Draft FY 2013 Final Report

COG Contract 12-006:

Assistance with Development and Application of the National Capital Region Transportation Planning Board Travel Demand Model

Submitted to:

National Capital Region Transportation Planning Board
Metropolitan Washington Council of Governments

Submitted by:

AECOM with Stump/Hausman Partnership

July 1, 2013

This page is intentionally blank

Table of Contents

List of Figures

List of Tables

1 Introduction

The National Capital Region Transportation Planning Board (NCRTPB or simply TPB) is the federally designated Metropolitan Planning Organization (MPO) for the Washington, D.C. metropolitan area and is also one of several policy boards that operate at the Metropolitan Washington Council of Governments (MWCOG or simply COG). The TPB is staffed by COG's Department of Transportation Planning (DTP). Since FY 2006, COG/TPB staff has maintained a consultant-assisted project to evaluate the travel forecasting practices used by the TPB. The objectives of the project are to ensure that the TPB's modeling methods are in line with the practices of other MPOs and to provide guidance and advice in the area of travel demand modeling. The project contract was designed to operate on a fiscal-year basis and to be renewable for up to two additional fiscal years. This arrangement would allow the selected consultant to hold the contract for up to three years in total, at which time re-bidding of the contract would occur. The contract was re-bid at the end of FY 2011 and AECOM was chosen by a selection committee to be the consultant for FY 2012 (July 1, 2011 to June 30, 2012). About a year later, the contract was renewed for a second year (FY 2013, running from July 1, 2012 to June 30, 2013). Finally, at the end of FY 2012, the contract has been renewed for its third and final year (FY 2014). In FY 2012, there were six task orders. During the course of FY 2013, three task orders were developed collaboratively between TPB staff and AECOM:

- Task Order 7: Attend relevant meetings, such as the Travel Forecasting Subcommittee, and respond to ad hoc requests (this is a continuation of Task Order #1 from FY 2012);
- Task Order 8: Traffic assignment improvements:
    - Evaluate potential changes to the procedures used to model high-occupancy/toll (HOT) lanes[1] and high-occupancy vehicle (HOV) lanes, with an emphasis on simplifying the code and reducing overall model run time;
    - Investigate the use of INRIX observed link speed data; and
- Task Order 9: Mode choice and transit modeling: Enhance transit modeling capabilities
    - Begin the process of migrating to a new transit path building process (from Citilabs' TRNBUILD to Citilabs' PT)
    - Begin the process of moving to a new mode choice application program (from AEMS to ModeChoice).

This report documents the work done by AECOM and its subcontractor, Stump/Hausman Partnership, to fulfill these task orders. Chapters 2 through 10 address Task Orders 7 through 9. Chapter 11 presents a summary of consultant recommendations.

[1] Also known as express toll lanes or, in Northern Virginia, the I-495 Express Lanes or the I-95 Express Lanes.

This page is intentionally blank.

2 Meetings and Technical Assistance (Task Order 7)

AECOM participated in each of the six Travel Forecasting Subcommittee meetings of FY 2012 and prepared presentations for the July 20, 2012, November 30, 2012, March 22, 2013, and May 17, 2013 committee meetings. AECOM also met several times with the MWCOG staff to discuss work tasks, findings and recommendations.

2.1 Interpreting AEMS Market Shares

The current regional travel demand model used by MWCOG (the TPB Version 2.3 Travel Model) uses a FORTRAN mode choice model application program, known as AEMS. MWCOG staff is considering moving to a new mode choice model application program, called ModeChoice, which is discussed later in this report. In the meantime, however, MWCOG staff wanted a better understanding of the access market definitions that are used in AEMS. At the Travel Forecasting Subcommittee meeting on September 21, 2012, MWCOG requested that AECOM provide a description of the variables used in the AEMS mode choice control file that define access market segments for each zone-to-zone interchange. This resulted in a memo, dated September 24, 2012. The material below is based on that memo.

Starting at line 83 of the AEMS control files are a series of comment statements that identify the data fields expected in the input zone data file (file 8). The fields of interest for this discussion are listed below:

7. Percentage of the zone within a short walk (0.5 miles) of Metrorail or Light Rail (modes 3 and 5)
8. Percentage of the zone within a long walk (1.0 miles) of Metrorail or Light Rail (modes 3 and 5)
9. Percentage of the zone within a short walk (0.5 miles) of any transit during the AM peak period
10. Percentage of the zone within a long walk (1.0 miles) of any transit during the AM peak period
11. Percentage of the zone within a short walk (0.5 miles) of any transit during the off peak period
12. Percentage of the zone within a long walk (1.0 miles) of any transit during the off peak period

The model then defines four variables to store these values for each trip purpose. Work trips use the AM peak period percentages and non-work models using the off peak percentages. Metrorail and light rail are assumed to be available for all trip purposes and time periods. The variables are defined below:

- PCMI – percent walk access to Metrorail/light rail (LRT) at the production zone
  = (i807 + 0.25 * (i808 – i807)) / 100
- PCMJ – percent walk access to Metrorail/LRT at the attraction zone
  = (j807 + 0.25 * (j808 – j807)) / 100
- PCTI – percent walk access to all transit at the production zone
  = (i809 + 0.25 * (i810 – i809)) / 100 for work
  = (i811 + 0.25 * (i812 – i811)) / 100 for non-work
- PCTJ – percent walk access to all transit at the attraction zone
  = (j809 + 0.25 * (j810 – j809)) / 100 for work
  = (j811 + 0.25 * (j812 – j811)) / 100 for non-work

The 0.25 (or 25%) in the equations above is the percentage of productions in the long-walk area that are assumed to walk (i.e., 75% of productions in long-walk area are assumed to drive). These percentages are then used to calculate transit access and Metrorail access market segments at the zone-to-zone interchange level. The model defines four access markets based on the share of the origin zone that can walk to transit. It defines two access markets for trips that must drive to transit and one market for trips that cannot be made by transit. The four markets that can walk are defined using four percentages:

- WSWM – walk segment that can walk to Metrorail/LRT at the production and attraction end
  = PCMI * PCMJ
- WSW1 – walk segment that can walk to transit other than Metrorail/LRT at the attraction end, but not at the production end
  = (PCTI – PCMI) * PCMJ
- WSW2 – walk segment that cannot walk to Metrorail/LRT at the production or the attraction end
  = (PCTI – PCMI) * (PCTJ – PCMJ)
- WSW3 – walk segment that can walk to Metrorail/LRT at the production, but not the attraction
  = PCMI * (PCTJ – PCMJ)

The two access markets that must drive to transit are defined as follows:

- WSM1 – walk segment that must drive at the production end, but can walk to Metrorail/LRT at the attraction end
  = (1 – PCTI) * PCMJ
- WSM2 – walk segment that must drive at the production end, but cannot walk to Metrorail/LRt at the attraction end
  = (1 – PCTI) * (PCTJ – PCMJ)

The access market that cannot use transit is defined as:

- WSNT – walk segment with no access to transit
  = (1 – WSWM – WSW1 – WSW2 – WSW3 – WSM1 – WSM2)[2]

For a given zone-to-zone interchange, the sum of these seven percentages must be 100%. The seven access market segments are then applied to specific mode combinations within the nested logit mode choice model in the following ways:

- WSWM – SOV, 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, and KNR-MR
- WSW1 – SOV, SR2, SR3+, WK-CR, WK-BUS, WK-BU/MR, PNR-CR, KNR-CR, PNR-BUS, KNR-BUS, PNR-BU/MR, KNR-BU/MR, PNR-MR, and KNR-MR

[2] In other words, the percent of travelers that has no access to transit is equal to 1 minus the percent-walk-to-transit found at the attraction end: P(NT) = (1 – PWT(j)).

- WSW2 – SOV, SR2, SR3+, WK-CR, WK-BUS, WK-BU/MR, PNR-CR, KNR-CR, PNR-BUS, KNR-BUS, PNR-BU/MR, and KNR-BU/MR
- WSW3 – SOV, SR2, SR3+, WK-CR, WK-BUS, WK-BU/MR, PNR-CR, KNR-CR, PNR-BUS, KNR-BUS, PNR-BU/MR, and KNR-BU/MR
- WSM1 – SOV, SR2, SR3+, PNR-CR, KNR-CR, PNR-BUS, KNR-BUS, PNR-BU/MR, KNR-BU/MR, PNR-MR, and KNR-MR
- WSM2 – SOV, SR2, SR3+, PNR-CR, KNR-CR, PNR-BUS, KNR-BUS, PNR-BU/MR, and KNR-BU/MR
- WSNT – SOV, SR2, and SR3+

where

- SOV = Drive Alone / single occupancy vehicle
- SR2 = Shared Ride with 2 persons
- SR3+ = Shared Ride with 3 or more persons
- WK-CR = walk to commuter rail
- WK-BUS = walk to bus
- WK-BU/MR = walk to/from bus and Metrorail/Light Rail
- WK-MR = walk to/from Metrorail/Light Rail only
- PNR-CR = park-n-ride to commuter rail
- KNR-CR = kiss-n-ride to commuter rail
- PNR-BUS = park-n-ride to bus
- KNR-BUS = kiss-n-ride to bus
- PNR-BU/MR = park-n-ride to bus and Metrorail/Light Rail
- KNR-BU/MR = kiss-n-ride to bus and Metrorail/Light Rail
- PNR-MR = park-n-ride to Metrorail/Light Rail
- KNR-MR = kiss-n-ride to Metrorail/Light Rail

This page is intentionally blank.

3 HOV Modeling (Task Order 8)

The goal of Task Order 8 was to make modifications to the traffic assignment process used in the MWCOG Version 2.3 Travel Model that would result in model run-time savings and simpler scripts. The primary focus of this effort was on potential changes to the HOT-lane and HOV modeling procedures that would simplify the model code and reduce overall model run time. A secondary focus was speed validation using the INRIX observed travel time data, given the fact that MWCOG can access this data via its membership on the I-95 Corridor Coalition. This chapter focuses on HOV modeling. HOT-lane modeling and INRIX data are the focus of later chapters.

The Version 2.3 Travel Model assigns HOT-lane traffic using a multi-run traffic assignment. The main goal of this technique is to ensure that travel speeds for the HOV 3+ traffic on the HOT lanes are not degraded by the other traffic using the HOT lanes. Since this technique, known as the "multi-run" process or the "HOV3+ skim substitution technique," requires two full model runs for each scenario, it doubles model run times (this issue is discussed in Chapter 4). In addition, the model uses a multi-user-class assignment with six user classes and four time periods, with the two peak-periods split into two assignments (HOV3+ and non-HOV3+; this is known as the "two-step" assignment process). This technique was found to improve the assignment of HOV traffic on I-395/Shirley Highway and the Capital Beltway in Virginia, but it also results in longer run times.

The key to properly modeling HOV facilities is to model travelers who choose to form a carpool or use transit in order to take advantage of the travel time or cost savings offered by the HOV lane differently from travelers that are already traveling in multiple/high-occupancy vehicles. What this means from a behavioral modeling perspective is that an HOV lane should be considered differently in a mode choice model than it should in an assignment model. In the assignment model, any vehicle that qualifies under the occupancy restriction can choose to use the HOV lane if it saves them travel time or cost. The mode choice model, however, needs to limit the HOV choice to those trips that have an HOV alternative and model general auto occupancy as a household travel decision. A mode choice model calibrated with indistinguishable modes is not likely to perform effectively as a predictive tool, especially given HOT lane incentives.

It makes sense that auto occupancy will vary by trip purpose and time of day. Unfortunately, it is hard to find observed data that shows these variations, since traffic counts typically cannot indicate trip purpose. Consequently, we will use modeled data to highlight a few points. Table 3-1 shows typical mode shares for single- and high-occupancy vehicles (SOV and HOV) in the Washington D.C. region.[3] This table shows how auto occupancy varies by trip purpose and time of day. Despite the variation in mode share, very few of these HOV trips have different travel characteristics from the corresponding SOV trips. The vast majority of HBO and NHB trips are joint household trips that are formed without regard to the relative performance of competing modes. This includes parents taking their kids to

[3] The estimated data is from a model run, representing year-2040 conditions, conducted for the WMATA Regional Transit System Plan (RTSP).

school, family members going out to eat or shop, and business associates traveling to a meeting or to lunch. For all practical purposes, only peak period HBW trips in relatively narrow geographic markets have travel time and cost differences between SOV and HOV alternatives and sufficient trip density to facilitate carpool formation.

