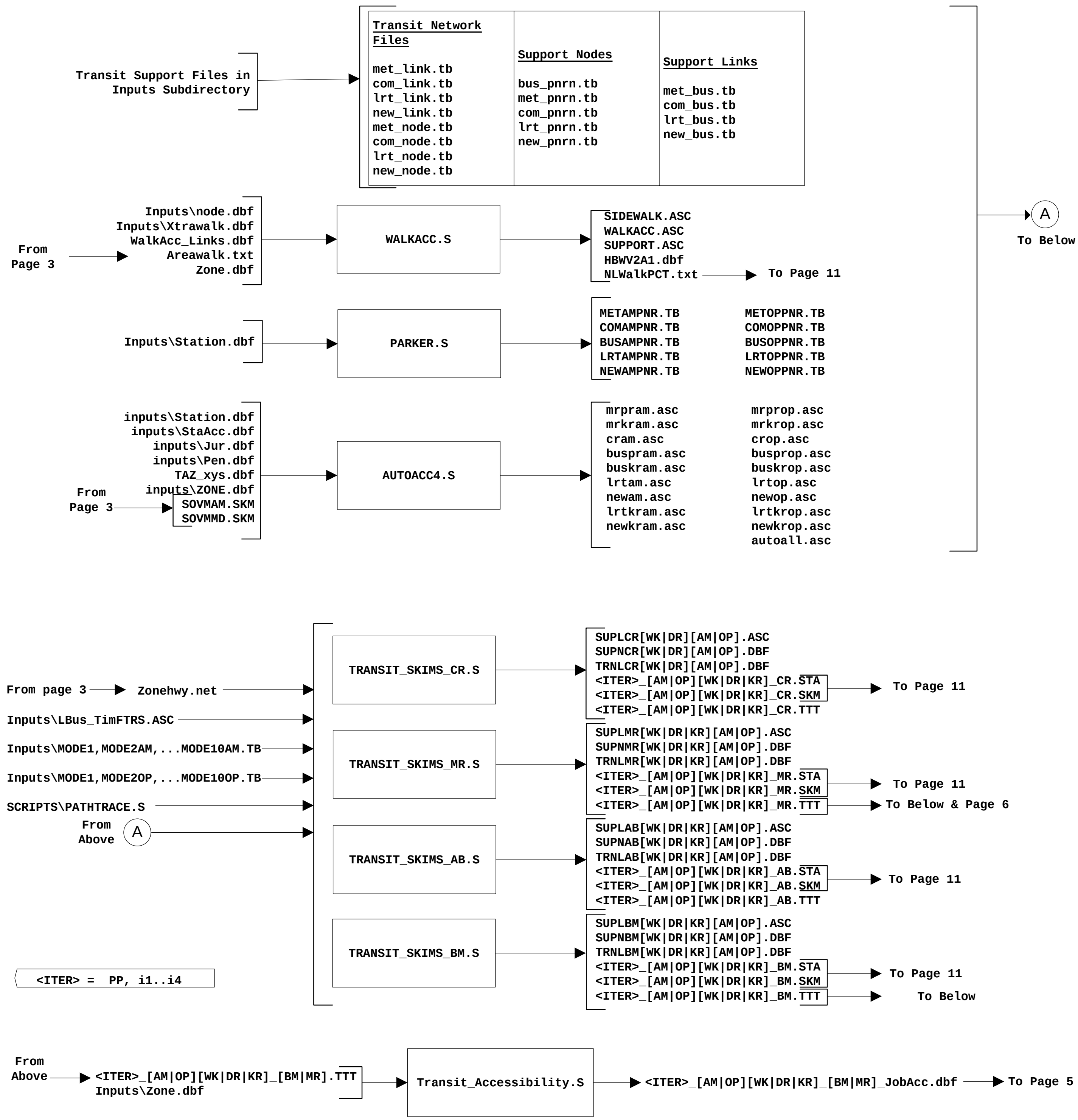
PP_Highway_Skims.bat



Report Files Generated by PP_Highway_Skims.bat:

pp_Highway_skims.rpt
pp_Joinskims.rpt
ppModnet.rpt
pp_Highway_skims_mod.rpt
pp_RemovePPSpeed.rpt

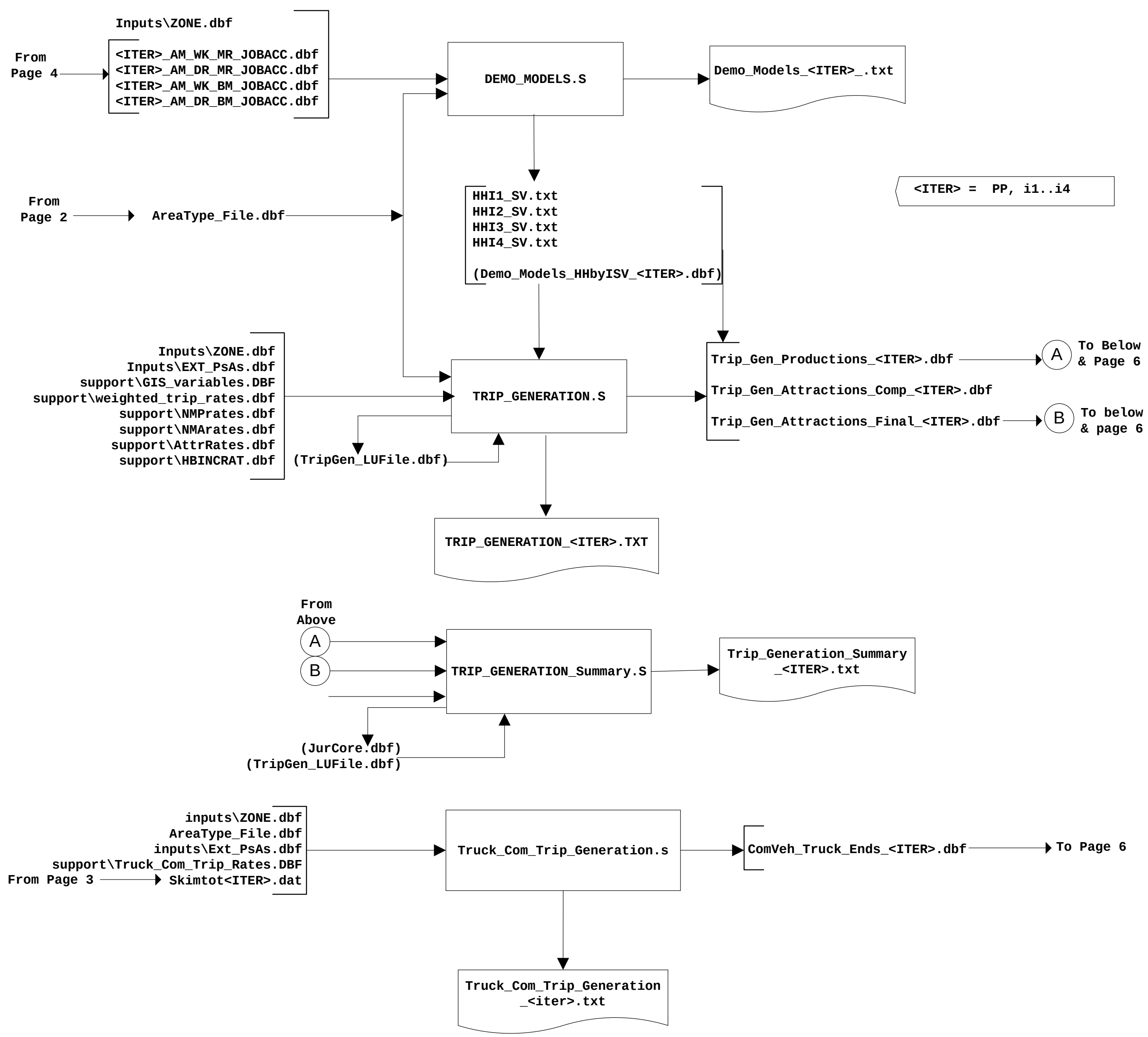
TRANSIT_Skim_All_Modes.bat



Report Files Generated by TRANSIT_Skim_All_Modes.bat:

Walkacc.rpt
Parker.rpt
Autoacc4.rpt
Transit_Skims_CR.rpt, Transit_Skims_MR.rpt, Transit_Skims_AB.rpt, Transit_Skims_BM.rpt
Transit Accessibility.rpt

Trip_Generation.bat



Report Files Generated by Trip_Generation.bat:

<ITER>_Demo_Models.rpt
<ITER>_Trip_Generation.rpt
<ITER>_Trip_Generation_Summary.rpt
<ITER>_Truck_Com_Trip_Generation.rpt

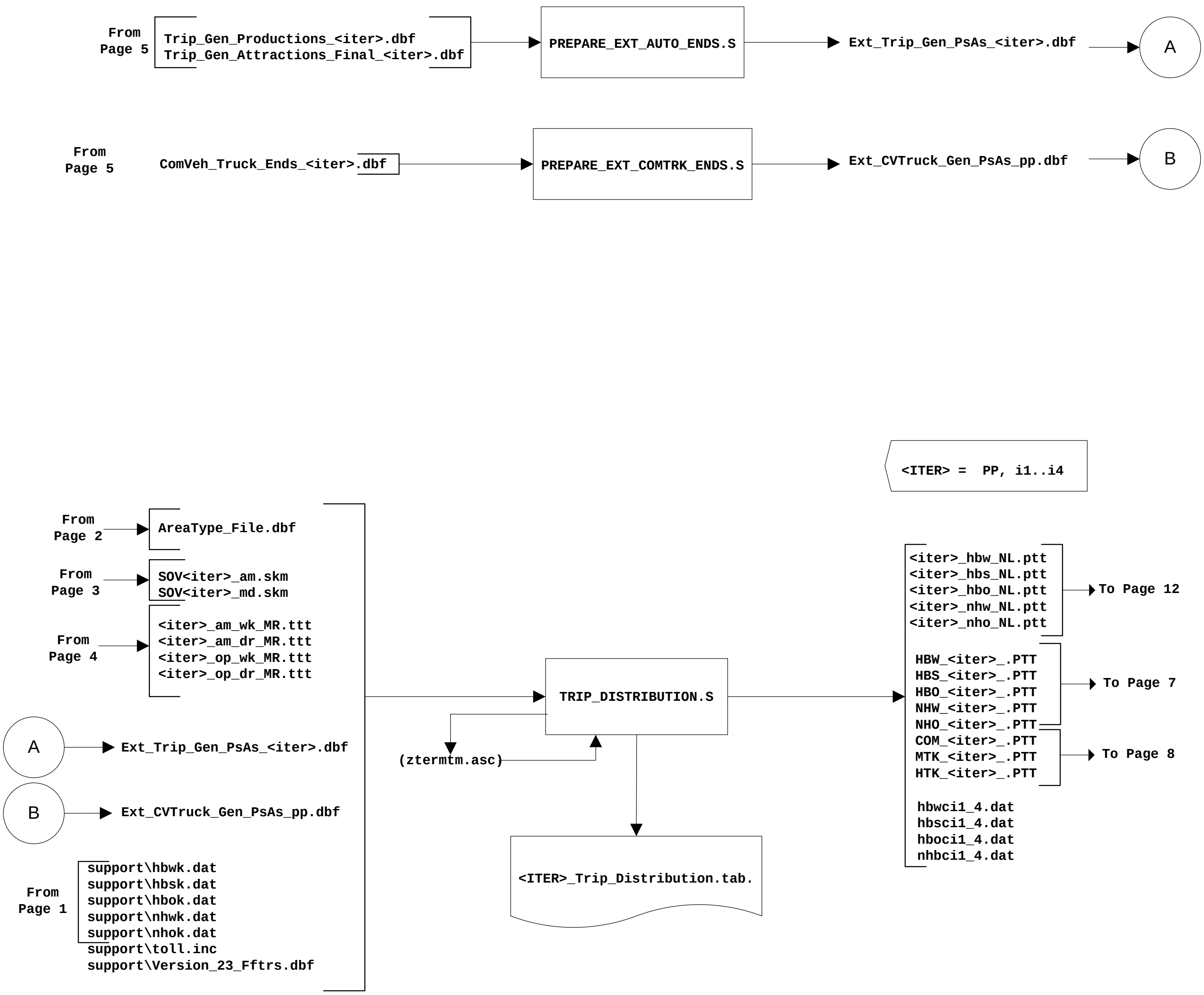
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB CREATOR: RM/MS

DATE: 2011-11 PG: 6 OF 16

FILENAME: V2.3_flowChart_2011-11-01.vsd

Trip_Distribution.bat



Report Files Generated by Trip_Distribution.bat:

Trip_Distribution.rpt

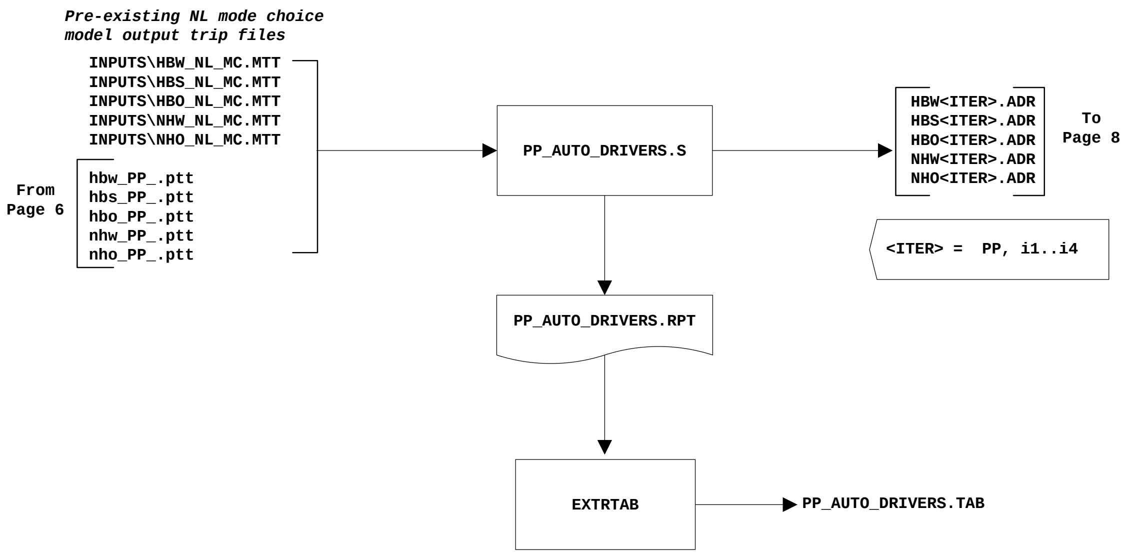
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR: RM/MS**

DATE: 2011-11

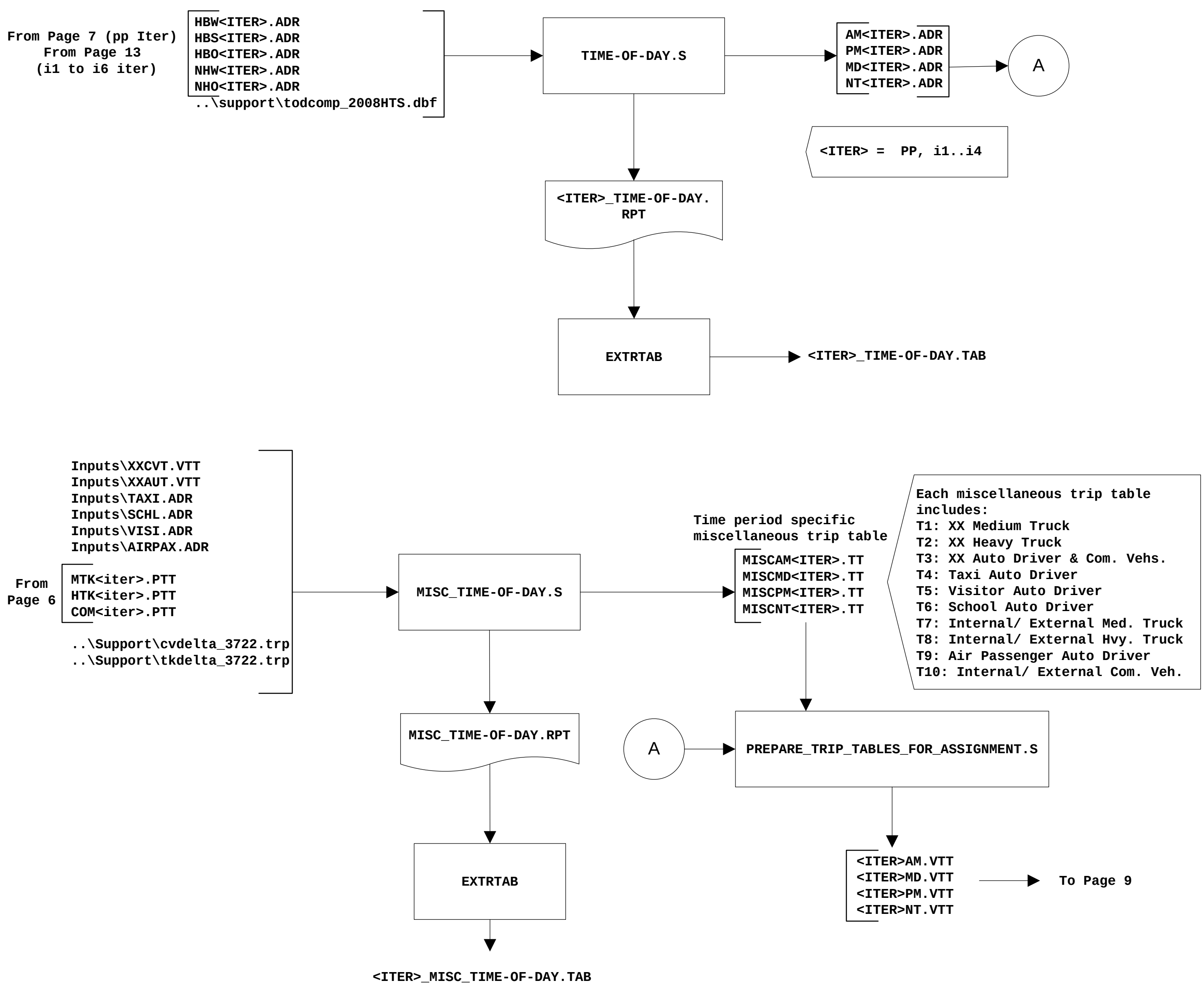
FILENAME: V2.3_flowChart_2011-11-01.vsd

PP_Auto_Drivers.bat:
Pump Prime Auto Driver Trips



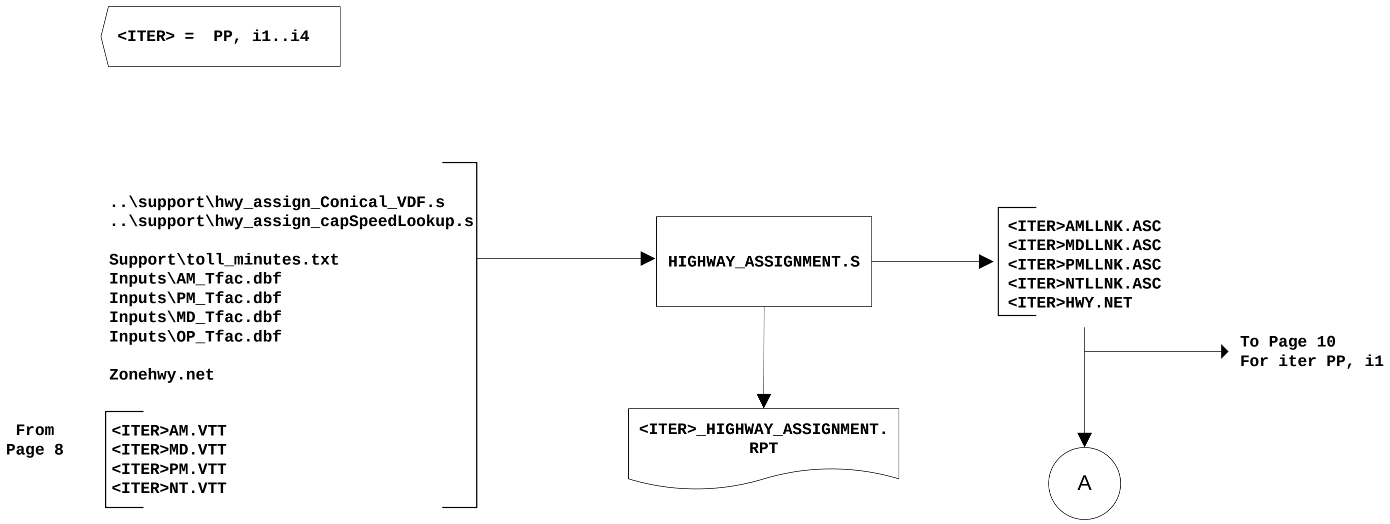
Report Files Generated by PP_Auto_Drivers.bat:
PP_Auto_Drivers.rpt

Time-of-Day.bat

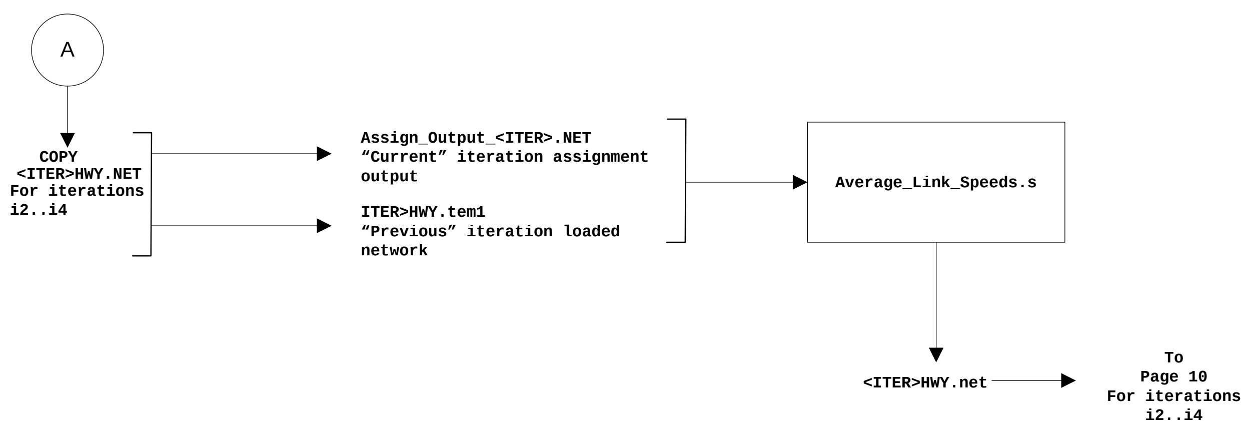


Report Files Generated by Time-of-Day.bat:
<ITER>_Time-of-Day.rpt
<ITER>_Misc_Time-of-Day.rpt

Highway_Assignment.bat



Average_Link_Speeds.bat



Report Files Generated by Highway_Assignment.bat:
<ITER>_Highway_assignment.rpt
Average_Link_Speeds.rpt

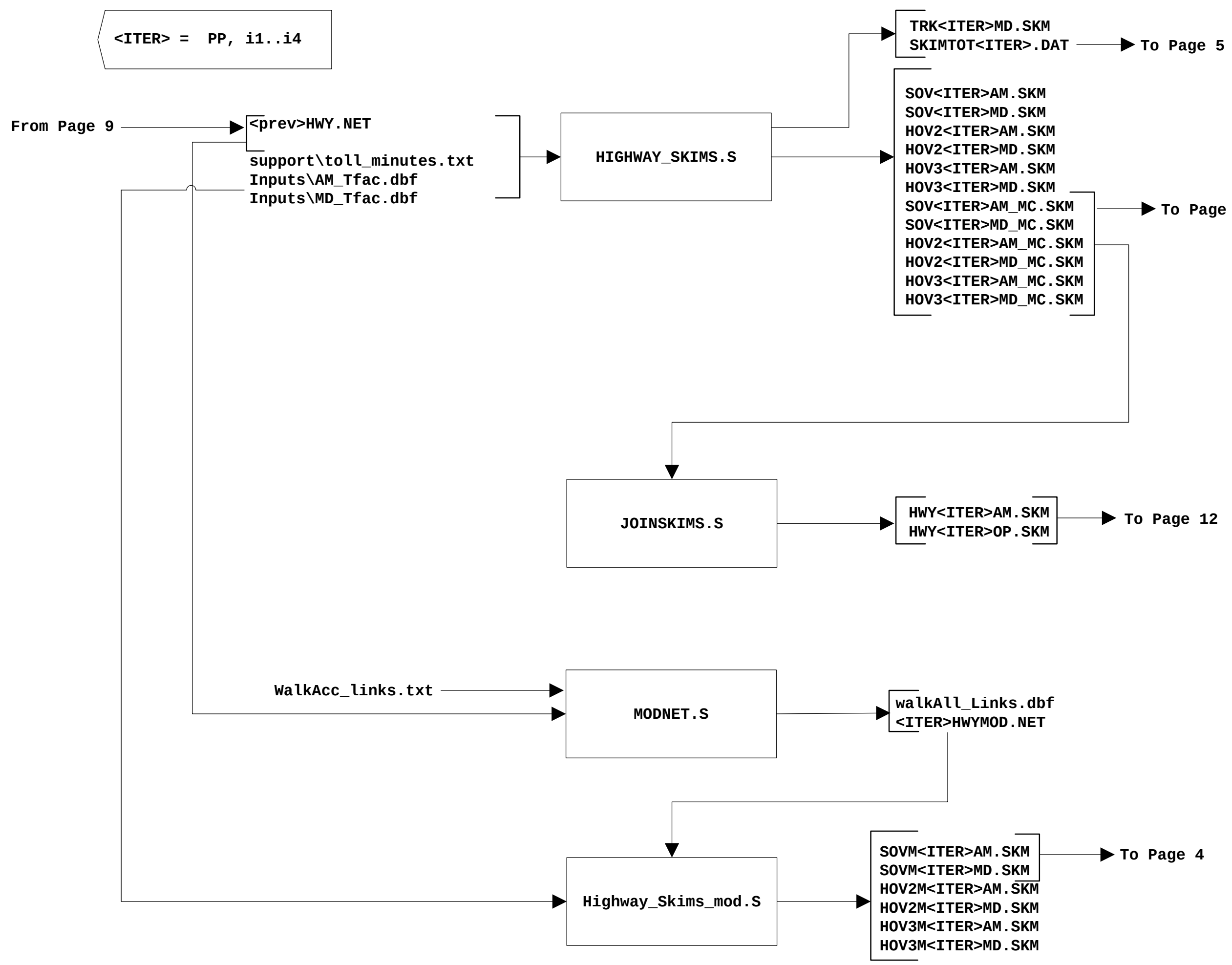
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR:** RM/MS

DATE: 2011-11

FILENAME: V2.3_flowChart_2011-11-01.vsd

Highway_Skims.bat




```
Report Files Generated by Highway_Skims.bat:
                                        <ITER>_Highwayskims.rpt
                                        <ITER>_Joinskims.rpt
                                        <ITER>_Modnet.rpt
                                        <ITER>_Highway_skims_mod.rpt
```

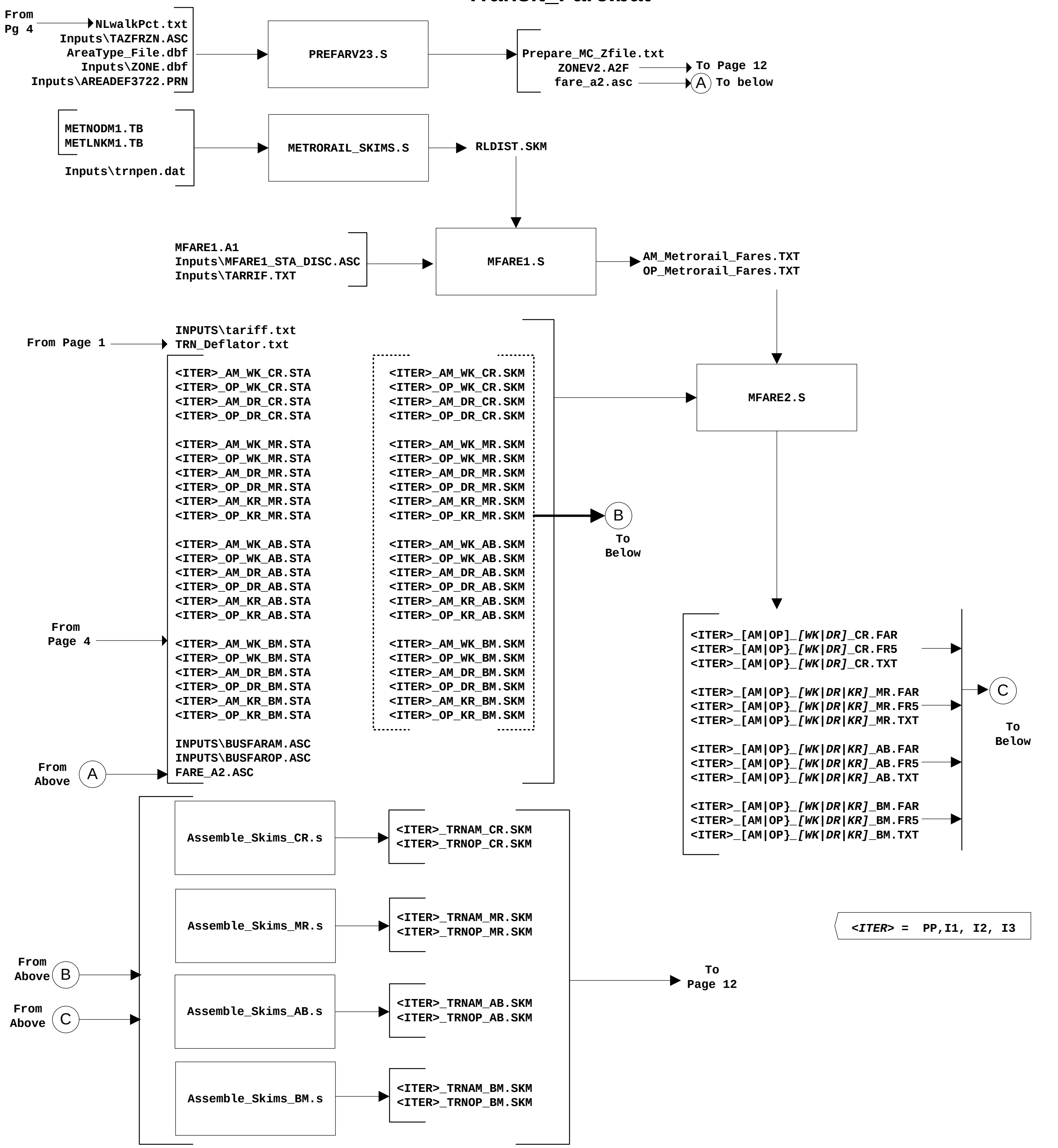
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR: RM/MS**

DATE: 2011-11 PG: 11 OF 16

FILENAME: V2.3_flowChart_2011-11-01.vsd

Transit_Fare.bat



Report Files Generated by Transit_Fare.bat:
<ITER>_prefarV23.rpt
<ITER>_Metrorail_Skims.rpt
<ITER>_Mfare1.rpt
<ITER>_Mfare2.rpt
<ITER>_Assemble_Skims_CR.rpt, Assemble_Skims_MR.rpt, Assemble_Skims_AB.rpt, Assemble_Skims_BM.rpt