Table 3-1: Typical Mode Shares for SOV and HOV trips

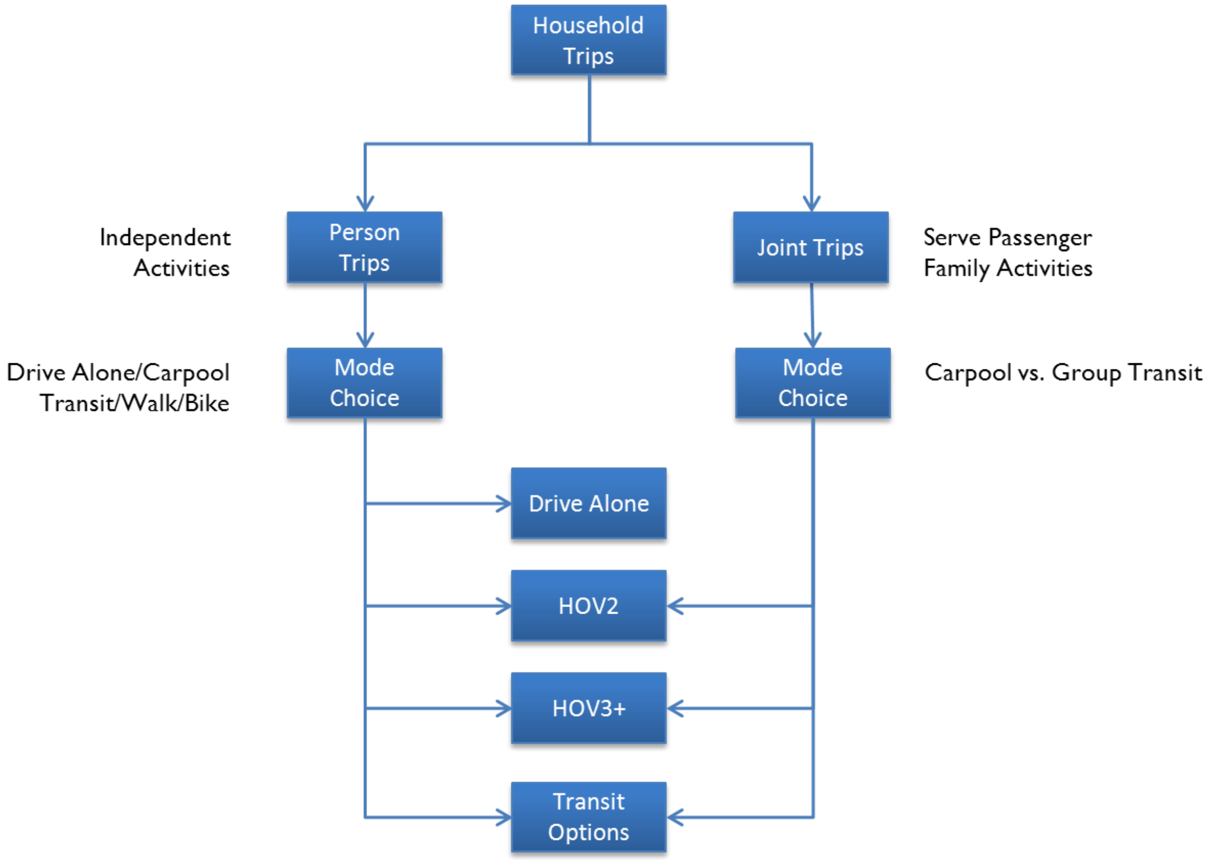
Mode	HBW	HBO	NHB	Total
	Peak Periods			
SOV	50.9%	27.6%	48.6%	39.3%
HOV	22.8%	69.1%	46.2%	49.9%
	Off-peak Periods			
SOV	76.1%	47.5%	51.9%	52.7%
HOV	11.2%	50.3%	45.4%	43.6%
	Daily Total			
SOV	60.7%	40.3%	50.8%	47.3%
HOV	18.3%	57.1%	45.7%	46.1%

Source: WMATA RTSP Model for 2040 Base Conditions

3.1 HOV Choice Models

The general structure of a comprehensive HOV choice model is shown in Figure 3-1. In this case, the household trips are assigned to independent person trips or joint household trips based on the trip purpose and joint travel rates recorded in the household survey. The joint trip probability model should be estimated using travel markets that do not provide a significant travel time or cost advantage for HOV trips. In other words, the objective is to estimate group travel based on household and general trip characteristics, rather than mode-specific information.

Figure 3-1: HOV Choice Structure: Comprehensive HOV Choice Model

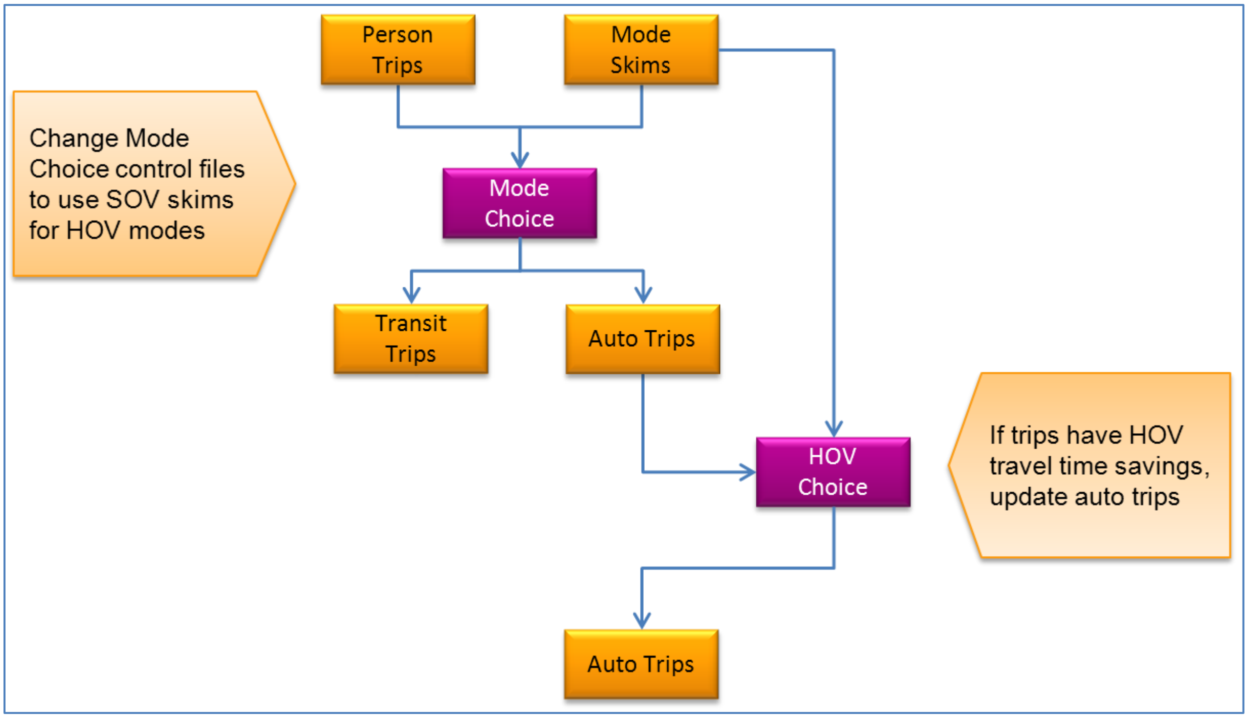


For joint or group trips, the mode choice model can be relatively simple. It does not include drive-alone trips, so the basic choice is between a standard carpool trip and traveling as a group on transit. In other words, the HOV travel time and cost are compared to the transit travel time and group fares. Since the market for family travel by transit is quite limited (zero-car households and activity locations where parking is extremely limited or expensive (e.g., sports events), etc.), the vast majority of joint household trips will be made by automobile.

The independent person trip mode choice model includes the full array of travel modes, but will generally limit the HOV2 and HOV3+ options to trip interchanges with a travel time or cost advantage. (This is not a hard and fast rule, but it does simplify the model calibration to some extent). If calibration data are limited, the HOV sub-model might only be implemented for peak period HBW trips.

Given the resources of this task order, it was not possible to estimate the comprehensive HOV choice model shown in Figure 3-1. Consequently, a simple HOV choice model was developed and tested, as shown in Figure 3-2.

Figure 3-2 HOV Choice Structure: Simplified HOV Choice Model



Source: Roden, D. B. (2013, March 22). *HOV/HOT Lane Modeling and Public Transport Research*. Presented at the March 22, 2013 meeting of the Travel Forecasting Subcommittee, held at the Metropolitan Washington Council of Governments, Washington, D.C.

This simplified HOV choice model was calibrated using the following procedure:

1. Use the LOV skims in the current mode choice models for HOV modes (i.e., assume no travel time advantage for HOV trips)
2. Assign all trips to the network using appropriate HOV restrictions.
3. Compare the Shirley Highway HOV volumes to observed counts (they should be low).
4. Compare the AM peak period LOV skims to the HOV skims in the Shirley Highway corridor and identify zone-to-zone interchanges that have an HOV travel time advantage.
5. Estimate a binary choice model that splits drive-alone HBW trips into drive-alone and HOV3+ trips based on the travel time difference. Calibrated the model to match the Shirley Highway volumes to the observed counts.

3.2 HOV Assignment Models

The output of the HOV choice and auto occupancy models are SOV, HOV2 and HOV3+ trips that can be aggregated by time of day for traffic assignment. Given the improved method of estimating HOV demand, it is likely the HOV3+ volumes assigned to the Shirley Highway HOV lanes during a single multi-class assignment will match the observed counts reasonably well. This should eliminate the need for a two-step assignment process and improve the overall consistency of the traffic assignment.

The probability of success could be further enhanced by modeling HOV links using perceived travel times rather than actual travel times in the generalized cost function. This perceived travel time is likely to reflect the reliability of the travel time on the HOV lanes as opposed to the general purpose lanes. The day-to-day impacts of non-recurring congestion in the general purpose lanes are frequently considered by travelers as they select roadways and schedule trips. If the lane is a managed HOT lane, a reliable speed should be maintained 95 percent of the time.

The use of perceived travel time and user-based tolls in a multi-class assignment imply that each vehicle class has a different generalized cost function for each link in the network. There are also link-use restrictions that determine the types of links the model can use to construct the path. These include HOV2, HOV3+, bus-only, and truck restrictions. They are also likely to include toll and no-toll restrictions if a toll choice model is included.

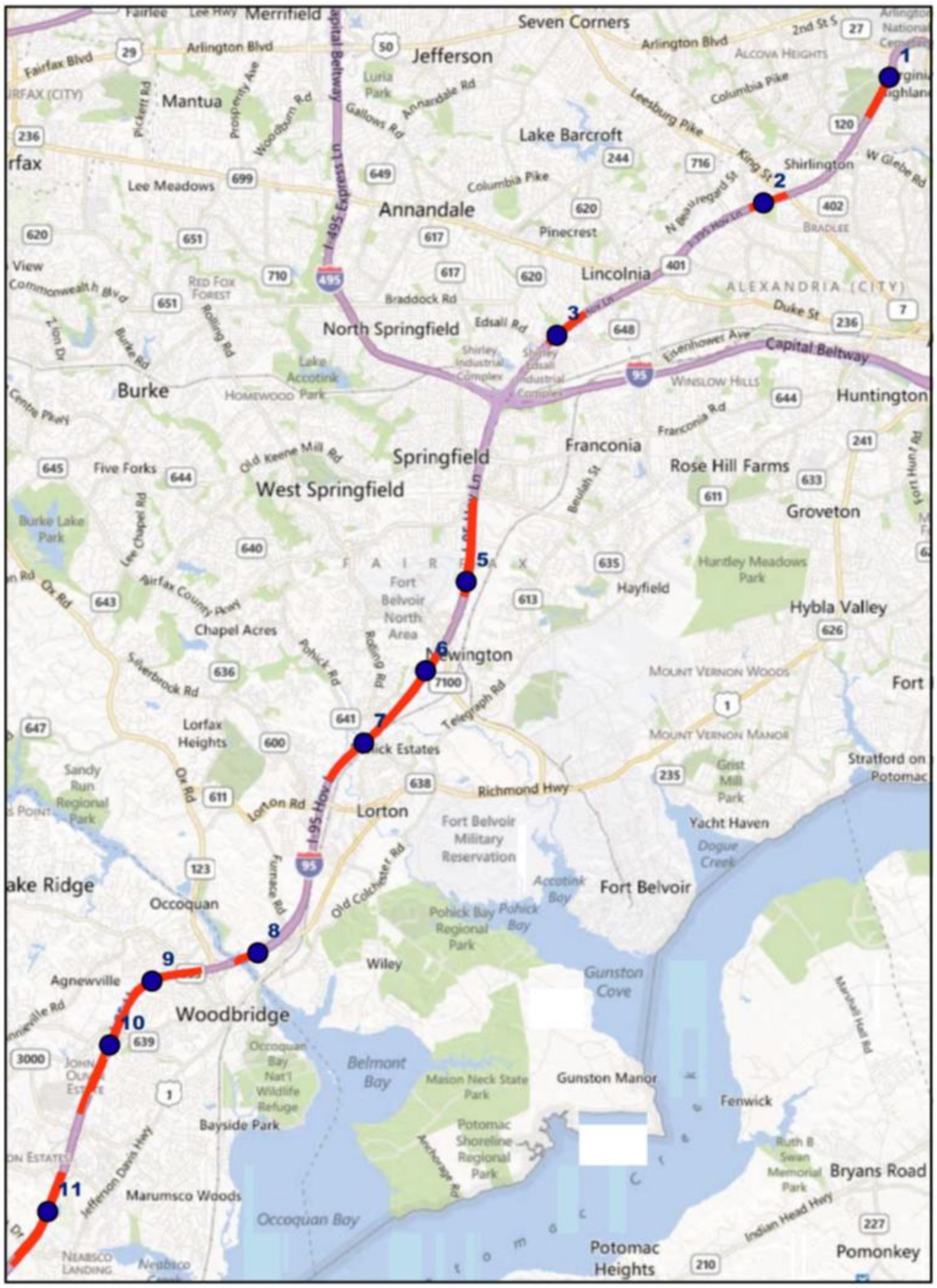
3.3 HOV Model Calibration

The calibration effort for the simplified HOV choice model involved the five steps outlined above. The HOV demand resulting from steps 1 and 2 is considered the background HOV demand. These trips chose HOV modes naturally and not due to travel time savings associated with that mode. Under step 2, the highway assignment was performed using a single multi-class assignment in which HOV3+ trips are assigned along with the rest of the modes.

The resulting volumes in the LOV and HOV lanes were compared with the available count data. Daily HOV counts were available for the ten locations along I-95/I-395 shown on Figure 3-3. HOV counts during the AM peak period were available at one location on I-395 at Glebe Road. A comparison of the observed and model-estimated daily volumes is shown in Table 3-2. Count data is not available for location # 4.

AECOM

Figure 3-3: I-95/I-395 Traffic Count Locations



Source: VDOT traffic counts compiled by MWCOG staff

Table 3-2: 2010 Background LOV and HOV3+ Daily Assignments along the I-95/I-395 Corridor

Loc	LOV_OBS	LOV_EST	EST/OBS	HOV_OBS	HOV_EST	EST/OBS	OBS	EST	EST/OBS
1	87,000	80,210	92%	21,500	21,490	100%	108,500	101,700	94%
2	82,000	83,060	101%	19,500	19,710	101%	101,500	102,770	101%
3	76,500	72,800	95%	19,000	18,870	99%	95,500	91,670	96%
5	89,500	102,030	114%	16,000	17,190	107%	105,500	119,220	113%
6	82,000	81,850	100%	25,000	19,130	77%	107,000	100,980	94%
7	80,000	83,480	104%	22,000	17,500	80%	102,000	100,980	99%
8	83,000	82,800	100%	21,000	16,620	79%	104,000	99,420	96%
9	82,000	81,980	100%	14,500	15,120	104%	96,500	97,100	101%
10	77,000	69,380	90%	12,000	13,720	114%	89,000	83,100	93%
11	68,500	75,900	111%	12,000	12,420	104%	80,500	88,320	110%
All	**80,750**	**81,350**	**101%**	**18,250**	**17,180**	**94%**	**99,000**	**98,530**	**100%**

Loc – Count Location, OBS – OBSERVED, EST - ESTIMATED

Observed counts are daily traffic counts from VDOT, assembled by MWCOG staff. Daily counts were taken on the general purpose (GP) lanes and HOV lanes. The GP lane counts include both LOV (1-occupant and 2-occupant) vehicles and HOV (3+ occupant) vehicles. The HOV lane counts include both LOV and HOV, since the counts are daily and the HOV lanes run north in the AM peak period, south in the PM peak period, and are open to all vehicles outside of the peak periods. Additionally, the HOV lanes may include some LOV violators.

The comparison indicates that the daily LOV and HOV model volumes are comparable to the observed counts. The model volumes are higher than the counts in many cases. This suggests that the Shirley Highway volumes at the daily level need not be increased to match the observed counts.

The model volumes (with LOV skims for HOV modes) were also compared to the available AM peak period counts at Glebe Road (Table 3-3).

Table 3-3: 2010 AM Peak Period Assignment Results at on I-395 Northbound at Glebe Road

		Background HOV	
	OBSERVED	**ESTIMATED**	**EST/OBS**
SOV	20,275	17,643	87%
HOV2	1,464	544	37%
HOV3+	6,266	3,167	51%
Total	28,005	21,354	76%

OBS – OBSERVED, EST - ESTIMATED

Source for observed counts: Zilliacus, C. P., & Reschovsky, C. (2011). 2010 Performance of High-Occupancy Vehicle Facilities on Freeways in the Washington Region. Washington, D.C.: National Capital Region Transportation Planning Board.

The comparison suggests that the estimated HOV3+ volumes in the AM peak period are considerably lower than the counts. One of the reasons for the discrepancy between AM peak and daily model volumes could be that the model does not account for the closure of HOV lanes for a few hours during the day.

An incremental logit model was developed to increase the AM peak HOV3+ volumes on Shirley Highway. The shift from SOV and HOV2 to HOV3+ is based on network travel time benefits for HOV3+ from the HOV lanes. The model was calibrated with the objective to match the AM peak HOV3+ model volumes to the counts. The logit model has two λ parameters. One is applied to interchanges with more than half-a-minute travel time benefit. The other is applied to interchanges with less travel time benefit. The model was designed with two λ parameters in order to provide flexibility to vary the influence on interchanges based on travel time criteria. If it is desired to keep the model simple, only one parameter can be retained. The calculation for revised HOV demand using the HOV choice model is shown below:

$$\mathrm{HOV3p} \;=\; \mathrm{SOV} + \mathrm{HOV2} + \mathrm{HOV3p} \;*\; \frac{(\mathrm{HOV3p}) * (\exp(-\lambda * \;\Delta\mathrm{TT}\;)}{(\mathrm{SOV} + \mathrm{HOV2}) + (\mathrm{HOV3p} * \exp(-\lambda * \;\Delta\mathrm{TT}\;)}$$

where $\Delta\mathrm{TT}$ is the travel time saving from using the HOV lane network.

The additional HOV3+ demand is deducted proportionally from SOV and HOV2 demand sets.

Two such sets of λ parameters produce the LOV and HOV volumes on Shirley Highway shown in Table 3-4.

Table 3-4: AM Peak Period HOV3+ Demand by HOV Choice Model

		Background HOV		Adjusted HOV		Adjusted HOV	
				λ1=0.15, λ2=0.10		λ1=0.20, λ2=0.10	
	OBS	EST	EST/OBS	EST	EST/OBS	EST	EST/OBS
SOV	20,275	17,643	87%	15,152	75%	14,493	71%
HOV2	1,464	544	37%	986	67%	1,128	77%
HOV3+	6,266	3,167	51%	6,541	104%	7,193	115%
Total	28,005	21,354	76%	22,679	81%	22,814	81%

OBS – OBSERVED, EST - ESTIMATED

The HOV choice model results in AM peak HOV3+ volumes closer to the count. However, the total volume on Shirley Highway is lower than the count. This suggests that a general lack of attraction to the Shirley Highway corridor is a reason for low HOV volumes, and not just the HOV mode share.

Table 3-5 is a comparison of estimated and observed (INRIX) speeds on LOV and HOV lanes during the AM peak period. One should keep in mind, however, that four-step travel models are not typically validated to speed.

Table 3-5: I-395 AM Peak Speeds at Glebe Road

	LOV	HOV
Observed (INRIX)	23 mph	48 mph
Background HOV	7 mph	59 mph
HOV Choice1	13 mph	57 mph
HOV Choice2	14 mph	56 mph

3.4 Calibrating the HOV Model to the MWCOG Assignment

The HOV choice model was also calibrated to match the HOV volumes on Shirley Highway generated by the original MWCOG two-step assignment. The model was calibrated with lambda of 0.06. Table 3-6 is a comparison of the multi-class assignment using the HOV choice model with the MWCOG 2010 traffic assignment.

Table 3-6: I-95/I-395 AM Peak Calibration Results using MWCOG Targets

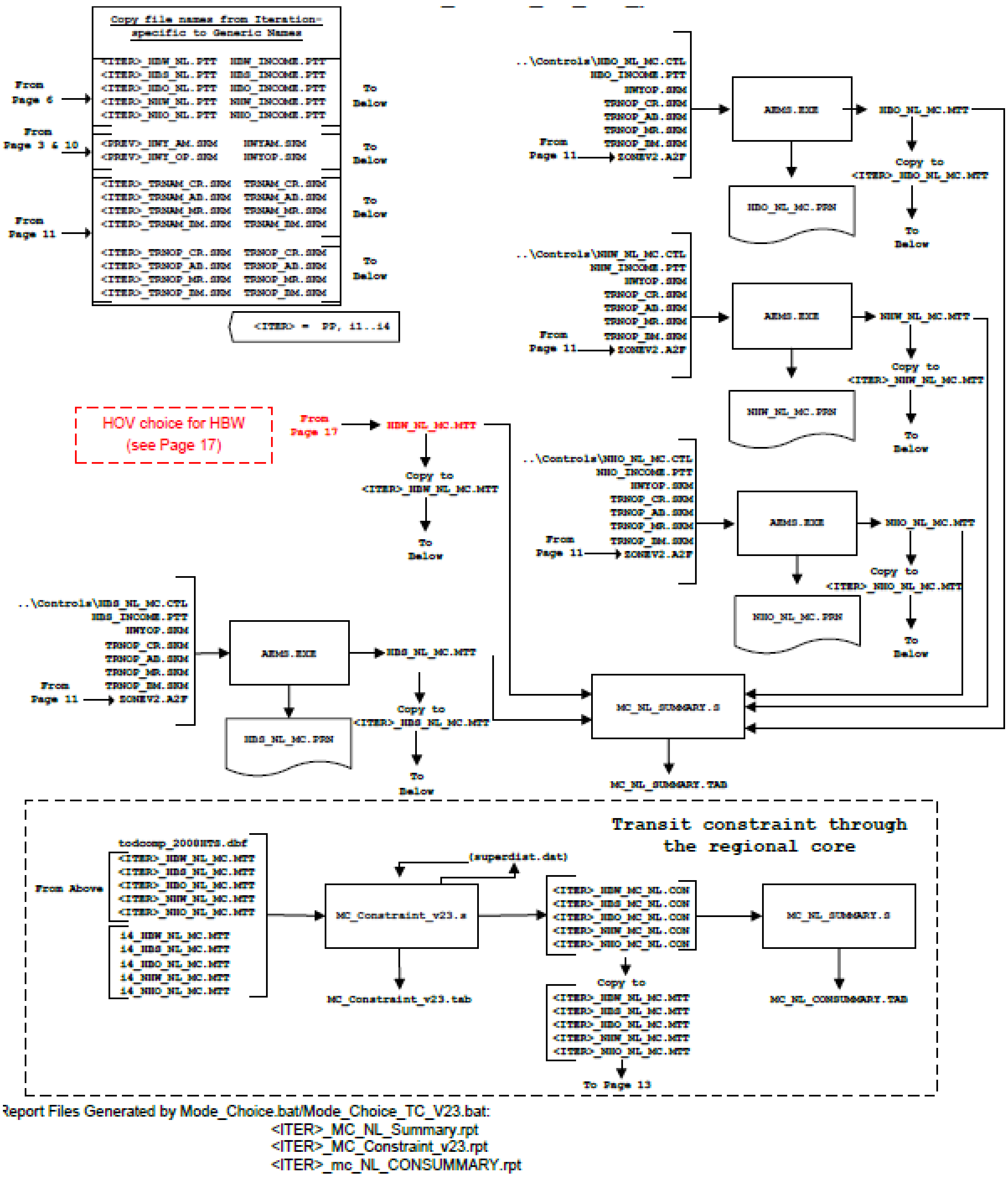
Loc	LOV_COG	LOV_EST	EST/COG	HOV_COG	HOV_EST	EST/COG	COG	EST	EST/COG
1	17,300	17,310	100%	5,690	5,380	95%	22,990	22,690	99%
2	17,910	17,930	100%	5,250	4,910	94%	23,160	22,840	99%
3	16,500	16,430	100%	4,350	4,140	95%	20,850	20,570	99%
5	19,270	18,950	98%	4,060	3,890	96%	23,330	22,840	98%
6	17,260	17,110	99%	3,840	3,760	98%	21,100	20,870	99%
7	17,260	17,110	99%	3,840	3,760	98%	21,100	20,870	99%
8	18,900	18,480	98%	3,650	3,570	98%	22,550	22,050	98%
9	15,930	15,750	99%	3,260	3,220	99%	19,190	18,970	99%
10	14,980	14,650	98%	3,260	3,220	99%	18,240	17,870	98%
11	14,810	14,560	98%	1,760	1,810	103%	16,570	16,370	99%
All	**17,010**	**16,830**	**99%**	**3,900**	**3,770**	**97%**	**20,910**	**20,590**	**98%**

Loc – Count Location, COG – v 2.3.48 MWCOG Model, EST - ESTIMATED

In summary, the HOV choice model can be calibrated to achieve desired HOV volumes on the HOV facilities. However, a careful review of the HOV count data must be conducted before applying the HOV choice model.