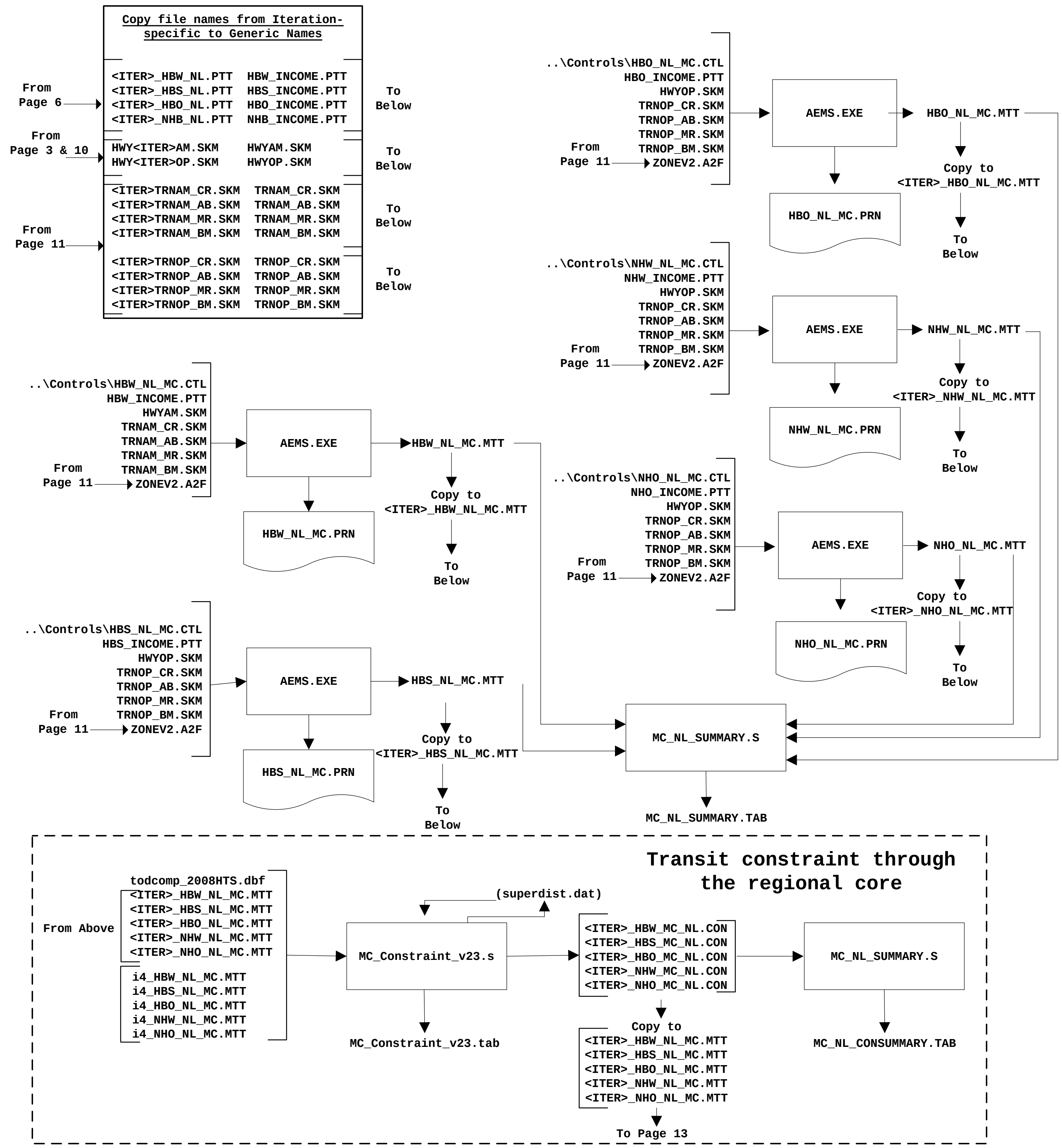
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR: RM/MS**

DATE: 2011-11

FILENAME: V2.3_flowChart_2011-11-01.vsd

Mode_Choice.bat/ Mode_Choice_TC_V23.bat



Report Files Generated by Mode_Choice.bat/Mode_Choice_TC_V23.bat:
<ITER>_MC_NL_Summary.rpt
<ITER>_MC_Constraint_v23.rpt
<ITER>_mc_NL_CONSUMMARY.rpt

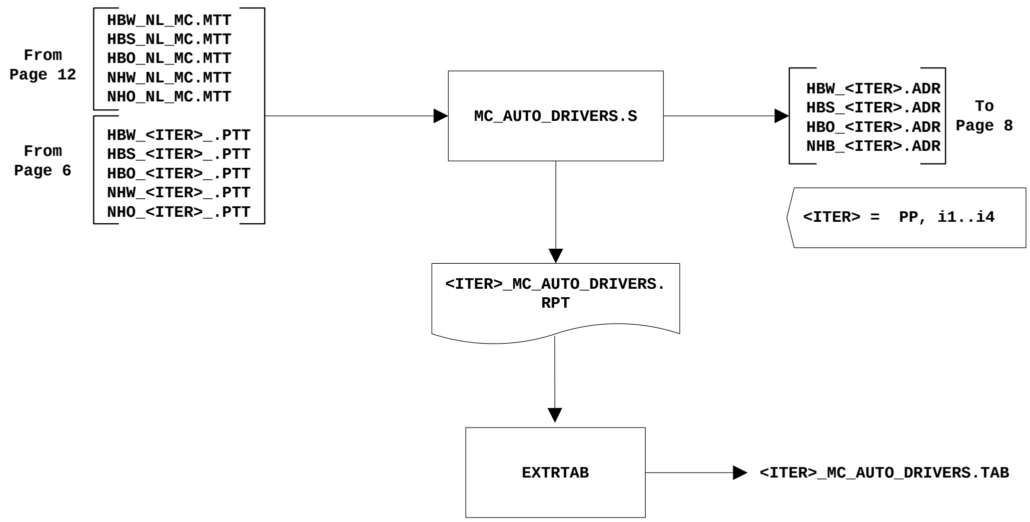
TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR: RM/MS**

DATE: 2011-11 PG: 13 OF 16

FILENAME: V2.3_flowChart_2011-11-01.vsd

Auto_Driver.bat



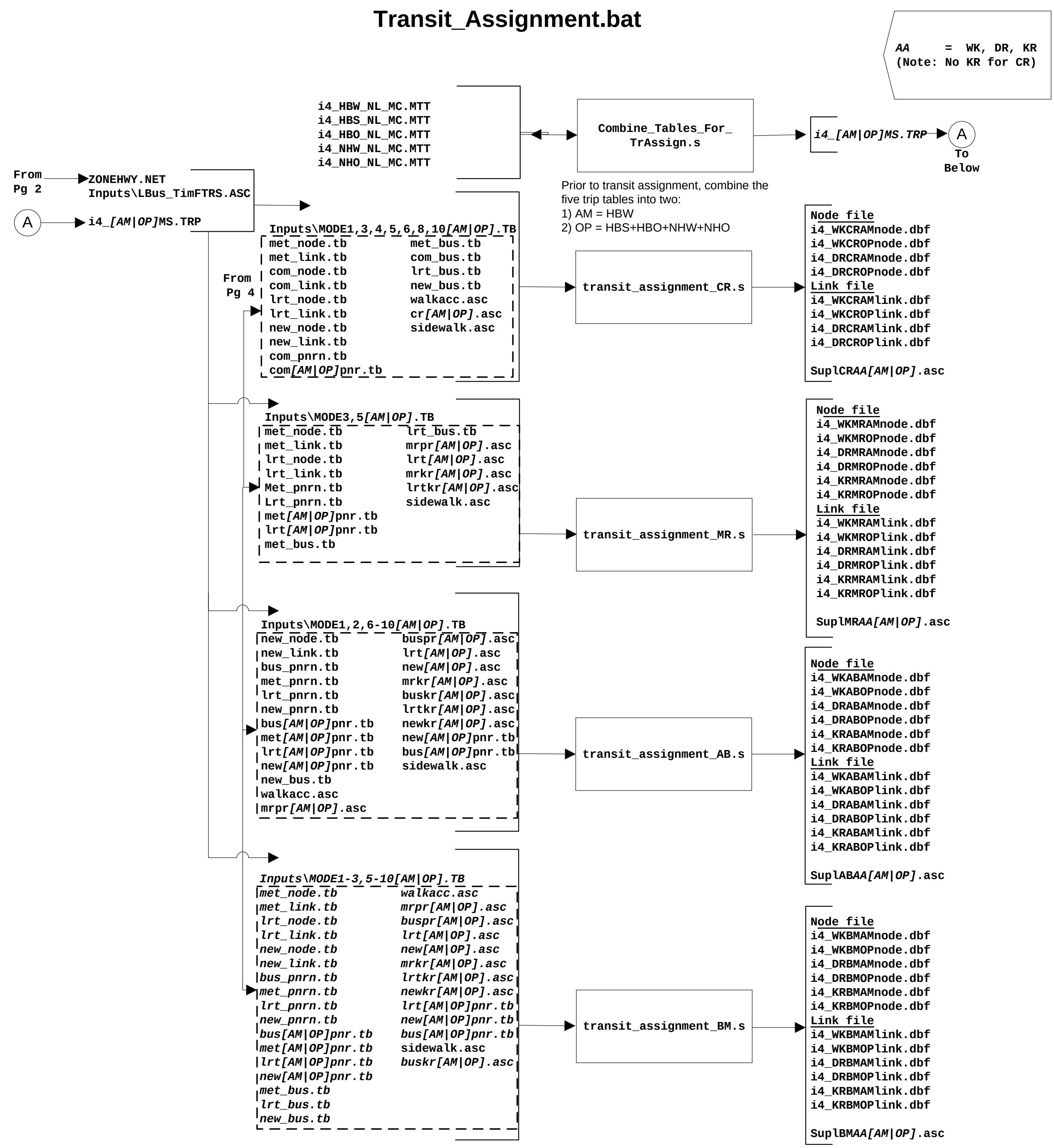
Report Files Generated by Auto_driver.bat:
<ITER>_MC_Auto_Drivers.rpt

TITLE: Application of the TPB Ver. 2.3 Travel Model (3,722-TAZ area system)

COMPANY: COG/TPB **CREATOR: RM/MS**

DATE: 2011-11

FILENAME: V2.3_flowChart_2011-11-01.vsd



Report Files Generated by Transit_Assignment.bat:

Combine _Tables_for_TrAssign.rpt

transit_assignment_CR.rpt, transit_assignment_MR.rpt, transit_assignment_AB.rpt, transit_assignment_BM.rpt

Transum.bat

`pk_vol.ctl`

```
i4_DRABAMlink.dbf
i4_DRBMAMlink.dbf
i4_DRCRAMlink.dbf
i4_DRMRAMlink.dbf
i4_KRABAMlink.dbf
i4_KRBMAMlink.dbf
i4_KRMRAMlink.dbf
i4_WKABAMlink.dbf
i4_WKBMAMlink.dbf
i4_WKCRAMlink.dbf
i4_WKMRAMlink.dbf
```

DBF files containing AM transit volumes on transit links. All HBW trips are assumed to occur in the AM peak period.

LINEVOL.EXE

LINEVOL consolidates the 11 TRNBUILD transit line volume DBF files into one DBF file for each time-of-day period: (AM and midday ("off peak")

`PK_VOL.DBF`

B

See below and on subsequent pages

One consolidated link output file (LINKO) with transit loads on each transit link.

This file contains transit loads for HBW (i.e., the AM peak period)

See previous page of flowchart for the description of LINKO and NODEO files.

`op_vol.ctl`

```
i4_DRABOPlink.dbf
i4_DRBMOPlink.dbf
i4_DRCROPlink.dbf
i4_DRMROPlink.dbf
i4_KRABOPlink.dbf
i4_KRBMOPlink.dbf
i4_KRMROPlink.dbf
i4_WKABOPlink.dbf
i4_WKBMOPlink.dbf
i4_WKCROPlink.dbf
i4_WKMROPlink.dbf
```

DBF files containing midday ("off peak") transit volumes on transit links. It is assumed that all HBS, HBO, NHW, and NHO trips occur in the off-peak period.

LINEVOL.EXE

LINEVOL consolidates the 11 TRNBUILD transit line volume DBF files into one DBF file for each time-of-day period: (AM and midday ("off peak")

`OP_VOL.DBF`

C

See below and on subsequent pages

One consolidated link output file (LINKO) with transit loads on each transit link.

This file contains transit loads for HBS, HBO, NHW, and NHO trips (i.e., the off-peak period)

See previous page of flowchart for the description of LINKO and NODEO files.

`linesumMR.ctl`

B C

From above

```
PK_VOL.DBF
OP_VOL.DBF
STANAME.prn
```

STANAME.prn is an equivalency between node/station number and the node name/station name.

LINESUM.EXE

LINESUM summarizes transit line volume data stored in a TRNBUILD loaded link DBF file. It can be used to create the following summaries:

* Boarding/alighting information
* Station access information
* Link-based summaries
  (i.e., between stations).
* Transit route/line summaries

In this case, we are using the "LINE" keyword, so the program produces boarding and alighting information.

`linesumMR.prn`

Peak and off-peak line summaries for Metrorail (ons and offs by station, summarized by line). Also includes passenger-mile statistics.

Title: ORANGE LINE;VIENNA STATION;NEW CARROLLTON STATION
Lines: WMORNA

			A->B Direction (Read Down)								
	Dist	Time	Peak			Off-Peak			Daily		
Stop	(miles)	(min)	On	Off	Ride	On	Off	Ride	On	Off	Ride
Vienna	2.39	3.71	29069	0	29069	4032	0	4032	33101	0	33101
Dunn Lorin	2.49	3.97	3836	191	32713	689	118	4605	4525	309	37318
West Falls	2.09	2.93	8929	523	41118	2655	423	6839	11584	946	47957
East Falls	2.51	4.00	5889	203	46806	1559	60	8338	7448	263	55144
Ballston	0.49	1.66	13803	3667	56940	5845	639	13544	19648	4306	70484
Virginia S	0.49	2.01	2922	928	58938	1104	318	14329	4026	1246	73267
Clarendon	0.67	2.56	2460	870	60525	1659	483	15502	4119	1353	76027
Court Hous	0.91	1.69	5887	2516	63895	2647	1369	16781	8534	3885	80676

Transum.bat cont'd

access.ctl

B C

From previous pages

PK_VOL.DBF
OP_VOL.DBF

Files representing loaded transit links from the AM peak and off-peak (midday) transit networks

LINESUM.EXE

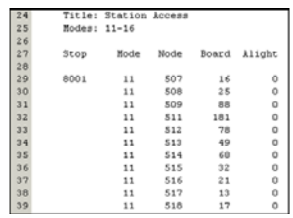
LINESUM summarizes transit line volume data stored in a TRNBUILD loaded link DBF file. It can be used to create the following summaries:

* Boarding/alighting information
* Station access information
* Link-based summaries
   (i.e., between stations).
* Transit route/line summaries

In this case, we are using the "ACCESS" keyword, so the program summarizes the number of riders accessing a station.
Transit assignment is in P/A format.

access.prn

Number of passengers accessing one or more stations, i.e., number of passengers boarding and alighting each station, stratified by access mode:
11 = KNR
12 = walk access to Metrorail and commuter rail
15 = PNR
16 = walk access to surface transit (bus, BRT, etc.)
"Boardings" = productions
"Alightings" = attractions



```
Title: Station Access
Modes: 11-16

Stop      Mode    Node    Board  Alight

8001        11     507       16       0
            11     508       25       0
            11     509       88       0
            11     511      181       0
            11     512       78       0
            11     513       49       0
            11     514       60       0
            11     515       32       0
            11     516       21       0
            11     517       13       0
            11     518       17       0
```

Appendix B. Batch files

1 Runall

1.1 run_Model_Ver2.3.36_2007_pseu.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_Model_Ver2.3.36_2007_pseu.bat
:: 2011-10-28 Fri 15:21:08

set root=.
set scenar=2007_pseu
set runbat=run_ModelSteps_Ver2.3.36_2007_pseu.bat
:: Environment variables for (intrastep) distributed processing:
::     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 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
::

:: Start Cube Cluster, create N sub-nodes for processing (these will work with the
main node already running)
Cluster.exe %root%\%scenar%\mwcog %subnode% start exit

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

:: Close the N sub-nodes
Cluster.exe %root%\%scenar%\mwcog %subnode% close exit

:: 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%\searchForErrs.txt      start %root%\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

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

1.2 run_ModelSteps_Ver2.3.36_2007_pseu.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_ModelSteps_Ver2.3.36_2007_pseu.bat
:: 2011-10-28 Fri 15:16:54

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2007
set _alt_=Ver2.3.36_2007_pseu

:: Location of substitute HOV 3+ skims/ null location for this year
:: This will eventually be handled in the mode choice batch file
cd %1
set _HOV3PATH_=
cd..

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

set _iter_=pp
set _prev_=pp

call Set_CPI.bat                %1
call PP_Highway_Build.bat       %1
call PP_Highway_Skims.bat       %1
call Transit_Skim_All_Modes.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.bat     %1
call Highway_Skims.bat          %1

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

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Highway_Skims.bat          %1

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

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i3
```

```
set _prev_=i2

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i4
set _prev_=i3

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

:: rem ====== Transit assignment ============================================

call Transit_Assignment.bat %1
call TranSum.bat %1

:: rem ====== End of batch file =============================================

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

1.3 run_Model_Ver2.3.36_2040_base.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_Model_Ver2.3.36_2040_base.bat
:: 2011-10-28 Fri 15:20:41

set root=.
set scenar=2040_base
set runbat=run_ModelSteps_Ver2.3.36_2040_base.bat
:: Environment variables for (intrastep) distributed processing:
::     use IDP = true(t) or false(f)
::     Number of subnodes:  1-3 => 3 subnodes and one main node = 4 nodes in total
set useIdp=t
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
::     (Tee.exe is part of the Windows 2000 Resource Kit,
::             Tee32 3.2  Copyright 2002 Brian Friesen, 47,104 bytes)

:: Start Cube Cluster, create N sub-nodes for processing (these will work with the
main node already running)
Cluster.exe %root%\%scenar%\mwcog %subnode% start exit

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

:: Close the N sub-nodes
Cluster.exe %root%\%scenar%\mwcog %subnode% close exit

:: 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%\searchForErrs.txt      start %root%\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

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

1.4 run_ModelSteps_Ver2.3.36_2040_base.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_ModelSteps_Ver2.3.36_2040_base.bat
:: 2011-10-28 Fri 15:14:03

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2040
set _alt_=Ver2.3.36_2040_base

:: Location of substitute HOV 3+ skims/ null location for this year
:: This will eventually be handled in the mode choice batch file
cd %1
set _HOV3PATH_=
cd..

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

set _iter_=pp
set _prev_=pp

call Set_CPI.bat                %1
call PP_Highway_Build.bat       %1
call PP_Highway_Skims.bat       %1
call Transit_Skim_All_Modes.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.bat     %1
call Highway_Skims.bat          %1

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

```
set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Highway_Skims.bat          %1

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

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i3
set _prev_=i2

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

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

set _iter_=i4
set _prev_=i3

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
call Highway_Skims.bat          %1

:: rem ====== Transit assignment ============================================

call Transit_Assignment.bat %1
call TranSum.bat %1

:: rem ====== End of batch file =============================================

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

1.5 run_Model_Ver2.3.36_2040_final.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_Model_Ver2.3.36_2040_final.bat
:: 2011-10-28 Fri 15:17:38

set root=.
set scenar=2040_final
set runbat=run_ModelSteps_Ver2.3.36_2040_final.bat
:: Environment variables for (intrastep) distributed processing:
::     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 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
::

:: Start Cube Cluster, create N sub-nodes for processing (these will work with the
main node already running)
Cluster.exe %root%\%scenar%\mwcog %subnode% start exit

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

:: Close the N sub-nodes
Cluster.exe %root%\%scenar%\mwcog %subnode% close exit

:: 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%\searchForErrs.txt      start %root%\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

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

1.6 run_ModelSteps_Ver2.3.36_2040_final.bat

```
:: E:\modelRuns\fy12\Ver2.3.36\run_ModelSteps_Ver2.3.36_2040_final.bat
:: 2011-10-28 Fri 15:12:58

:: Version 2.3 TPB Travel Model on 3722 TAZ System

set _year_=2040
set _alt_=Ver2.3.36_2040_final

:: Location of substitute HOV 3+ skims/ null location for this year
:: This will eventually be handled in the mode choice batch file
cd %1
set _HOV3PATH_=..\2040_base
cd..

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

set _iter_=pp
set _prev_=pp

call Set_CPI.bat                %1
call PP_Highway_Build.bat       %1
call PP_Highway_Skims.bat       %1
call Transit_Skim_All_Modes.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.bat     %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat    %1

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

set _iter_=i1
set _prev_=pp

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat    %1

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

set _iter_=i2
set _prev_=i1

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat    %1

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

set _iter_=i3
set _prev_=i2

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat    %1

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

set _iter_=i4
set _prev_=i3

call Transit_Skim_All_Modes.bat %1
call Transit_Fare.bat           %1
call Trip_Generation.bat        %1
call Trip_Distribution.bat      %1
call Mode_Choice.bat            %1
call Auto_Driver.bat            %1
call Time-of-Day.bat            %1
call Highway_Assignment.bat     %1
call Average_Link_Speeds.bat    %1
rem  call Highway_Skims.bat          %1
call HSR_Highway_Skims.bat    %1

:: rem ====== Transit assignment ============================================

call Transit_Assignment.bat %1
call TranSum.bat %1

:: rem ====== End of batch file =============================================

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

2 ‘Pump-Prime’ Iterations

2.1 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


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
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

```
CD %1

REM  Highway Skims

:: COPY ZONEHWY.NET TEMPORARILY TO PPHWY.NET

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

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

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

if exist SOVPPam.SKM   copy SOVPPam.SKM     SOVPPam_Initial.SKM /y
if exist SOVPPmd.SKM   copy SOVPPmd.SKM     SOVPPmd_Initial.SKM /y
if exist HOV2PPam.SKM  copy HOV2PPam.SKM    HOV2PPam_Initial.SKM /y
if exist HOV2PPmd.SKM  copy HOV2PPmd.SKM    HOV2PPmd_Initial.SKM /y
if exist HOV3PPam.SKM  copy HOV3PPam.SKM    HOV3PPam_Initial.SKM /y
if exist HOV3PPmd.SKM  copy HOV3PPmd.SKM    HOV3PPmd_Initial.SKM /y

if exist SOVMPPam.SKM   copy SOVMPPam.SKM     SOVMPPam_Initial.SKM /y
if exist SOVMPPmd.SKM   copy SOVMPPmd.SKM     SOVMPPmd_Initial.SKM /y
if exist HOV2MPPam.SKM  copy HOV2MPPam.SKM    HOV2MPPam_Initial.SKM /y
if exist HOV2MPPmd.SKM  copy HOV2MPPmd.SKM    HOV2MPPmd_Initial.SKM /y
if exist HOV3MPPam.SKM  copy HOV3MPPam.SKM    HOV3MPPam_Initial.SKM /y
if exist HOV3MPPmd.SKM  copy HOV3MPPmd.SKM    HOV3MPPmd_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_%_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

:: DELETE TEMPORARY ppHWY.NET, THIS WILL BE CREATED AFTER the PP HIGHWAY ASSIGNMENT

if exist PPHWY.NET del PPHWY.NET


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


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

2.4 PP_Auto_Drivers.bat

```
CD %1

REM  Pump Prime Auto Driver Trips

if exist voya*.*                   del voya*.*
if exist %_iter_%_Auto_Drivers.rpt del %_iter_%_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_%_Auto_Drivers.rpt /y

..\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.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

CD %1

::copy transit lines and support files from the inputs subdir.
copy inputs\*.TB /y
copy inputs\mfare1.a1 /y

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

if exist sovmam.skm   del sovmam.skm
if exist sovmmd.skm   del sovmmd.skm


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

if exist sovm%_prev_%am.skm  copy sovm%_prev_%am.skm  sovmam.skm /y
if exist sovm%_prev_%MD.skm  copy sovm%_prev_%md.skm  sovmmd.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
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

:skipwalk

if exist voya*.*  del voya*.*
if exist autoacc4.rpt  del autoacc4.rpt
start /w Voyager.exe  ..\scripts\autoacc4.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn  copy voya*.prn autoacc4.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\tppl*.*

CD..

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

::  Transit Network Building (Final) Commuter Rail

CD %1

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



CD %1

REM  Transit Network Building (Final) Metrorail

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

```
CD %1

REM  Transit Network Building (Final) All Bus

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


CD %1

REM  Transit Network Building (Final) Bus+Rail

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

CD..

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
start /w Voyager.exe  ..\scripts\MFARE2.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
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


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

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

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


goto end

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

CD..
```

3.5 Mode_Choice.bat

```
::  Nested Logit Mode Choice Model Application

CD %1

:: 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 Hwy%_prev_%AM.SKM  copy Hwy%_prev_%AM.SKM        HWYAM.SKM /y
if exist Hwy%_prev_%OP.SKM  copy Hwy%_prev_%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 exist hbw_NL_MC.* del hbw_NL_MC.*
..\software\AEMS   ..\controls\HBW_NL_MC.ctl
if errorlevel 1 goto error

if exist hbs_NL_MC.* del hbs_NL_MC.*
..\software\AEMS   ..\controls\HBS_NL_MC.ctl
```

```
if errorlevel 1 goto error

if exist hbo_NL_MC.* del hbo_NL_MC.*
..\software\AEMS   ..\controls\hbo_NL_MC.ctl
if errorlevel 1 goto error

if exist nhw_NL_MC.* del nhw_NL_MC.*
..\software\AEMS   ..\controls\nhw_NL_MC.ctl
if errorlevel 1 goto error

if exist nho_NL_MC.* del nho_NL_MC.*
..\software\AEMS   ..\controls\nho_NL_MC.ctl
if errorlevel 1 goto error

::
::  COPY GENERIC MODE CHOICE OUTPUT FILES
::  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


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 *.tb1 copy *.tb1          %_iter_%_mc_NL_summary.txt /y
if exist *.tb1 del  *.tb1

goto end

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

3.6 Mode_Choice_TC_V23.bat

```
::--------------------------------------------------
::  Nested Logit Mode Choice Model Application
::  ** WITH TRANSIT CONSTRAINT **
::--------------------------------------------------

CD %1
:: Set transit constraint path and files

:: Subdirectory containing the constraining transit trips
:: ORIGINAL SET COMMAND: set _tcpath_=..\2020_final

set _tcpath_=..\2020_final

:: constraining transit trip files
set _TChbw_=%_tcpath_%\i4_HBW_NL_MC.MTT
set _TChbs_=%_tcpath_%\i4_HBS_NL_MC.MTT
set _TChbo_=%_tcpath_%\i4_HBO_NL_MC.MTT
set _TCnhw_=%_tcpath_%\i4_NHW_NL_MC.MTT
set _TCnho_=%_tcpath_%\i4_NHO_NL_MC.MTT

:: check on existence of constraining trip files
if not exist    %_tcpath_%\i4_HBW_NL_MC.MTT goto error
if not exist    %_tcpath_%\i4_HBS_NL_MC.MTT goto error
if not exist    %_tcpath_%\i4_HBO_NL_MC.MTT goto error
if not exist    %_tcpath_%\i4_NHW_NL_MC.MTT goto error
if not exist    %_tcpath_%\i4_NHO_NL_MC.MTT goto error

:: Now, go forward with standard mode choice modeling process
:: 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 Hwy%_prev_%AM.SKM  copy Hwy%_prev_%AM.SKM        HWYAM.SKM /y
if exist Hwy%_prev_%OP.SKM  copy Hwy%_prev_%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 exist hbw_NL_MC.* del hbw_NL_MC.*
..\software\AEMS   ..\controls\HBW_NL_MC.ctl
if errorlevel 1 goto error

if exist hbs_NL_MC.* del hbs_NL_MC.*
..\software\AEMS   ..\controls\HBS_NL_MC.ctl
if errorlevel 1 goto error

if exist hbo_NL_MC.* del hbo_NL_MC.*
..\software\AEMS   ..\controls\hbo_NL_MC.ctl
if errorlevel 1 goto error

if exist nhw_NL_MC.* del nhw_NL_MC.*
..\software\AEMS   ..\controls\nhw_NL_MC.ctl
if errorlevel 1 goto error

if exist nho_NL_MC.* del nho_NL_MC.*
..\software\AEMS   ..\controls\nho_NL_MC.ctl
if errorlevel 1 goto error

::
::  COPY GENERIC MODE CHOICE OUTPUT FILES
::  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

if exist HBW_NL_MC.MTT del   HBW_NL_MC.MTT
if exist HBS_NL_MC.MTT del   HBS_NL_MC.MTT
if exist HBO_NL_MC.MTT del   HBO_NL_MC.MTT
if exist NHW_NL_MC.MTT del   NHW_NL_MC.MTT
```

```
if exist NHO_NL_MC.MTT del   NHO_NL_MC.MTT

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

..\software\extrtab    %_iter_%_mc_NL_summary.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 *.tb1 copy *.tb1              %_iter_%_mc_NL_summary.txt /y
if exist *.tb1 del  *.tb1


::-----------------------------------------------
:: - End of standard mode choice modeling process -
:: - Begin transit constraint process below      -
::-----------------------------------------------

if exist voya*.*                     del voya*.*
if exist %_iter_%mc_constraint_V23.rpt   del %_iter_%mc_constraint_V23.rpt

start /w Voyager.exe  ..\scripts\mc_constraint_v23.s /start -Pvoya -S..\%1
if errorlevel 1 goto error

if exist voya*.prn copy voya*.prn       %_iter_%_mc_constraint_v23.rpt /y
..\software\extrtab                     %_iter_%_mc_constraint_V23.rpt
if exist extrtab.out copy extrtab.out   %_iter_%_mc_constraint_v23.tab /y
if exist extrtab.out del  extrtab.out

::
:: Now save the unconstrained mode choice output files (*.ucn)
:: for the purpose of checking, and then, replace the unconstrained
:: files with their constrained counterparts
::

if exist  %_iter_%_HBW_NL_MC.MTT    copy %_iter_%_HBW_NL_MC.MTT
%_iter_%_HBW_NL_MC.ucn /y
if exist  %_iter_%_HBS_NL_MC.MTT    copy %_iter_%_HBS_NL_MC.MTT
%_iter_%_HBS_NL_MC.ucn /y
if exist  %_iter_%_HBO_NL_MC.MTT    copy %_iter_%_HBO_NL_MC.MTT
%_iter_%_HBO_NL_MC.ucn /y
if exist  %_iter_%_NHW_NL_MC.MTT    copy %_iter_%_NHW_NL_MC.MTT
%_iter_%_NHW_NL_MC.ucn /y
if exist  %_iter_%_NHO_NL_MC.MTT    copy %_iter_%_NHO_NL_MC.MTT
%_iter_%_NHO_NL_MC.ucn /y

del       %_iter_%_HBW_NL_MC.MTT
del       %_iter_%_HBS_NL_MC.MTT
del       %_iter_%_HBO_NL_MC.MTT
del       %_iter_%_NHW_NL_MC.MTT
del       %_iter_%_NHO_NL_MC.MTT

if exist  %_iter_%_HBW_NL_MC.con    copy %_iter_%_HBW_NL_MC.con
%_iter_%_HBW_NL_MC.MTT /y
if exist  %_iter_%_HBS_NL_MC.con    copy %_iter_%_HBS_NL_MC.con
%_iter_%_HBS_NL_MC.MTT /y
if exist  %_iter_%_HBO_NL_MC.con    copy %_iter_%_HBO_NL_MC.con
%_iter_%_HBO_NL_MC.MTT /y
if exist  %_iter_%_NHW_NL_MC.con    copy %_iter_%_NHW_NL_MC.con
%_iter_%_NHW_NL_MC.MTT /y
if exist  %_iter_%_NHO_NL_MC.con    copy %_iter_%_NHO_NL_MC.con
%_iter_%_NHO_NL_MC.MTT /y

::
:: Finally rerun the **constrained** mode choice output files
:: through the summary process and give the RPT, TAB, and TXT files new name
prefixes
::  '%_iter_%_MC_NL_CONSUMMARY' instead of '%_iter_%_MC_NL_SUMMARY'
::

if exist voya*.*  del voya*.*
if exist %_iter_%_MC_NL_CONSUMMARY.rpt   del       %_iter_%_MC_NL_CONSUMMARY.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_CONSUMMARY.rpt

..\software\extrtab                    %_iter_%_mc_NL_CONSUMMARY.rpt
if exist extrtab.out copy extrtab.out  %_iter_%_mc_NL_CONSUMMARY.tab /y
if exist extrtab.out del  extrtab.out
if exist temp.rpt del  temp.rpt
if exist *.tb1 copy *.tb1              %_iter_%_mc_NL_CONSUMMARY.txt /y
if exist *.tb1 del  *.tb1

goto end

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

3.7 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...
```

3.8 Time-of-Day.bat

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

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

```
if exist voya*.*  del voya*.*
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
        ..\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
..\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
/y
..\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..
```

3.9 Highway_Assignment.bat

```
CD %1

REM  Highway Assignment

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

copy Voya*.prn       %_iter_%_Highway_Assignment.rpt /y

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

3.10 Average_Link_Speeds.bat - Iterations (2- 4)

```
CD %1
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    Assign_Output_%_iter_%.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 /y
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 Assigned_%_iter_%HWY.net

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


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

3.11 Highway_Skims.bat

```
CD %1

REM  Highway Skims

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

3.12 HSR_Highway_Skims.bat

```
CD %1

REM  Highway Skims

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

:: Treatment of HOT lane facilities in Virginia
:: HOV3+ Skim Replacement (HSR)
:: 2011-03-28 Mon 10:29:57
:: We need to delete the HOV3+ highway skims that come out of highway_skims.s
:: then, we replace them with their counterpart from the base iteration, which
:: reflect HOV3+ operations, not HOT-lane operations.

:: AM skims
if exist hov3%_iter_%am_mc.skm del hov3%_iter_%am_mc.skm
copy %_HOV3PATH_%\hov3%_iter_%am_mc.skm hov3%_iter_%am_mc.skm /y

:: Midday skims
if exist hov3%_iter_%md_mc.skm del hov3%_iter_%md_mc.skm
copy %_HOV3PATH_%\hov3%_iter_%md_mc.skm hov3%_iter_%md_mc.skm /y
:: End of HOV3+ skim replacment


:: 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.rpt  del %_iter_%_Highway_Skims_mod.rpt
start /w Voyager.exe  ..\scripts\Highway_Skims_mod.s /start -Pvoya -S..\%1
if errorlevel 1 goto error
if exist voya*.prn  copy voya*.prn %_iter_%_Highway_Skims_Mod.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..
```

3.13 Transit_Assignment.bat

```
CD %1

:: Combine Mode Choice Output for Transit Assignment

if exist voya*.* del voya*.*
if exist %_iter_%_Combine_Tables_For_TrAssign.RPT del
%_iter_%_Combine_Tables_For_TrAssign.RPT
start /w voyager.exe ..\Scripts\Combine_Tables_For_TrAssign.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Combine_Tables_For_TrAssign.RPT /y

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

::  Transit Assignment Commuter Rail

if exist voya*.* del voya*.*
if exist %_iter_%_Transit_Assgn_CR.RPT del %_iter_%_Transit_Assgn_CR.RPT
start /w voyager.exe ..\Scripts\transit_assignment_CR.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Transit_Assgn_CR.RPT /y


::  Transit Assignment Metrorail

if exist voya*.* del voya*.*
if exist %_iter_%_Transit_Assgn_MR.RPT del %_iter_%_Transit_Assgn_MR.RPT
```

```
start /w voyager.exe ..\Scripts\transit_assignment_MR.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Transit_Assgn_MR.RPT /y


::  Transit Assignment All Bus

if exist voya*.* del voya*.*
if exist %_iter_%_Transit_Assgn_AB.RPT del %_iter_%_Transit_Assgn_AB.RPT
start /w voyager.exe ..\Scripts\transit_assignment_AB.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Transit_Assgn_AB.RPT /y

::  Transit Assignment Bus and Metrorail

if exist voya*.* del voya*.*
if exist %_iter_%_Transit_Assgn_BM.RPT del %_iter_%_Transit_Assgn_BM.RPT
start /w voyager.exe ..\Scripts\transit_assignment_BM.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
if exist voya*.prn copy voya*.prn %_iter_%_Transit_Assgn_BM.RPT /y

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

3.14 Transit_Skim_Select_Paths.bat

```
CD %1

REM  Transit Network Building (Final)
REM    -- for Selected Paths Only --
del tppl*.*
del transit_skims.rpt
start /w TPPLUS.EXE  ..\scripts\transit_skims_Select_Paths.s /start -Ptppl -S..\%1
if errorlevel 2 goto error
copy tppl*.prn  %_iter_%_TRANSIT_SKIMS_Select_Paths.RPT /y
goto end
:error
REM  Processing Error......
PAUSE
:end
CD..
```

3.15 TranSum.bat

```
:: TranSum.bat

::CD %1\TransitAssignment\Transum
CD %1\Transum
::CD %1

REM Step: Transit Volume Summary
REM Consolidate Peak and Off-Peak Volumes
::..\..\..\software\LINEVOL ..\..\..\controls\pk_vol.ctl
::..\..\..\software\LINEVOL ..\..\..\controls\op_vol.ctl
..\..\software\LINEVOL ..\..\controls\pk_vol.ctl
..\..\software\LINEVOL ..\..\controls\op_vol.ctl
REM       if errorlevel 1 goto error

REM Metro Rail Line Summary
::..\..\..\software\LINESUM ..\..\..\controls\linesumMR.ctl
..\..\software\LINESUM ..\..\controls\linesumMR.ctl
REM       if errorlevel 1 goto error

REM Metro Bus Line Summary
::..\..\..\software\LINESUM ..\..\..\controls\linesumMB.ctl
..\..\software\LINESUM ..\..\controls\linesumMB.ctl
REM       if errorlevel 1 goto error

REM Other Bus Line Summary
::..\..\..\software\LINESUM ..\..\..\controls\linesumOB.ctl
..\..\software\LINESUM ..\..\controls\linesumOB.ctl
REM       if errorlevel 1 goto error

REM Metro Station Access
::..\..\..\software\LINESUM ..\..\..\controls\access.ctl
..\..\software\LINESUM ..\..\controls\access.ctl
..\..\software\LINESUM ..\..\controls\accessPkHbw.ctl
REM       if errorlevel 1 goto error

REM Link Summary
::..\..\..\software\LINESUM ..\..\..\controls\linksum.ctl
..\..\software\LINESUM ..\..\controls\linksum.ctl
REM       if errorlevel 1 goto error

REM Total Transit Summary
::..\..\..\software\LINESUM ..\..\..\controls\total.ctl
..\..\software\LINESUM ..\..\controls\total.ctl
REM       if errorlevel 1 goto error

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

Appendix C. Cube Voyager Scripts

1 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)
                 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
;  doucumented 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 overide 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)

                 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
```

2 Assemble_Skims_AB.s

```
;--------------------------------------------------------------------------
;Assemble_Skims_AB.s
;MWCOG Version 2.3  Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter_%) = 'i1',..,'i6'
;   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,
  FORMAT = MINUTP,
  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 ----
```

3 Assemble_Skims_BM.s

```
;-------------------------------------------------------------------------
;Assemble_Skims_BM.s
;MWCOG Version 2.3  Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter_%) = 'i1',..,'i6'
;   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,
  FORMAT = MINUTP,
  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 ----
```

4 Assemble_Skims_CR.s

```
;-------------------------------------------------------------------------
;Assemble_Skims_CR.s
;MWCOG Version 2.3  Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter_%) = 'i1',..,'i6'
;   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,
  FORMAT = MINUTP,
  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        (1980 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 ----
```

5 Assemble_Skims_MR.s

```
;---------------------------------------------------------------------------
;Assemble_Skims_MR.s
;MWCOG Version 2.3  Model
;Assemble Transit Skims by Time Period
; Input Files:
;   iteration (%_iter_%) = 'i1',..,'i6'
;   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

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,
  FORMAT = MINUTP,
```

```
  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 ----
```

6 Autoacc4.s

```
*del voya*.prn
; AutoAcc4.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)


; Dimensions:
;
TAZSTASize  = 7000
IZsize      = 3675
FrstStaCen  = 5001
Stasize     = 1000


;;Input Files:
  AMSkimFile  =  'sovmam.skm'
  OPSkimFile  =  'sovmmd.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    =  'mrpram.asc  ' ;unit 21
  M_Knr_AM    =  'mrkram.asc  ' ;     22
  C_Pnr_AM    =  'cram.asc    ' ;     23
  B_Pnr_AM    =  'buspram.asc ' ;     24 renamed file
  B_Knr_AM    =  'buskram.asc ' ;     new file
  L_Pnr_AM    =  'lrtam.asc   ' ;     25
  N_Pnr_AM    =  'newam.asc   ' ;     26
  L_Knr_AM    =  'lrtkram.asc ' ;     43
  N_Knr_AM    =  'newkram.asc ' ;     44

  M_Pnr_OP    =  'mrprop.asc  ' ;     31
  M_Knr_OP    =  'mrkrop.asc  ' ;     32
  C_Pnr_OP    =  'crop.asc    ' ;     33
  B_Pnr_OP    =  'busprop.asc ' ;     34 renamed file
  B_Knr_OP    =  'buskrop.asc ' ;     new file
  L_Pnr_OP    =  'lrtop.asc   ' ;     35
  N_Pnr_OP    =  'newop.asc   ' ;     36
  L_Knr_OP    =  'lrtkrop.asc ' ;     47
  N_Knr_OP    =  'newkrop.asc ' ;     48
;
  AutoAll     =  'autoall.asc' ;      40
;
; Params:
```

```
  BackD   = 1000.00
  BackPC  =    0.30
  Divpc   =    1.30
  NCBD    =   35.00       ; Representative TAZ of the region's CBD

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@,
      ;STAOPCost  = @STASize@,
      ;STAPKShad  = @STASize@,
      ;STAOPShad  = @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
   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= extra1.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= tazxys.dbg

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

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

                       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))


                           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)

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

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

                               Print PRINTO=3 list = ' SUPPORT N=',I(5),'-
',stat[STADX](5),' DIST=',opDIST(6),
                                            ' ONEWAY=Y MODE=11 SPEED=',opSPD(4)
                           ENDIF                                ;;; 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)
                           ENDIF                                ;;; 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)

                               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
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 voya*.prn AutoAcc4.rpt
*copy extra1.xy+extra2.XY+extra3.xy extra.xy
*del  extra1.xy
*del  extra2.xy
*del  extra3.xy
```

7 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@AMVC,     @iter@PMVC,     @iter@MDVC,     @iter@NTVC,
                     @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] = Assign_Output_@iter@.net           ;; current  iteration LL file w/
traffic assigned speeds
NETO = Averaged_@iter@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,SPDCLASS)                                       ;
Freeflow AM speed
  @iter@PMFFSPD  =SPEEDFOR(PMLANE,SPDCLASS)                                       ;
Freeflow PM speed
  @iter@MDFFSPD  =SPEEDFOR(OPLANE,SPDCLASS)                                       ;
Freeflow MD speed
  @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

  AMHRLNCAP=CAPACITYFOR(1,CAPCLASS)                                               ;
Hrly Lane capacity
  PMHRLNCAP=CAPACITYFOR(1,CAPCLASS)                                               ;
Hrly Lane capacity
  MDHRLNCAP=CAPACITYFOR(1,CAPCLASS)                                               ;
Hrly Lane capacity
  NTHRLNCAP=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.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= Average_Link_Speeds_@iter@.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
     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
```

8 Combine_Tables_For_TrAssign.s

```
; COMBINE TRIPS FOR ALL PURPOSES INTO ONE FOR EACH SUB TRANSIT MODE
RUN PGM=MATRIX
MATI[1] = '%_iter_%_HBW_NL_MC.MTT' ;AECOM HBW Mode Choice file           (Input)
MATI[2] = '%_iter_%_HBS_NL_MC.MTT' ;AECOM HBS Mode Choice file           (Input)
MATI[3] = '%_iter_%_HBO_NL_MC.MTT' ;AECOM HBO Mode Choice file           (Input)
MATI[4] = '%_iter_%_NHW_NL_MC.MTT' ;AECOM NHW 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
```

9 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

ZNFILE_IN1  = 'inputs\ZONE.dbf'   ;  Input  Zonal Land Use File

ZNFILE_IN2  = 'AreaType_File.dbf'    ;  Input  Zonal Area Type File from network
building

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

ZNFILE_INa1  = '%_iter_%_AM_WK_MR_JOBACC.dbf'  ;  Input  Jobs accessible within 45
min. by AM WalkAcc  Metrorail Only Service
ZNFILE_INa2  = '%_iter_%_AM_DR_MR_JOBACC.dbf'  ;  Input  Jobs accessible within 45
min. by AM DriveAcc Metrorail Only Service
ZNFILE_INa3  = '%_iter_%_AM_WK_BM_JOBACC.dbf'  ;  Input  Jobs accessible within 45
min. by AM WalkAcc  Bus&Metrorail  Service
ZNFILE_INa4  = '%_iter_%_AM_DR_BM_JOBACC.dbf'  ;  Input  Jobs accessible within 45
min. by AM DriveAcc Bus&Metrorail  Service

ZNFILE_OU1  = 'HHI1_SV.txt'              ;  Output Zonal Income 1 HH by Size& VehAv
Classes: i1s1v1,i1s1v2,...,i1s4v4
ZNFILE_OU2  = 'HHI2_SV.txt'              ;  Output Zonal Income 2 HH by Size& VehAv
Classes: i2s1v1,i2s1v2,...,i2s4v4
ZNFILE_OU3  = 'HHI3_SV.txt'              ;  Output Zonal Income 3 HH by Size& VehAv
Classes: i3s1v1,i3s1v2,...,i3s4v4
ZNFILE_OU4  = 'HHI4_SV.txt'              ;  Output Zonal Income 4 HH by Size& VehAv
Classes: i4s1v1,i4s1v2,...,i4s4v4

ZonalCCHHs  = 'Demo_Models_HHbyISV_%_iter_%.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,                    ;  proportion of
             Interpolate = N, FAIL=0,0,0,                   ;  zonal median inc.
; inc level:   QRT1      QRT2     QRT3     QRT4             ; to regional median
income
 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
    "  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'
;             -----   ----                  'X'          '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





;===================================================================================
==================
; End of Lookups-  Now read the input files
=
;===================================================================================
==================

;
; read Zonal land use files  into Z-File
;

ZDATI[1] = @ZNFILE_IN1@ ;; variables in DBF file: TAZ,  HH, HHPOP, JURCODE, HHINCIDX

; Zonal Area Type File
ZDATI[2] = @ZNFILE_IN2@ ;; variables in DBF file: TAZ, ATYPE


; Zonal Transit Access. Files
  ZDATI[3] = @ZNFILE_INa1@ ;   TAZ, emp45
  ZDATI[4] = @ZNFILE_INa2@ ;   TAZ, emp45
  ZDATI[5] = @ZNFILE_INa3@ ;   TAZ, emp45
  ZDATI[6] = @ZNFILE_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
to AT 6
                                                    ; 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
;* 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
; 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
                   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=@ZNFILE_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 Iloop
;  --Print out input files to Trip Generation
;     4 income based files written in text form TAZ, HH by size&VeAv
s1v1,s1v2,...,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=@ZNFILE_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=@ZNFILE_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=@ZNFILE_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=@ZNFILE_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]
 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@ ;
   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@
;
   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@ ;
   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
```

10 Highway_Assignment.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:
  _iter_  (Iteration indicator = 'pp','i1' - 'i6')

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

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

PAGEHEIGHT=32767 ; preclude insertion of page headers

; Set up tokens to either use or comment out commands for Cube Cluster (distributed processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
  dp_token = ' '
else
  dp_token = ';'
endif

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

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

rel_gap =   0.001         ;  Relative gap threshold, normally set to 10E-3
mxIters = 300            ;  Max. number of user equilibrium (UE) iterations,
                         ;    normally set to 200

;;;*****************************************************************************
;;; 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'

LOOP PERIOD = 1,2  ; Loop thru 1 (AM) and 2 (PM);  Each pk per. includes NonHOV3+ and HOV3+

IF    (Period==1)        ; AM Peak Period
   PRD    = 'AM'       ;
   PCTADT =   41.7       ; %_AMPF_%  AM PHF (% of traffic in pk hr of period)
ELSE                   ; PM Peak Period
   PRD    = 'PM'       ;
   PCTADT =   29.4       ; %_PMPF_%  PM PHF (% of traffic in pk hr of period)
ENDIF

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 NonHOV3+ trip tables only
;;;          (SOV, HOV2, CV, TRUCK & AIRPORT PASSENGER TRIPS)
;;;*****************************************************************************

  RUN PGM=HIGHWAY  ; NonHOV3+ traffic assignment
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
  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=@rel_gap@   ; Set a relative gap tolerance
  PARAMETERS MAXITERS=@mxIters@      ; 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:       $
;------------------------------------------------------$
;
;
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.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),
;;       ' 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 1 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==1)
    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    + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ;SOV   IMP
       LW.HV2@PRD@IMP = TIME    + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ;HOV 2 IMP
;      LW.HV3@PRD@IMP = TIME    + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
;;-->>   LW.HV3@PRD@IMP = TIME    + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
       LW.CV@PRD@IMP  = TIME    + (LW.CV@PRD@TOLL/100.0) * CV@PRD@EQM ;CV    IMP
       LW.TRK@PRD@IMP = TIME    + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ;Truck IMP
       LW.APX@PRD@IMP = TIME    + (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),
           ' 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,        VOL[6]=MI.1.6   ;  Airport

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
 Fileo Printo[1] = "UE_Iteration_Report_NonHOV_%_iter_%_@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

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

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

 RUN PGM=HIGHWAY  ; HOV3+    traffic assignment
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 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 ;
 ;
 FILEO NETO=TEMP2@PRD@.NET         ; Output loaded network of current iter/time prd.
 PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
 PARAMETERS RELATIVEGAP=@rel_gap@   ; Set a relative gap tolerance
 PARAMETERS MAXITERS=@mxIters@      ; 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
;;;     ENDIF
;;;
   ENDIF


;$
   ;
   ;   The highway network is coded with limit codes from 1 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==1)
     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.HV3@PRD@IMP = TIME    + (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),
         ' 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] = "UE_Iteration_Report_NonHOV_%_iter_%_@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


ENDLOOP          ; Loop thru 1 (AM) and 2 (PM);  Each pk per. includes NonHOV3+ and HOV3+

;;;*****************************************************************************
;;; Step 2: Execute off-peak-period traffic assignments (midday/MD & night/NT)
;;;        All 6 trip tables are assigned together.
;;;*****************************************************************************


LOOP PERIOD = 3,4  ; Loop thru 3 (midday, MD) and 4 (Night, NT)

IF  (period==3)           ; Off-Peak Period
   PRD   =  'MD'        ;
   PCTADT =   17.7       ; %_MDPF_%  Midday PHF (% of traffic in pk hr of period)
ENDIF
IF  (period==4)           ; Off-Peak Period
   PRD   =  'NT'        ;
   PCTADT =   15.0       ; %_NTPF_%  NT PHF (% of traffic in pk hr of period)
ENDIF

CAPFAC=1/(PCTADT/100)               ; Capacity Factor = 1/(PCTADT/100)
; Turnpen  = 'inputs\turnpen.pen'       ; Turn penalty

 RUN PGM=HIGHWAY  ; Off-peak (midday & evening) traffic assignment
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 FILEI NETI     = @INPNET@                         ; 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 ;
 ;
 FILEO NETO=temp2@PRD@.net         ; Output loaded network of current iter/time prd. FOR OFF PEAK
 PARAMETERS COMBINE=EQUI ENHANCE=@assignType@
 PARAMETERS RELATIVEGAP=@rel_gap@   ; Set a relative gap tolerance
 PARAMETERS MAXITERS=@mxIters@      ; 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.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),
;;;       ' 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 1 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==1)
    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    + (LW.SOV@PRD@TOLL/100.0)* SV@PRD@EQM ;SOV   IMP
       LW.HV2@PRD@IMP = TIME    + (LW.HV2@PRD@TOLL/100.0)* H2@PRD@EQM ;HOV 2 IMP
       LW.HV3@PRD@IMP = TIME    + (LW.HV3@PRD@TOLL/100.0)* H3@PRD@EQM ;HOV 3+IMP
       LW.CV@PRD@IMP  = TIME    + (LW.CV@PRD@TOLL /100.0)* CV@PRD@EQM ;CV    IMP
       LW.TRK@PRD@IMP = TIME    + (LW.TRK@PRD@TOLL/100.0)* TK@PRD@EQM ;Truck IMP
       LW.APX@PRD@IMP = TIME    + (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),
           ' 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,        VOL[6]=MI.1.6   ;  Airport

;$

ENDPHASE

PHASE=ADJUST

ENDPHASE

PHASE=CONVERGE
 Fileo Printo[1] = "UE_Iteration_Report_NonHOV_%_iter_%_@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

ENDLOOP          ; Loop thru 3 (midday, MD) and 4 (evening/off-peak, OP)


;
; 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
  %_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
  @prd@HRLNCAP=CAPACITYFOR(1,CAPCLASS)                  ;  Hrly 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,WFSPD,_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@HRLNCAP(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@LLNK.ASC

;;
```

```
ENDRUN
ENDLOOP          ; Loop thru 1 (AM) and 2 (PM);  Each pk per. includes NonHOV3+ and HOV3+

;;;*****************************************************************************
;;; 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,SPDCLASS)        ; Freeflow speed
 @prd@HRLKCAP=CAPACITYFOR(@prd@LANE,CAPCLASS)             ; Hrly LINK capacity
 @prd@HRLNCAP=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,WFSPD,_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@HRLNCAP(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@LLNK.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]=tempAM.net
  FILEI NETI[2]=tempMD.net
  FILEI NETI[3]=tempPM.net
  FILEI NETI[4]=tempNT.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
      %_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
```

11 Highway_Skims.s

```
;//////////////////////////////////////////////////////////////;
;  Highway_Skims.S                                          //;
;  MWCOG Version 2.3 Model                                  //;
;                                                           //;
;  Build AM Peak/Off-Peak Highway Skims                     //;
;  the Current Iteration Asssignment                        //;
;  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
;
;//////////////////////////////////////////////////////////////;
;
;
;   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=1,2         ;  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   = 'sov%_iter_%am.skm '
    MATOUT2   = 'hov2%_iter_%am.skm'
    MATOUT3   = 'hov3%_iter_%am.skm'

    MATOUTMC1 = 'sov%_iter_%am_MC.skm '
    MATOUTMC2 = 'hov2%_iter_%am_MC.skm'
    MATOUTMC3 = 'hov3%_iter_%am_MC.skm'

    MYID      = '%_iter_% AM skims'

    TT        = ';'
    MATOUT4   = ' '
    SKMTOT    = ' '

  ELSEIF (Period=2)     ; MD Highway Skim tokens
    PRD       = 'MD'
    MATOUT1   = 'sov%_iter_%md.skm '
    MATOUT2   = 'hov2%_iter_%md.skm'
    MATOUT3   = 'hov3%_iter_%md.skm'

    MATOUTMC1 = 'sov%_iter_%md_MC.skm '
    MATOUTMC2 = 'hov2%_iter_%md_MC.skm'
    MATOUTMC3 = 'hov3%_iter_%md_MC.skm'

    TT        = ' '
    MATOUT4   = 'trk%_iter_%md.skm'
    SKMTOT    = 'skimtot%_iter_%.dat'

    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)
          T1 = 0
       ELSE
          T1 = LI.DISTANCE / SPEED * 60.0
       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
;
; 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]

      MW[103] = MAX(0,MW[103])
      MW[203] = MAX(0,MW[203])
      MW[303] = MAX(0,MW[303])
 ENDRUN
```

```
; end of truck access section
ENDLOOP
```

12 Highway_Skims_mod.s

```
;//////////////////////////////////////////////////////////////;
;  Highway_Skims_Mod.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')
;//////////////////////////////////////////////////////////////;
;
;
;   Environment Variables:
;     _iter_  (Iteration indicator = 'pp','i1'-'i4')
;
;
pageheight=32767  ; Preclude header breaks
NETIN    = '%_iter_%hwymod.net'

LOOP Period=1,2         ;  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 = 'sovm%_iter_%am.skm '
    MATOUT2 = 'hov2m%_iter_%am.skm'
    MATOUT3 = 'hov3m%_iter_%am.skm'
    MYID    = '%_iter_% AM skims'
  ELSE                  ; MD Highway Skim tokens
    PRD     = 'MD'
    MATOUT1 = 'sovm%_iter_%MD.skm '
    MATOUT2 = 'hov2m%_iter_%MD.skm'
    MATOUT3 = 'hov3m%_iter_%MD.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
       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.
;---------------------------------------------------------------------


     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
```

13 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.
;
; _HOV3Path_  environment variable is used to override HOV3 Skims from another
Subdirectory
;
pageheight=32767  ; Preclude header breaks


RUN PGM=MATRIX
    MATI[1]=         sov%_iter_%am_MC.skm
    MATI[2]=         hov2%_iter_%am_MC.skm
    MATI[3]=         hov3%_iter_%am_MC.skm

    MATI[4]=         sov%_iter_%md_MC.skm
    MATI[5]=         hov2%_iter_%md_MC.skm
    MATI[6]=         hov3%_iter_%md_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] = hwy%_iter_%am.skm, MO=1-9,
                                 name=SovTime,SOVDst10,SOVToll,
                                      Hv2Time,Hv2Dst10,HV2Toll,
                                      Hv3Time,Hv3Dst10,HV3Toll

   MATO[2] = hwy%_iter_%op.skm, MO=10-18,
                                 name=SovTime,SOVDst10,SOVToll,
                                      Hv2Time,Hv2Dst10,HV2Toll,
                                      Hv3Time,Hv3Dst10,HV3Toll
ENDRUN
```

14 MC_Auto_Drivers.s

```
; ===================================================================
;  Version 2.3
;  MC_Auto_Drivers.s
;  This program is used to develop 1-occ, 2-occ, and 3+occ auto driver
;  trip tables, by purpose (HBW, HBS, HBO, and NHB).  The script reads two files:
;  1) Internal Auto Person Trips - The AECOM NL Mode choice output, each file
;     contains auto person trips by occupancy group (1,2,and 3+ Occupant Vehicles).
;  2) External Auto Person trips - the trip distibution output containing
;     total auto person trips.
; ===================================================================
;
;
;///////////////////////////////////////////////////////////////////
;

Zonesize   = 3722
FstExtZn   = 3676

; First, establish Input/Output filenames:
LOOP PURP=1,5   ;  We'll Loop 5 times, for each purpose
                ;
IF (PURP=1)    ; HBW Loop
 MCFILE   = '%_iter_%_HBW_NL_MC.MTT' ;AECOM Mode Choice file           (Input)
 TDFILE   = 'HBW_%_iter_%_.PTT'      ;Trip distibution output          (Input)
 MC123OCC = 'HBW%_iter_%.ADR'        ;HBW Auto Drv trips- 1,2,3+ Occ.  (Output)
 PURPOSE  = 'HBW'                    ;
 Avg3P_Occ=  3.50                    ; Avg Auto Occupancy for autos w/ 3+ person
 ExtCarOcc=  1.15                    ; Avg External Auto Occ.
 TDTab    =  '6'                     ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=2) ; HBS Loop
 MCFILE   = '%_iter_%_HBS_NL_MC.MTT' ;AECOM Mode Choice file           (Input)
 TDFILE   = 'HBS_%_iter_%_.PTT'     ;Trip distibution output          (Input)
 MC123OCC = 'HBS%_iter_%.ADR'        ;HBW Auto Drv trips- 1,2,3+ Occ.  (Output)
 PURPOSE  = 'HBS'                    ;
 Avg3P_Occ=  3.50                    ; Avg Auto Occupancy for autos w/ 3+ person
 ExtCarOcc=  1.64                    ; Avg External Auto Occ.
 TDTab    =  '6'                     ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=3) ; HBO Loop
 MCFILE   = '%_iter_%_HBO_NL_MC.MTT' ;AECOM Mode Choice file           (Input)
 TDFILE   = 'HBO_%_iter_%_.PTT'     ;Trip distibution output          (Input)
 MC123OCC = 'HBO%_iter_%.ADR'        ;HBW Auto Drv trips- 1,2,3+ Occ.  (Output)
 PURPOSE  = 'HBO'                    ;
 Avg3P_Occ=  3.50                    ; Avg Auto Occupancy for autos w/ 3+ person
 ExtCarOcc=  1.61                    ; Avg External Auto Occ.
 TDTab    =  '6'                     ; Total Psn Trip tab no. in Trip Dist. output
file

ELSEIF (PURP=4) ; NHW Loop
 MCFILE   = '%_iter_%_NHW_NL_MC.MTT' ;AECOM Mode Choice file           (Input)
 TDFILE   = 'NHW_%_iter_%_.PTT'     ;Trip distibution output          (Input)
 MC123OCC = 'NHW%_iter_%.ADR'        ;HBW Auto Drv trips- 1,2,3+ Occ.  (Output)
 PURPOSE  = 'NHW'                    ;
 Avg3P_Occ=  3.50                    ; Avg Auto Occupancy for autos w/ 3+ person
 ExtCarOcc=  1.28                    ; Avg External Auto Occ.
 TDTab    =  '3'                     ; Total Psn Trip tab no. in Trip Dist. output
file


ELSEIF (PURP=5) ; NHO Loop
 MCFILE   = '%_iter_%_NHO_NL_MC.MTT' ;AECOM Mode Choice file           (Input)
 TDFILE   = 'NHO_%_iter_%_.PTT'     ;Trip distibution output           (Input)
 MC123OCC = 'NHO%_iter_%.ADR'        ;HBW Auto Drv trips- 1,2,3+ Occ.  (Output)
 PURPOSE  = 'NHO'                    ;
 Avg3P_Occ=  3.50                    ; Avg Auto Occupancy for autos w/ 3+ person
 ExtCarOcc=  1.28                    ; Avg External Auto Occ.
 TDTab    =  '3'                     ; Total Psn Trip tab no. in Trip Dist. output
file

ENDIF
;
;///////////////////////////////////////////////////////////////////

RUN PGM=MATRIX
  PAGEHEIGHT= 32767

    MATI[1]=@MCFILE@           ;  MODE CHOICE MODEL OUTPUT FILE (for INTL TRIPS)
```

```
    MATI[2]=@TDfILE@          ;  TRIP DISTRIBUTION OUTPUT FILE (for EXTL TRIPS)

    ; put INTERNAL 1,2,3+ OCC AUTO PERSON TRIPS IN MTX 1,2,3
        FILLMW MW[1]  = MI.1.1,2,3

    ; compute internal auto driver trips, by occ group in mtx 11,12,13
        MW[11]  = MW[1] / 1.0              ;;  intl 1-occ. auto drivers
        MW[12]  = MW[2] / 2.0              ;;  intl 2-occ. auto drivers
        MW[13]  = MW[3] / @Avg3P_Occ@      ;;  intl 3+occ. auto drivers

    ; put TOTAL motorized person trips in mtx 20.
        MW[20] = MI.2.@TDtab@

    ; the external portion(auto person trips) will be extracted from mtx 20, and put
into 30
    ; .
        IF (I <  @FstExtZn@)  MW[22]  = 1.0, include = @FstExtZn@-@Zonesize@ ;
        IF (I >= @FstExtZn@)  MW[22]  = 1.0, exclude = @FstExtZn@-@Zonesize@ ;

        MW[30] = MW[20] * MW[22]        ;; Extl auto person trips

    ; compute external auto driver trips in mtx 40, and apportion among occ groups
    ; using standard occ. curves

        MW[40] = MW[30] / @ExtCarOcc@    ;; Extl Auto driver trips

JLOOP
  XCarOcc =@ExtCarOcc@
; Determine LOV Vehicles in 1,2,3&4+ occupant groups using model
; COG's disaggrgegation model.