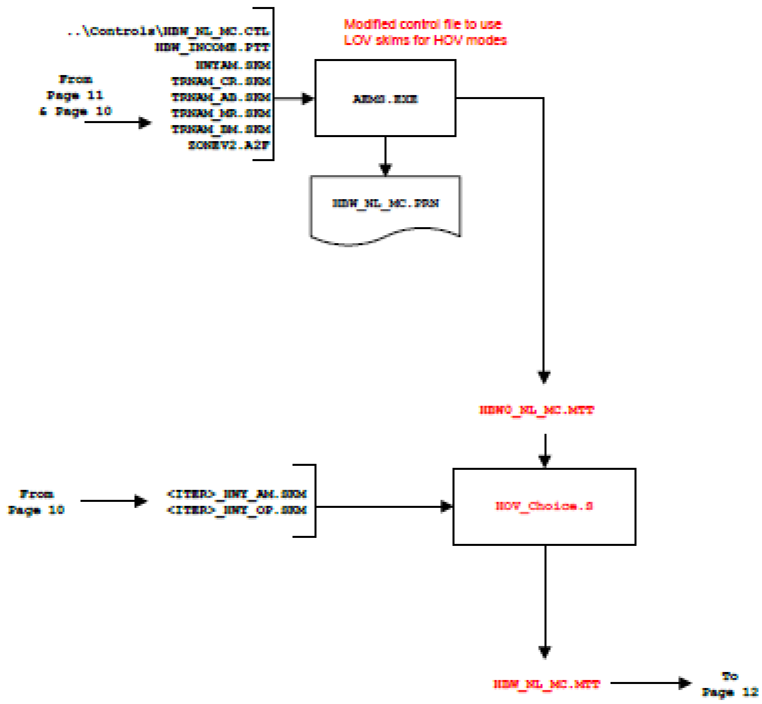
For application purposes, the HOV choice model and the multi-class assignment procedures were integrated into the current mode choice and assignment setups. Modified model procedure flow charts are shown in Figures 3-4 and 3-5. Figure 3-4 shows the change in the current mode choice set up. Figure 3-5 shows the HOV choice process that generates the revised HOV and SOV shares.

Figure 3-4: Change to Mode Choice Process with Proposed HOV Choice Model



Source: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 48) – Page A-12

Figure 3-5: Proposed HOV Choice Model



Source: Process flowchart: Application of the TPB Version 2.3 Travel Model (Build 48) – Page A-17

This page is intentionally blank.

4 HOT Lane Modeling (Task Order 8)

The Version 2.3 Travel Model assigns HOT-lane traffic using a multi-run traffic assignment. The main goal of this technique is to ensure that travel speeds for the HOV 3+ traffic on the HOT lanes are not degraded by the other traffic using the HOT lanes. Since this technique requires two full model runs for each scenario, it doubles model run times. In addition, the model uses a multi-class assignment with six user classes and four time periods with the two peak-periods split into two assignments (HOV3+ and non-HOV3+). This technique was found to improve the assignment of HOV traffic on I-395/Shirley Highway and the Capital Beltway in Virginia.

4.1 Toll Modeling

There are a number of different ways tolls can be modeled within a regional modeling process. Pricing considerations are sometimes implemented as a toll choice model or a sub-mode within a nested logit mode choice model. In this case, the choice of paying a toll or not paying a toll is based on the attributes of the traveler, the trip purpose, and the generalized cost of the two alternatives. Such models can capture traveler behavior effectively, but they are difficult to estimate and calibrate because observed toll choice data at the household level is not generally available.

The overall assumption of a toll choice model, however, is that the decision or choice is made for the whole trip. The model compares a minimum impedance path that includes toll options to a minimum impedance path that does not. The probability of choosing one path over the other is determined by the toll choice model. As with the HOV choice, it is best to limit the estimation and application of a toll choice model to trips that have at least two viable alternatives.

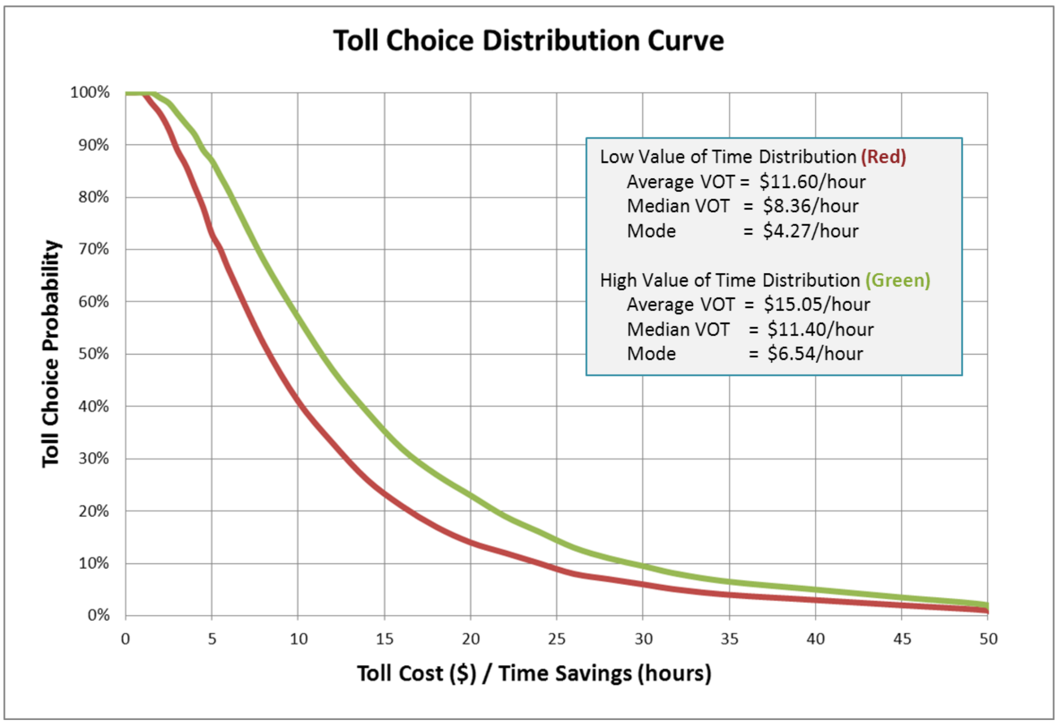
Tolls may also be modeled using a path builder or assignment model. In this case, the decision to pay a toll is based on the impact of the toll on the generalized cost function of the minimum impedance path. This becomes a simple trade-off between travel time and cost based on the value of time. In other words, a single path is built that minimizes the weighted combination of travel time and cost for each trip interchange and all trips on that interchange use that path. In this approach, the toll is buried in the link impedance and is typically unaffected by the attributes of the traveler or the trip purpose.

In reality, different travelers have significantly different values of time for different types of trips. In addition, the value of time for a given trip is not a simple function of the traveler's income. A low income person may have a high value of time for their trip to work or their trip to pick-up their child at the day care center because there are significant costs or other impacts associated with being late. This means that the decision to pay a toll is not an all-or-nothing choice based on the minimum impedance between two points, but a choice probability based on a value of time distribution.

A reasonable compromise between the all-or-nothing nature of a path-based toll choice approach and the complexity of a toll nest within a mode choice model is to apply a probability function based on a value of time distribution to the toll / no-toll path options. The travel time savings provided by the toll-based path is used to determine the probability that trips on that interchange will choose the toll or no-toll option. The trips are split accordingly and assigned to the corresponding path. This makes the assignment approach sensitive to the input value of time distribution and distributes the trips between toll and no-toll options more realistically.

Figure 4-1 shows an example of a toll probability function. This example shows a high and low value of time distribution. The low distribution was derived from a small stated-preference survey about people's willingness to pay tolls and the high distribution was estimated using a regional income distribution. In this example, if the toll option costs $10 but saves an hour of travel time, 40 percent of the travelers would pay the toll under the low value of time distribution and 57 percent of the travelers would pay the toll under the high value of time distribution. In other words, under the low value of time example, 40 percent of the trips on this interchange will be assigned to the toll path and 60 percent of the trips will be assigned to the no-toll path.

Figure 4-1: Toll Probability Example



Source: Tampa-Hillsborough Expressway Authority Bus Toll Lane Feasibility Study

4.2 HOT Lanes

HOT lanes combine HOV choice with managed toll lanes. HOV travelers can use the toll lanes for free or at a reduced price, but other travelers must pay the toll. If the lanes are managed, the toll is dynamically set to ensure that the speed on the toll lanes are maintained at a reasonably high level, typically 50 mph or better. As demand increases, the toll is increased to discourage toll-paying vehicles from using the facility. The toll rate assigned to a given vehicle is based on the time of day and location where the vehicle enters the facility. This means that different vehicles can have different toll rates while traveling on the same link at the same time.

Replicating this level of complexity is well beyond the capabilities of traditional travel demand forecasting models. The best approximation is to estimate the average toll for a given link over the assignment period. For the MWCOG model, assigning trips in four time periods implies that four toll rates will need to be estimated for each link to maintain the desired level of service.

4.3 Dynamically Managed Lanes

Implementing a toll choice model or a multi-class traffic assignment using fixed tolls is relatively straightforward[4]. The toll becomes a component of the generalized cost function using one or more values of time for different vehicle classes. The more difficult challenge is identifying a toll that maintains a given performance level within a dynamically managed lane. For a HOT lane facility, the managed lanes include both tolled and non-tolled traffic (i.e., HOV vehicles). The system manager sets the toll rate for the tolled vehicles based on the speed of the combined traffic volume. Since reducing tolled traffic often increases HOV traffic (due to increased congestion on the un-managed facilities and lanes), setting the toll rate is an iterative process of converging to an optimal solution.

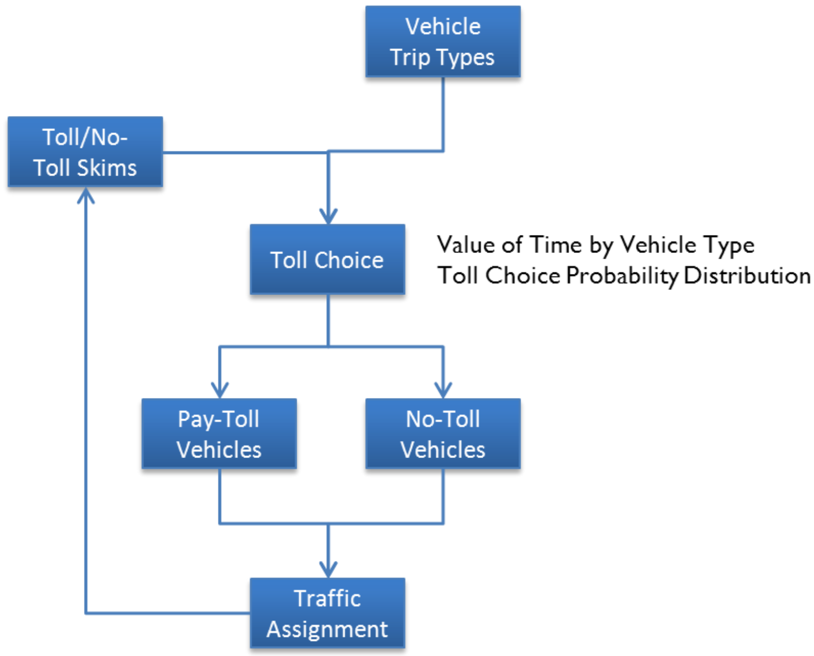
Currently MWCOG identifies the toll rate for managed lanes through an off-line process that iterates through a series of traffic assignments with toll adjustments during each iteration. The toll is incrementally adjusted until the volume-to-capacity ratio on each managed link or group of links is between 0.95 and 1.01 (i.e., the minimum speed threshold). The resulting toll rate is then inserted as a fixed toll on each link by time period for use in a full application of the model.

This task investigated several options for integrating a similar toll setting process with HOV and toll choice models within the overall traffic assignment step in the standard modeling process. The overall design of the traffic assignment process is outlined in Figure 4-2. The primary challenge is to develop a procedure that identifies a reasonable toll rate in a computationally efficient way.

[4] Though, according to Boyce (2010), "Consistent forecasts of multiple-class link flows are needed for the evaluation of proposed facilities for which vehicle classes are treated differently, such as toll roads, bridges, HOV lanes and HOT lanes... However, class-specific link flows ... are not uniquely determined by the standard user-equilibrium traffic assignment model deployed in currently available software systems" (p. 1)

Boyce, David E., Yu (Marco) Nie, Hillel Bar-Gera, Yang Liu, and Yucong Hu. *Field Test of a Method for Finding Consistent Route Flows and Multiple-Class Link Flows in Road Traffic Assignments*. Travel Model Improvement Program (TMIP). Washington, D.C.: U.S. Department of Transportation, Federal Highway Administration, March 8, 2010. http://www.transportation.northwestern.edu/docs/research/Boyce_FieldTestConsistentRouteFlows.pdf.

Figure 4-2: Overall Traffic Assignment Process



The assignment progress outlined in Figure 4-2 differs from the current MWCOG model in two primary ways. The first difference is that the toll choice model in the proposed process uses value of time distributions instead of a single value of time to split the vehicle trip tables into toll and no-toll classes for input to the multi-class assignment. The second difference is the iterative process that is needed to identify a toll rate that achieves the objective function (e.g., maximize revenue or maximize throughput given a minimum performance standard).

The toll choice model requires a toll and no-toll path skim as input. This is typically done using link-use restrictions to prohibit the no-toll path from using toll links. The roadway travel times and costs would be the output of a previous iteration or model application. For interchanges with a significant difference between the generalized cost of the toll and no-toll paths, a toll probability function similar to that in Figure 4-1 would be used to split the vehicle trips into toll and no-toll vehicles for traffic assignment. As explained later in this chapter, the value of time distribution used for this feasibility test was derived from average wage data for the Washington D.C. region from the Bureau of Labor Statistics.

For analysis years with fixed tolls (i.e., no dynamic pricing), it is desirable to re-build the toll / no-toll paths using travel times from the current assignment to identify any trip interchanges where the toll probability changes significantly. This is important if the original input speeds are not realistic (e.g., free flow speeds). In this case it is more efficient to use a series of all-or-nothing assignments with toll probability updates to identify an approximate set of link travel times prior to performing a complete equilibrium assignment.

For analysis years with managed lanes, the assignment results are used to set the toll rates for the next toll-setting iteration. This could be done using the volume-to-capacity ratio method included in the current MWCOG toll setting process or a critical-link approach. The critical-link method uses the toll path from the previous toll-setting iteration to identify the managed link on the path with the greatest performance problems (i.e., lowest speed). The toll rate for the trip is increased to reduce the probability of selecting a toll path for this trip interchange. This approach links the toll rate to the trip, which is more consistent with actual operations. However, it is more difficult and time consuming to implement than the link-based approach. For an initial feasibility test, the link-based approach should be adequate.

The iterative process that identifies an appropriate toll rate for each link is likely to require many iterations to converge. To keep the computer processing time to a manageable level, it is desirable to reduce the traffic assignment processing time as much as possible. Various assignment methods were tested to identify a process that generates approximate link volumes for use in the toll setting procedure. These tests included:

- a series of all-or-nothing assignments with toll updates;
- equilibrium assignments with low convergence criteria followed by toll updates; and
- iterative assignment techniques with imbedded toll updates.

The expectation was that earlier iterations and speed feedback loops will use simplified assignment techniques that would gradually increase in complexity as the process converges. In the end, the complexity of the process must be weighed against the processing time benefits to select the most desirable solution.

The remaining part of this chapter presents changes made to the MWCOG model scripts, and summaries of the performance tests and recommendations to integrate toll setting and a toll choice model into the standard highway assignment process.

4.4 Modifications to the MWCOG Highway Assignment Process

All of the HOT lane assignment results presented in this report are based on year 2020 data from the MWCOG Version 2.3.48 Travel Model with Round 8.1 Cooperative Land-Use Forecasts. The year 2020 was chosen due to the availability of coded HOT lanes on the Beltway and I-95 in Virginia to enable testing variably priced facilities.

The current MWCOG model utilizes Windows batch files to manage the execution of Cube Voyager based scripts and other custom programs for a typical model application. All of these batch files are generic except two, a “wrapper” batch file and a “model-steps” batch file, that are year or scenario specific. As part of this research effort, the overall structure of the model execution was retained while streamlining three areas: the wait-process between sub-steps, Cube Cluster management, and the overall highway assignment process. All of these changes were implemented by modifying the model-steps batch file, relevant Voyager scripts, and the associated “helper” batch files where applicable. Details of changes to the batch files and the Voyager scripts are presented later in this chapter.

The following is a summary of the miscellaneous changes made as part of the highway assignment model development effort in order to streamline the model system management and execution:

1. Currently, the Windows “ping” command is used to pause processing for a 10-second period to prevent possible conflicts of certain processes running in parallel. We have changed the “ping” command to the “choice” command, as shown in Figure 4-3, to increase reliability and reduce clutter in the log and batch files. Using the ping command to pause processing is somewhat obscure, whereas using the choice command is more straightforward. For example, in the choice command, one can specify the number of seconds to wait (in this example, “/T:10” causes the choice command to wait 10 seconds), whereas, the ping command depends on network status and hardware performance, and it requires an estimate of number of retries, which generates multiple output lines in the log file.

Figure 4-3: Wait-process changed from PING to CHOICE

```
BEFORE:
@ping -n 11 127.0.0.1
AFTER:
CHOICE /C C /M "Performing a brief wait ... ; press C to resume immediately" /D:C /T:10
```

2. Four Cygwin[5] DLLs (Figure 4-4) required to execute commands included in the Ver.2.3.48 model such as “tail” and “head” were identified and placed in the root folder of the model.

Figure 4-4: Cygwin DLLs Added to Root Folder

cyggcc_s-1.dll	8/14/2010 8:54 PM	Application extension	46 KB
cygiconv-2.dll	12/23/2009 8:33 AM	Application extension	983 KB
cygintl-8.dll	4/3/2009 1:04 AM	Application extension	31 KB
cygwin1.dll	8/31/2010 4:00 AM	Application extension	2,587 KB

3. Some programs, when they finish running, produce a “return code” that indicates whether the program completed successfully or not. In Windows batch files, the return code is called ERRORLEVEL. After a command is run in a Windows batch file, one may check the return code by using the IF ERRORLEVEL command directly after the program whose return code is being checked. In the original helper batch files, the IF ERRORLEVEL command was placed after the Cluster.exe command, which means it would have checked the return code of Cluster.exe. We have moved the IF ERRORLEVEL command so that it comes immediately after the command that runs the Voyager script (see Figure 4-5). In this way, the IF ERRORLEVEL checks the return code of the Cube Voyage script, not the Cube Cluster command that is used to close the Cube Cluster node.

4. The use of Cube Cluster was modified in two ways: 1) the cluster ID was given a unique name based on the scenario name instead of fixed names - ‘AM’, ‘MD’ and ‘MWCOG’ and 2) the Cube Cluster windows were changed to launch in minimized mode using the key “starthide” option, instead of the “start” option. This change minimizes screen clutter during runtime. These two

[5] Cygwin is a collection of tools that allow one to execute Linux/UNIX commands in Microsoft Windows. As an alternative to this step, one can also install Cygwin and update the Windows PATH environment variable so that it finds the four DLL files in the Cygwin installation folder.

changes corrected the originally misplaced checks to detect failure of sub-steps in the model while unlocking the potential to perform simultaneous model runs within the same root folder by providing separate and unique nodes for each scenario run. These changes can also be noticed in Figure 4-5.

Figure 4-5: Cube Cluster and Exit-Code Check Fix

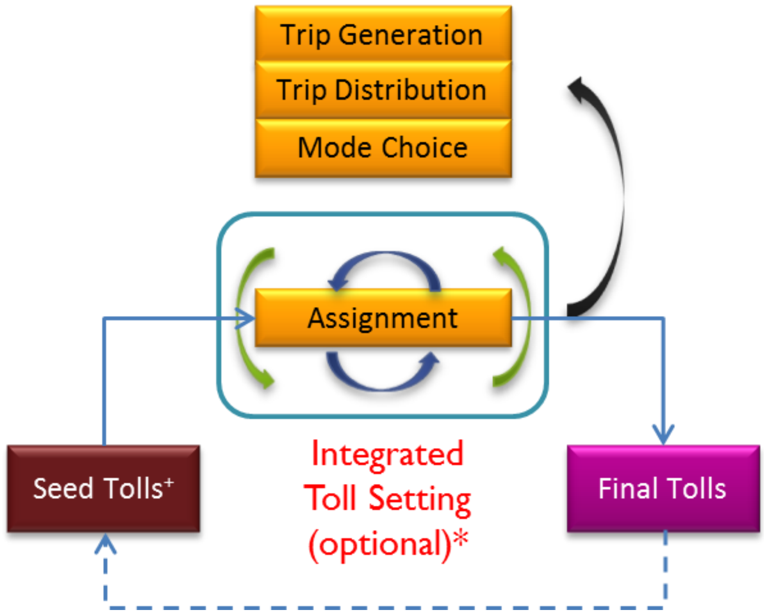
```
BEFORE:
Cluster.exe MWCOG %subnode% start exit
start /w voyager.exe ..\Scripts\Combine_Tables_For_TrAssign_Parallel.s /start -Pvoya -S..\%1
Cluster.exe MWCOG %subnode% close exit

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

AFTER:
Cluster.exe %MWCOGClusterID% %TRNsubnode% starthide exit
start /w voyager.exe ..\Scripts\Combine_Tables_For_TrAssign_Parallel.s /start -Pvoya -S..\%1
if errorlevel 2 goto error
Cluster.exe %MWCOGClusterID% %TRNsubnode% close exit

if exist voya*.prn copy voya*.prn %_iter_%_Combine_Tables_For_TrAssign_Parallel.RPT /y
```

The HOV choice model development and the highway assignment model development processes were initially implemented independently to test various options and were later merged to form a single process. Both of these efforts began by replacing the “two-step” assignment with a single multi-user/class assignment. The “two-step” assignment, used in the Version 2.3 Travel Model for the AM and PM peak periods, involves assigning non-HOV3+ trips first, then assigning HOV3+ trips as a second assignment.

One of the key changes to the highway assignment process involved adapting and integrating MWCOG’s off-line toll setting procedure into the standard assignment procedure. A toll file and a value of time distribution were added to the inputs to facilitate the integrated toll-choice modeling and highway assignment. Additional parameters were introduced in the model-steps batch file and the assignment scripts to allow the user the ability to control and customize toll-setting behavior, including the option to turn it off. These changes are discussed in relevant sub-sections.

Figure 4-6 presents an overview of the highway assignment process showing speed-feedback iterations with a black arrow, all-or-nothing convergence assignments with blue arrows and the newly integrated toll-setting process with green arrows.

Figure 4-6: Proposed Highway Assignment Process

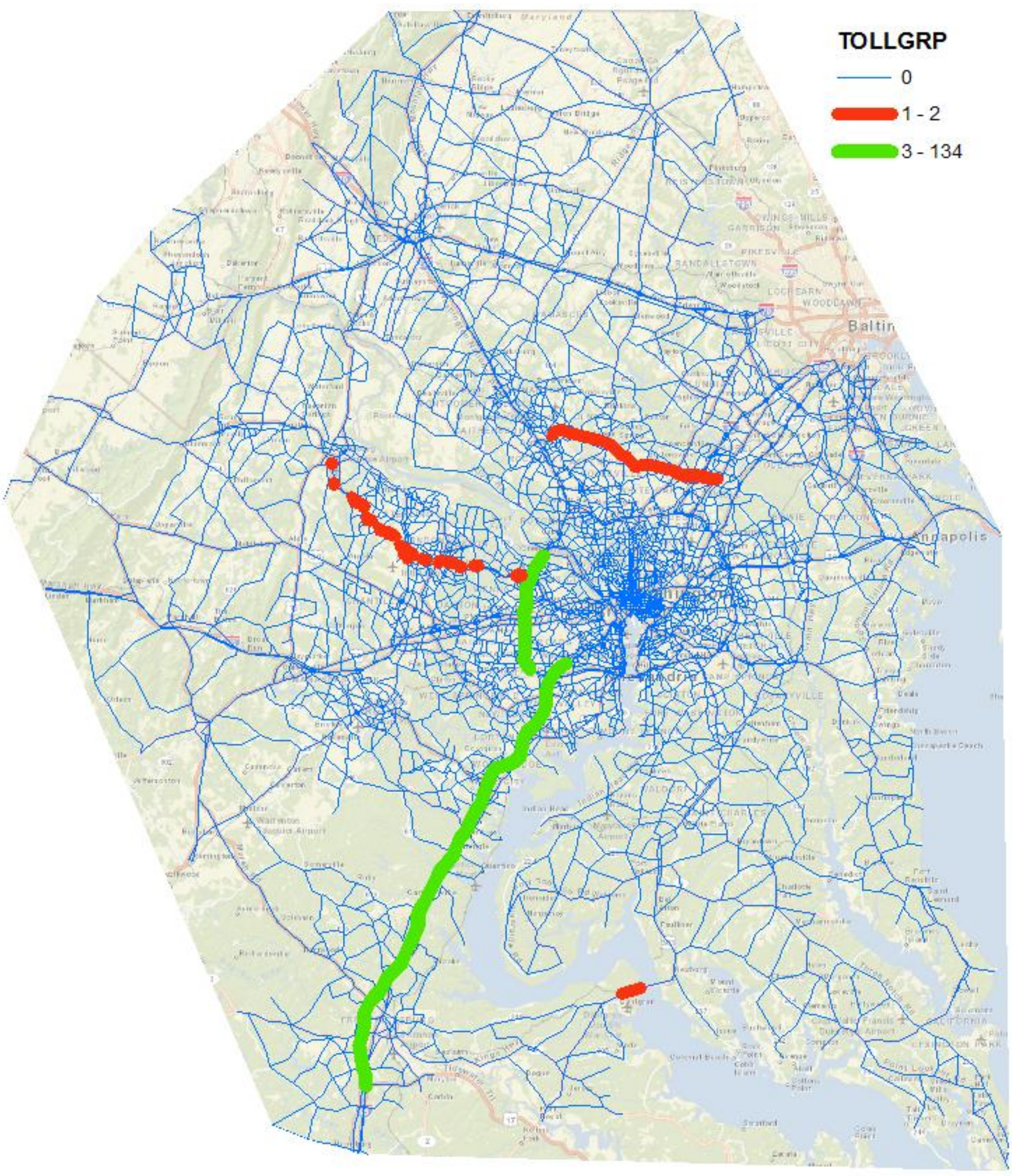


+ Fixed tolls or outputs from the toll setting process of the previous speed-feedback iteration

* Two levels of toll setting convergence criteria and search methods

The integrated toll-setting procedure was designed to adjust only the non-fixed tolls in the network; however the toll-choice model considered all tolls in the generalized cost function to fully account for all costs associated with paths. Currently there are 134 toll groups in the model that include all toll links in the network. The toll groups numbered 1 and 2 use static tolls, whereas toll groups numbered 3 and above use dynamic tolls (i.e., HOT lanes). Figure 4-7 depicts these toll groups in the MWCOG region. All toll groups were retained to avoid introducing additional complexity during evaluation and comparison of results with “out of the box” model runs.