        IF     (XCarOcc  < 1.0050) ;  Make sure the computed Car Occ.
                XCarOcc  = 1.0050  ;  is between 1.005 and 2.500
        ELSEIF (XCarOcc  > 2.5000) ;  -- if not establish boundary
                XCarOcc  = 2.5000  ;      conditions
        ENDIF
;
; Apply Car Occ. Pct Model-Computes Pct Vehs.in Occ groups as function
; of avg auto occ.  The function is continuous  but piecewise.
;
        IF     (XCarOcc = 1.0050 - 1.1199999)
         MW[21] =  2.00264 - (0.9989 * XCarOcc) ; Shr of 1-Occ Vehs
         MW[22] = -1.00050 + (0.9952 * XCarOcc) ; Shr of 2-Occ Vehs
         MW[23] = -0.00158 + (0.0029 * XCarOcc) ; Shr of 3-Occ Vehs
         MW[24] = -0.00056 + (0.0008 * XCarOcc) ; Shr of 4-Occ Vehs
        ELSEIF (XCarOcc = 1.1200 - 2.5000)
         MW[21] =  1.59600 - (0.6357 * XCarOcc) ; Shr of 1-Occ Vehs
         MW[22] = -0.31143 + (0.3800 * XCarOcc) ; Shr of 2-Occ Vehs
         MW[23] = -0.17082 + (0.1540 * XCarOcc) ; Shr of 3-Occ Vehs
         MW[24] = -0.11375 + (0.1017 * XCarOcc) ; Shr of 4-Occ Vehs
        ENDIF

        ;
        ;   Apply Modeled Shares to the Extl Auto Drivers in mtx 51-54

         MW[51] =(MW[21] * MW[40]) ; Estimated Extl 1 occ vehicles
         MW[52] =(MW[22] * MW[40]) ; Estimated Extl 2 occ vehicles
         MW[53] =(MW[23] * MW[40]) ; Estimated Extl 3 occ vehicles
         MW[54] =(MW[24] * MW[40]) ; Estimated Extl 4+occ vehicles


; compute add intl and extl 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 Extl 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='        '
 LIST='===  END OF ','@PURPOSE@',' ITERATION: ','%_iter_%',' AUTO DRV RESULTS ==='
 LIST='/et     '
ENDIF

MATO=@MC123OCC@,MO=61,62,63 ; output file designation

ENDRUN
ENDLOOP
```

15 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
;     8 KNR-CR
;     9 PNR-BUS
;    10 KNR-BUS
;    11 PNR-BU/MR
;    12 KNR-BU/MR
;    13 PNR-MR
;    14 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 ; Total Purpose Mode Choice Trips
ENDRUN


;---------------------------------------------------------------
; 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.tb1'
    MCOUTTAB='%_iter_%_HBW_NL_MC.MTT'
    PURPOSE ='Internal HBW Trips'
ELSEIF (PURP=2)
    pur = 'HBS'
    purfile = 'B_HBS.tb1'
    MCOUTTAB='%_iter_%_HBS_NL_MC.MTT'
    PURPOSE ='Internal HBS Trips'
ELSEIF (PURP=3)
    pur = 'HBO'
    purfile = 'C_HBO.tb1'
    MCOUTTAB='%_iter_%_HBO_NL_MC.MTT'
    PURPOSE ='Internal HBO Trips'
ELSEIF (PURP=4)
    pur = 'NHW'
    purfile = 'D_NHW.tb1'
    MCOUTTAB='%_iter_%_NHW_NL_MC.MTT'
    PURPOSE ='Internal NHW Trips'
ELSEIF (PURP=5)
    pur = 'NHO'
    purfile = 'E_NHO.tb1'
    MCOUTTAB='%_iter_%_NHO_NL_MC.MTT'
    PURPOSE ='Internal NHO Trips '
ELSEIF (PURP=6)
    pur = 'ALL'
    purfile = 'F_ALL.tb1'
    MCOUTTAB='%_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 Jurisiction-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 Jurisiction-to-TAZ equivalency --
ENDCOPY
```

```
RUN PGM=MATRIX
  PAGEHEIGHT= 32767
  ZONES=3722
  MATI[1]= @MCOUTTAB@


  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/placemarker 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)

    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@;
         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
  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
       ENDLOOP
     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],
    '      ',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 'Inner' Loop
ENDLOOP ; End 'Outer' Loop
```

16 Metrorail_skims.s

```
;==================================================================
;  Metrorail_skims.S
;  MWCOG Version 2.3 Model
;
;  Step 1: Build Metrorail Staion to Station Network
;  Step 2: Build Distance skims (in 1/100s mi) to be used in the
;  MFARE1 process
;  set metrorail link file to new input name
;==================================================================
; max 'zones' (stations changed from 116 to 150)

;  Global variables:

NZONES = 150           ; Max. no. of Stations
NNODES = 10000         ; Max. no. of NODES

NODIN='METNODM1.TB'    ; Input Station Nodes
LNKIN='METLNKM1.TB'   ; Input Station Links
DSKMO='rldist.skm'     ; Output Distance Skim File
TPENS='inputs\trnpen.dat'     ; Turn Penalty file


;==================================================================
; Step 1: Build Metrorail Network
;==================================================================

RUN PGM=NETWORK
;
ZONES=@NZONES@
NODES=@NNODES@

; Node Coordinate File
; XY Units are NAD83 (in whole feet)
FILEI NODEI=@NODIN@,
        VAR=N,11-14,
        VAR=X,20-27,
        VAR=Y,34-40

; Metrorail Links
FILEI LINKI=@LNKIN@,
        VAR=A,13-17,         ; A-Node Number
        VAR=B,22-26,         ; B-Node Number
        VAR=REV,35-35,       ; Reverse Code
        VAR=DISTANCE,43-47,  ; Distance in 1/100ths of Miles
        VAR=SPEED,67-71      ; Speed Value (mph)

; output network in TP+ format
NETO=metrail.TPN
;

;==================================================================
; Step 2: Build Station Level Distance Skims
;==================================================================

RUN PGM=HIGHWAY
  NETI   =metrail.tpn        ; Metrorail Network
  MATO[1]=@DSKMO@,MO=1,
          FORMAT=MINUTP
  TURNPENI=@TPENS@


  PHASE=LINKREAD
       SPEED   = LI.SPEED            ; Use Link Coded Speed
       DISTANCE= LI.DISTANCE / 100   ; Set Distance in 1/100ths of mi to true mi
  ENDPHASE
;
; Now create station-to-station distance skims over minimum time
; paths.  The distance skims are in 100ths of miles
; (e.g. a skim value of '145' indicates 1.45 miles)
;
;
  PHASE=ILOOP

   PATHLOAD PATH=TIME, PENI=1,  TRACE=(I=64 && J=37),

              MW[1]=PATHTRACE(LI.DISTANCE), noaccess = 0
;-------------------------------------------------------------------
; I will print selected rows of skim files
;-------------------------------------------------------------------
     IF (i = 1-2)                       ;  for select rows (Is)
         printrow MW=1, j=1-@NZONES@   ;  print work matrices 1-3
     ENDIF                              ;  row value to all Js.
   ENDPHASE
ENDRUN
```

17 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/js that are 'unused

;     Calculate airline distance (MW[2]) in 100s of miles
      IxCrd  = StaXYS(1,I)
      JxCrd  = StaXYS(1,J)
      IyCrd  = StaXYS(2,I)
      JyCrd  = StaXYS(2,J)
      MW[2]  = ((IxCrd-JxCrd)^2 + (IyCrd-JyCrd)^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-defalted 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]
```

```
                    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
```

18 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 = ' '
```

```
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 '    ;
                                                     ;; METRORAIL 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 '    ;
```

```
      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 '    ;

                                                    ;; ALL BUS/METRORAIL FARES
=============================================
    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='mwcog', 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  =
;                                                             =
;===============================================================
```

```
;
     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 pcts
('1000'=100%)
            LOOKUP[04]   = 1,Result = 5,   ;  Dest Walk Pct in 10ths of pcts
('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(undefl)    /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
```

```
          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 acces 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  ;
undeflated transit fare, Bus-Only paths
         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

    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:

      RJBFZ1 = TAZLOOK(5,JSTA)
```

```
      RJBFZ2 = TAZLOOK(6,JSTA)
      IF (RJBFZ2 = 0) RJBFZ2 = RJBFZ1

       _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,RJBFZ1),BusFrMTX(JBFZ1,RJBFZ2))
       _EgrFare2 = MIN(BusFrMTX(JBFZ2,RJBFZ1),BusFrMTX(JBFZ2,RJBFZ2))
       _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
```

19 Misc_Time-of-Day.S

```
; ===================================================================
;  Misc_Time-of-Day.s
;  MWCOG 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','i1'-'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       //
SchlADR  = 'inputs\schl.adr'        ;  School  A.Dr Trips       //
;
COMTDOUT = 'COM_%_iter_%_.pTT'        ;  Comm Vehs t1-Intl, t2-Extl
MTKTDOUT = 'MTK_%_iter_%_.pTT'        ;  Med  Trks t1-Intl, t2-Extl
HTKTDOUT = 'HTK_%_iter_%_.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  = 'MISCAM%_iter_%.TT'           ; AM     Non-Modeled Trips  //
MISCMD  = 'MISCMD%_iter_%.TT'           ; Midday Non-Modeled Trips  //
MISCPM  = 'MISCPM%_iter_%.TT'           ; PM     Non-Modeled Trips  //
MISCOP  = 'MISCNT%_iter_%.TT'           ; Night  Non-Modeled Trips  //
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; Each output file contains 8 tables -                              //
; 1/xx truck,2/xx autodr,3/taxi adr,4/visitor adr,
; 5/med. truck, 6/hvy truck, 7/air passenger adr, 8/comm veh
;////////////////////////////////////////////////////////////////////
;-------------------------------------------------------------------
;;=======================================================
;;  Begin com veh, med, hvy truck time of day processing
;;=======================================================
run pgm=matrix
  pageheight=32767  ; Preclude header breaks
  id = "Commercial time of day + delta

  mati[1] = @COMTDOUT@
  mati[2] = @XXCVTRK@
  mati[3] = @CVDELTA@

  mato    = tempcom.trp, mo=61-68


 ; set up mtx 100, 200 to identify I-X, and X-I ijs respectively
  MW[100] = 0
  MW[200] = 0
  if (I=1-@LastIzn@)
      MW[100] = 1, include= @FExt@-@zonesize@
     else
      MW[200] = 1, include= 1-@LastIzn@
  endif

; I/I trips are already balanced, so we can apply a single factor
;  to all trips.  Apply separate P/A and A/P factors to externals.
;  Assume externals are 70/30 inbound (X/I, or A/P) in the morning,
;  70/30 outbound (I/X, P/A) in the evening.  Off-peak is 50/50.
;
; Note: the External(I-X,X-I) trips are multiplied by 2.0 as the CV model
; (i.e., delta table) was developed this way - rm 4/30/08

  mw[1] = mi.1.1              ; I/I CV trips
  mw[2] = mi.1.2   * mw[100]  ; Int P/ Ext A (outbound)     Delta trip table
reflects 1/2 total trips
  mw[3] = mi.1.2   * mw[200]  ; Ext A/ Int P (inbound)      Delta trip table
reflects 1/2 total trips

; Also add in the X/X's.
  mw[4] = mi.2.1

; Read and transpose the external delta
  mw[11] = mi.3.1 ; I/I
  mw[12] = mi.3.2 ; Int P/ Ext A (outbound)
  mw[13] = mi.3.2.t ; Ext A/ Int P (inbound)

; Add in the deltas.  First, for I/I and I/X.
  if (i = 1-@LastIZN@)
    jloop
      mw[21] = max(mw[1] + mw[11],0)
      mw[22] = max(mw[2] + mw[12],0)
    endjloop
  endif

  if (i > @LastIZN@)
; Now for Ext transposed (X/I).
    mw[23] = max(mw[3] + mw[13],0), include = 1-@LastIZN@
```

```
; Now for X/X.
    mw[24] = max(mw[4] + mw[12],0), include = @FExt@-@ZONESIZE@
  endif

; Sum I/I and External here (Total auto drv. distribution from 2007/08 HTS)
  mw[61]  = 0.18700  * (mw[21] + 0.70 * mw[23] + 0.30 * mw[22])  ; AM  Commercial
Vehs.
  mw[62]  = 0.32630  * (mw[21] + 0.50 * mw[23] + 0.50 * mw[22])  ; MD  Commercial
Vehs.
  mw[63]  = 0.32890  * (mw[21] + 0.30 * mw[23] + 0.70 * mw[22])  ; PM  Commercial
Vehs.
  mw[64]  = 0.15780  * (mw[21] + 0.50 * mw[23] + 0.50 * mw[22])  ; OP  Commercial
Vehs.

; Keep X/X separate
  mw[65]  = 0.18700  *  mw[24]
  mw[66]  = 0.32630  *  mw[24]
  mw[67]  = 0.32890  *  mw[24]
  mw[68]  = 0.15780  *  mw[24]

endrun

;-------------------------------------------------------------------
run pgm=matrix

  id = "Truck time of day + delta

  mati[1] = @MTKTDOUT@
  mati[4] = @HTKTDOUT@
  mati[2] = @XXCVTRK@
  mati[3] = @TKDELTA@

  mato    = temptrk.trp, mo=71-86


 ; set up mtx 100, 200 to identify I-X, and X-I ijs respectively
  MW[100] = 0
  MW[200] = 0
  if (I=1-@LastIzn@)
      MW[100] = 1, include= @FExt@-@zonesize@
     else
      MW[200] = 1, include= 1-@LastIzn@
  endif

; I/I trips are already balanced, so we can apply a single factor
;  to all trips.  Apply separate P/A and A/P factors to externals.
;  Assume externals are 70/30 inbound (X/I, or A/P) in the morning,
;  70/30 outbound (I/X, P/A) in the evening.  Off-peak is 50/50.
  mw[1] = mi.1.1                  ; MTK I/I
  mw[2] = mi.1.2    * mw[100]     ; MTK Int P/ Ext A (outbound)  Delta trip table
reflects 1/2 total trips
  mw[3] = mi.1.2    * mw[200]     ; MTK Ext A/ Int P (inbound)   Delta trip table
reflects 1/2 total trips

  mw[4] = mi.4.1                  ; HTK I/I
  mw[5] = mi.4.2    * mw[100]     ; HTK Int P/ Ext A (outbound)  Delta trip table
reflects 1/2 total trips
  mw[6] = mi.4.2    * mw[200]     ; HTK Ext A/ Int P (inbound)   Delta trip table
reflects 1/2 total trips

; Also add in the X/X's.
  mw[7] = mi.2.2 ; MTK
  mw[8] = mi.2.3 ; HTK

; Read and transpose the external delta.
  mw[21] = mi.3.1          ;  mi.3.mtkii
  mw[22] = mi.3.2          ;  mi.3.mtkext
```

```
  mw[23] = mi.3.2.t         ;  mi.3.mtkext.t
  mw[24] = mi.3.3           ;  mi.3.mtkxx
                            ;
  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]=@SchlADR@            ; 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

    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)

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_APAdr,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_APAdr,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_APAdr,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_APAdr,OP_ComV
e
ENDRUN
;
*del tempcom.trp
*del temptrk.trp
```

20 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
```

21 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
;  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 =
37, 283, 623, 662, 717, 1342, 1472, 1496, 1501, 1599, 1679, 1843, 2112)
```

22 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    HBWPP.ADR
;        2    HBSPP.ADR
;        3    HBOPP.ADR
;        4    NHWPP.ADR
;        5    NHOPP.ADR
;;                  ... 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     = 'hbw_%_iter_%_.ptt'            ;Trip distibution output
(Input)
 MC123OCC   = 'HBW%_iter_%.ADR'        ;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
 TDTab      =  '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     = 'hbs_%_iter_%_.ptt'            ;Trip distibution output
(Input)
 MC123OCC   = 'HBS%_iter_%.ADR'        ;HBW 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
 TDTab      =  '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     = 'hbo_%_iter_%_.ptt'            ;Trip distibution output
(Input)
 MC123OCC   = 'HBO%_iter_%.ADR'        ;HBW 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
 TDTab      =  '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     = 'nhw_%_iter_%_.ptt'            ;Trip distibution output
(Input)
 MC123OCC   = 'NHW%_iter_%.ADR'        ;HBW 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
 TDTab      =  '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     = 'nho_%_iter_%_.ptt'            ;Trip distibution output
(Input)
 MC123OCC   = 'NHO%_iter_%.ADR'        ;HBW 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
 TDTab      =  '3'                     ; Total Psn Trip tab no. in Trip Dist. output
file

ENDIF
;
;//////////////////////////////////////////////////////////////////
; Step 1:
```

```
;  - First read trip distribution person trips (from which auto drivers aree 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
  array  TDmkt_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 1occ  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 idex 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

       TDmkt_trips[5][mkt] =  TDmkt_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)
          TDmkt_trips[1][mkt] = TDmkt_trips[5][mkt] *  NLmkt_trips[1][mkt] /
NLmkt_trips[5][mkt]; est 1 occapsn Trip Dist Psn trips
          TDmkt_trips[2][mkt] = TDmkt_trips[5][mkt] *  NLmkt_trips[2][mkt] /
NLmkt_trips[5][mkt]; est 2 occapsn Trip Dist Psn trips
          TDmkt_trips[3][mkt] = TDmkt_trips[5][mkt] *  NLmkt_trips[3][mkt] /
NLmkt_trips[5][mkt]; est 3+occpsn  Trip Dist Psn trips
          TDmkt_trips[4][mkt] = TDmkt_trips[5][mkt] *  NLmkt_trips[4][mkt] /
NLmkt_trips[5][mkt]; est Transit   Trip Dist Psn trips
       ELSE
          TDmkt_trips[1][mkt] = TDmkt_trips[5][mkt] * @adroccshr1@
          TDmkt_trips[2][mkt] = TDmkt_trips[5][mkt] * @adroccshr2@
          TDmkt_trips[3][mkt] = TDmkt_trips[5][mkt] * @adroccshr3@
      ENDIF
ENDLOOP


;; compute TD auto driver shares
loop Mkt= 1,21
     IF ( TDmkt_trips[5][mkt] > 0)
        TDmkt_share[1][mkt] =  TDmkt_trips[1][mkt] /TDmkt_trips[5][mkt]
        TDmkt_share[2][mkt] =  TDmkt_trips[2][mkt] /TDmkt_trips[5][mkt]
        TDmkt_share[3][mkt] =  TDmkt_trips[3][mkt] /TDmkt_trips[5][mkt]
        TDmkt_share[4][mkt] =  TDmkt_trips[4][mkt] /TDmkt_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    = TDmkt_trips[1][mkt]  ; auto psn 1 occ trips
                    ro.TDPsn2    = TDmkt_trips[2][mkt]  ; auto psn 2 occ trips
                    ro.TDPsn3    = TDmkt_trips[3][mkt]  ; auto psn 3+occ trips
                    ro.TDTrn     = TDmkt_trips[4][mkt]  ; transit        trips
                    ro.TDPsn     = TDmkt_trips[5][mkt]  ; person         trips
```

```
                    ro.TDPsn1shr = TDmkt_share[1][mkt]  ; auto psn 1 occ trips share
                    ro.TDPsn2shr = TDmkt_share[2][mkt]  ; auto psn 2 occ trips share
                    ro.TDPsn3shr = TDmkt_share[3][mkt]  ; auto psn 3+occ trips share
                    ro.TDTrnshr  = TDmkt_share[4][mkt]  ; transit trip        share

   WRITE RECO=1
ENDLOOP
endif

  FILEO MATO[1]  = Market1_21.Mtx, MO=99
ENDRUN
;

;///////////////////////////////////////////////////////////////////
; Step 2:
;  - 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
      ENDLOOP
  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
         mw[302] = MW[101] * TDmkt_share[2][mkt]                ; zonal 2-occ auto
persons
         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
      ELSE                                  ;; apply external default pcts
         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] = @Purpose@%_iter_%.ADR,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
                    ro.OTDpsn3    = OTDmkt_trips[3][mkt]  ; OUTPUT  auto drv 3+occ
trips
                    ro.OTDTrn     = OTDmkt_trips[4][mkt]  ; OUTPUT  transit
trips
                    ro.OTDadr1    = OTDmkt_trips[5][mkt]  ; OUTPUT  auto drv 1 occ
trips
                    ro.OTDadr2    = OTDmkt_trips[6][mkt]  ; OUTPUT  auto drv 2 occ
trips
                    ro.OTDadr3    = OTDmkt_trips[7][mkt]  ; OUTPUT  auto drv 3+occ
trips
                    ro.OTDPsn     = OTDmkt_trips[8][mkt]  ; Output  person
trips

                    ro.ITDPsn1    = TDmkt_share[5][mkt]  ; INPUT auto Psn1occ
                    ro.ITDPsn2    = TDmkt_share[6][mkt]  ; INPUT auto Psn2occ
                    ro.ITDPsn3    = TDmkt_share[7][mkt]  ; INPUT auto Psn3+occ
                    ro.ITDtrn     = TDmkt_share[8][mkt]  ; INPUT transit
                    ro.ITDpsn     = TDmkt_share[9][mkt]  ; INPUT person


         WRITE RECO=1
     ENDLOOP
  ENDIF
ENDRUN
ENDLOOP
```

23 prefarV23.s

```
;==================================================================
;  PREFAREV23.S
;   Program to read Zone File Used for MFARE2 Program (without walk pcts)
;        and to merge in walk pct. information
;       (Conversion of FORTRAN program Prefaretp.FOR)
;        Program also prepares the Z-file for the NL Mode Choice model (File 8)
;
;   Programmer: Milone
;   Date:        12/11/10
;
;    The program reads 3 files:
;                - a GIS-based walk area file containing short and
;                  long walk areas to all rail stations
;                  (rail includes metro & commuter rail).  The file also
;                  contains the sht,lng distances to the nearest metrorail
;                  station.  Note: the walk distance is based on 1.0 mile
;                  radius per the V2 models (NOT 7/10 mile per V1 models)
;                - a zone file containing bus fare zone/station equivs and
;                  jurisdiction code information.  This is essentially
;                  an A2 deck without walk percentages
;                - the 'final' zonal walk percentage file written
;                  by the wlklnktp.exe program.  This will suppress
;                  metrorail walk percentages to be consistent with
;                  the walk access links built previously
;    It writes out:
;               - A 'complete' A2 file for the MFARE2.S
;                  process
;    1/31/08 rm / a quality control check section added at the bottom
;    4/10/08 rm / added procedure to prepare the A1 file for the NL Mode choice
;                 application (Note: must use updated Ctl files)

;
ZONESIZE        =  3675                  ;  internal zones
ZNFILE_TrPcts   = 'NLwalkPct.txt' ;  Input Zonal Transit Walk Pcts
Fare_Zone_File  = 'INPUTS\tazfrzn.asc'   ;  from \INPUTS SD
ATYPFILE        = 'AreaType_File.dbf'    ;  Zonal Area Type file    (I/P file)

out_file        = 'fare_a2.asc'

RUN PGM=MATRIX
ZONES=@ZONESIZE@

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Initialize current metrorail walk pct and final pct walk
;;=======================================================
ZDATI[1] = @ZNFILE_TrPcts@ , Z            =  #1,
                             MetroShort   =  #2,
                             MetroLong    =  #3,
                             AMShort      =  #4,
                             AMLong       =  #5,
                             OPShort      =  #6,
                             OPLong       =  #7
                   ;; Convert Metrorail Long walk proportion to 1/10s of pcts (i.e.,
1.00 will be 1000.)
                   ;; as expected in the MFARE process

                            Metwkpct     =  Round(zi.1.MetroLong[I]   * 1000.0)


;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;  Lets double check that the computed metrorail walk pct (in tenths)
;  is within the expected range, if not then abort and write msg.

      if ((metwkpct <  0) || (metwkpct  > 1000.0)) ABORT

;;  print list = I(5),' ', larea(10.7),' ',swrarea(10.7),' ',lwrarea(10.7),' ',
;;                 smetdst(10.3),' ',lmetdst(10.3),
;;                       ' ',metwkpct(6.2)

ZDATI[3] = @Fare_Zone_File@,
           Z        =   4- 8,
           bfz1     =   9-16,
           bfz2     =  17-24,
           rfz1     =  41-48,
           rfz2     =  49-56,
           jur      =  57-64,
           pdsc     =  65-72,
           adsc     =  73-80



;
;  Print Out zonal data
;  -- Only if input bus fare zone 1 is nonzero
;     -this ensures that a consistent record count will be maintained w/ I&O
  IF (zi.3.bfz1 > 0)


       Print list = i(8), zi.3.bfz1(8),zi.3.bfz2(8),
```

```
                          metwkpct(8),metwkpct(8),
                          zi.3.rfz1(8),zi.3.rfz2(8),
                          zi.3.JUR(8),zi.3.pdsc(8),zi.3.adsc(8),file=@out_file@

  ENDIF

ENDRUN


;==================================================================
;  Prepare_MC_Zfile.S
;
;
;
;
;   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
OFilem       = ' ZONEV2.A2F  '         ; Output ZFile for the NL Mode Choice Model


;==================================================================
; Set Input Files:
;==================================================================

ZNFILE_LU      = 'inputs\zone.dbf'         ;  Input Zonal Land Use File
ZNFILE_MCMrkts = '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
         NHBParkCost = 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] = @ZNFILE_LU@;Z
                      ;EMP         TOTEMP  INDEMP  RETEMP  OFFEMP  OTHEMP  JURCODE
LANDAREA
                      ;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] = @ZNFILE_MCMrkts@, Z            =  #1,
                             MCDistrict   =  #2
                             MCDistrict   =  zi.2.MCDistrict[I]
```

```
; Zonal Transit Walk Shares
ZDATI[3] = @ZNFILE_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
                   ;; 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
                             IF (AMSHORT = 100.0 )
AMMELONG = 0.0
                             IF (AMSHORT >   0.0 && AMSHORT < 100.0  &&  AMLONG > 0)
AMMELONG = AMLONG - AMShort
                             IF (AMSHORT =   0.0 && AMLONG  >   0.0)
AMMELONG = AMLONG

 ;; 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  ;; indexs 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   ;; indexs 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
              ELSEIF (_Atype = 2)
                 HrNonWkPkCost =  100.0
              ELSEIF (_Atype = 3)
                 HrNonWkPkCost =   25.0
              ELSE
                 HrNonWkPkCost =    0.0
           ENDIF


           HBSParkCost = HrNonWkPkCost        ; Assume 1-Hour parking duration for
HBS trips
           HBOParkCost = HrNonWkPkCost * 2.0  ; Assume 2-Hour parking duration for
HBO trips
           NHBParkCost = HrNonWkPkCost * 2.0  ; Assume 2-Hour parking duration for
NHB trips


;---------------------------------------------------------------------
;Write out zonal files here  ...
;---------------------------------------------------------------------

 Print file=@ofilem@, form = 5 List=  I,
                                     HBWParkCost,
                                     HBSParkCost,
                                     HBOParkCost,
                                     NHBParkCost,
                                     HB_TermTime,
                                     NHB_TermTime,
                                     MetroShort,
                                     MetroLong,
                                     AMShort,
                                     AMLong,
                                     OPShort,
                                     OPLong,
                                     MCDistrict


IF (I=@Zonesize@)
     Print form=10.5csv file=@Rept@ list = '    Total Employment:  ',
totemp(10.0csv) ,'\n','\n'






;;-----------------------------------------------------------------------------------
----------------------------
     Print file=@Rept@ list = ' Jurisdictional Summary of Walk Shed Area (sq mi) by
Shed Type ','\n','\n',
                              '  Walk_Pct  MetroSh   MetroLg   AMShort   AMLong
OPShort   OPLong     TOTAL  ','\n',
                              ' --------- --------- --------- --------- --------- --
------- --------- ---------','\n'
 LOOP Idx   = 1, 24
  IF (Idx=1)
   CURDIST=STR(Idx,2,1)+' DC   '+ '|' ; Make row header
  ELSEIF (Idx=2)
   CURDIST=STR(Idx,2,1)+' MTG  '+ '|' ; Make row header
  ELSEIF (Idx=3)
   CURDIST=STR(Idx,2,1)+' PG   '+ '|' ; Make row header
  ELSEIF (Idx=4)
   CURDIST=STR(Idx,2,1)+' ARL  '+ '|' ; Make row header
  ELSEIF (Idx=5)
   CURDIST=STR(Idx,2,1)+' ALX  '+ '|' ; Make row header
  ELSEIF (Idx=6)
   CURDIST=STR(Idx,2,1)+' FFX  '+ '|' ; Make row header
  ELSEIF (Idx=7)
   CURDIST=STR(Idx,2,1)+' LDN  '+ '|'; Make row header
  ELSEIF (Idx=8)
   CURDIST=STR(Idx,2,1)+' PW   '+ '|' ; Make row header
  ELSEIF (Idx=9)
   CURDIST=STR(Idx,2,1)+' --   '+ '|' ; Make row header
  ELSEIF (Idx=10)
   CURDIST=STR(Idx,2,1)+' FRD  '+ '|' ; Make row header
  ELSEIF (Idx=11)
   CURDIST=STR(Idx,2,1)+' HOW  '+ '|' ; Make row header
  ELSEIF (Idx=12)
   CURDIST=STR(Idx,2,1)+' AAR  '+ '|' ; Make row header
  ELSEIF (Idx=13)
   CURDIST=STR(Idx,2,1)+' CHS  '+ '|' ; Make row header
  ELSEIF (Idx=14)
   CURDIST=STR(Idx,2,1)+' --   '+ '|' ; Make row header
  ELSEIF (Idx=15)
   CURDIST=STR(Idx,2,1)+' CAR  '+ '|' ; Make row header
  ELSEIF (Idx=16)
   CURDIST=STR(Idx,2,1)+' CAL  '+ '|' ; Make row header
  ELSEIF (Idx=17)
   CURDIST=STR(Idx,2,1)+' STM  '+ '|' ; Make row header
  ELSEIF (Idx=18)
   CURDIST=STR(Idx,2,1)+' KG   '+ '|' ; Make row header
  ELSEIF (Idx=19)
   CURDIST=STR(Idx,2,1)+' FBG  '+ '|' ; Make row header
  ELSEIF (Idx=20)
   CURDIST=STR(Idx,2,1)+' STF  '+ '|' ; Make row header
  ELSEIF (Idx=21)
   CURDIST=STR(Idx,2,1)+' SPTS '+ '|' ; Make row header
  ELSEIF (Idx=22)
   CURDIST=STR(Idx,2,1)+' FAUQ '+ '|' ; Make row header
  ELSEIF (Idx=23)
   CURDIST=STR(Idx,2,1)+' CLK  '+ '|' ; Make row header
  ELSE
   CURDIST=STR(Idx,2,1)+' JEFF '+ '|' ; Make row header
  ENDIF
     Print form=10.2csv, file=@Rept@,  list = CURDIST,
       MetroShortj[idx],
       MetroLongj[idx],
       AMShortj[idx],
       AMLongj[idx],
       OPShortj[idx],
       OPLongj[idx],
       TOTAL_Area[idx]
 ENDLOOP
     Print form=10.2csv, file=@Rept@ list = '\n', '\n',
                              ' --------- --------- --------- --------- --------- --
-------  --------- ---------','\n',
                              '     Total',MetroShortArea,MetroLongArea, AMShortArea
, AMLongArea,
                                          OPShortArea, OPLongArea,
totarea,'\n','\n','\n'
```

```
;;------------------------------------------------------------------------------------
----------------------------