Figure 4-7: Existing Toll Groups in 2020



4.5 Toll File

The existing MWCOG model uses a single input "Toll_Esc.dbf" file[6] to store the tolls for both the fixed- and variable-toll facilities. This file includes toll rates specified in cents/mile and factors for adjusting these toll rates for each of the four time periods (AM, MD, PM and NT) by toll group. These toll rates are then posted on individual links, based on its coded toll group, during the highway network preparation step, as a toll cost per link in cents. While this file structure provides convenient access to the tolls across all four time periods, it is not convenient for parallel processing of time periods. Hence a new toll file, based on the structure previously in use in the original offline toll setting procedure, was used to store and update tolls during integrated toll setting. The tolls from this file are used to encode the "TOLL" and "TOLL_VP" fields on the links and override any previously coded values. The structure of this file is show in Table 4-1. This file stores tolls as toll rates in cents/mile for only variably prices facilities (toll group ≥ 3), and has place holders for two sets of tolls, old and new, to aid in evaluating toll change. The last field in this file contains a flag that indicates satisfying the toll-termination criteria (satisfied when set to '1' and un-satisfied when set to '0').

Table 4-1: Toll File Structure

TOLL_GROUP	NUM_LINKS	AVG_FTYPE	OLD_TOLL	NEW_TOLL	AVG_SPD	AVG_VC	VMT	DIFF_VC	FLAG
3	0	0	0	0	0	0	0	-1.01	1
4	0	0	0	0	0	0	0	-1.01	1
5	0	0	0	0	0	0	0	-1.01	1
6	1	1	224.5356	224.5356	31.7646	1.002	1922.35	-0.008	1
7	1	1	20	20	45.9461	0.9179	1496.83	-0.0921	1
8	3	1	224.5356	224.5356	31.7646	1.002	4950.05	-0.008	1
9	1	1	20	20	59.7217	0.7114	5732.11	-0.2986	1
10	1	1	20	20	62.7909	0.501	3364.11	-0.509	1
11	8	1	20	20	53.9948	0.8442	24049.84	-0.1658	1

The following is a description of various toll files that are input or output during the toll-setting assignment process. They all have the identical structure described earlier.

<TOD> = AM/PM/MD/NT, <itr> = pp/i1/i2/i3/i4 and <tsitr> = 1/2/3/4/5… etc.

1. 'Inputs\SEED_TOLLS_<TOD>.TXT'
   For example: 'inputs\SEED_TOLLS_AM.TXT'
   These four files corresponding to the four time-of-day periods, and are included in the 'inputs' directory for every scenario and are the first set of tolls that are evaluated. The values used for the seed tolls are critical to the computational efficiency of a model run, due to their impact on the number of toll setting iterations required to reach convergence.

2. '<scenario>\LATEST_TOLLS_<TOD>.TXT'
   For example: '2020_ NewHighwayAssignment \LATEST_TOLLS_AM.TXT'

[6] See page 2-3 of *User's Guide for the TPB Travel Forecasting Model, Version 2.3, Build 38, on the 3,722-Zone Area System*. Final Report. Washington, D.C.: National Capital Region Transportation Planning Board, January 20, 2012.

Similar to the four seed toll files, these files are created by the assignment script at the end of every toll setting iteration for potential use in the next toll iteration or speed feedback iteration depending on the user-configuration.

3. '<scenario>\<itr>_OUT<tsiter><TOD>_TOLLS.TXT'
   For example: '2020_NewHighwayAssignment\i4_OUT4AM_TOLLS.TXT'
   These are intermediate toll files that are newly determined ("output") and have not been evaluated yet.

4. '<scenario>\<itr>_IN<tsiter><TOD>_TOLLS.TXT'
   For example: '2020_NewHighwayAssignment\i4_IN2AM_TOLLS.TXT'
   These are also intermediate toll files created for the purpose of checking what tolls have been coded on the network. These "input" toll files from a current toll setting iteration should be identical to the "output" toll files from a previous toll setting iteration.

Also, all tolls (rates) in these files are saved in deflated dollars to avoid the process of deflating them when reading from these files and inflating them while writing to them. This simplification helps avoid errors when updating assignment scripts and keeps the generalized cost computation consistent with the rest of the model.

The latest tolls from preliminary testing and initial applications were used to create seed tolls for performance tests as described later. These seed tolls are presented in Table 4-2 for AM, in Table 4-3 for PM and in Table 4-4 for off-peak (MD and NT). The AM seed tolls range from 20 cents/mile (default/minimum toll) to over $6.0/mile (corresponding to toll group 87) whereas the PM seed tolls range from 20 cents/mile (default/minimum) to over $6.2/mile (corresponding to toll group 82). The MD and NT seed tolls are set to a flat value of 15 cents/mile. In both the peak period seed tolls, only 18 have toll groups have tolls higher than the default/minimum toll rate of 20 cents/mile. Also, some toll groups appear "lumped" together sharing common tolls, for example, toll groups 33, 35 and 37.

Table 4-2: Seed Tolls (AM) for HOT Lanes

3	0	0	0	0	0	0	0	-1.0100	1
4	0	0	0	0	0	0	0	-1.0100	1
5	0	0	0	0	0	0	0	-1.0100	1
6	1	1.00	224.5356	224.5356	31.7646	1.0020	1922.35	-0.0080	1
7	1	1.00	20.0000	20.0000	45.9461	0.9179	1496.83	-0.0921	1
8	3	1.00	224.5356	224.5356	31.7646	1.0020	4950.05	-0.0080	1
9	1	1.00	20.0000	20.0000	59.7217	0.7114	5732.11	-0.2986	1
10	1	1.00	20.0000	20.0000	62.7909	0.5010	3364.11	-0.5090	1
11	8	1.00	20.0000	20.0000	53.9948	0.8442	24049.84	-0.1658	1
12	1	1.00	20.0000	20.0000	62.7909	0.5010	865.06	-0.5090	1
13	5	1.00	44.3779	44.3779	33.2829	0.9934	11053.68	-0.0166	1
14	1	1.00	20.0000	20.0000	64.1305	0.2760	1508.95	-0.7340	1
15	2	1.00	20.0000	20.0000	58.3813	0.7560	2538.19	-0.2540	1
16	1	1.00	20.0000	20.0000	64.2376	0.2478	285.25	-0.7622	1
17	1	1.00	20.0000	20.0000	61.6614	0.6039	3418.02	-0.4061	1
18	1	1.00	20.0000	20.0000	64.4970	0.1760	675.23	-0.8340	1
19	3	1.00	126.5138	126.5138	32.7547	0.9978	13399.96	-0.0122	1
20	1	1.00	20.0000	20.0000	64.5823	0.1493	343.82	-0.8607	1
21	4	1.00	20.0000	20.0000	53.7896	0.8473	7720.81	-0.1627	1
22	4	1.00	20.0000	20.0000	64.1997	0.2578	2373.68	-0.7522	1
23	1	1.00	20.0000	20.0000	63.5362	0.3893	616.18	-0.6207	1
24	3	1.00	20.0000	20.0000	63.9596	0.3141	4579.24	-0.6959	1
25	1	1.00	20.0000	20.0000	63.4785	0.3996	1150.05	-0.6104	1

26	1	1.00	20.0000	20.0000	64.4496	0.1908	805.26	-0.8192	1
27	1	1.00	20.0000	20.0000	63.4249	0.4076	586.45	-0.6024	1
28	2	1.00	20.0000	20.0000	64.1066	0.2823	1164.30	-0.7277	1
29	1	1.00	20.0000	20.0000	64.0746	0.2907	1617.40	-0.7193	1
30	5	1.00	20.0000	20.0000	63.6446	0.3700	4152.10	-0.6400	1
31	2	1.00	20.0000	20.0000	62.7256	0.5070	3939.29	-0.5030	1
32	8	1.00	20.0000	20.0000	64.5535	0.1583	4206.89	-0.8517	1
33	1	1.00	463.3321	463.3321	28.0140	1.0140	2528.93	0.0040	1
34	1	1.00	20.0000	20.0000	64.6498	0.1283	1231.14	-0.8817	1
35	2	1.00	463.3322	463.3322	28.0140	1.0140	3307.06	0.0040	1
36	1	1.00	20.0000	20.0000	63.8587	0.3319	127.35	-0.6781	1
37	1	1.00	463.3321	463.3321	28.0140	1.0140	1069.93	0.0040	1
38	0	0	0	0	0	0	0	-1.0100	1
39	0	0	0	0	0	0	0	-1.0100	1
40	0	0	0	0	0	0	0	-1.0100	1
41	0	0	0	0	0	0	0	-1.0100	1
42	0	0	0	0	0	0	0	-1.0100	1
43	0	0	0	0	0	0	0	-1.0100	1
44	0	0	0	0	0	0	0	-1.0100	1
45	0	0	0	0	0	0	0	-1.0100	1
46	0	0	0	0	0	0	0	-1.0100	1
47	0	0	0	0	0	0	0	-1.0100	1
48	0	0	0	0	0	0	0	-1.0100	1
49	0	0	0	0	0	0	0	-1.0100	1
50	0	0	0	0	0	0	0	-1.0100	1
51	0	0	0	0	0	0	0	-1.0100	1
52	0	0	0	0	0	0	0	-1.0100	1
53	0	0	0	0	0	0	0	-1.0100	1
54	0	0	0	0	0	0	0	-1.0100	1
55	0	0	0	0	0	0	0	-1.0100	1
56	0	0	0	0	0	0	0	-1.0100	1
57	0	0	0	0	0	0	0	-1.0100	1
58	0	0	0	0	0	0	0	-1.0100	1
59	0	0	0	0	0	0	0	-1.0100	1
60	0	0	0	0	0	0	0	-1.0100	1
61	2	1.00	20.0000	20.0000	64.8472	0.0589	175.13	-0.9511	1
62	0	0	0	0	0	0	0	-1.0100	1
63	3	1.00	20.0000	20.0000	64.8127	0.0723	887.23	-0.9377	1
64	0	0	0	0	0	0	0	-1.0100	1
65	3	1.00	20.0000	20.0000	64.7201	0.1065	4208.59	-0.9035	1
66	0	0	0	0	0	0	0	-1.0100	1
67	1	1.00	20.0000	20.0000	63.7269	0.3553	5964.26	-0.6547	1
68	0	0	0	0	0	0	0	-1.0100	1
69	3	1.00	20.0000	20.0000	63.8693	0.3300	6964.65	-0.6800	1
70	0	0	0	0	0	0	0	-1.0100	1
71	3	1.00	20.0000	20.0000	63.5416	0.3884	12702.93	-0.6216	1
72	0	0	0	0	0	0	0	-1.0100	1
73	5	1.00	20.0000	20.0000	60.4800	0.6750	25832.63	-0.3350	1
74	0	0	0	0	0	0	0	-1.0100	1
75	2	1.00	20.0000	20.0000	52.2714	0.8708	15703.86	-0.1392	1
76	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
77	1	1.00	20.0000	20.0000	52.2714	0.8708	7267.21	-0.1392	1
78	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
79	3	1.00	20.0000	20.0000	52.2714	0.8708	30655.94	-0.1392	1
80	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
81	4	1.00	20.0000	20.0000	59.4729	0.7195	23397.57	-0.2905	1
82	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
83	5	1.00	20.0000	20.0000	59.4729	0.7195	26641.49	-0.2905	1
84	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
85	4	1.00	20.0000	20.0000	52.4557	0.8679	17648.98	-0.1421	1
86	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
87	2	1.00	604.3795	604.3795	27.9431	1.0143	6421.22	0.0043	1
88	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
89	1	1.00	131.9558	131.9558	31.2128	1.0036	10541.48	-0.0064	1
90	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
91	2	1.00	60.8508	60.8508	37.9421	0.9598	23614.40	-0.0502	1
92	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
93	3	1.00	192.4230	192.4230	31.0089	1.0042	20517.57	-0.0058	1
94	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
95	4	1.00	192.5539	192.5539	30.7141	1.0051	26609.45	-0.0049	1
96	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
97	4	1.00	233.2719	233.2719	25.7233	1.0122	15838.03	0.0022	1
98	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
99	1	1.00	232.4830	232.4830	31.5033	1.0028	18612.51	-0.0072	1
100	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
101	3	1.00	57.6610	57.6610	38.1012	0.9588	13519.25	-0.0512	1
102	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
103	1	1.00	20.0000	20.0000	42.6668	0.9332	18663.21	-0.0768	1
104	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
105	1	1.00	20.0000	20.0000	59.5250	0.7178	309.85	-0.2922	1
106	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1

107	1	1.00	162.9745	162.9745	38.1471	0.9585	5654.32	-0.0515	1
108	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
109	1	1.00	170.6977	170.6977	37.8567	0.9603	4421.57	-0.0497	1
110	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
111	2	1.00	518.9239	518.9239	31.2944	1.0034	7074.11	-0.0066	1
112	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
113	1	1.00	20.0000	20.0000	61.8045	0.5928	5117.64	-0.4172	1
114	5	1.00	20.0000	20.0000	0	0	0	-1.0100	1
115	3	1.00	20.0000	20.0000	60.5629	0.6699	14265.00	-0.3401	1
116	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
117	0	0	0	0	0	0	0	-1.0100	1
118	5	1.00	20.0000	20.0000	0	0	0	-1.0100	1
119	0	0	0	0	0	0	0	-1.0100	1
120	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
121	0	0	0	0	0	0	0	-1.0100	1
122	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
123	0	0	0	0	0	0	0	-1.0100	1
124	5	1.00	20.0000	20.0000	0	0	0	-1.0100	1
125	0	0	0	0	0	0	0	-1.0100	1
126	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
127	0	0	0	0	0	0	0	-1.0100	1
128	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
129	0	0	0	0	0	0	0	-1.0100	1
130	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
131	0	0	0	0	0	0	0	-1.0100	1
132	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
133	0	0	0	0	0	0	0	-1.0100	1
134	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1

Table 4-3: Seed Tolls (PM) for HOT Lanes

3	0	0	0	0	0	0	0	-1.0100	1
4	0	0	0	0	0	0	0	-1.0100	1
5	0	0	0	0	0	0	0	-1.0100	1
6	1	1.00	345.7130	345.7130	32.7088	0.9982	2716.22	-0.0118	1
7	1	1.00	20.0000	20.0000	57.9574	0.7706	1782.25	-0.2394	1
8	3	1.00	345.7131	345.7131	32.7088	0.9982	6994.26	-0.0118	1
9	1	1.00	20.0000	20.0000	62.5790	0.5205	5948.41	-0.4895	1
10	1	1.00	20.0000	20.0000	62.8079	0.4991	4753.38	-0.5109	1
11	8	1.00	20.0000	20.0000	61.6501	0.6046	24431.16	-0.4054	1
12	1	1.00	20.0000	20.0000	62.8079	0.4991	1222.30	-0.5109	1
13	5	1.00	20.0000	20.0000	46.0347	0.9175	14480.72	-0.0925	1
14	1	1.00	20.0000	20.0000	64.2363	0.2481	1924.40	-0.7619	1
15	2	1.00	20.0000	20.0000	58.6021	0.7485	3564.41	-0.2615	1
16	1	1.00	20.0000	20.0000	63.7924	0.3437	561.08	-0.6663	1
17	1	1.00	20.0000	20.0000	63.2958	0.4266	3424.31	-0.5834	1
18	1	1.00	20.0000	20.0000	64.1069	0.2822	1535.71	-0.7278	1
19	3	1.00	20.0000	20.0000	60.4315	0.6779	12912.83	-0.3321	1
20	1	1.00	20.0000	20.0000	64.1552	0.2695	879.87	-0.7405	1
21	4	1.00	20.0000	20.0000	62.5123	0.5266	6806.92	-0.4834	1
22	4	1.00	20.0000	20.0000	56.1410	0.8133	10622.54	-0.1967	1
23	1	1.00	20.0000	20.0000	64.0511	0.2969	666.53	-0.7131	1
24	3	1.00	152.4436	152.4436	29.3881	1.0093	20871.77	-0.0007	1
25	1	1.00	20.0000	20.0000	63.9095	0.3229	1318.08	-0.6871	1
26	1	1.00	20.0000	20.0000	62.0135	0.5731	3430.83	-0.4369	1
27	1	1.00	20.0000	20.0000	64.1565	0.2691	549.22	-0.7409	1
28	2	1.00	20.0000	20.0000	60.2168	0.6911	4043.45	-0.3189	1
29	1	1.00	20.0000	20.0000	64.1172	0.2795	2205.34	-0.7305	1
30	5	1.00	20.0000	20.0000	35.6500	0.9752	15523.83	-0.0348	1
31	2	1.00	20.0000	20.0000	62.7878	0.5013	5524.50	-0.5087	1
32	8	1.00	20.0000	20.0000	57.7388	0.7781	29325.87	-0.2319	1
33	1	1.00	253.6588	253.6588	31.5633	1.0026	3546.60	-0.0074	1
34	1	1.00	20.0000	20.0000	60.2570	0.6887	9369.62	-0.3213	1
35	2	1.00	253.6587	253.6587	31.5633	1.0026	4637.86	-0.0074	1
36	1	1.00	20.0000	20.0000	59.5144	0.7182	390.84	-0.2918	1
37	1	1.00	253.6588	253.6588	31.5633	1.0026	1500.48	-0.0074	1
38	0	0	0	0	0	0	0	-1.0100	1
39	0	0	0	0	0	0	0	-1.0100	1
40	0	0	0	0	0	0	0	-1.0100	1
41	0	0	0	0	0	0	0	-1.0100	1
42	0	0	0	0	0	0	0	-1.0100	1
43	0	0	0	0	0	0	0	-1.0100	1
44	0	0	0	0	0	0	0	-1.0100	1
45	0	0	0	0	0	0	0	-1.0100	1
46	0	0	0	0	0	0	0	-1.0100	1
47	0	0	0	0	0	0	0	-1.0100	1

48	0	0	0	0	0	0	0	-1.0100	1
49	0	0	0	0	0	0	0	-1.0100	1
50	0	0	0	0	0	0	0	-1.0100	1
51	0	0	0	0	0	0	0	-1.0100	1
52	0	0	0	0	0	0	0	-1.0100	1
53	0	0	0	0	0	0	0	-1.0100	1
54	0	0	0	0	0	0	0	-1.0100	1
55	0	0	0	0	0	0	0	-1.0100	1
56	0	0	0	0	0	0	0	-1.0100	1
57	0	0	0	0	0	0	0	-1.0100	1
58	0	0	0	0	0	0	0	-1.0100	1
59	0	0	0	0	0	0	0	-1.0100	1
60	0	0	0	0	0	0	0	-1.0100	1
61	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
62	0	0	0	0	0	0	0	-1.0100	1
63	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
64	0	0	0	0	0	0	0	-1.0100	1
65	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
66	0	0	0	0	0	0	0	-1.0100	1
67	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
68	0	0	0	0	0	0	0	-1.0100	1
69	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
70	0	0	0	0	0	0	0	-1.0100	1
71	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
72	0	0	0	0	0	0	0	-1.0100	1
73	5	1.00	20.0000	20.0000	0	0	0	-1.0100	1
74	0	0	0	0	0	0	0	-1.0100	1
75	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
76	1	1.00	20.0000	20.0000	63.9723	0.3118	1781.85	-0.6982	1
77	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
78	3	1.00	20.0000	20.0000	60.1166	0.6973	21062.82	-0.3127	1
79	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
80	1	1.00	20.0000	20.0000	61.9603	0.5781	7078.80	-0.4319	1
81	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
82	2	1.00	623.0384	623.0384	29.8479	1.0078	10077.70	-0.0022	1
83	5	1.00	20.0000	20.0000	0	0	0	-1.0100	1
84	1	1.00	86.0662	79.4106	42.9207	0.9319	6085.85	-0.0781	1
85	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
86	1	1.00	64.7035	59.3485	44.0586	0.9264	3214.08	-0.0836	1
87	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
88	1	1.00	20.0000	20.0000	60.6832	0.6625	6625.38	-0.3475	1
89	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
90	1	1.00	20.0000	20.0000	51.6119	0.8815	21047.49	-0.1285	1
91	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
92	3	1.00	20.0000	20.0000	47.9888	0.9095	18189.29	-0.1005	1
93	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
94	1	1.00	44.5572	44.5572	30.8657	1.0046	26448.43	-0.0054	1
95	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
96	4	1.00	44.5572	44.5572	30.8657	1.0046	25423.30	-0.0054	1
97	4	1.00	20.0000	20.0000	0	0	0	-1.0100	1
98	1	1.00	44.5572	44.5572	30.8657	1.0046	7175.93	-0.0054	1
99	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
100	1	1.00	44.5572	44.5572	30.8657	1.0046	20092.61	-0.0054	1
101	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
102	1	1.00	44.5572	44.5572	30.8657	1.0046	12301.60	-0.0054	1
103	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
104	3	1.00	63.3121	63.3121	34.5170	0.9911	25714.11	-0.0189	1
105	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
106	3	1.00	145.3382	145.3382	28.9404	1.0108	38573.76	0.0008	1
107	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
108	2	1.00	112.7811	112.7811	33.6388	0.9905	21022.94	-0.0195	1
109	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
110	1	1.00	20.0000	20.0000	52.5297	0.8667	3065.91	-0.1433	1
111	2	1.00	20.0000	20.0000	0	0	0	-1.0100	1
112	3	1.00	22.9764	22.9764	30.3556	1.0062	21355.61	-0.0038	1
113	1	1.00	20.0000	20.0000	0	0	0	-1.0100	1
114	5	1.00	20.0000	20.0000	43.4163	0.9295	47548.48	-0.0805	1
115	3	1.00	20.0000	20.0000	0	0	0	-1.0100	1
116	1	1.00	20.0000	20.0000	63.0268	0.4664	6599.84	-0.5436	1
117	0	0	0	0	0	0	0	-1.0100	1
118	5	1.00	20.0000	20.0000	44.5416	0.9242	44888.11	-0.0858	1
119	0	0	0	0	0	0	0	-1.0100	1
120	1	1.00	20.0000	20.0000	44.5416	0.9242	30805.56	-0.0858	1
121	0	0	0	0	0	0	0	-1.0100	1
122	3	1.00	20.0000	20.0000	55.1584	0.8271	31284.68	-0.1829	1
123	0	0	0	0	0	0	0	-1.0100	1
124	5	1.00	20.0000	20.0000	62.1246	0.5627	30086.92	-0.4473	1
125	0	0	0	0	0	0	0	-1.0100	1
126	3	1.00	20.0000	20.0000	62.5244	0.5255	24381.23	-0.4845	1
127	0	0	0	0	0	0	0	-1.0100	1
128	3	1.00	20.0000	20.0000	63.3374	0.4204	12642.09	-0.5896	1

129	0	0	0	0	0	0	0	-1.0100	1
130	4	1.00	20.0000	20.0000	63.9501	0.3158	25260.38	-0.6942	1
131	0	0	0	0	0	0	0	-1.0100	1
132	3	1.00	20.0000	20.0000	64.7013	0.1123	1956.43	-0.8977	1
133	0	0	0	0	0	0	0	-1.0100	1
134	2	1.00	20.0000	20.0000	64.7803	0.0848	357.59	-0.9252	1