;;------------------------------------------------------------------------------------
----------------------------
     Print file=@Rept@ list = ' # of "Active"  TAZs by Shed Type and Walk Percentage
(0% to 100%) ','\n','\n',
                              '  Walk_Pct  MetroSh   MetroLg   AMShort   AMLong
OPShort   OPLong  ','\n',
                              ' --------- --------- --------- --------- --------- --
------- ---------','\n'
 LOOP Idx  = 1, 101
     value = idx - 1
     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
```

24 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 ='Ext_Trip_Gen_PsAs_%_iter_%.dbf'  ;; OUTPUT external zonal Ps,As
file, HBW,HBS,HBO,NHW,NHO purposes

RUN PGM=MATRIX
ZONES=1

Fileo printo[1] ='Ext_Trip_Gen_PsAs_%_iter_%.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] = "Trip_Gen_productions_%_iter_%.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] = "Trip_Gen_Attractions_Final_%_iter_%.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)
              ZAttra[1][di.2.TAZ]    =   di.2.HBW_Mtr_As
              ZAttra[2][di.2.TAZ]    =   di.2.HBS_Mtr_As
              ZAttra[3][di.2.TAZ]    =   di.2.HBO_Mtr_As
              ZAttra[4][di.2.TAZ]    =   di.2.NHW_Mtr_As
              ZAttra[5][di.2.TAZ]    =   di.2.NHO_Mtr_As

;;       Accumulate total, internal and external P's by purpose
              TotAttra[1]  =  TotAttra[1]  +  ZAttra[1][di.2.TAZ]
              TotAttra[2]  =  TotAttra[2]  +  ZAttra[2][di.2.TAZ]
              TotAttra[3]  =  TotAttra[3]  +  ZAttra[3][di.2.TAZ]
              TotAttra[4]  =  TotAttra[4]  +  ZAttra[4][di.2.TAZ]
              TotAttra[5]  =  TotAttra[5]  +  ZAttra[5][di.2.TAZ]
              TotAttraSum  =  TotAttraSum  +  ZAttra[1][di.2.TAZ] +
ZAttra[2][di.2.TAZ] + ZAttra[3][di.2.TAZ] + ZAttra[4][di.2.TAZ] +
ZAttra[5][di.2.TAZ]

          IF (K <= @LastIZn@)
              IntAttra[1]  =  IntAttra[1]  +  ZAttra[1][di.2.TAZ]
              IntAttra[2]  =  IntAttra[2]  +  ZAttra[2][di.2.TAZ]
              IntAttra[3]  =  IntAttra[3]  +  ZAttra[3][di.2.TAZ]
              IntAttra[4]  =  IntAttra[4]  +  ZAttra[4][di.2.TAZ]
              IntAttra[5]  =  IntAttra[5]  +  ZAttra[5][di.2.TAZ]
              IntAttraSum  =  IntAttraSum  +  ZAttra[1][di.2.TAZ] +
ZAttra[2][di.2.TAZ] + ZAttra[3][di.2.TAZ] + ZAttra[4][di.2.TAZ] +
ZAttra[5][di.2.TAZ]
           ELSE
              ExtAttra[1]  =  ExtAttra[1]  +  ZAttra[1][di.2.TAZ]
              ExtAttra[2]  =  ExtAttra[2]  +  ZAttra[2][di.2.TAZ]
              ExtAttra[3]  =  ExtAttra[3]  +  ZAttra[3][di.2.TAZ]
              ExtAttra[4]  =  ExtAttra[4]  +  ZAttra[4][di.2.TAZ]
              ExtAttra[5]  =  ExtAttra[5]  +  ZAttra[5][di.2.TAZ]
              ExtAttraSum  =  ExtAttraSum  +  ZAttra[1][di.2.TAZ] +
ZAttra[2][di.2.TAZ] + ZAttra[3][di.2.TAZ] + ZAttra[4][di.2.TAZ] +
ZAttra[5][di.2.TAZ]
          ENDIF
 ENDLOOP

;; compute scaling  factors by purpose

    Loop pp= 1, @Purps@

         If (IntProda[pp]!= 0)  Pscale[pp] = ExtAttra[pp]/IntProda[pp]
         If (IntAttra[pp]!= 0)  Ascale[pp] = ExtProda[pp]/IntAttra[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: ',
IntAttra[1],  IntAttra[2], IntAttra[3], IntAttra[4], IntAttra[5],  IntAttraSum
   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: ',
TotAttra[1],  TotAttra[2], TotAttra[3], TotAttra[4], TotAttra[5],  TotAttraSum
   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),
                             NHWIIAs(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_ZAttra[pp][zz] =  ZAttra[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_ZAttra[pp][zz]

            IntScalePSum  = IntScalePSum  +   S_ZProda[pp][zz]
            IntScaleASum  = IntScaleASum  +   S_ZAttra[pp][zz]

           ELSE                    ;; Else TAZ is external, final scaled P/S equals
input P,A
             S_ZProda[pp][zz] =  ZProdA[pp][zz]
             S_ZAttra[pp][zz] =  ZAttrA[pp][zz]

          ENDIF                   ;;
```

```
            ;; Accum. total of scaled intls and untouched extls for reporting, by
purpose and for total
            TotScaleP[pp] = TotScaleP[pp] +   S_ZProda[pp][zz]
            TotScaleA[pp] = TotScaleA[pp] +   S_ZAttra[pp][zz]

            TotScalePSum  = TotScalePSum  +   S_ZProda[pp][zz]
            TotScaleASum  = TotScaleASum  +   S_ZAttra[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]
          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
```

25 Prepare_Ext_ComTrk_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  ='Ext_CVTruck_Gen_PsAs_%_iter_%.dbf'  ;; OUTPUT external zonal Ps,As
file, HBW,HBS,HBO,NHW,NHO purposes

RUN PGM=MATRIX
ZONES=1

Fileo printo[1] ='Ext_CVTruck_Gen_PsAs_%_iter_%.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] = "ComVeh_Truck_Ends_%_iter_%.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
 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
             ZAttra[3][di.1.TAZ]    =   di.1.Hvy_Truck

;;      Accumulate total, internal and external P's by purpose
             TotAttra[1]  =  TotAttra[1]  +  ZAttra[1][di.1.TAZ]
             TotAttra[2]  =  TotAttra[2]  +  ZAttra[2][di.1.TAZ]
             TotAttra[3]  =  TotAttra[3]  +  ZAttra[3][di.1.TAZ]
             TotAttraSum  =  TotAttraSum  +  ZAttra[1][di.1.TAZ] +
ZAttra[2][di.1.TAZ] + ZAttra[3][di.1.TAZ]

         IF (K <= @LastIZn@)
             IntAttra[1]  =  IntAttra[1]  +  ZAttra[1][di.1.TAZ]
             IntAttra[2]  =  IntAttra[2]  +  ZAttra[2][di.1.TAZ]
             IntAttra[3]  =  IntAttra[3]  +  ZAttra[3][di.1.TAZ]
             IntAttraSum  =  IntAttraSum  +  ZAttra[1][di.1.TAZ] +
ZAttra[2][di.1.TAZ] + ZAttra[3][di.1.TAZ]
          ELSE
             ExtAttra[1]  =  ExtAttra[1]  +  ZAttra[1][di.1.TAZ]
             ExtAttra[2]  =  ExtAttra[2]  +  ZAttra[2][di.1.TAZ]
             ExtAttra[3]  =  ExtAttra[3]  +  ZAttra[3][di.1.TAZ]
             ExtAttraSum  =  ExtAttraSum  +  ZAttra[1][di.1.TAZ] +
ZAttra[2][di.1.TAZ] + ZAttra[3][di.1.TAZ]
         ENDIF
 ENDLOOP

;; compute scaling  factors by purpose

    Loop pp= 1, @Purps@

         If (IntProda[pp]!= 0)  Pscale[pp] = ExtAttra[pp]/IntProda[pp]
         If (IntAttra[pp]!= 0)  Ascale[pp] = ExtProda[pp]/IntAttra[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: ',
IntAttra[1],  IntAttra[2], IntAttra[3], IntAttraSum
   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: ',
TotAttra[1],  TotAttra[2], TotAttra[3], TotAttraSum
   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_ZAttra[pp][zz] =  ZAttra[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_ZAttra[pp][zz]

            IntScalePSum  = IntScalePSum  +   S_ZProda[pp][zz]
            IntScaleASum  = IntScaleASum  +   S_ZAttra[pp][zz]

           ELSE                   ;; Else TAZ is external, final scaled P/S equals
input P,A
             S_ZProda[pp][zz] =  ZProdA[pp][zz]
             S_ZAttra[pp][zz] =  ZAttrA[pp][zz]

          ENDIF                   ;;
            ;; Accum. total of scaled intls and untouched extls for reporting, by
purpose and for total
            TotScaleP[pp] = TotScaleP[pp] +   S_ZProda[pp][zz]
            TotScaleA[pp] = TotScaleA[pp] +   S_ZAttra[pp][zz]

            TotScalePSum  = TotScalePSum  +   S_ZProda[pp][zz]
            TotScaleASum  = TotScaleASum  +   S_ZAttra[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_ZAttra[1][zz]
          ro.SMed_TrkAs = S_ZAttra[2][zz]
          ro.SHvy_TrkAs = S_ZAttra[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 = '            '
   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
```

26 Prepare_Trip_Tables_for_Assignment.s

```
;-------------------------------------------------------------- ;------------------
-------------------------------------------
;   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  = 'AM%_iter_%.ADR'
ADRMD  = 'MD%_iter_%.ADR'
ADRPM  = 'PM%_iter_%.ADR'
ADRNT  = 'NT%_iter_%.ADR'
;
; I/P MISC Auto Dr.Tables:
MISCAM  = 'MISCAM%_iter_%.TT'
MISCMD  = 'MISCMD%_iter_%.TT'
MISCPM  = 'MISCPM%_iter_%.TT'
MISCNT  = 'MISCNT%_iter_%.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@
    MATI[3]=@ADRPM@
    MATI[4]=@ADRNT@

;;  Miscellaneaous 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
vehs

      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
   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 1occ 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 1occ 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
   ammtrk   = ammtrk   + 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
   mdvehs   = mdvehs   +(MW[251]+MW[252]+MW[253]+MW[254]+MW[255]+MW[256]) ; all md
vehs
   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 1occ 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 1occ 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
   pmvehs   = pmvehs   +(MW[351]+MW[352]+MW[353]+MW[354]+MW[355]+MW[356]) ; all pm
vehs
   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 1occ 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 1occ 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
   opvehs   = opvehs   +(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 1occ 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 1occ 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      '
Print list = '%_iter_% Iter. Pre-Traffic Assignment Trip Table Prep.
Report',file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt

Print list = 'AM-Peak Totals: '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am modeled 1-occveh's ',am1occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am modeled 2-occveh's ',am2occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am modeled 3+occveh's ',am3occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am trucks             ',amtrks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am 1occ adr           ',am1occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am 2occ adr           ',am2occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am 3+occ adr          ',am3occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am total adr(modeled) ',amadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am Thru     Truck     ',amxxtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am Thru 1occ Adr      ',amxxad1
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am Thru 2occ Adr      ',amxxad2
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am Thru 3+occAdr      ',amxxad3
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am total xx adr       ',amxxadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am Taxi     ADr       ',amtaxi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am visitor  ADr       ',amvisi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am School   ADr       ',amschl
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am int,ext MedTk      ',ammtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
```

```
Print form= 12.0csv list = '   am int,ext HvyTk      ',amhtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am air pax auto dr    ',amairpax
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   am int,ext,ComVeh     ',amcomveh
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   all am vehs           ',amvehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt


   ;MD group
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = 'Midday  Totals: '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md modeled 1-occveh's ',md1occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md modeled 2-occveh's ',md2occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md modeled 3+occveh's ',md3occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md trucks             ',mdtrks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md 1occ adr           ',md1occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md 2occ adr           ',md2occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md 3+occ adr          ',md3occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md total adr(modeled) ',mdadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md Thru     Truck     ',mdxxtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md Thru 1occ Adr      ',mdxxad1
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md Thru 2occ Adr      ',mdxxad2
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md Thru 3+occAdr      ',mdxxad3
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md total xx adr       ',mdxxadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md Taxi     ADr       ',mdtaxi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md visitor  ADr       ',mdvisi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md school   ADr       ',mdschl
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md int,ext MedTk      ',mdmtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md int,ext HvyTk      ',mdhtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md air pax auto dr    ',mdairpax
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   md int,ext,ComVeh     ',mdcomveh
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   all md vehs           ',mdvehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt


   ;PM group
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = 'PM-Peak Totals: '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm modeled 1-occveh's ',pm1occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm modeled 2-occveh's ',pm2occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm modeled 3+occveh's ',pm3occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm trucks             ',pmtrks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm 1occ adr           ',pm1occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm 2occ adr           ',pm2occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm 3+occ adr          ',pm3occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm total adr(modeled) ',pmadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm Thru     Truck     ',pmxxtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm Thru 1occ Adr      ',pmxxad1
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm Thru 2occ Adr      ',pmxxad2
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm Thru 3+occAdr      ',pmxxad3
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm total xx adr       ',pmxxadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm Taxi     ADr       ',pmtaxi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm visitor  ADr       ',pmvisi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm school   ADr       ',pmschl
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm int,ext MedTk      ',pmmtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm int,ext HvyTk      ',pmhtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm air pax auto dr    ',pmairpax
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   pm int,ext,ComVeh     ',pmcomveh
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   all pm vehs           ',pmvehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt


   ;OP group
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = 'Off-Peak Totals: '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt modeled 1-occveh's ',op1occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt modeled 2-occveh's ',op2occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt modeled 3+occveh's ',op3occ
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt trucks             ',optrks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt 1occ adr           ',op1occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt 2occ adr           ',op2occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt 3+occ adr          ',op3occad
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt total adr(modeled) ',opadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt Thru     Truck     ',opxxtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt Thru 1occ Adr      ',opxxad1
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt Thru 2occ Adr      ',opxxad2
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
```

```
Print form= 12.0csv list = '   nt Thru 3+occAdr      ',opxxad3
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt total xx adr       ',opxxadr
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt Taxi     ADr       ',optaxi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt visitor  ADr       ',opvisi
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt school   ADr       ',opschl
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt int,ext MedTk      ',opmtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt int,ext HvyTk      ',ophtrk
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt air pax auto dr    ',opairpax
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   nt int,ext,ComVeh     ',opcomveh
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   all nt vehs           ',opvehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt

Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   SUM OF ALL VEHICLES:  ',vehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '%_iter_% Trip Table Output Totals: '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
;  AM
Print form= 12.0csv list = '   AMSOVs                ',AMSOVs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   AMHOV2s               ',AMHOV2s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   AMHOV3s               ',AMHOV3s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   AMComVehs
',AMComVehs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   AMTrucks              ',AMTrucks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   AMAirPaxs
',AMAirPaxs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt

;  MD
Print form= 12.0csv list = '   MDSOVs                ',MDSOVs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   MDHOV2s               ',MDHOV2s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   MDHOV3s               ',MDHOV3s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   MDComVehs
',MDComVehs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   MDTrucks              ',MDTrucks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   MDAirPaxs
',MDAirPaxs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
```

```
;  PM
Print form= 12.0csv list = '   PMSOVs                ',PMSOVs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   PMHOV2s               ',PMHOV2s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   PMHOV3s               ',PMHOV3s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   PMComVehs
',PMComVehs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   PMTrucks              ',PMTrucks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   PMAirPaxs
',PMAirPaxs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt

;  OP
Print form= 12.0csv list = '   NTSOVs                ',OPSOVs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   NTHOV2s               ',OPHOV2s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   NTHOV3s               ',OPHOV3s
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   NTComVehs
',OPComVehs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   NTTrucks              ',OPTrucks
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   NTAirPaxs
',OPAirPaxs,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print form= 12.0csv list = '   SUM OF ALL VEHICLES:  ',vehs
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.txt
Print list = '  '
,file=Prepare_Trip_Tables_For_Assignment%_Iter_%.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

 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

 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

 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

ENDRUN
```

27 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.seqno       =di.1.seqno
            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
 ENDLOOP


ENDRUN
```

28 RemovePPSpeed.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
```

29 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.
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
          _BseGrowRate = CPI_Table(2,@ModeledYear@) ;    then use the defaltion
rate from table
          _AltGrowRate = 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
          _BseGrowRate = CPI_Table(2,@ModeledYear@)                             ;
then use the defaltion rate from table
          _AltGrowRate = CPI_Table(2,@ModeledYear@) * INFLATIONFTR

          _FutBseCPI   = CPI_Table(1,BaseCPIYear) * ((1.0 +
_BseGrowRate)^(@ModeledYear@ - BaseCPIYear))
          _FutAltCPI   = CPI_Table(1,BaseCPIYear) * ((1.0 +
_AltGrowRate)^(@ModeledYear@ - BaseCPIYear))

          DEFLATIONFTR = (_FutAltCPI / _FutBseCPI)  *  CPI_Table(3,@ModeledYear@)


        ELSE
          _BseGrowRate = ( (CPI_Table(1,CurrCPIYear) / CPI_Table(1,BaseCPIYear)) ^
(1.0/ (CurrCPIYear - BaseCPIYear)) ) - 1.0
          _AltGrowRate = _BseGrowRate * INFLATIONFTR

          _FutBseCPI   = CPI_Table(1,BaseCPIYear) * ((1.0 +
_BseGrowRate)^(@ModeledYear@ - BaseCPIYear))
          _FutAltCPI   = CPI_Table(1,BaseCPIYear) * ((1.0 +
_AltGrowRate)^(@ModeledYear@ - BaseCPIYear))

          DEFLATIONFTR =  (_FutAltCPI / _FutBseCPI)  *  CPI_Table(3,CurrCPIYear)
        ENDIF

              ;  print out small text file containing deflation factor deriviation:
          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)',
_BseGrowRate(8.5)  , _FutBseCPI(8.1),'(forecasts years only)','\n',file=@CPI_rept@
          Print List='Modeled Year growth rate & CPI w/ Infla. Factor: (B)',
_AltGrowRate(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                                 =
;===============================================================
```

30 Set_Factors.s

```
;-------------------------------------------------------------------  ;-------------
---------------------------------------------------
; SET_FACTORS.S   Version 2.3 Model-3722 TAZ system
;-------------------------------------------------------------------

; MWCOG Version 2.3  Model
;  Set up K-factor files used in Trip Distribution
;
; Zonal K-factor Files created by this script (files written to the support subdir.)
;
HBWK     = '..\support\hbwk.dat'    ;
HBSK     = '..\support\hbsk.dat'    ;
HBOK     = '..\support\hbok.dat'    ;
NHWK     = '..\support\nhwk.dat'    ;
NHOK     = '..\support\nhok.dat'    ;
;
;
; Output Files:
;
;  HBWK.DAT  =     HBW      zonal K-Factor Matrix
;  HBSK.DAT  =     HBS      zonal K-Factor Matrix
;  HBOK.DAT  =     HBO      zonal K-Factor Matrix
;  NHWK.DAT  =     NHW      zonal K-Factor Matrix
;  NHOK.DAT  =     NHO      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
\\
; \\\\\\\\\       The will be applied across income strata in trip   \\
; \\\\\\\\\       distribution.                                      \\
; //////////////////////////////////////////////////////////////////////

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/Extl

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/Extl
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
NHWK          NHOK
;;;                                         -----         -----         -----
-----         -----
  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)
   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)
   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
```

31 Time-of-Day.s

```
; ====================================================================
; Time-of-Day.s
; MWCOG Version 2.3 Model
;
```

```
;
;              Distribute Modeled Pump Prime Auto Driver Trips, i.e,
;              4 Purposes (HBW,HBS,HBO,NHB), 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','i1'-'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 = 'HBW%_iter_%.ADR' ; HBW  1,2,3+ Occ Adr Trips (t1-3)          //
 HBSADR = 'HBS%_iter_%.ADR' ; HBS  1,2,3+ Occ Adr Trips (t1-3)          //
 HBOADR = 'HBO%_iter_%.ADR' ; HBO  1,2,3+ Occ Adr Trips (t1-3)          //
 NHWADR = 'NHW%_iter_%.ADR' ; NHW  1,2,3+ Occ Adr Trips (t1-3)          //
 NHOADR = 'NHO%_iter_%.ADR' ; NHO  1,2,3+ Occ Adr Trips (t1-3)          //
;                                                                       //
; O/P Auto Dr. Pct. tables:                                             //
 ADRAM  = 'AM%_iter_%.ADR'  ; AM     Modeled Total Auto Drivers         //
 ADRPM  = 'PM%_iter_%.ADR'  ; PM     Modeled Total Auto Drivers         //
 ADRMD  = 'MD%_iter_%.ADR'  ; Midday Modeled Total Auto Drivers         //
 ADRNT  = 'NT%_iter_%.ADR'  ; 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


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][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][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][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][1]/100.00)) / 2.0  ;   HBS / DA             *   *
   mw[505] =  (MW[122] * (TODFtrs[2][3][1][1]/100.00) + MW[222]*
(TODFtrs[2][3][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][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][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][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][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][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][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][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][1]/100.00)) / 2.0  ;   NHO / DA             *   *
   mw[514] =  (MW[152] * (TODFtrs[5][3][1][1]/100.00) + MW[252]*
(TODFtrs[5][3][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][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]/100.00)) / 2.0  ;   HBW / DA             *****
   mw[517] =  (MW[112] * (TODFtrs[1][3][1][2]/100.00) + MW[212]*
(TODFtrs[1][3][2][2]/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]/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]/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]/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]/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]/100.00)) / 2.0  ;   HBO / DA             *   *
   mw[523] =  (MW[132] * (TODFtrs[3][3][1][2]/100.00) + MW[232]*
(TODFtrs[3][3][2][2]/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]/100.00)) / 2.0  ;   HBO / 3+occ carpool  * P *

;                      * 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

; 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 = '  '
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       ',inhw(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

   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

   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

   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

ENDRUN
;;
```

32 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

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: ',
                                                                    '  Iteration:
','%_iter_%',
                                                                    '  Period:
','@per@' ,
                                                                    '  AccType:
','@Acc@' ,
                                                                    '  PathType:
','@Path@',
                                                                    ' #Connected
IJs:    ', Connected,
                                                                    ' #Disconnected
IJs: ', NotConnected
ENDIF


ENDRUN
ENDLOOP
ENDLOOP
ENDLOOP
```

33 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
```

```
;


;--------------------------------------------------------------------------
; Loop through each period and access mode
;--------------------------------------------------------------------------

;
;  Read in time factors to increase local bus times
;  based on increasing arterial hwy congestion

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
pageheight=32767  ; Preclude header breaks
;

LOOP PERIOD = 1, 2

 IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR

  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'

 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR

  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'
  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
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
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,  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 = suplAB@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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10

;@KR_MODEL@ READ FILE = lrt@TIME_PERIOD@pnr.tb  ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new@TIME_PERIOD@pnr.tb  ;---- Mode10-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 ----
```

34 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: WKBMAMnode.dbf; WKBMAMlink.dbf
; Step 2: AM Peak Drive Assignment
;  Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
;  Output Files: DRBMAMnode.dbf; DRBMAMlink.dbf
; Step 3: AM Peak K/R Assignment
;  Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB, '%_iter_%_AMMS.TRP'
;  Output Files: KRBMAMnode.dbf; KRBMAMlink.dbf
; Step 4: Off Peak Walk Assignment
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
;  Output Files: WKBMOPnode.dbf; WKBMOPlink.dbf
; Step 5: Off Peak Drive Assignment
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
;  Output Files: DRBMOPnode.dbf; DRBMOPlink.dbf
; Step 6: Off Peak K/R Assignment
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB, '%_iter_%_OPMS.TRP'
;  Output Files: KRBMOPnode.dbf; KRBMOPlink.dbf
;
;
;  Read in time factors to increase local bus times
;  based on increasing arterial hwy congestion

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
  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
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
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,  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 = suplBM@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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10

@KR_MODEL@ READ FILE = lrt@TIME_PERIOD@pnr.tb  ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new@TIME_PERIOD@pnr.tb  ;---- Mode10-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_%_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_%_KRBMAMnode.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_%_KRBMAMlink.dbf ; output link file
@KR_MODEL@@OP@FILEO LINKO    = %_iter_%_KRBMOPlink.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 ----
```

35 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

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
   MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
   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@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
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,  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 = suplCR@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)
;@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

;@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus@TIME_PERIOD@.asc;--- drive to bus
;@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

;---- 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_%_WKCRAMnode.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_%_WKCRAMlink.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 ----
```

36 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

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
```

```
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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
  MATIN='%_iter_%_AMMS.TRP'
  AM=' '
  OP=';'
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
  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
@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
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,  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 = suplMR@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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
;KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
;@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10

@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_%_WKMROPnode.dbf ; output node file
@DRIVE_MODEL@@AM@FILEO NODEO = %_iter_%_DRMRAMnode.dbf ; output node file
@DRIVE_MODEL@@OP@FILEO NODEO = %_iter_%_DRMROPnode.dbf ; output node file
@KR_MODEL@@AM@FILEO NODEO    = %_iter_%_KRMRAMnode.dbf ; output node file
@KR_MODEL@@OP@FILEO NODEO    = %_iter_%_KRMROPnode.dbf ; output node file

@WALK_MODEL@@AM@FILEO LINKO  = %_iter_%_WKMRAMlink.dbf ; output link file
@WALK_MODEL@@OP@FILEO LINKO  = %_iter_%_WKMROPlink.dbf ; output link file
@DRIVE_MODEL@@AM@FILEO LINKO = %_iter_%_DRMRAMlink.dbf ; output link file
@DRIVE_MODEL@@OP@FILEO LINKO = %_iter_%_DRMROPlink.dbf ; output link file
@KR_MODEL@@AM@FILEO LINKO    = %_iter_%_KRMRAMlink.dbf ; output link file
@KR_MODEL@@OP@FILEO LINKO    = %_iter_%_KRMROPlink.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 ----
```

37 Transit_Skims_AB.s

```
;---------------------------------------------------------------------------
;Transit_Skims_AB.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.SKM
; Step 2: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 4: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 6: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 8: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 10: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 12: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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                          ;
; Added  table #19 (Total Transit time in min.) to output transit.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)

;---------------------------------------------------------------------------
; Set up tokens to either use or comment out commands for Cube Cluster (distributed
processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
  dp_token = ' '
else
  dp_token = ';'
endif
; useIdp = t (true) or f (false);  this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=f

;---------------------------------------------------------------------------
;         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

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
;

LOOP PERIOD = 1, 2

 IF (PERIOD = 1)
  TIME_PERIOD = 'AM'
  COMBINE = 5.0
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
 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.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
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,  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 = suplAB@access_mode@@time_period@.asc modes=11-16 oneway=t fixed=y
fileo nodeo    = supnAB@access_mode@@time_period@.dbf
fileo linko    = trnlAB@access_mode@@time_period@.dbf   ; Can be used to create
transit shapefile


;---- specify output skims ----


MATRICES NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, IWAIT, XWAIT, WACCT, WLKT,
 XADD, BRDS, DACCT, DACCD, PRKI, PRKC, ISTOS, JSTOS,
; MW[1] = TIME(1,6,8),
```

```
 MW[1] = TIME(1) * @_IBFTR@ +
         TIME(6) * @_IBFTR@ +
         TIME(8) * @_OBFTR@ ,                ;---- 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   ;---- 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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10
```

```
;@KR_MODEL@ READ FILE = lrt@TIME_PERIOD@pnr.tb  ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new@TIME_PERIOD@pnr.tb  ;---- Mode10-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)

/* 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
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 MATI[1]=TRANSIT.SKM
 MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.SKM, MO = 1-16,
  FORMAT = MINUTP,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC
 MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.STA, MO = 17-18,
  FORMAT = MINUTP,
 NAME = ISTOS, JSTOS
                                                                          ;3
 MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_AB.ttt, MO = 100,           ;
  FORMAT = MINUTP,                                                        ;
 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 ----
```

38 Transit_Skims_BM.s

```
;---------------------------------------------------------------------------
;Transit_Skims_BM.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.SKM
; Step 2: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 4: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 6: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 8: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 10: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 12: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_OP_KR_BM.SKM, %_iter_%_OP_KR_BM.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                      ;
; Added  table #19 (Total Transit time in min.) to output transit.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)

;---------------------------------------------------------------------------
; Set up tokens to either use or comment out commands for Cube Cluster (distributed
processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
  dp_token = ' '
else
  dp_token = ';'
endif
; useIdp = t (true) or f (false);  this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=f
;
;  Read in time factors to increase local bus times
;  based on increasing arterial hwy congestion

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
;

;---------------------------------------------------------------------------
;         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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
 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.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
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,  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 = suplBM@access_mode@@time_period@.asc modes=11-16 oneway=t fixed=y
fileo nodeo    = supnBM@access_mode@@time_period@.dbf
fileo linko    = trnlBM@access_mode@@time_period@.dbf   ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, IWAIT, XWAIT, WACCT, WLKT,
 XADD, BRDS, DACCT, DACCD, PRKI, PRKC, ISTOS, JSTOS,
; MW[1] = TIME(1,6,8),
 MW[1] = TIME(1) * @_IBFTR@ +
         TIME(6) * @_IBFTR@ +
         TIME(8) * @_OBFTR@ ,               ;---- 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   ;---- 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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car
```

```
READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
@KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10

@KR_MODEL@ READ FILE = lrt@TIME_PERIOD@pnr.tb  ;---- LRT-PNR connectors (links)
@KR_MODEL@ READ FILE = new@TIME_PERIOD@pnr.tb  ;---- Mode10-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)

/* 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
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 MATI[1]=TRANSIT.SKM
 MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.SKM, MO = 1-16,
  FORMAT = MINUTP,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC
 MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.STA, MO = 17-18,
  FORMAT = MINUTP,
 NAME = ISTOS, JSTOS

 MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_BM.ttt, MO = 100,                ;
  FORMAT = MINUTP,                                                             ;
 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[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 ----
```

39 Transit_Skims_CR.s

```
;--------------------------------------------------------------------------
;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.SKM
; Step 2: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 4: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 6: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 8: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.skm file     ;
; create total transit time skims named:                                       ;
;          %_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.ttt                         ;
;
; Set up tokens to either use or comment out commands for Cube Cluster (distributed
processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
  dp_token = ' '
else
  dp_token = ';'
endif
; useIdp = t (true) or f (false);  this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=f

;  Read in time factors to increase local bus times
;  based on increasing arterial hwy congestion

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
;

;--------------------------------------------------------------------------
;          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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
 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.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
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,  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 = suplCR@access_mode@@time_period@.asc modes=11-16 oneway=t fixed=y
fileo nodeo    = supnCR@access_mode@@time_period@.dbf
fileo linko    = trnlCR@access_mode@@time_period@.dbf   ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, IWAIT, XWAIT, WACCT, WLKT,
 XADD, BRDS, DACCT, DACCD, PRKI, PRKC, ISTOS, JSTOS,
; MW[1] = TIME(1,6,8),
 MW[1] = TIME(1) * @_IBFTR@ +
         TIME(6) * @_IBFTR@ +
         TIME(8) * @_OBFTR@ ,                ;---- 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   ;---- 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)
;@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

;@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = bus@TIME_PERIOD@.asc;--- drive to bus
;@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

;---- 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
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 MATI[1]=TRANSIT.SKM
 MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.SKM, MO = 1-16,
  FORMAT = MINUTP,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC
 MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.STA, MO = 17-18,
  FORMAT = MINUTP,
 NAME = ISTOS, JSTOS
                                                                               ;3
 MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_CR.ttt, MO = 100,                ;
  FORMAT = MINUTP,                                                             ;
 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 ----
```

40 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.SKM
; Step 2: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 4: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 6: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 8: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_OP_WK_MR.SKM, %_iter_%_OP_WK_MR.STA,
; Step 9: Off Peak Drive Skims
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
;  Output Files: TRANSIT.SKM
; Step 10: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.SKM
; Step 12: Condition & Split Skims into Multiple Files
;  Input Files: TRANSIT.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.skm file     ;
; create total transit time skims named:                                       ;
;          %_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_MR.ttt                         ;

;
; Set up tokens to either use or comment out commands for Cube Cluster (distributed
processing)
if ('%useIdp%'='t' || '%useIdp%'='T')
  dp_token = ' '
else
  dp_token = ';'
endif
; useIdp = t (true) or f (false);  this is set in the wrapper batch file
distribute intrastep=%useIdp% multistep=f

;  Read in time factors to increase local bus times
;  based on increasing arterial hwy congestion

READ FILE=INPUTS\LBus_TimFTRS.ASC   ; Local Bus Time Factors
```

```
;

;--------------------------------------------------------------------------
;         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
  _IBFTR=AMIBFTR
  _OBFTR=AMOBFTR
 ELSE
  TIME_PERIOD = 'OP'
  COMBINE = 10.0
  _IBFTR=OPIBFTR
  _OBFTR=OPOBFTR
 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.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
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,  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 = suplMR@access_mode@@time_period@.asc modes=11-16 oneway=t fixed=y
fileo nodeo    = supnMR@access_mode@@time_period@.dbf
fileo linko    = trnlMR@access_mode@@time_period@.dbf   ; Can be used to create
transit shapefile
;

;---- specify output skims ----

MATRICES NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, IWAIT, XWAIT, WACCT, WLKT,
 XADD, BRDS, DACCT, DACCD, PRKI, PRKC, ISTOS, JSTOS,
; MW[1] = TIME(1,6,8),
 MW[1] = TIME(1) * @_IBFTR@ +
         TIME(6) * @_IBFTR@ +
         TIME(8) * @_OBFTR@ ,                ;---- 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   ;---- 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)
@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  ;---- Mode10-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   ;--- Mode10 bus-LRT links&xfer car

READ FILE = walkacc.asc ;--- walk to local transit

@DRIVE_MODEL@READ FILE = mrpr@TIME_PERIOD@.asc;--- drive to metrorail
;@DRIVE_MODEL@READ FILE = cr@TIME_PERIOD@.asc;--- drive to Commuter rail
;@DRIVE_MODEL@READ FILE = buspr@TIME_PERIOD@.asc;--- drive to bus
@DRIVE_MODEL@READ FILE = lrt@TIME_PERIOD@.asc;--- drive to LRT
;@DRIVE_MODEL@READ FILE = new@TIME_PERIOD@.asc;--- drive to Mode10

@KR_MODEL@READ FILE = mrkr@TIME_PERIOD@.asc;--- k/r to metrorail
;KR_MODEL@READ FILE = buskr@TIME_PERIOD@.asc;--- k/r to bus
@KR_MODEL@READ FILE = lrtkr@TIME_PERIOD@.asc;--- k/r to LRT
;@KR_MODEL@READ FILE = newkr@TIME_PERIOD@.asc;--- k/r to Mode10

@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)

/* 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
@dp_token@distributeIntrastep processId='mwcog', ProcessList=%subnode%
 MATI[1]=TRANSIT.SKM
 MATO[1]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_MR.SKM, MO = 1-16,
  FORMAT = MINUTP,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC
 MATO[2]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_MR.STA, MO = 17-18,
  FORMAT = MINUTP,
 NAME = ISTOS, JSTOS

 MATO[3]=%_iter_%_@TIME_PERIOD@_@ACCESS_MODE@_MR.ttt, MO = 100,                ;
  FORMAT = MINUTP,                                                             ;
 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 ----
```

41 Transit_Skims_Select_Paths.s

```
;-------------------------------------------------------------------------
;Transit_Skims_Select_Paths.s
;MWCOG Version 2.1D 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        = WALK_pp.TB, PNR_pp.TB
;  Walk Sidewalk Network        = WLKNET.TB
;  Zone Employment              = ZONE.ASC
; Output Files:
;  Walk and Drive Access Skims  = %_iter_%_pp_aa.SKM
;  Walk and Drive Station Data  = %_iter_%_pp_aa.STA
;  Walk and Drive Travel Time   = %_iter_%_pp_aa.TTT
;  Transit Access to Employment = JOBACC.ASC
;
; Step 1: AM Peak Walk Skims
;  Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
;  Output Files: TRANSIT.SKM
; Step 2: Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_AM_WK.SKM, %_iter_%_AM_WK.STA, %_iter_%_AM_WK.TTT
; Step 3: AM Peak Drive Skims
;  Input Files: ZONEHWY.NET, MODE?_AM.TB, *.TB
;  Output Files: TRANSIT.SKM
; Step 4: Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_AM_DR.SKM, %_iter_%_AM_DR.STA, %_iter_%_AM_DR.TTT
; Step 5: Off Peak Walk Skims
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
;  Output Files: TRANSIT.SKM
; Step 6: Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_OP_WK.SKM, %_iter_%_OP_WK.STA, %_iter_%_OP_WK.TTT
; Step 7: Off Peak Drive Skims
;  Input Files: ZONEHWY.NET, MODE?_OP.TB, *.TB
;  Output Files: TRANSIT.SKM
; Step 8: Split Skims into Multiple Files
;  Input Files: TRANSIT.SKM
;  Output Files: %_iter_%_OP_DR.SKM, %_iter_%_OP_DR.STA, %_iter_%_OP_DR.TTT
```

```
; Step 9: Sum the Jobs by Transit Travel Time
;  Input Files: %_iter_%_AM_WK.TTT, %_iter_%_AM_DR.TTT
;  Output Files: JOBACC.ASC
;
;---------------------------------------------------------------------------
;
; Global Variables
;
;        _iter_ (= PP,i1-i6)
;
;---------------------------------------------------------------------------
;         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

;---------------------------------------------------------------------------
; Steps 1, 3, and 5: Build Transit Paths
;---------------------------------------------------------------------------

RUN PGM = TRNBUILD
 NETI = ZONEHWY.NET
;; MATO = TRANSIT.SKM

 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@)

;---- factors to convert actual time to perceived time ----

MODEFAC[1]  = 10*1.00 ;---- in-vehicle time
MODEFAC[11] = 1.00    ;---- drive access time
MODEFAC[12] = 2.50    ;---- transit transfer time
MODEFAC[13] = 2.50    ;---- walk network time
MODEFAC[14] = 2.50    ;---- unused (used to be dummy link to station)
MODEFAC[15] = 2.50    ;---- park-&-ride transfer time
MODEFAC[16] = 2.50    ;---- walk access time

;---- initial and transfer wait factors ----

IWAITFAC[1] = 10*2.50
XWAITFAC[1] = 10*2.50
IWAITMAX[1] = 10*60.0

;---- boarding penalty - limit to three transfers ----

BOARDPEN[1] = 0.0, 6.0, 6.0, 60.0

;---- 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,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[2]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[3]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[4]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[5]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[6]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  n,  Y,  n,  n,  n,  Y,  n
NOX[7]  =  n,  n,  n,  n,  n,  n,  n,  n,  n,  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,  n,  n,  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,  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,  Y,  n,  Y,  Y

;---- Parameters ----

LISTINPUT = N         ;--- echo input files

MAXPATHTIME = 240.0   ;--- 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@access_mode@@time_period@.asc modes=11-16 oneway=t fixed=y
fileo nodeo    = supn@access_mode@@time_period@.dbf
;


;---- specify output skims ----
```

```
;;MATRICES NAME = WLKT, DACCT, INIT, XFERT, IVNMT, IVMT, TOTT, ISTOS, JSTOS,
;; MW[1] = TIME(12,13,14,15)*0.01,            ;---- xfer walk time   (min)
;; MW[2] = TIME(11)*0.01,                     ;---- drv acc time     (min)
;; MW[3] = IWAIT*0.01,                        ;---- ini.wait time    (min)
;; MW[4] = XWAIT(1,2,3,4,5,6,7,8,9,10)*0.01,  ;---- xfr wait time    (min)
;; MW[5] = TIME(1,2,4,5,6,7,8,9,10)*0.01,     ;---- ivt-nonmetrorail (min)
;; MW[6] = TIME(3)*0.01,                      ;---- ivt-metrorail    (min)
;; MW[7] = (IWAIT + TIME (0) + XWAIT (0))*0.01,;---- total time       (min)
;; MW[8] = NODE0(3) - 7300.0,                 ;---- metro board  sta (1-116)
;; MW[9] = NODEL(3) - 7300.0                  ;---- metro alight sta (1-116)

;---- 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

;---- 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 = bus_pnrl.tb  ;---- Bus-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = met_pnrl.tb  ;---- Metro-PNR connectors (links)
@DRIVE_MODEL@ READ FILE = com_pnrl.tb  ;---- Commuter Rail-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 = walk_@TIME_PERIOD@.tb ;--- walk to local transit

@DRIVE_MODEL@READ FILE = pnr_@TIME_PERIOD@.tb;--- drive to transit

;---- Dummy Centroid Access Links (mode 14) ----

;---- Sidewalk Network (mode 13) ----

READ FILE = wlknet.tb;--- walk network for transfers

;---- Transit Line Cards (modes 1-9) ----

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

;---- Reports -----
; Path Tracing
; Consider these "i"s            to      these "j"s
; -----------------------------         ---------
;   8 Downtwn      1236 Rosslyn               8  Downtwn
;  64 Union Sta    1337 Alexandria           64  Union Sta
; 345 Bethesda     1537 Tysons Crnr         345  Bethesda
; 362 Silver Spr   1554 Ft Belvoir          362  Silver Spr
; 464 N.SilverSpr  1619 Vienna             1231  Pentagon
; 578 Shady Gr Rd  1698 Dulles AP          1236  Rosslyn
; 829 Andrews AFB  1716 Reston             1337  Alexandria
; 927 New Carrltn  1842 Leesburg           1537  Tysons
;1043 Frederick    1942 Dale City
;1231 Pentagon     1967 Manassas

Select   i = 451, 692
trace = (i = 451, 692  &
         j = 8, 64)
;; REPORT LINES = NAME, MODE  ; added by rm 4/09/04 to ensure line listings
                           ; with or without 'RT=' commands in transit line files
ENDRUN

ENDLOOP

ENDLOOP
```

42 Trip_Distribution.s

```
*del voya*.prn



; Trip_Distribution.s  - Version 2.3 Trip Distribution
;
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 = 'SOV%_prev_%am.skm'             ;  AM  HWY TIME SKIMS
MDSOVSKM = 'SOV%_prev_%md.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)


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$) by income level (for toll modeling)
toll_inc = '..\support\toll.inc'    ; Equivalent minutes (min/'07$) by period &
income level (for toll modeling)

; Zonal K-factor Files
;
HBWK     = '..\support\hbwk.dat'    ;
```

```
HBSK     = '..\support\hbsk.dat'    ;
HBOK     = '..\support\hbok.dat'    ;
NHWK     = '..\support\nhwk.dat'    ;
NHOK     = '..\support\nhok.dat'    ;
;
;------------------------------------------------------------------
;
;
FFsFile   = '..\SUPPORT\Version_23_FFtrs.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  EXTMTK  EXTHTK ;
;
; Trip-End (P/A) Input Files:
;
AutoProds = 'Trip_Gen_Productions_%_iter_%.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
;

AutoAttrs = 'Trip_Gen_Attractions_Final_%_iter_%.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   = 'Ext_Trip_Gen_PsAs_%_iter_%.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 = 'ComVeh_Truck_Ends_%_iter_%.dbf'         ; Intl Comm.Veh/Truck TripEnds
;;Variables in dbf file:
; TAZ    ICOMM_VEH  IMED_TRUCK  IHVY_TRUCK
;
;
Ext_TrkEnds = 'Ext_CVTruck_Gen_PsAs_pp.dbf'
;;Variables in dbf file:
;  TAZ SCOM_VEHPS      SMED_TRKPS      SHVY_TRKPS      SCOM_VEHAS      SMED_TRKAS
SHVY_TRKAS
;
;; OUTPUT TRIP TABLES
HBWTDOUT  = 'HBW_%_iter_%_.PTT';
HBSTDOUT  = 'HBS_%_iter_%_.PTT';
HBOTDOUT  = 'HBO_%_iter_%_.PTT';
NHWTDOUT  = 'NHW_%_iter_%_.PTT';
NHOTDOUT  = 'NHO_%_iter_%_.PTT';
COMTDOUT  = 'COM_%_iter_%_.PTT';
MTKTDOUT  = 'MTK_%_iter_%_.PTT';
HTKTDOUT  = 'HTK_%_iter_%_.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)



; //////////////////////////////////////////////////////////////////////
; \\\\\\\\\    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

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]
    ENDIF
    IF  (MW[2] > 0)
         MW[4] = MW[2] + zi.1.hterm[I] + zi.1.hterm[J]
         ELSE
         MW[4] = MW[2]
    ENDIF
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))

 ENDIF
ENDJLOOP
; WRITE OUT FINAL TIME SKIMS

 MATO[1] = SOVAMTT.SKF, MO=3; output am sov time(min) w/ o&d term&intra times
 MATO[2] = SOVMDTT.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
   ENDIF

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] = SOVAMTT.SKF  ;  AM  PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
MATI[2] = SOVMDTT.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
  ENDIF
  IF (MW[2] = 0.0)
      MW[2] = 1.0
  ENDIF
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
; 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] = HBWCI1_4.DAT, MO=15,16,17,18 ;HBW COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[2] = HBSCI1_4.DAT, MO=20,21,22,23 ;HBS COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[3] = HBOCI1_4.DAT, MO=25,26,27,28 ;HBO COMP.IMPEDANCES-INC.LEVELS 1-4
MATO[4] = NHBCI.DAT,    MO=50,51       ;NHW/NHO  COMP.IMPEDANCES
MATO[5] = SOVAMTTX.SKF, MO=76          ; AM Peak  X-I, I-X impedances with tolls
MATO[6] = SOVMDTTX.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] = SOVAMTTX.SKF  ;  AM  PK HWY TIME FILE W/ TERM&INTRAZNL VALUES
MATI[2] = SOVMDTTX.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 ij's
;  MW[12] - Midday  Period, External ij'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] = SOVAMTTE.skf, MO=11 ;  AM -PK Time skims for Extl trip dist.
MATO[2] = SOVMDTTE.skf, MO=12 ;  Midday Time skims for Extl trip dist.
ENDRUN
;

; End of Composite Impedance Development

;-------------------------------------------------------------------------------
; Trip Distribution Model Calibration Process
;-------------------------------------------------------------------------------
;

; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
; |//////      Start HBW Trip Distribution Here:            /////|
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|



RUN PGM=DISTRIBUTION
  zones= 3722
  MATI= HBWCI1_4.DAT,   ; Composite Time Impedances HBW Inc.Levels 1-4  #1
        @HBWK@,         ; HBW Kfactors (Scaled by 1000.0)               #2
        SOVAMTTE.skf,   ; AM -PK Time skims for Extl trip dist.         #3
        SOVMDTTE.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]     = @AutoProds@ ;  internal auto  productions file
ZDATI[2]     = @AutoAttrs@ ;  internal auto  attractions file
ZDATI[3]     = @ExtPsAs@   ;  External Ps,As attractions 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,   ;
       LOOKUP[5]  = IMP, RESULT=HBWEI,     ;
       LOOKUP[6]  = IMP, RESULT=HBWEA,     ;
       INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

;  Establish production and attraction vectors here:

SETPA P[1]=ZI.1.HBWMTRP_I1, P[2]=ZI.1.HBWMTRP_I2, P[3]=ZI.1.HBWMTRP_I3,
P[4]=ZI.1.HBWMTRP_I4, P[5]=ZI.3.SHBW_MtrPs, P[6]=ZI.3.SHBW_MtrPs
SETPA A[1]=ZI.2.HBWMTRA_I1, A[2]=ZI.2.HBWMTRA_I2, A[3]=ZI.2.HBWMTRA_I3,
A[4]=ZI.2.HBWMTRA_I4, A[5]=ZI.3.SHBW_MtrAs, A[6]=ZI.3.SHBW_MtrAs

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]
GRAVITY  PURPOSE  = 5, LOS=MW[21], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21-am-HBW 31-md/nonHBW
GRAVITY  PURPOSE  = 6, LOS=MW[21], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21/am-HBW 31-op/nonHBW

;REPORT ZDAT = Y
;REPORT ACOMP=1-6

MATO = HBW.TEM,MO=1-6   ;   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)
                         ;      T4 - HBW Inc. Level 4 (i-i)
                         ;      T5 - externals/ using interstate facility FFactors
                         ;      T6 - externals/ using arterial   facility FFactors
ENDRUN

;; --ENB HBW Trip Dist---;;
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
; |//////      Start HBS Trip Distribution Here:            /////|
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|

RUN PGM=DISTRIBUTION
  zones= 3722
  MATI= HBSCI1_4.DAT,   ; Composite Time Impedances HBW Inc.Levels 1-4  #1
        @HBSK@,         ; HBW Kfactors (Scaled by 1000.0)               #2
        SOVAMTTE.skf,   ; AM -PK Time skims for Extl trip dist.         #3
        SOVMDTTE.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]     = @AutoProds@ ;  internal auto  productions file
ZDATI[2]     = @AutoAttrs@ ;  internal auto  attractions file
ZDATI[3]     = @ExtPsAs@   ;  External Ps,As attractions 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,   ;
       LOOKUP[5]  = IMP, RESULT=HBSEI,     ;
       LOOKUP[6]  = IMP, RESULT=HBSEA,     ;
       INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

;  Establish production and attraction vectors here:

SETPA P[1]=ZI.1.HBSMTRP_I1, P[2]=ZI.1.HBSMTRP_I2, P[3]=ZI.1.HBSMTRP_I3,
P[4]=ZI.1.HBSMTRP_I4, P[5]=ZI.3.SHBS_MtrPs, P[6]=ZI.3.SHBS_MtrPs
SETPA A[1]=ZI.2.HBSMTRA_I1, A[2]=ZI.2.HBSMTRA_I2, A[3]=ZI.2.HBSMTRA_I3,
A[4]=ZI.2.HBSMTRA_I4, A[5]=ZI.3.SHBS_MtrAs, A[6]=ZI.3.SHBS_MtrAs

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]
GRAVITY  PURPOSE  = 5, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21-am-HBW 31-md/nonHBW
GRAVITY  PURPOSE  = 6, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21/am-HBW 31-op/nonHBW

;REPORT ZDAT = Y
;REPORT ACOMP=1-6

MATO = HBS.TEM,MO=1-6   ;   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)
                         ;      T5 - externals/ using interstate facility FFactors
```

```
                         ;       T6 - externals/ using arterial   facility FFactors
ENDRUN

;; --ENB HBS Trip Dist---;;
;

; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
; |//////      Start HBO Trip Distribution Here:             /////|
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|


RUN PGM=DISTRIBUTION
  zones= 3722
  MATI= HBOCI1_4.DAT,   ; Composite Time Impedances HBW Inc.Levels 1-4  #1
        @HBOK@,         ; HBW Kfactors (Scaled by 1000.0)               #2
        SOVAMTTE.skf,   ; AM -PK Time skims for Extl trip dist.         #3
        SOVMDTTE.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]     = @AutoProds@ ;  internal auto  productions file
ZDATI[2]     = @AutoAttrs@ ;  internal auto  attractions file
ZDATI[3]     = @ExtPsAs@   ;  External Ps,As attractions 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,   ;
       LOOKUP[5]  = IMP, RESULT=HBOEI,     ;
       LOOKUP[6]  = IMP, RESULT=HBOEA,     ;
       INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

;  Establish production and attraction vectors here:

SETPA P[1]=ZI.1.HBOMTRP_I1, P[2]=ZI.1.HBOMTRP_I2, P[3]=ZI.1.HBOMTRP_I3,
P[4]=ZI.1.HBOMTRP_I4, P[5]=ZI.3.SHBO_MtrPs, P[6]=ZI.3.SHBO_MtrPs
SETPA A[1]=ZI.2.HBOMTRA_I1, A[2]=ZI.2.HBOMTRA_I2, A[3]=ZI.2.HBOMTRA_I3,
A[4]=ZI.2.HBOMTRA_I4, A[5]=ZI.3.SHBO_MtrAs, A[6]=ZI.3.SHBO_MtrAs

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]
GRAVITY  PURPOSE  = 5, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21-am-HBW 31-md/nonHBW
GRAVITY  PURPOSE  = 6, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;21/am-HBW 31-op/nonHBW
```

```
;REPORT ZDAT = Y
;REPORT ACOMP=1-6

MATO = HBO.TEM,MO=1-6   ;   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)
                         ;       T5 - externals/ using interstate facility FFactors
                         ;       T6 - externals/ using arterial   facility FFactors
ENDRUN

;; --ENB HBO Trip Dist---;;
;

; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
; |//////      Start NHW/NHO Trip Distribution Here:         /////|
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|

RUN PGM=DISTRIBUTION
  zones= 3722
  MATI=   NHBCI.DAT,      ; Composite Time Impedances NHW/NHO T1&2   file 1
          SOVMDTTE.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]     = @ExtPsAs@   ;  External Ps,As attractions 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,       ;
       LOOKUP[3]  = IMP, RESULT=NHBEI,     ;
       LOOKUP[4]  = IMP, RESULT=NHBEA,     ;
       LOOKUP[5]  = IMP, RESULT=NHBEI,     ;
       LOOKUP[6]  = IMP, RESULT=NHBEA,     ;
       INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

;  Establish production and attraction vectors here:

SETPA P[1]=ZI.1.NHWIIAs, P[2]=ZI.1.NHOIIAs, P[3]=ZI.1.SNHW_MtrAs,
P[4]=ZI.1.SNHW_MtrAs, P[5]=ZI.1.SNHO_MtrAs, P[6]=ZI.1.SNHO_MtrAs
SETPA A[1]=ZI.1.NHWIIAs, A[2]=ZI.1.NHOIIAs, A[3]=ZI.1.SNHW_MtrAs,
A[4]=ZI.1.SNHW_MtrAs, A[5]=ZI.1.SNHO_MtrAs, A[6]=ZI.1.SNHO_MtrAs

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
GRAVITY  PURPOSE  = 3, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;NHW /INTERSTATE FFS
GRAVITY  PURPOSE  = 4, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[21],LOSRANGE=2-250.
;;NHW /ARTERIAL   FFS
GRAVITY  PURPOSE  = 5, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[20],LOSRANGE=2-250.
;;NHO /INTERSTATE FFS
GRAVITY  PURPOSE  = 6, LOS=MW[31], FFACTORS= FF, KFACTORS = MW[21],LOSRANGE=2-250.
;;NHO /ARTERIAL   FFS

;REPORT ZDAT = Y
;REPORT ACOMP=1-6

MATO = NHB.TEM,MO=1-6   ;   Final NHB  trip table(s)
                        ;       T1 - NHW    INTL
                        ;       T2 - NHO    INTL
                        ;       T3 - NHW    EXTL      interstate facility FFactors
                        ;       T4 - NHW    EXTL      arterial   facility FFactors
                        ;       T5 - NHO    EXTL      interstate facility FFactors
                        ;       T6 - NHO    EXTL      arterial   facility FFactors
ENDRUN
;; --ENB NHB Trip Dist---;;
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|
; |//////      Start COM/TRK Trip Distribution Here:               /////|
; |\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\|

RUN PGM=TRIPDIST
  MATI[1] = SOV%_prev_%MD.SKM       ; Off-Pk Time Imped. for COM
  MATI[2] = trk%_prev_%md.skm       ; Off-Pk Truck Time for MTK/HTK
  MATI[3] = SOVMDTTE.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]  = @TruckEnds@
ZDATI[2]  = @Ext_TrkEnds@

;  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
       LOOKUP[4]  = IMP, RESULT=EXTCOM,     ; Ext CV
       LOOKUP[5]  = IMP, RESULT=EXTMTK,     ; Ext Mtk
       LOOKUP[6]  = IMP, RESULT=EXTHTK,     ; Ext Htk
       INTERPOLATE=N,SETUPPER=T,FAIL=0,0,0

;  Establish production and attraction vectors here:

SETPA P[1]=ZI.1.IComm_Veh, P[2]=ZI.1.IMed_Truck, P[3]=ZI.1.IHvy_Truck,
P[4]=ZI.2.SCOM_VEHPS, P[5]=ZI.2.SMED_TRKPS, P[6]=ZI.2.SHVY_TRKPS
SETPA A[1]=ZI.1.IComm_Veh, A[2]=ZI.1.IMed_Truck, A[3]=ZI.1.IHvy_Truck,
A[4]=ZI.2.SCOM_VEHAS, A[5]=ZI.2.SMED_TRKAS, A[6]=ZI.2.SHVY_TRKAS

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
GRAVITY  PURPOSE  = 4, LOS=MW[13], FFACTORS= FF,losrange=2-250     ; COM External
GRAVITY  PURPOSE  = 5, LOS=MW[13], FFACTORS= FF,losrange=2-250     ; MTK External
GRAVITY  PURPOSE  = 6, LOS=MW[13], FFACTORS= FF,losrange=2-250     ; HTK External

MATO[1] = COM.TEM,MO=1,4 ; Final COM trip tables: 1 = I/I, 2 = Extl
MATO[2] = MTK.TEM,MO=2,5 ; Final MTK trip tables: 1 = I/I, 2 = Extl
MATO[3] = HTK.TEM,MO=3,6 ; Final HTK trip tables: 1 = I/I, 2 = 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] = HBW.TEM    ;    6 HBW  trip tables:  Inc1,..,Inc4, Ext/InterstFFs,
Extls/ArterFFs
MATI[2] = HBS.TEM    ;    6 HBS  trip tables:  Inc1,..,Inc4, Ext/InterstFFs,
Extls/ArterFFs
MATI[3] = HBO.TEM    ;    6 HBO  trip tables:  Inc1,..,Inc4, Ext/InterstFFs,
Extls/ArterFFs
MATI[4] = NHB.TEM    ;    6 NHB  trip tables:  NHW I-I, NHO I-I, NHW Extl/IntFFs,NHW
Extl/ArtFFs, NHO Extl/IntFFs,NHO Extl/ArtFFs
MATI[5] = COM.TEM    ;    2 Com  trip tables:  I/I, Extl
MATI[6] = MTK.TEM    ;    2 Mtk  trip tables:  I/I, Extl
MATI[7] = HTK.TEM    ;    2 Htk  trip tables:  I/I, Extl

FillMW MW[101]=mi.1.1,2,3,4,5,6  ; HBW tabs in mw 101-106
FillMW MW[201]=mi.2.1,2,3,4,5,6  ; HBS tabs in mw 201-206
FillMW MW[301]=mi.3.1,2,3,4,5,6  ; HBO tabs in mw 301-306
FillMW MW[401]=mi.4.1,2,3,4,5,6  ; NHW/NHO tabs in mw 401-406

FillMW MW[601]=mi.5.1,2          ; Com tabs in mw 501-506
FillMW MW[701]=mi.6.1,2          ; Mtk tabs in mw 601-606
FillMW MW[801]=mi.7.1,2          ; Htk tabs in mw 701-706

;; 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[105] * mw[11]) + (MW[106] * mw[22])  ; Final HBW External trip
tables
MW[207] =   (MW[205] * mw[11]) + (MW[206] * mw[22])  ;       HBS External trip
tables
MW[307] =   (MW[305] * mw[11]) + (MW[306] * mw[22])  ;       HBO External trip
tables

MW[407] =   (MW[403] * mw[11]) + (MW[404] * mw[22])  ;       NHW External trip
tables
MW[507] =   (MW[405] * mw[11]) + (MW[406] * mw[22])  ;       NHO External trip
tables

;;
;;Compute Total Person Trips matrix
MW[108] =    MW[101] + MW[102] + MW[103] + MW[104] + MW[107]    ; Total HBW
Motorized Person Trip tabs (II,IX,XI)
MW[208] =    MW[201] + MW[202] + MW[203] + MW[204] + MW[207]    ; Final HBS
Motorized Person Trip tabs (II,IX,XI)
MW[308] =    MW[301] + MW[302] + MW[303] + MW[304] + MW[307]    ; Final HBO
Motorized Person Trip tabs (II,IX,XI)
MW[408] =    MW[401] +                             MW[407]    ; Final NHW
Motorized Person Trip tabs (II,IX,XI)
MW[508] =    MW[402] +                             MW[507]    ; Final NHO
Motorized Person Trip tabs (II,IX,XI)
MW[608] =    MW[601] + MW[602]                                ; Final Commercial
Vehicle Trips       (II,IX,XI)
MW[708] =    MW[701] + MW[702]                                ; Final Medium Truck
Trips             (II,IX,XI)
MW[808] =    MW[801] + MW[802]                                ; Final Heavy  Truck
Trips             (II,IX,XI)

 ;; write out final matrices comprehensive tabs
MATO[1] = @HBWTDOUT@ ,
MO=101,102,103,104,107,108,name=HBWI1Psn,HBWI2Psn,HBWI3Psn,HBWI4Psn,HBW_Ext,HBWAllPs
n
MATO[2] = @HBSTDOUT@ ,
MO=201,202,203,204,207,208,name=HBSI1Psn,HBSI2Psn,HBSI3Psn,HBSI4Psn,HBS_Ext,HBSAllPs
n
MATO[3] = @HBOTDOUT@ ,
MO=301,302,303,304,307,308,name=HBOI1Psn,HBOI2Psn,HBOI3Psn,HBOI4Psn,HBO_Ext,HBOAllPs
n
MATO[4] = @NHWTDOUT@ , MO=401,407,408            ,name=NHWPsnII,NHWPsnExt
NHWAllPsn
MATO[5] = @NHOTDOUT@ , MO=402,507,508            ,name=NHOPsnII,NHOPsnExt
NHOAllPsn
MATO[6] = @COMTDOUT@ , MO=601,602,608            ,name=COM_Int ,COM_Ext,
COMAllVeh
MATO[7] = @MTKTDOUT@ , MO=701,702,708            ,name=MTK_Int ,MTK_Ext,
MTKAllVeh
MATO[8] = @HTKTDOUT@ , MO=801,802,808            ,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 (INC1..INC4)
MATO[10]= @HBSforMC@ ,MO=201,202,203,204,name=HBSI1Psn,HBSI2Psn,HBSI3Psn,HBSI4Psn
; HBS Person   Trips-4TABS (INC1..INC4)
MATO[11]= @HBOforMC@ ,MO=301,302,303,304,name=HBOI1Psn,HBOI2Psn,HBOI3Psn,HBOI4Psn
; HBO Person   Trips-4TABS (INC1..INC4)
MATO[12]= @NHWforMC@ ,MO=401            ,name=NHWPsnII
; NHW Person   Trips-1TAB  (INTERNAL)
MATO[13]= @NHOforMC@ ,MO=402            ,name=NHOPsnII
; NHO Person   Trips-1TAB  (INTERNAL)

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 Jurisiction-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 Jurisiction-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
  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

  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
     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],
    '      ',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
```

43 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 addjustments 9/14/2011
;
;==================================================================
;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  = 'Demo_Models_HHbyISV_%_iter_%.dbf'     ;  HHs by Income Size Vehs
Avail

ZNFILE_IN5  = 'TripGen_LUFile.dbf'                  ;  Consolidated zonal input
file (intermediate I/O file)

ReportFile  = 'Trip_Generation_%_iter_%.txt'            ; Trip Gen. Report file
TripPros    = 'Trip_Gen_Productions_%_iter_%.dbf'       ; Zonal Trip productions by
purpose
TripAttsCom = 'Trip_Gen_Attractions_Comp_%_iter_%.dbf'   ; Zonal Trip Attractions -
Initial/computed
TripAttsFin = 'Trip_Gen_Attractions_Final_%_iter_%.dbf'  ; Zonal Trip Attractions -
Final/scaled
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.
NMArate_in  ='..\support\NMArates.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  ;
MHBSAAdj1=0.7952  MHBSAAdj2=1.0967  MHBSAAdj3=1.1577  MHBSAAdj4=0.8770
MHBSAAdj5=0.9437   MHBSAAdj6=0.5187  ;
MHBOAAdj1=1.1542  MHBOAAdj2=1.1304  MHBOAAdj3=0.9307  MHBOAAdj4=1.0635
MHBOAAdj5=1.0480   MHBOAAdj6=0.8032  ;
MNHWAAdj1=1.1457  MNHWAAdj2=0.8686  MNHWAAdj3=0.9843  MNHWAAdj4=1.5731
MNHWAAdj5=1.1860   MNHWAAdj6=1.0919  ;
MNHOAAdj1=0.7953  MNHOAAdj2=1.0652  MNHOAAdj3=1.0724  MNHOAAdj4=0.9180
MNHOAAdj5=1.0899   MNHOAAdj6=0.7224  ;

;; NONMOTORIZED PRODUCTIONS
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  ;
NNHOPAdj1=1.0000  NNHOPAdj2=1.0000  NNHOPAdj3=1.0000  NNHOPAdj4=1.0000
NNHOPAdj5=1.0000   NNHOPAdj6=1.0000  ;

;; NONMOTORIZED ATTRACTIONS
NHBWAAdj1=1.0300  NHBWAAdj2=1.0000  NHBWAAdj3=1.1100  NHBWAAdj4=1.1100
NHBWAAdj5=1.1300   NHBWAAdj6=1.1000  ;
NHBSAAdj1=1.8400  NHBSAAdj2=1.2900  NHBSAAdj3=1.0900  NHBSAAdj4=1.1000
NHBSAAdj5=1.0000   NHBSAAdj6=1.0000  ;
NHBOAAdj1=0.6000  NHBOAAdj2=1.0600  NHBOAAdj3=1.1100  NHBOAAdj4=1.0900
NHBOAAdj5=1.1000   NHBOAAdj6=1.0800  ;
NNHWAAdj1=1.0000  NNHWAAdj2=1.0000  NNHWAAdj3=1.0000  NNHWAAdj4=1.0000
NNHWAAdj5=1.0000   NNHWAAdj6=1.0000  ;
NNHOAAdj1=1.6600  NNHOAAdj2=1.0000  NNHOAAdj3=0.7000  NNHOAAdj4=0.7000
NNHOAAdj5=0.7000   NNHOAAdj6=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]    = "@ZNFile_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] = "@ZNFILE_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] = "@ZNFILE_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] = "@ZNFILE_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

           Xdiff   = abs(Xi-Xj)                            ; calc. airline distance
           Ydiff   = abs(Yi-Yj)                            ;
                                                           ;
           d_ft    = sqrt(xdiff*xdiff + Ydiff*Ydiff)       ;
           d_mi    = d_ft/5280.0                           ;
                                                           ;
           ;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 IniAttra   =@InCl@,@PrCl@
Array FinAttra   =@InCl@,@PrCl@
Array IniAtot    =@PrCl@
Array FinAtot    =@PrCl@
Array Scaltot    =@PrCl@
Array Mscale     =@PrCl@
Array NMscale    =@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@,
      XMTotProda         =@PrCl@,
      NMTotProda         =@PrCl@,
      MTotAttra          =@PrCl@,
      XMTotAttra         =@PrCl@,
      NMTotAttra         =@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@,

      HBWCompATT         =@zonesize@,
      HBSCompATT         =@zonesize@,
      HBOCompATT         =@zonesize@,
      NHWCompATT         =@zonesize@,
      NHOCompATT         =@zonesize@,

      HBWScalATT         =@zonesize@,
      HBSScalATT         =@zonesize@,
      HBOScalATT         =@zonesize@,
      NHWScalATT         =@zonesize@,
      NHOScalATT         =@zonesize@

 Array HBWATTInca        =@zonesize@,@InCl@
 Array HBSATTInca        =@zonesize@,@InCl@
 Array HBOATTInca        =@zonesize@,@InCl@
 Array NHWATTInca        =@zonesize@
 Array NHOATTInca        =@zonesize@
;;-----------------------------------------------
;;-----------------------------------------------

Array MPro_Adj = @PrCl@,@ArCl@
Array MAtt_Adj = @PrCl@,@ArCl@
Array NPro_Adj = @PrCl@,@ArCl@
Array NAtt_Adj = @PrCl@,@ArCl@

; fill purpose and area type adjustments
;;motorized adjustments
MPro_Adj[1][1]=@MHBWPAdj1@  MPro_Adj[2][1]=@MHBSPAdj1@   MPro_Adj[3][1]= @MHBOPAdj1@
MPro_Adj[4][1]= @MNHWPAdj1@  MPro_Adj[5][1]=@MNHOPAdj1@
MPro_Adj[1][2]=@MHBWPAdj2@  MPro_Adj[2][2]=@MHBSPAdj2@   MPro_Adj[3][2]= @MHBOPAdj2@
MPro_Adj[4][2]= @MNHWPAdj2@  MPro_Adj[5][2]=@MNHOPAdj2@
MPro_Adj[1][3]=@MHBWPAdj3@  MPro_Adj[2][3]=@MHBSPAdj3@   MPro_Adj[3][3]= @MHBOPAdj3@
MPro_Adj[4][3]= @MNHWPAdj3@  MPro_Adj[5][3]=@MNHOPAdj3@
MPro_Adj[1][4]=@MHBWPAdj4@  MPro_Adj[2][4]=@MHBSPAdj4@   MPro_Adj[3][4]= @MHBOPAdj4@
MPro_Adj[4][4]= @MNHWPAdj4@  MPro_Adj[5][4]=@MNHOPAdj4@
MPro_Adj[1][5]=@MHBWPAdj5@  MPro_Adj[2][5]=@MHBSPAdj5@   MPro_Adj[3][5]= @MHBOPAdj5@
MPro_Adj[4][5]= @MNHWPAdj5@  MPro_Adj[5][5]=@MNHOPAdj5@
MPro_Adj[1][6]=@MHBWPAdj6@  MPro_Adj[2][6]=@MHBSPAdj6@   MPro_Adj[3][6]= @MHBOPAdj6@
MPro_Adj[4][6]= @MNHWPAdj6@  MPro_Adj[5][6]=@MNHOPAdj6@

MAtt_Adj[1][1]=@MHBWAAdj1@  MAtt_Adj[2][1]=@MHBSAAdj1@   MAtt_Adj[3][1]= @MHBOAAdj1@
MAtt_Adj[4][1]= @MNHWAAdj1@  MAtt_Adj[5][1]=@MNHOAAdj1@
MAtt_Adj[1][2]=@MHBWAAdj2@  MAtt_Adj[2][2]=@MHBSAAdj2@   MAtt_Adj[3][2]= @MHBOAAdj2@
MAtt_Adj[4][2]= @MNHWAAdj2@  MAtt_Adj[5][2]=@MNHOAAdj2@
MAtt_Adj[1][3]=@MHBWAAdj3@  MAtt_Adj[2][3]=@MHBSAAdj3@   MAtt_Adj[3][3]= @MHBOAAdj3@
MAtt_Adj[4][3]= @MNHWAAdj3@  MAtt_Adj[5][3]=@MNHOAAdj3@
MAtt_Adj[1][4]=@MHBWAAdj4@  MAtt_Adj[2][4]=@MHBSAAdj4@   MAtt_Adj[3][4]= @MHBOAAdj4@
MAtt_Adj[4][4]= @MNHWAAdj4@  MAtt_Adj[5][4]=@MNHOAAdj4@
MAtt_Adj[1][5]=@MHBWAAdj5@  MAtt_Adj[2][5]=@MHBSAAdj5@   MAtt_Adj[3][5]= @MHBOAAdj5@
MAtt_Adj[4][5]= @MNHWAAdj5@  MAtt_Adj[5][5]=@MNHOAAdj5@
```

```
MAtt_Adj[1][6]=@MHBWAAdj6@  MAtt_Adj[2][6]=@MHBSAAdj6@   MAtt_Adj[3][6]= @MHBOAAdj6@
MAtt_Adj[4][6]= @MNHWAAdj6@  MAtt_Adj[5][6]=@MNHOAAdj6@


;;nonmotorized adjustments
NPro_Adj[1][1]=@NHBWPAdj1@  NPro_Adj[2][1]=@NHBSPAdj1@   NPro_Adj[3][1]= @NHBOPAdj1@
NPro_Adj[4][1]= @NNHWPAdj1@  NPro_Adj[5][1]=@NNHOPAdj1@
NPro_Adj[1][2]=@NHBWPAdj2@  NPro_Adj[2][2]=@NHBSPAdj2@   NPro_Adj[3][2]= @NHBOPAdj2@
NPro_Adj[4][2]= @NNHWPAdj2@  NPro_Adj[5][2]=@NNHOPAdj2@
NPro_Adj[1][3]=@NHBWPAdj3@  NPro_Adj[2][3]=@NHBSPAdj3@   NPro_Adj[3][3]= @NHBOPAdj3@
NPro_Adj[4][3]= @NNHWPAdj3@  NPro_Adj[5][3]=@NNHOPAdj3@
NPro_Adj[1][4]=@NHBWPAdj4@  NPro_Adj[2][4]=@NHBSPAdj4@   NPro_Adj[3][4]= @NHBOPAdj4@
NPro_Adj[4][4]= @NNHWPAdj4@  NPro_Adj[5][4]=@NNHOPAdj4@
NPro_Adj[1][5]=@NHBWPAdj5@  NPro_Adj[2][5]=@NHBSPAdj5@   NPro_Adj[3][5]= @NHBOPAdj5@
NPro_Adj[4][5]= @NNHWPAdj5@  NPro_Adj[5][5]=@NNHOPAdj5@
NPro_Adj[1][6]=@NHBWPAdj6@  NPro_Adj[2][6]=@NHBSPAdj6@   NPro_Adj[3][6]= @NHBOPAdj6@
NPro_Adj[4][6]= @NNHWPAdj6@  NPro_Adj[5][6]=@NNHOPAdj6@

NAtt_Adj[1][1]=@NHBWAAdj1@  NAtt_Adj[2][1]=@NHBSAAdj1@   NAtt_Adj[3][1]= @NHBOAAdj1@
NAtt_Adj[4][1]= @NNHWAAdj1@  NAtt_Adj[5][1]=@NNHOAAdj1@
NAtt_Adj[1][2]=@NHBWAAdj2@  NAtt_Adj[2][2]=@NHBSAAdj2@   NAtt_Adj[3][2]= @NHBOAAdj2@
NAtt_Adj[4][2]= @NNHWAAdj2@  NAtt_Adj[5][2]=@NNHOAAdj2@
NAtt_Adj[1][3]=@NHBWAAdj3@  NAtt_Adj[2][3]=@NHBSAAdj3@   NAtt_Adj[3][3]= @NHBOAAdj3@
NAtt_Adj[4][3]= @NNHWAAdj3@  NAtt_Adj[5][3]=@NNHOAAdj3@
NAtt_Adj[1][4]=@NHBWAAdj4@  NAtt_Adj[2][4]=@NHBSAAdj4@   NAtt_Adj[3][4]= @NHBOAAdj4@
NAtt_Adj[4][4]= @NNHWAAdj4@  NAtt_Adj[5][4]=@NNHOAAdj4@
NAtt_Adj[1][5]=@NHBWAAdj5@  NAtt_Adj[2][5]=@NHBSAAdj5@   NAtt_Adj[3][5]= @NHBOAAdj5@
NAtt_Adj[4][5]= @NNHWAAdj5@  NAtt_Adj[5][5]=@NNHOAAdj5@
NAtt_Adj[1][6]=@NHBWAAdj6@  NAtt_Adj[2][6]=@NHBSAAdj6@   NAtt_Adj[3][6]= @NHBOAAdj6@
NAtt_Adj[4][6]= @NNHWAAdj6@  NAtt_Adj[5][6]=@NNHOAAdj6@
;;-------------------------------------------------
;;-------------------------------------------------
;; Read in Consolidated zone file

 ZDATI[1] = @ZNFILE_IN5@ ;; variables in DBF file: TAZ,  HH, HHPOP, JURCODE,
HHINCIDX
                         ;; TAZ,       HH,     TOTPOP, TOTEMP,  RETEMP,  NRETEMP,
OFFEMP,  OTHEMP,
                         ;; INDEMP,    POP_10, EMP_10, AREA_10,
POPDEN10,EMPDEN10,BLOCKS05,AREA05,
                         ;; BLOCKDEN05,JURCODE,ATYPE,ADISTTOX

Atypea[i] = zi.1.Atype   ; populate zonal area type array

;; Identify 'core' TAZs to be used for P-A-mod adjustments

coreflag = 0
 IF (I=1-4,6-47,49-50,52-63,65,181-209,282-287,374-381 ) coreflag = 1   ;DC core
 IF (I=1471-1476, 1486-1489, 1495-1497                ) coreflag = 1   ;Arl Core


;;==================================================================================
================
; Define Jurisdiction Motorized Production, Attraction Adjustment Lookup
; index = jurcode * 10 + core flag        jurcode ranges from 0-23 and core flag is
binary 0(non-core) or 1(core)
;                                          so index is from    0 - 230
;;==================================================================================
================
;
LOOKUP NAME=P_JurAdj,   ;
       LOOKUP[1] = 1, RESULT=2, ;  HBW Production Adjustment
       LOOKUP[2] = 1, RESULT=3, ;  HBS Production Adjustment
       LOOKUP[3] = 1, RESULT=4, ;  HBO Production Adjustment
       LOOKUP[4] = 1, RESULT=5, ;  NHW Production Adjustment
       LOOKUP[5] = 1, RESULT=6, ;  NHO Production Adjustment
       INTERPOLATE=N, FAIL= 1.0,1.0,1.0,
;;
;;          HBWPs  HBSPs  HBOPs  NNWPs  NHOPs
R="      0,  1.00,  0.85,  1.20,  1.00,  1.00,",  ;;dc  NONCORE
  "      1,  1.00,  0.85,  1.20,  1.00,  1.00,",  ;;dc  CORE
  "     10,  0.95,  1.00,  1.05,  1.00,  1.00,",  ;;mtg
  "     20,  1.00,  0.88,  0.97,  1.00,  1.00,",  ;;pg
  "     30,  1.00,  1.11,  1.08,  1.00,  1.00,",  ;;arl NONCORE
  "     31,  1.00,  1.11,  1.08,  1.00,  1.00,",  ;;arl CORE
  "     40,  1.00,  1.00,  1.00,  1.00,  1.00,",  ;;alx
  "     50,  1.02,  1.02,  1.02,  1.00,  1.00,",  ;;ffx
  "     60,  1.00,  0.95,  0.92,  1.00,  1.00,",  ;;ldn
  "     70,  1.04,  1.15,  0.94,  1.00,  1.00,",  ;;pw
  "     80,  1.00,  1.00,  1.00,  1.00,  1.00,",  ;;
  "     90,  1.13,  1.00,  1.04,  1.00,  1.00,",  ;;frd
  "    100,  1.00,  1.00,  0.94,  1.00,  1.00,",  ;;how
  "    110,  1.00,  1.12,  1.03,  1.00,  1.00,",  ;;aa
  "    120,  1.00,  1.00,  0.93,  1.00,  1.00,",  ;;chs
  "    130,  1.00,  1.00,  1.00,  1.00,  1.00,",  ;;
  "    140,  1.00,  1.00,  0.92,  1.00,  1.00,",  ;;car
  "    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,
;;
;;            HBWAs  HBSAs  HBOAs  NNWAs  NHOAs
R="       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
           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] = @ZNFILE_IN4@ ;; variables in DBF file:
                         ;; HHSISV111       HHSISV112       HHSISV113
HHSISV114
                         ;; HHSISV211       HHSISV212       HHSISV213
HHSISV214
                         ;; HHSISV311       HHSISV312       HHSISV313
HHSISV314
                         ;; HHSISV411       HHSISV412       HHSISV413
HHSISV414
                         ;; HHSISV121       HHSISV122       HHSISV123
HHSISV124
                         ;; HHSISV221       HHSISV222       HHSISV223
HHSISV224
                         ;; HHSISV321       HHSISV322       HHSISV323
HHSISV324
                         ;; HHSISV421       HHSISV422       HHSISV423
HHSISV424
                         ;; HHSISV131       HHSISV132       HHSISV133
HHSISV134
                         ;; HHSISV231       HHSISV232       HHSISV233
HHSISV234
                         ;; HHSISV331       HHSISV332       HHSISV333
HHSISV334
                         ;; HHSISV431       HHSISV432       HHSISV433
HHSISV434
                         ;; HHSISV141       HHSISV142       HHSISV143
HHSISV144
                         ;; HHSISV241       HHSISV242       HHSISV243
HHSISV244
                         ;; HHSISV341       HHSISV342       HHSISV343
HHSISV344
                         ;; HHSISV441       HHSISV442       HHSISV443
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
   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]
; HHs 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 HHs

             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


;;
;;  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
;;----------------------------------------------------------------------------------
----


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]                   +
               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 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) * ( 1.0 -  IX_ShareHBW )
* MPro_Adj[1][At] *  P_JurAdj(1,jurcode2) ;; compute internal HBW Motorized
productions
   NMZProda[1][i] =     Zproda[1][i] *        NMP_ShareHBW  * ( 1.0 -  IX_ShareHBW)
* MPro_Adj[1][At] *  P_JurAdj(1,jurcode2)  * NPro_Adj[1][At]  ;;       ;; 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) * ( 1.0 -  IX_ShareHBS )
* MPro_Adj[2][At] *  P_JurAdj(2,jurcode2)  ;; compute internal HBS Motorized
productions
   NMZProda[2][i] =     Zproda[2][i] *        NMP_ShareHBS  * ( 1.0 -  IX_ShareHBS)
* MPro_Adj[2][At] *  P_JurAdj(2,jurcode2)   * NPro_Adj[2][At]  ;;       ;; 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) * ( 1.0 -  IX_ShareHBO )
* MPro_Adj[3][At] *  P_JurAdj(3,jurcode2) ;; compute internal HBO Motorized
productions
   NMZProda[3][i] =     Zproda[3][i] *        NMP_ShareHBO  * ( 1.0 -  IX_ShareHBO)
* MPro_Adj[3][At] *  P_JurAdj(3,jurcode2) * NPro_Adj[3][At] ;;       ;; 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) * ( 1.0 -  IX_ShareNHW)
* MPro_Adj[4][At] *  P_JurAdj(4,jurcode2)  ;; compute internal NHW Motorized
productions
   NMZProda[4][i] =     Zproda[4][i] *        NMP_ShareNHW  * ( 1.0 -  IX_ShareNHW)
* MPro_Adj[4][At] *  P_JurAdj(4,jurcode2)  * NPro_Adj[4][At] ;;        ;; compute
internal NHW Non-Motorized productions

;;---------------NHO-----------
   MZProda[5][i]  =     Zproda[5][i] * (1.0 - NMP_ShareNHO) * ( 1.0 -  IX_ShareNHO)
* MPro_Adj[5][At] *  P_JurAdj(5,jurcode2)  ;; compute internal NHO Motorized
productions
   NMZProda[5][i] =     Zproda[5][i] *        NMP_ShareNHO  * ( 1.0 -  IX_ShareNHO)
* MPro_Adj[5][At] *  P_JurAdj(5,jurcode2)   * NPro_Adj[5][At];;         ;; compute
internal NHO Non-Motorized productions


;;--;; Debug productions calculations
if (I <300)
 print  form=6.4 list ='taz: ',i,'  IX  shares by purp: ', IX_ShareHBW,' ',
IX_ShareHBS,' ', IX_ShareHBO,' ', IX_ShareNHW,' ', IX_ShareNHO, file=
debug_P_Shares.txt
 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=6.2 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 = 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 = 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]

 ro.HBO_Mtr_Ps = MZProda[3][i]      ro.HBO_NMt_Ps = NMZProda[3][i]
ro.HBO_All_Ps = 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 = ZProda[4][i]

 ro.NHO_Mtr_Ps = MZProda[5][i]      ro.NHO_NMt_Ps = NMZProda[5][i]
ro.NHO_All_Ps = 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

   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

   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

   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

   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


;; Accumulate Regional Motor/NonMotor 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]
;;==================================================================================
========================================


   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)

   NHOCompATT[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 Internal Motor/NonMotor ATTRACTIONS by purpose

   MZAttra[1][i]  =     HBWCompATT[i] * (1.0 - NMA_ShareHBW) * MAtt_Adj[1][at]  *
A_JurAdj(1,jurcode2)     ;; compute internal HBW Motorized attractions
   NMZAttra[1][i] =     HBWCompATT[i] *       NMA_ShareHBW  * MAtt_Adj[1][at]  *
A_JurAdj(1,jurcode2)     *  NAtt_Adj[1][at]         ;; compute internal HBW Non-
Motorized attractions

   MZAttra[2][i]  =     HBSCompATT[i] * (1.0 - NMA_ShareHBS) * MAtt_Adj[2][at]  *
A_JurAdj(2,jurcode2)     ;; compute internal HBS Motorized attractions
   NMZAttra[2][i] =     HBSCompATT[i] *       NMA_ShareHBS  * MAtt_Adj[2][at]  *
A_JurAdj(2,jurcode2)     *  NAtt_Adj[2][at]         ;; compute internal HBS Non-
Motorized attractions

   MZAttra[3][i]  =     HBOCompATT[i] * (1.0 - NMA_ShareHBO) * MAtt_Adj[3][at]  *
A_JurAdj(3,jurcode2)     ;; compute internal HBO Motorized attractions
   NMZAttra[3][i] =     HBOCompATT[i] *       NMA_ShareHBO  * MAtt_Adj[3][at]  *
A_JurAdj(3,jurcode2)     *  NAtt_Adj[3][at]         ;; compute internal HBO Non-
Motorized attractions

   MZAttra[4][i]  =     NHWCompATT[i] * (1.0 - NMA_ShareNHW) * MAtt_Adj[4][at]  *
A_JurAdj(4,jurcode2)     ;; compute internal NHW Motorized attractions
   NMZAttra[4][i] =     NHWCompATT[i] *       NMA_ShareNHW  * MAtt_Adj[4][at]  *
A_JurAdj(4,jurcode2)     *  NAtt_Adj[4][at]         ;; compute internal NHW Non-
Motorized attractions

   MZAttra[5][i]  =     NHOCompATT[i] * (1.0 - NMA_ShareNHO) * MAtt_Adj[5][at]  *
A_JurAdj(5,jurcode2)     ;; compute internal NHO Motorized attractions
   NMZAttra[5][i] =     NHOCompATT[i] *       NMA_ShareNHO  * MAtt_Adj[5][at]  *
A_JurAdj(5,jurcode2)     *  NAtt_Adj[5][at]         ;; compute internal NHO Non-
Motorized attractions


;; Accumulate Regional Motor/NonMotor Totals by purpose

   MTotAttra[1]   =     MTotAttra[1] +  MZAttra[1][i]
;; compute internal HBW Motorized attractions
   NMTotAttra[1]  =    NMTotAttra[1] + NMZAttra[1][i]
;; compute internal HBW Non-Motorized attractions

   MTotAttra[2]   =     MTotAttra[2] +  MZAttra[2][i]
;; compute internal HBS Motorized attractions
   NMTotAttra[2]  =    NMTotAttra[2] + NMZAttra[2][i]
;; compute internal HBS Non-Motorized attractions
```

```
   MTotAttra[3]   =     MTotAttra[3]  + MZAttra[3][i]
;; compute internal HBO Motorized attractions
   NMTotAttra[3]  =    NMTotAttra[3] + NMZAttra[3][i]
;; compute internal HBO Non-Motorized attractions

   MTotAttra[4]   =     MTotAttra[4]  + MZAttra[4][i]
;; compute internal NHW Motorized attractions
   NMTotAttra[4]  =    NMTotAttra[4] + NMZAttra[4][i]
;; compute internal NHW Non-Motorized attractions

   MTotAttra[5]   =     MTotAttra[5]  + MZAttra[5][i]
;; compute internal NHO Motorized attractions
   NMTotAttra[5]  =    NMTotAttra[5] + NMZAttra[5][i]
;; compute internal NHO Non-Motorized attractions


;; Accumulate Regional Motor/NonMotor Totals
   MTotAttr       =  MTotAttr      +   MZAttra[1][i] +   MZAttra[2][i] +
MZAttra[3][i] +   MZAttra[4][i]  +   MZAttra[5][i]
   NMTotAttr      =  NMTotAttr     +  NMZAttra[1][i] +  NMZAttra[2][i] +
NMZAttra[3][i] +  NMZAttra[4][i]  +  NMZAttra[5][i]

;; write out dbf files for computed Trip Attractions by purpose and mode
;;
FILEO RECO[2]    = "@TripAttsCom@",fields =
                 TAZ(5),
                 HBW_Mtr_As@ofmt@, HBW_NMt_As@ofmt@, HBW_All_As@ofmt@,
                 HBS_Mtr_As@ofmt@, HBS_NMt_As@ofmt@, HBS_All_As@ofmt@,
                 HBO_Mtr_As@ofmt@, HBO_NMt_As@ofmt@, HBO_All_As@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 = i
 ro.HBW_Mtr_As = MZAttra[1][i]     ro.HBW_NMt_As = NMZAttra[1][i]   ro.HBW_All_As =
HBWCompAtt[i]
 ro.HBS_Mtr_As = MZAttra[2][i]     ro.HBS_NMt_As = NMZAttra[2][i]   ro.HBS_All_As =
HBSCompAtt[i]
 ro.HBO_Mtr_As = MZAttra[3][i]     ro.HBO_NMt_As = NMZAttra[3][i]   ro.HBO_All_As =
HBOCompAtt[i]
 ro.NHW_Mtr_As = MZAttra[4][i]     ro.NHW_NMt_As = NMZAttra[4][i]   ro.NHW_All_As =
NHWCompAtt[i]
 ro.NHO_Mtr_As = MZAttra[5][i]     ro.NHO_NMt_As = NMZAttra[5][i]   ro.NHO_All_As =
NHOCompAtt[i]

 WRITE RECO=2


;;=================================================================================
=======================================
;; 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_ShareHBW,     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),MZAttra[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),MZAttra[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),MZAttra[3][i](10), HBOATTInca[i][1](10),
HBOATTInca[i][2](10),HBOATTInca[i][3](10),HBOATTInca[i][4](10),
                                    file= debugaHBOas.txt


;;=================================================================================
===============================================
;; Disaggregate Motorized Attractions by Income
;;=================================================================================
===============================================


Loop Pr = 1,3
     IniAtot[Pr] = 0
     FinAtot[Pr] = 0
     Scaltot[Pr] = 0
     Loop In = 1,4
             IniAttra[In][Pr]= MZAttra[Pr][i]* AincRatio[In][zi.1.Atype][Pr] *
AincShare[In][zi.1.Atype][Pr]  ;; compute initial attractions by income
             IniAtot[Pr]     =                 IniAtot[Pr] + IniAttra[In][Pr]
;; accum.  initial attractions by purpose
     EndLoop
EndLoop



Loop Pr = 1,3

     if (IniAtot[Pr] = 0)
              Scaltot[Pr] = 0
         else
              Scaltot[Pr] =  MZAttra[Pr][i] / IniAtot[Pr]        ;; compute scaling
factor by purpose
     endif

     Loop In = 1,4
             FinAttra[In][Pr]= 0
             FinAttra[In][Pr]= IniAttra[In][Pr] * Scaltot[Pr]   ;; compute final
attractions by purp/income level (apply scaling factor)
             FinAtot[Pr]     = FinAtot[Pr] + FinAttra[In][Pr]   ;; accumu. final
attractions by income level
     EndLoop
EndLoop
;;; ---
  print list = 'comp HBW attractions    ',MZAttra[1][i], ' Area Type: ', zi.1.Atype
,file= debug_incdisagg.txt
  print list = 'Initial  HBW attractions ',IniAttra[1][1] , IniAttra[2][1]
,IniAttra[3][1] ,IniAttra[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 ',FinAttra[1][1] , FinAttra[2][1]
,FinAttra[3][1] ,FinAttra[4][1] , ' Sum: ', FinAtot[1] ,file= debug_incdisagg.txt

  ;;                         in  pr
  HBWATTInca[i][1] =FinAttra[1][1]
  HBWATTInca[i][2] =FinAttra[2][1]
  HBWATTInca[i][3] =FinAttra[3][1]
  HBWATTInca[i][4] =FinAttra[4][1]

  ;;                         in  pr
  HBSATTInca[i][1] =FinAttra[1][2]
```

```
  HBSATTInca[i][2] =FinAttra[2][2]
  HBSATTInca[i][3] =FinAttra[3][2]
  HBSATTInca[i][4] =FinAttra[4][2]

  ;;                        in  pr
  HBOATTInca[i][1] =FinAttra[1][3]
  HBOATTInca[i][2] =FinAttra[2][3]
  HBOATTInca[i][3] =FinAttra[3][3]
  HBOATTInca[i][4] =FinAttra[4][3]



;;--------------------------------------------------------------------------------
---------------------------------------------
ENDIF ; if I <= last internal zone
;;--------------------------------------------------------------------------------
---------------------------------------------

IF (I=@Zonesize@)  ;; If at last TAZ
;;================================================================================
===============================================
;; Now at the end of the internal TAZs-- now read external Ps& As, and then compute
scaling factor for attractions
;;================================================================================
===============================================
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@

              MZAttra[1][di.6.TAZ]    =   di.6.HBW_IX * @XOccHBW@
              MZAttra[2][di.6.TAZ]    =   di.6.HBS_IX * @XOccHBS@
              MZAttra[3][di.6.TAZ]    =   di.6.HBO_IX * @XOccHBO@
              NHBAttrs                =   di.6.NHB_IX

              MZAttra[4][di.6.TAZ]    =   NHBAttrs * @XNHW_Share@ * @XOccNHW@
              MZAttra[5][di.6.TAZ]    =   NHBAttrs * @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]

              XMTotAttra[1]           =   XMTotAttra[1] +  MZAttra[1][di.6.TAZ]
              XMTotAttra[2]           =   XMTotAttra[2] +  MZAttra[2][di.6.TAZ]
              XMTotAttra[3]           =   XMTotAttra[3] +  MZAttra[3][di.6.TAZ]
              XMTotAttra[4]           =   XMTotAttra[4] +  MZAttra[4][di.6.TAZ]
              XMTotAttra[5]           =   XMTotAttra[5] +  MZAttra[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]

           ;; zero out all of the unknown external production 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 =  0
           ro.HBS_ALL_PS =  0
           ro.HBO_ALL_PS =  0
           ro.NHW_ALL_PS =  0
           ro.NHO_ALL_PS =  0

           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
ENDLOOP



print form=12csv list =
            'Total extl productions by purpose:  ',XMTotProda[1],' ',XMTotProda[2],'
',XMTotProda[3],' ',XMTotProda[4],' ',XMTotProda[5],
            'Total extl attractions by purpose:  ',XMTotAttra[1],' ',XMTotAttra[2],'
',XMTotAttra[3],' ',XMTotAttra[4],' ',XMTotAttra[5],
            file= debug_Extlsr.txt






print list = '  HBW attrs ',  TotHBWCompAtt,
             '  HBS attrs ',  TotHBSCompAtt,
             '  HBO attrs ',  TotHBOCompAtt,
             '  NHW attrs ',  TotNHWCompAtt,
             '  NHO attrs ',  TotNHOCompAtt

print list = 'Motorized attractions by purp.    ',      MTotAttra[1]  MTotAttra[2]
MTotAttra[3]  MTotAttra[4]  MTotAttra[5]   MTotAttr
print list = 'Nonmotorized attractions by purp. ',      NMTotAttra[1] NMTotAttra[2]
NMTotAttra[3] NMTotAttra[4] NMTotAttra[5]  NMTotAttr


;; Compute scaling factors for Motorized attractions and Non-motorized attractions,
by purpose

loop m=1,5
```

```
      IF (M=1) purpo=' HBW '
      IF (M=2) purpo=' HBS '
      IF (M=3) purpo=' HBO '
      IF (M=4) purpo=' NHW '
      IF (M=5) purpo=' NHO '

      IF ( MTotAttra[M]  > 0)    Mscale[M] =  (MTotProda[M] + XMTotProda[M] -
XMTotAttra[M])/ MTotAttra[M]
      IF ( NMTotAttra[M] > 0)   NMscale[M] =  NMTotProda[M] / NMTotAttra[M]

      PRINT Printo = 1               LIST = 'Purpose:  ',purpo(a6)
      PRINT Printo = 1 FORM= 12.2csv LIST = '           Total Motorized Internal
Productions:    ',MTotProda[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           External  Productions:
',XMTotProda[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           External Attractions:
',XMTotAttra[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           Internal Computed
Attractions:           ',MTotAttra[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           Scaling factor:
',MScale[M](12.6)

      PRINT Printo = 1 FORM= 12.2csv LIST = '    '

      PRINT Printo = 1 FORM= 12.2csv LIST = '           Total NonMotorized
Productions:          ', NMTotProda[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           Total Computed NonMotorized
Attractions: ', NMTotAttra[M]
      PRINT Printo = 1 FORM= 12.2csv LIST = '           NonMotor Scaling Factor:
', NMScale[M](12.6)
      PRINT Printo = 1 FORM= 12.2csv LIST = '    '
      PRINT Printo = 1 FORM= 12.2csv LIST = '    '


endloop


;; Apply scaling factors to INTERNAL attractions
LOOP ZZ= 1,@Zonesize@
     IF (ZZ<= @LastIZN@)
         Loop PP= 1,5
               MZAttra[PP][ZZ]  = MZAttra[PP][ZZ]   *  MScale[PP]
              NMZAttra[PP][ZZ]  = NMZAttra[PP][ZZ]  * NMScale[PP]
         ENDLOOP

              HBWAttInca[ZZ][1] = HBWAttInca[ZZ][1] *  MScale[1]
              HBWAttInca[ZZ][2] = HBWAttInca[ZZ][2] *  MScale[1]
              HBWAttInca[ZZ][3] = HBWAttInca[ZZ][3] *  MScale[1]
              HBWAttInca[ZZ][4] = HBWAttInca[ZZ][4] *  MScale[1]

              HBSAttInca[ZZ][1] = HBSAttInca[ZZ][1] *  MScale[2]
              HBSAttInca[ZZ][2] = HBSAttInca[ZZ][2] *  MScale[2]
              HBSAttInca[ZZ][3] = HBSAttInca[ZZ][3] *  MScale[2]
              HBSAttInca[ZZ][4] = HBSAttInca[ZZ][4] *  MScale[2]

              HBOAttInca[ZZ][1] = HBOAttInca[ZZ][1] *  MScale[3]
              HBOAttInca[ZZ][2] = HBOAttInca[ZZ][2] *  MScale[3]
              HBOAttInca[ZZ][3] = HBOAttInca[ZZ][3] *  MScale[3]
              HBOAttInca[ZZ][4] = HBOAttInca[ZZ][4] *  MScale[3]
     ENDIF

;; write out dbf files for Final/Scaled Trip Attractions by purpose and mode
;;
FILEO RECO[3]    = "@TripAttsFin@",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@,

                 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 = MZAttra[1][zz]    ro.HBW_NMt_As = NMZAttra[1][zz]   ro.HBW_All_As =
MZAttra[1][zz] + NMZAttra[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 = MZAttra[2][zz]    ro.HBS_NMt_As = NMZAttra[2][zz]   ro.HBS_All_As =
MZAttra[2][zz] + NMZAttra[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 = MZAttra[3][zz]    ro.HBO_NMt_As = NMZAttra[3][zz]   ro.HBO_All_As =
MZAttra[3][zz] + NMZAttra[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 = MZAttra[4][zz]    ro.NHW_NMt_As = NMZAttra[4][zz]   ro.NHW_All_As =
NHWCompAtt[zz]
 ro.NHO_Mtr_As = MZAttra[5][zz]    ro.NHO_NMt_As = NMZAttra[5][zz]   ro.NHO_All_As =
NHOCompAtt[zz]

 WRITE RECO=3




 ENDLOOP


;;
;; -------------------------------------------------------------------------------
;; 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]
   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, file=dud.dat
endloop



;;
;; ----------------------------------------------------------------------------
;;
ENDIF ;; If at last TAZ


ENDRUN
*copy voya*.prn mod2.rpt
```

44 Trip_Generation_Summary.s

```
*del Voya*.prn
;
ReportFile ='Trip_Generation_Summary_%_Iter_%.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,    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,    0,   AnnAr       ,
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,    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] = "Trip_Gen_productions_%_iter_%.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]="Trip_Gen_Attractions_Final_%_iter_%.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                                                   =
 ;;================================================================================

 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
      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
    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]
 ENDLOOP


  Print PRINTO=1 LIST='                          '
```

```
  Print form=10csv PRINTO=1 LIST='      TOTAL                ',
                                 TOTHBWMTRPs, ' ',TOTHBSMTRPs,' ',TOTHBOMTRPs,'
',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                ',
                                 TOTHBWNMTPs, ' ',TOTHBSNMTPs,' ',TOTHBONMTPs,'
',TOTNHWNMTPs,' ',TOTNHONMTPs,' ',TOTNMTPs


;; ------------------------------------------------------------------------------
----------------------------------------------------
 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                ',
                                 TOTHBWNMTPs, ' ',TOTHBSNMTPs,' ',TOTHBONMTPs,'
',TOTNHWNMTPs,' ',TOTNHONMTPs,' ',TOTNMTPs

; -------------------------------------------------------------------------------
---------------------------------------------------
 Print PRINTO=1 LIST= '\n','\n',' Home-Based Motorized Trip Productions 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_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                ',
                                 TOTHBWMTRPs_i1, ' ',TOTHBWMTRPs_i2,'
',TOTHBWMTRPs_i3,' ',TOTHBWMTRPs_i4,' ',
                                 TOTHBSMTRPs_i1, ' ',TOTHBSMTRPs_i2,'
',TOTHBSMTRPs_i3,' ',TOTHBSMTRPs_i4,' ',
                                 TOTHBOMTRPs_i1, ' ',TOTHBOMTRPs_i2,'
',TOTHBOMTRPs_i3,' ',TOTHBOMTRPs_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_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_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                ',
                                 TOTHBWMTRPs_i1, ' ',TOTHBWMTRPs_i2,'
',TOTHBWMTRPs_i3,' ',TOTHBWMTRPs_i4,' ',
                                 TOTHBSMTRPs_i1, ' ',TOTHBSMTRPs_i2,'
',TOTHBSMTRPs_i3,' ',TOTHBSMTRPs_i4,' ',
                                 TOTHBOMTRPs_i1, ' ',TOTHBOMTRPs_i2,'
',TOTHBOMTRPs_i3,' ',TOTHBOMTRPs_i4



;;###############################################################################
##########################################################
 ;;=============================================================================
 ;; process Trip Attractions next                                              =
 ;;=============================================================================

 LOOP K = 1,dbi.4.NUMRECORDS
      x = DBIReadRecord(4,k)
      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   TotHBONmtAs = TotHBOnmtAs +
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   TotNHONmtAs = TotNHOnmtAs +
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
    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)  EMPTARate_j[jdx]     =  MTR_ATot_Ja[jdx]  /
TotEMP_ja[jdx]
  ENDLOOP

      if (TotEMP_Tot  > 0)   EMPTARate_p[1]    = TotHBWMtrAs    /    TotEMP_Tot
      if (TotEMP_Tot  > 0)   EMPTARate_p[2]    = TotHBSMtrAs    /    TotEMP_Tot
      if (TotEMP_Tot  > 0)   EMPTARate_p[3]    = TotHBOMtrAs    /    TotEMP_Tot
      if (TotEMP_Tot  > 0)   EMPTARate_p[4]    = TotNHWMtrAs    /    TotEMP_Tot
      if (TotEMP_Tot  > 0)   EMPTARate_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

  Print PRINTO=1 LIST='                           '
  Print form=10csv PRINTO=1 LIST='      TOTAL               ',
```

```
                                TOTHBWNMTAs, ' ',TOTHBSNMTAs,' ',TOTHBONMTAs,'
',TOTNHWNMTAs,' ',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, ' ',TOTHBSNMTAs,' ',TOTHBONMTAs,'
',TOTNHWNMTAs,' ',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_i1, ' ',TOTHBWMTRAs_i2,'
',TOTHBWMTRAs_i3,' ',TOTHBWMTRAs_i4,' ',
                                 TOTHBSMTRAs_i1, ' ',TOTHBSMTRAs_i2,'
',TOTHBSMTRAs_i3,' ',TOTHBSMTRAs_i4,' ',
                                 TOTHBOMTRAs_i1, ' ',TOTHBOMTRAs_i2,'
',TOTHBOMTRAs_i3,' ',TOTHBOMTRAs_i4

; --------------------------------------------------------------------------------
---------------------------------------------------
;; --------------------------------------------------------------------------------
----------------------------------------------------
 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
```

45 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

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 = 'skimtot%_prev_%.dat'                ;  Zone file showing TAZs without
Truck Access (generation is suppressed)



ZnFile_Ou1  = 'ComVeh_Truck_Ends_%_iter_%.dbf'     ;  output comm, med trk, hvy
truck trip ends
ZnFile_Ou2  = 'ComVeh_Truck_dbg_%_iter_%.dbf'      ;  output debug file- zonal
inputs and outputs

Rates_in    ='..\support\Truck_Com_Trip_Rates.DBF'    ;  Truck, Comm.Veh trip rates
reportfile  ='Truck_Com_Trip_Generation_%_iter_%.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'
;;;;     end
```

```
; Define zonal Area Type File as a zonal lookup table
 FileI LOOKUPI[2] = "@ZNFILE_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
           OTHRateA[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]    = "@ZNFile_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]    = "@ZNFile_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                               ;
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)
                                                                     ; CV trip rates
```

```
            ro.COFF_Rate      = OFFRATEA[3][_ATYPE]                  ;
            ro.CRET_Rate      = RETRATEA[3][_ATYPE]                  ;
            ro.CIND_Rate      = INDRATEA[3][_ATYPE]                  ;
            ro.COTH_Rate      = OTHRATEA[3][_ATYPE]                  ;
            ro.CHH_Rate       = HH_RATEA[3][_ATYPE]

            ro.MOFF_Rate      = OFFRATEA[1][_ATYPE]   ro.HOFF_Rate =
OFFRATEA[2][_ATYPE] ; truck rates
            ro.MRET_Rate      = RETRATEA[1][_ATYPE]   ro.HRET_Rate =
RETRATEA[2][_ATYPE] ;
            ro.MIND_Rate      = INDRATEA[1][_ATYPE]   ro.HIND_Rate =
INDRATEA[2][_ATYPE] ;
            ro.MOTH_Rate      = OTHRATEA[1][_ATYPE]   ro.HOTH_Rate =
OTHRATEA[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@


            ro.TAZ            = k                 ;                 TAZ (extl
station)
            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
            ro.IHvy_Truck        = 0.0               ;               int  hvytk
trips
```

```
            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 '
   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
```

46 unbuild_net.s

```
*del tppl*.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  = 'I:\ateam'
inhwy     = 'zonehwy.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),spdc(7),capc(7),jur(7),Screen(5),ftype(7),toll(9),to
llgrp(5),

amlane(3),amlimit(3),pmlane(3),pmlimit(3),oplane(3),oplimit(3),edgeid(10),linkid(10)
,Networkyear(8),Shape_Length(7.2),
           projectid(10)

/* Write out node file */

nodeo= @basepath@\@out_node@,
  format=DBF,
  include=n(6),x(8),y(8)

endrun

*copy tppl*.prn  unbuild_net.rpt
```

47 V2.3_Highway_Build.s

```
*del voya*.prn
;=========================================================================
;  HIGHWAY_BUILD_TOLL.S
;
;  MWCOG Version 2.3 Model  - Highway Network Building Program
;  Toll-DBF lookup file used
;;=========================================================================
;
; 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\AMSPD.LKP'       ; AM       Speed lookup ATxFT   (I/P
file)
  MDSPD     = '..\support\MDSPD.LKP'       ; 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
 ;
     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
;
;
;  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 the default
;  highway deflator 'DEFLATION' (calculated in the SET_Factors.s script) is used.
;  The recommended appraoch 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
       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=@MDSPD@

        _IDX    =  FTYPE + 1
        PPAMSPD= AMSPD(_IDX,AType)
        PPOPSPD= OPSPD(_IDX,AType)

;
; ESTABLISH  AM/PM/MD/NT    Highway Times (for the transit Network)
;
PPPMSPD =  PPAMSPD                 ; assume PM spd is equal to AM
IF (PPAMSPD != 0 )
  AMHTIME    =  (DISTANCE/PPAMSPD)*60.00
  PMHTIME    =  (DISTANCE/PPPMSPD)*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
PPMDSPD     = PPOPSPD
MDHTIME     = OPHTIME

NTTOLL      = OPTOLL
NTTOLL_VP   = OPTOLL_VP
PPNTSPD     = PPOPSPD
NTHTIME     = OPHTIME

;  CREATE SOME FREQUENCY-CROSSTABS FOR CHECKING
_CNT= 1

CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=AMLANE,    RANGE=1-7-1
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=OPLANE,    RANGE=1-7-1
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=PMLANE,    RANGE=1-7-1

CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=AMLIMIT,   RANGE=0-9-1
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=OPLIMIT,   RANGE=0-9-1
CROSSTAB VAR=_CNT,ROW=FTYPE, RANGE=1-7-1, COL=PMLIMIT,   RANGE=0-9-1

;
;


ENDRUN
```

48 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

;; 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                ;;
;;                            1) HBWV2A1.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 pcts 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 reasonablility (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),
                                       _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@


;ARRAY  Type=c1  AD = @Xlinksize@

ARRAY  nx            = @nodesize@,
       ny            = @nodesize@,
       use           = @nodesize@,
       Dela1         = @Xlinksize@,
       Delb1         = @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
            Dela1[Ndel1] = AD_A
            Delb1[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 tenths 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 = Dela1[L] && B = Delb1[L]) GOTO SKIP
                IF (B = Dela1[L] && A = Delb1[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
;;


;;   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, Ndel2
                  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 hundrths 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, Ndel2
                 IF (ZZ = Dela2[M] && NN = Delb2[M])  GOTO NextNode1
                 IF (NN = Dela2[M] && ZZ = Delb2[M])  GOTO NextNode1
              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 tenths 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, Ndel2
                 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 tenths 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
     :NextNode2
     ENDLOOP

:NextTAZ
ENDLOOP
ENDRUN
*copy voya*.prn WalkAcc.rpt
```

Appendix D. AEMS Fortran Control files

1 Access.ctl

```
Station Access Summary
PK_VOL.DBF                              //---- peak transit network ----
OP_VOL.DBF                              //---- offpeak transit network ----

ACCESS
      8001, 8002, 8003, 8004, 8005, 8006, 8007, 8008, 8009, 8010, 8011, 8012, 8013,
      8014, 8015, 8016, 8017, 8018, 8019, 8020, 8021, 8022, 8023, 8024, 8025, 8026,
      8027, 8028, 8029, 8030, 8031, 8032, 8033, 8034, 8035, 8036, 8037, 8038, 8039,
      8040, 8041, 8042, 8043, 8044, 8045, 8046, 8047, 8048, 8049, 8050, 8051, 8052,
      8053, 8054, 8055, 8056, 8057, 8058, 8059, 8060, 8061, 8062, 8063, 8064, 8065,
      8066, 8067, 8068, 8069, 8070, 8071, 8072, 8073, 8074, 8075, 8076, 8077, 8078,
      8079, 8080, 8081, 8082, 8083, 8085, 8086, 8098

MODES=11-16, DETAIL=YES, NAME="Station Access", FILE=BOARDING.ASC
```

2 AccessPkHbw.ctl

```
Station Access Summary
PK_VOL.DBF                              //---- peak transit network ----

ACCESS
      8001, 8002, 8003, 8004, 8005, 8006, 8007, 8008, 8009, 8010, 8011, 8012, 8013,
      8014, 8015, 8016, 8017, 8018, 8019, 8020, 8021, 8022, 8023, 8024, 8025, 8026,
      8027, 8028, 8029, 8030, 8031, 8032, 8033, 8034, 8035, 8036, 8037, 8038, 8039,
      8040, 8041, 8042, 8043, 8044, 8045, 8046, 8047, 8048, 8049, 8050, 8051, 8052,
      8053, 8054, 8055, 8056, 8057, 8058, 8059, 8060, 8061, 8062, 8063, 8064, 8065,
      8066, 8067, 8068, 8069, 8070, 8071, 8072, 8073, 8074, 8075, 8076, 8077, 8078,
      8079, 8080, 8081, 8082, 8083, 8085, 8086, 8098

MODES=11-16, DETAIL=YES, NAME="Station Access", FILE=BOARDINGPKHBW.ASC
```

3 Hbw_nl_mc.ctl

```
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
SKIM01:IVTT       1>DAIV      S2IV      S3IV      WCIV      WBIV      WTIV      WMIV      PCIV      KCIV      PBIV      KBIV      PTIV      KTIV      PMIV      KMIV
COEF02:AUTO ACC   1>                                                                      -0.03192  -0.03192  -0.03192  -0.03192  -0.03192  -0.03192  -0.03192  -0.03192
SKIM02:AUTO ACC   1>                                                                      PCAA      KCAA      PBAA      KBAA      PTAA      KTAA      PMAA      KMAA
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
```

```
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
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
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
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
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
SKIM08:TRN XFERS  1>                            WCXF      WBXF      WTXF      WMXF      PCXF      KCXF      PBXF      KBXF      PTXF      KTXF      PMXF      KMXF
COEF09:TRN BRDPEN 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
SKIM09:TRN BRDPEN 1>                            WCXP      WBXP      WTXP      WMXP      PCXP      KCXP      PBXP      KBXP      PTXP      KTXP      PMXP      KMXP
*WALK WEIGHT
COEF10:TRN WLKWT  1>                            -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256  -0.04256
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
* 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+
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.
* 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-METRORAIL 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
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 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())
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)
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
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>                              2.000000  2.000000  2.000000  2.000000
PURP02 1INC 2     1>
PURP03 1INC 3     1>
PURP04 1INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 2INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 2INC 2     1>
PURP03 2INC 3     1>
PURP04 2INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 3INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 3INC 2     1>
PURP03 3INC 3     1>
PURP04 3INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 4INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 4INC 2     1>
PURP03 4INC 3     1>
PURP04 4INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 5INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 5INC 2     1>
PURP03 5INC 3     1>
PURP04 5INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 6INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 6INC 2     1>
PURP03 6INC 3     1>
PURP04 6INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 7INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 7INC 2     1>
PURP03 7INC 3     1>
PURP04 7INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 8INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 8INC 2     1>
PURP03 8INC 3     1>
PURP04 8INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 9INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 9INC 2     1>
PURP03 9INC 3     1>
PURP04 9INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0110INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0210INC 2     1>
PURP0310INC 3     1>
PURP0410INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0111INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0211INC 2     1>
PURP0311INC 3     1>
PURP0411INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0112INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0212INC 2     1>
PURP0312INC 3     1>
```

```
PURP0412INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0113INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0213INC 2     1>
PURP0313INC 3     1>
PURP0413INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0114INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0214INC 2     1>
PURP0314INC 3     1>
PURP0414INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0115INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0215INC 2     1>
PURP0315INC 3     1>
PURP0415INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0116INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0216INC 2     1>
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>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0118INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0218INC 2     1>
PURP0318INC 3     1>
PURP0418INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0119INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0219INC 2     1>
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      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 MODE1>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 MODE1>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 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG CW 4 PCT 1>WSW3
```

```
WALK SEG CW 4 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y         Y         Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODE1>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
NEST 2,1=         1>                              Y         Y         Y         Y
NEST 2,2=         1>                                                                      Y                   Y                   Y                   Y
NEST 2,3=         1>                                                                                Y                   Y                   Y                   Y
NEST 3,1=         1>                              Y
NEST 3,2=         1>                                        Y
NEST 3,3=         1>                                                  Y
NEST 3,4=         1>                                                            Y
NEST 4,1          1>                                                                      Y
NEST 4,2          1>                                                                                          Y
NEST 4,3          1>                                                                                                              Y
NEST 4,4          1>                                                                                                                                  Y
NEST 5,1          1>                                                                                Y
NEST 5,2          1>                                                                                                    Y
NEST 5,3          1>                                                                                                                        Y
NEST 5,4          1>                                                                                                                                            Y
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
*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    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
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
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
* 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
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
NSTC 5212KNR BUS   >      1.0   -0.17748
NSTC 5312KNR BU/MR >      1.0   -4.78017
NSTC 5412KNR 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 2313KNR 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 5013KNR TRN
NSTC 5113KNR CR    >      1.0   -4.43232
NSTC 5213KNR BUS   >      1.0   -6.67781
NSTC 5313KNR BU/MR >      1.0   -1.36864
NSTC 5413KNR 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 2314KNR 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 5014KNR TRN
NSTC 5114KNR CR    >      1.0   -9.21450
NSTC 5214KNR BUS   >      1.0   -1.14640
NSTC 5314KNR BU/MR >      1.0   -1.11396
NSTC 5414KNR 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
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
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
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

*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


*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
```

4 hbs_nl_mc.ctl

```
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>                                                                      -0.03252  -0.03252  -0.03252  -0.03252  -0.03252  -0.03252  -0.03252  -0.03252
SKIM02:AUTO ACC   1>                                                                      PCAA      KCAA      PBAA      KBAA      PTAA      KTAA      PMAA      KMAA
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
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>                              -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      PBXF      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
SKIM09:TRN BRDPEN 1>                              WCXP      WBXP      WTXP      WMXP      PCXP      KCXP      PBXP      KBXP      PTXP      KTXP      PMXP      KMXP
*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
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
* 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 3.25 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-METRORAIL 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
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 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())
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)
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
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>                             2.000000  2.000000  2.000000  2.000000
PURP02 1INC 2    1>
PURP03 1INC 3    1>
PURP04 1INC 4    1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 2INC 1    1>                             2.000000  2.000000  2.000000  2.000000
PURP02 2INC 2    1>
PURP03 2INC 3    1>
PURP04 2INC 4    1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 3INC 1    1>                             2.000000  2.000000  2.000000  2.000000
```

```
PURP02 3INC 2     1>
PURP03 3INC 3     1>
PURP04 3INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 4INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 4INC 2     1>
PURP03 4INC 3     1>
PURP04 4INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 5INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 5INC 2     1>
PURP03 5INC 3     1>
PURP04 5INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 6INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 6INC 2     1>
PURP03 6INC 3     1>
PURP04 6INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 7INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 7INC 2     1>
PURP03 7INC 3     1>
PURP04 7INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 8INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 8INC 2     1>
PURP03 8INC 3     1>
PURP04 8INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP01 9INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP02 9INC 2     1>
PURP03 9INC 3     1>
PURP04 9INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0110INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0210INC 2     1>
PURP0310INC 3     1>
PURP0410INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0111INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0211INC 2     1>
PURP0311INC 3     1>
PURP0411INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0112INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0212INC 2     1>
PURP0312INC 3     1>
PURP0412INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0113INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0213INC 2     1>
PURP0313INC 3     1>
PURP0413INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0114INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0214INC 2     1>
PURP0314INC 3     1>
PURP0414INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0115INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0215INC 2     1>
PURP0315INC 3     1>
PURP0415INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0116INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0216INC 2     1>
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>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0118INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0218INC 2     1>
PURP0318INC 3     1>
PURP0418INC 4     1>                             -2.00000  -2.00000  -2.00000  -2.00000
PURP0119INC 1     1>                             2.000000  2.000000  2.000000  2.000000
PURP0219INC 2     1>
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      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 MODE1>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 MODE1>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 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y         Y         Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODE1>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
NEST 2,1=         1>                              Y         Y         Y         Y
NEST 2,2=         1>                                                                      Y                   Y                   Y                   Y
NEST 2,3=         1>                                                                                Y                   Y                   Y                   Y
NEST 3,1=         1>                              Y
NEST 3,2=         1>                                        Y
NEST 3,3=         1>                                                  Y
NEST 3,4=         1>                                                            Y
NEST 4,1          1>                                                                      Y
```

```
NEST 4,2          1>                                                                                          Y
NEST 4,3          1>                                                                                                              Y
NEST 4,4          1>                                                                                                                                  Y
NEST 5,1          1>                                                                                Y
NEST 5,2          1>                                                                                                    Y
NEST 5,3          1>                                                                                                                        Y
NEST 5,4          1>                                                                                                                                            Y
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
*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
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
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
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
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
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
* 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
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
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

*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

*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
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
```

5 hbo_nl_mc.ctl

```
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>                                                                      -0.03483  -0.03483  -0.03483  -0.03483  -0.03483  -0.03483  -0.03483  -0.03483
SKIM02:AUTO ACC   1>                                                                      PCAA      KCAA      PBAA      KBAA      PTAA      KTAA      PMAA      KMAA
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>                              -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      PBXF      KBXF      PTXF      KTXF      PMXF      KMXF
COEF09:TRN BRDPEN 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
SKIM09:TRN BRDPEN 1>                              WCXP      WBXP      WTXP      WMXP      PCXP      KCXP      PBXP      KBXP      PTXP      KTXP      PMXP      KMXP
*WALK WEIGHT
COEF10:TRN WLKWT  1>                              -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644  -0.04644
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
* 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
* PERCENT WALKS-METRORAIL 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
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 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())
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)
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
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>                              2.000000  2.000000  2.000000  2.000000
PURP02 1INC 2    1>
PURP03 1INC 3    1>
PURP04 1INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 2INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 2INC 2    1>
PURP03 2INC 3    1>
PURP04 2INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 3INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 3INC 2    1>
PURP03 3INC 3    1>
PURP04 3INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 4INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 4INC 2    1>
PURP03 4INC 3    1>
PURP04 4INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 5INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 5INC 2    1>
PURP03 5INC 3    1>
PURP04 5INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 6INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 6INC 2    1>
PURP03 6INC 3    1>
PURP04 6INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 7INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 7INC 2    1>
PURP03 7INC 3    1>
PURP04 7INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP01 8INC 1    1>                              2.000000  2.000000  2.000000  2.000000
PURP02 8INC 2    1>
PURP03 8INC 3    1>
PURP04 8INC 4    1>                              -2.00000  -2.00000  -2.00000  -2.00000
```

```
PURP01 9INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP02 9INC 2     1>
PURP03 9INC 3     1>
PURP04 9INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0110INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0210INC 2     1>
PURP0310INC 3     1>
PURP0410INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0111INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0211INC 2     1>
PURP0311INC 3     1>
PURP0411INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0112INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0212INC 2     1>
PURP0312INC 3     1>
PURP0412INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0113INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0213INC 2     1>
PURP0313INC 3     1>
PURP0413INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0114INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0214INC 2     1>
PURP0314INC 3     1>
PURP0414INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0115INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0215INC 2     1>
PURP0315INC 3     1>
PURP0415INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0116INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0216INC 2     1>
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>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0118INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0218INC 2     1>
PURP0318INC 3     1>
PURP0418INC 4     1>                              -2.00000  -2.00000  -2.00000  -2.00000
PURP0119INC 1     1>                              2.000000  2.000000  2.000000  2.000000
PURP0219INC 2     1>
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      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 MODE1>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 MODE1>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 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y         Y         Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODE1>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
NEST 2,1=         1>                              Y         Y         Y         Y
NEST 2,2=         1>                                                                      Y                   Y                   Y                   Y
NEST 2,3=         1>                                                                                Y                   Y                   Y                   Y
NEST 3,1=         1>                              Y
NEST 3,2=         1>                                        Y
NEST 3,3=         1>                                                  Y
NEST 3,4=         1>                                                            Y
NEST 4,1          1>                                                                      Y
NEST 4,2          1>                                                                                          Y
NEST 4,3          1>                                                                                                              Y
NEST 4,4          1>                                                                                                                                  Y
NEST 5,1          1>                                                                                Y
NEST 5,2          1>                                                                                                    Y
NEST 5,3          1>                                                                                                                        Y
NEST 5,4          1>                                                                                                                                            Y
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
*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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
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
```

```
* 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
```

```
*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
```

```
*SELJ              >      362
*SELJ              >     1231
*SELJ              >     1236
*SELJ              >     1265
*SELJ              >     1337
*SELJ              >     1537
*SELI              >  523
*SELJ              >    9

TRACE             >        0
* OUTPUT %          >
*PROCSEL           >
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?   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
```

6 nhw_nl_mc.ctl

```
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>                                                                      -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
SKIM08:TRN XFERS  1>                              WCXF      WBXF      WTXF      WMXF      PCXF      KCXF      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
SKIM09:TRN BRDPEN 1>                              WCXP      WBXP      WTXP      WMXP      PCXP      KCXP      PBXP      KBXP      PTXP      KTXP      PMXP      KMXP
*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      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
```

```
* 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
* PERCENT WALKS-METRORAIL 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
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 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())
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)
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
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>
PURP03 1INC 3     1>
PURP04 1INC 4     1>

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
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 MODE1>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 MODE1>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 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y         Y         Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODE1>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
NEST 2,1=         1>                              Y         Y         Y         Y
```

```
NEST 2,2=         1>                                                                      Y                   Y                   Y                   Y
NEST 2,3=         1>                                                                                Y                   Y                   Y                   Y
NEST 3,1=         1>                              Y
NEST 3,2=         1>                                        Y
NEST 3,3=         1>                                                  Y
NEST 3,4=         1>                                                            Y
NEST 4,1          1>                                                                      Y
NEST 4,2          1>                                                                                          Y
NEST 4,3          1>                                                                                                              Y
NEST 4,4          1>                                                                                                                                  Y
NEST 5,1          1>                                                                                Y
NEST 5,2          1>                                                                                                    Y
NEST 5,3          1>                                                                                                                        Y
NEST 5,4          1>                                                                                                                                            Y
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
*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   -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
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
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
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
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
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
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
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
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
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

*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
*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
```

7 nho_nl_mc.ctl

```
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>                                                                      -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
SKIM08:TRN XFERS  1>                              WCXF      WBXF      WTXF      WMXF      PCXF      KCXF      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
SKIM09:TRN BRDPEN 1>                              WCXP      WBXP      WTXP      WMXP      PCXP      KCXP      PBXP      KBXP      PTXP      KTXP      PMXP      KMXP
*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      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
* 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
* PERCENT WALKS-METRORAIL 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
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 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())
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)
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
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>
PURP03 1INC 3     1>
PURP04 1INC 4     1>

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
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 MODE1>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 MODE1>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 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG CW 4 PCT 1>WSW3
WALK SEG CW 4 MODE1>Y         Y         Y         Y         Y         Y                   Y         Y         Y         Y         Y         Y
WALK SEG MD 5 PCT 1>WSM1
WALK SEG MD 5 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y         Y         Y
WALK SEG MD 6 PCT 1>WSM2
WALK SEG MD 6 MODE1>Y         Y         Y                                                 Y         Y         Y         Y         Y         Y
WALK SEG NT 7 PCT 1>WSNT
WALK SEG NT 7 MODE1>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
NEST 2,1=         1>                              Y         Y         Y         Y
NEST 2,2=         1>                                                                      Y                   Y                   Y                   Y
NEST 2,3=         1>                                                                                Y                   Y                   Y                   Y
NEST 3,1=         1>                              Y
NEST 3,2=         1>                                        Y
NEST 3,3=         1>                                                  Y
NEST 3,4=         1>                                                            Y
NEST 4,1          1>                                                                      Y
NEST 4,2          1>                                                                                          Y
NEST 4,3          1>                                                                                                              Y
NEST 4,4          1>                                                                                                                                  Y
NEST 5,1          1>                                                                                Y
NEST 5,2          1>                                                                                                    Y
NEST 5,3          1>                                                                                                                        Y
NEST 5,4          1>                                                                                                                                            Y
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
*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   -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
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
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
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
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
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
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
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
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
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

*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
*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
```