Table 4-4: Seed Tolls (MD & NT) for HOT Lanes

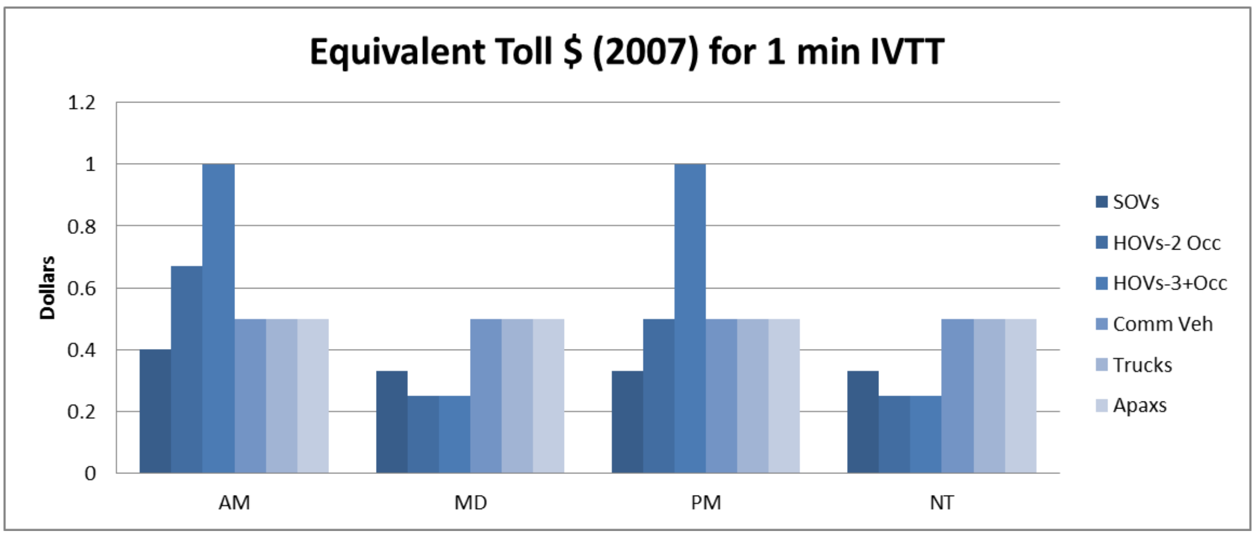
3	3	0	0	0	0	0	0	-1.0100	1
4	4	0	0	0	0	0	0	-1.0100	1
5	5	0	0	0	0	0	0	-1.0100	1
6	6	1	15.0000	15.0000	62.4812	0.5295	2393.31	-0.4805	1
7	7	1	15.0000	15.0000	63.1515	0.4479	1720.80	-0.5621	1
8	8	1	15.0000	15.0000	62.4812	0.5295	6162.78	-0.4805	1
9	9	1	15.0000	15.0000	64.2523	0.2440	4630.99	-0.7660	1
10	10	1	15.0000	15.0000	64.1731	0.2648	4188.30	-0.7452	1
11	11	1	15.0000	15.0000	63.9933	0.3081	20680.78	-0.7019	1
12	12	1	15.0000	15.0000	64.1731	0.2648	1076.99	-0.7452	1
13	13	1	15.0000	15.0000	61.5455	0.6108	16011.67	-0.3992	1
14	14	1	15.0000	15.0000	64.1731	0.2648	3410.47	-0.7452	1
15	15	1	15.0000	15.0000	63.5532	0.3863	3055.34	-0.6237	1
16	16	1	15.0000	15.0000	64.3881	0.2084	565.11	-0.8016	1
17	17	1	15.0000	15.0000	64.3015	0.2311	3080.77	-0.7789	1
18	18	1	15.0000	15.0000	64.7177	0.1072	969.32	-0.9028	1
19	19	1	15.0000	15.0000	62.6573	0.5133	16239.26	-0.4967	1
20	20	1	15.0000	15.0000	64.7736	0.0874	473.93	-0.9226	1
21	21	1	15.0000	15.0000	63.1047	0.4549	9765.25	-0.5551	1
22	22	1	15.0000	15.0000	64.4764	0.1824	3957.54	-0.8276	1
23	23	1	15.0000	15.0000	64.4291	0.1972	735.45	-0.8128	1
24	24	1	15.0000	15.0000	63.8899	0.3264	11211.45	-0.6836	1
25	25	1	15.0000	15.0000	64.2381	0.2477	1679.18	-0.7623	1
26	26	1	15.0000	15.0000	64.4718	0.1839	1828.28	-0.8261	1
27	27	1	15.0000	15.0000	63.6956	0.3609	1223.29	-0.6491	1
28	28	1	15.0000	15.0000	64.1661	0.2666	2590.56	-0.7434	1
29	29	1	15.0000	15.0000	63.3952	0.4120	5399.57	-0.5980	1
30	30	1	15.0000	15.0000	63.2789	0.4291	11345.23	-0.5809	1
31	31	1	15.0000	15.0000	63.2518	0.4331	7927.51	-0.5769	1
32	32	1	15.0000	15.0000	64.3547	0.2171	13591.43	-0.7929	1
33	33	1	15.0000	15.0000	52.5648	0.8662	5089.27	-0.1438	1
34	34	1	15.0000	15.0000	64.4561	0.1888	4266.21	-0.8212	1
35	35	1	15.0000	15.0000	52.5648	0.8662	6655.20	-0.1438	1
36	36	1	15.0000	15.0000	63.8744	0.3291	297.53	-0.6809	1
37	37	1	15.0000	15.0000	52.5648	0.8662	2153.15	-0.1438	1
38	38	0	0	0	0	0	0	-1.0100	1
39	39	0	0	0	0	0	0	-1.0100	1
40	40	0	0	0	0	0	0	-1.0100	1
41	41	0	0	0	0	0	0	-1.0100	1
42	42	0	0	0	0	0	0	-1.0100	1
43	43	0	0	0	0	0	0	-1.0100	1
44	44	0	0	0	0	0	0	-1.0100	1
45	45	0	0	0	0	0	0	-1.0100	1
46	46	0	0	0	0	0	0	-1.0100	1
47	47	0	0	0	0	0	0	-1.0100	1
48	48	0	0	0	0	0	0	-1.0100	1
49	49	0	0	0	0	0	0	-1.0100	1
50	50	0	0	0	0	0	0	-1.0100	1
51	51	0	0	0	0	0	0	-1.0100	1
52	52	0	0	0	0	0	0	-1.0100	1
53	53	0	0	0	0	0	0	-1.0100	1
54	54	0	0	0	0	0	0	-1.0100	1
55	55	0	0	0	0	0	0	-1.0100	1
56	56	0	0	0	0	0	0	-1.0100	1
57	57	0	0	0	0	0	0	-1.0100	1
58	58	0	0	0	0	0	0	-1.0100	1
59	59	0	0	0	0	0	0	-1.0100	1
60	60	0	0	0	0	0	0	-1.0100	1
61	61	1	15.0000	15.0000	64.7101	0.1096	383.89	-0.9004	1
62	62	0	0	0	0	0	0	-1.0100	1
63	63	1	15.0000	15.0000	64.6537	0.1271	1838.80	-0.8829	1
64	64	0	0	0	0	0	0	-1.0100	1
65	65	1	15.0000	15.0000	64.5193	0.1690	7867.88	-0.8410	1
66	66	0	0	0	0	0	0	-1.0100	1
67	67	1	15.0000	15.0000	64.1866	0.2612	5165.27	-0.7488	1
68	68	0	0	0	0	0	0	-1.0100	1
69	69	1	15.0000	15.0000	64.2983	0.2319	5764.28	-0.7781	1

70	70	0	0	0	0	0	0	-1.0100	1
71	71	1	15.0000	15.0000	64.0619	0.2941	11330.33	-0.7159	1
72	72	0	0	0	0	0	0	-1.0100	1
73	73	1	15.0000	15.0000	64.0051	0.3060	13797.72	-0.7040	1
74	74	0	0	0	0	0	0	-1.0100	1
75	75	1	15.0000	15.0000	63.4430	0.4049	8601.87	-0.6051	1
76	76	1	15.0000	15.0000	64.8466	0.0591	374.16	-0.9509	1
77	77	1	15.0000	15.0000	63.4430	0.4049	3980.65	-0.6051	1
78	78	1	15.0000	15.0000	64.5478	0.1601	5355.43	-0.8499	1
79	79	1	15.0000	15.0000	63.4430	0.4049	16791.95	-0.6051	1
80	80	1	15.0000	15.0000	64.6296	0.1346	1825.55	-0.8754	1
81	81	1	15.0000	15.0000	63.9577	0.3144	12043.18	-0.6956	1
82	82	1	15.0000	15.0000	60.9207	0.6481	7177.16	-0.3619	1
83	83	1	15.0000	15.0000	63.9577	0.3144	13712.88	-0.6956	1
84	84	1	15.0000	15.0000	62.5456	0.5236	3786.27	-0.4864	1
85	85	1	15.0000	15.0000	62.9736	0.4743	11362.93	-0.5357	1
86	86	1	15.0000	15.0000	63.7492	0.3513	1349.75	-0.6587	1
87	87	1	15.0000	15.0000	61.6535	0.6044	4507.49	-0.4056	1
88	88	1	15.0000	15.0000	64.6510	0.1280	1416.99	-0.8820	1
89	89	1	15.0000	15.0000	63.8465	0.3341	5511.35	-0.6759	1
90	90	1	15.0000	15.0000	64.5860	0.1482	3918.93	-0.8618	1
91	91	1	15.0000	15.0000	64.0898	0.2867	11078.99	-0.7233	1
92	92	1	15.0000	15.0000	64.5665	0.1543	3416.69	-0.8557	1
93	93	1	15.0000	15.0000	63.9996	0.3070	9852.46	-0.7030	1
94	94	1	15.0000	15.0000	64.5181	0.1694	4937.76	-0.8406	1
95	95	1	15.0000	15.0000	63.9360	0.3182	13233.22	-0.6918	1
96	96	1	15.0000	15.0000	64.5181	0.1694	4746.37	-0.8406	1
97	97	1	15.0000	15.0000	63.8338	0.3353	8285.02	-0.6747	1
98	98	1	15.0000	15.0000	64.5181	0.1694	1339.70	-0.8406	1
99	99	1	15.0000	15.0000	63.4590	0.4026	11736.18	-0.6074	1
100	100	1	15.0000	15.0000	64.5181	0.1694	3751.16	-0.8406	1
101	101	1	15.0000	15.0000	63.8966	0.3252	7202.39	-0.6848	1
102	102	1	15.0000	15.0000	64.5181	0.1694	2296.63	-0.8406	1
103	103	1	15.0000	15.0000	64.0606	0.2944	9247.84	-0.7156	1
104	104	1	15.0000	15.0000	64.4910	0.1752	4812.93	-0.8348	1
105	105	1	15.0000	15.0000	64.4082	0.2031	137.72	-0.8069	1
106	106	1	15.0000	15.0000	64.2050	0.2564	10833.93	-0.7536	1
107	107	1	15.0000	15.0000	61.5573	0.6101	5652.82	-0.3999	1
108	108	1	15.0000	15.0000	63.8292	0.3371	5942.88	-0.6729	1
109	109	1	15.0000	15.0000	57.0478	0.8009	5792.08	-0.2091	1
110	110	1	15.0000	15.0000	64.0814	0.2889	848.77	-0.7211	1
111	111	1	15.0000	15.0000	51.0731	0.8904	9859.72	-0.1196	1
112	112	1	15.0000	15.0000	63.7373	0.3535	6230.41	-0.6565	1
113	113	1	15.0000	15.0000	64.2187	0.2528	3427.44	-0.7572	1
114	114	1	15.0000	15.0000	63.8479	0.3338	14183.01	-0.6762	1
115	115	1	15.0000	15.0000	64.0937	0.2857	9555.01	-0.7243	1
116	116	1	15.0000	15.0000	64.5282	0.1662	1953.50	-0.8438	1
117	117	0	0	0	0	0	0	-1.0100	1
118	118	1	15.0000	15.0000	64.0602	0.2945	11880.36	-0.7155	1
119	119	0	0	0	0	0	0	-1.0100	1
120	120	1	15.0000	15.0000	64.0602	0.2945	8153.19	-0.7155	1
121	121	0	0	0	0	0	0	-1.0100	1
122	122	1	15.0000	15.0000	64.2334	0.2489	7818.91	-0.7611	1
123	123	0	0	0	0	0	0	-1.0100	1
124	124	1	15.0000	15.0000	64.1164	0.2797	12420.26	-0.7303	1
125	125	0	0	0	0	0	0	-1.0100	1
126	126	1	15.0000	15.0000	64.1457	0.2720	10479.68	-0.7380	1
127	127	0	0	0	0	0	0	-1.0100	1
128	128	1	15.0000	15.0000	64.3649	0.2144	5355.03	-0.7956	1
129	129	0	0	0	0	0	0	-1.0100	1
130	130	1	15.0000	15.0000	64.5888	0.1473	9788.90	-0.8627	1
131	131	0	0	0	0	0	0	-1.0100	1
132	132	1	15.0000	15.0000	64.6425	0.1306	1888.80	-0.8794	1
133	133	0	0	0	0	0	0	-1.0100	1
134	134	1	15.0000	15.0000	64.7207	0.1063	372.36	-0.9037	1

4.6 Toll Choice Distribution

The current MWCOG process uses equivalent toll minutes for converting toll costs to a generalized cost (time) for each of the six vehicle classes during path building as shown in Figure 4-8. As part of the toll choice model, a toll choice distribution curve based on the value of time distribution is used to separate the OD trips into two groups: toll trips and no-toll trips. Note that this split is performed only once per toll setting iteration in order to avoid affecting the highway assignment’s convergence. Toll and no-toll skims are prepared by building paths with and without access to all of the tolled links to assist in this process. The toll choice distribution for the Washington region was prepared using the average wage rate of $31.10/hour and a median wage rate of $24.10/hour for 2012 from the Bureau of Labor Statistics (BLS) for the MWCOG region. The resulting toll choice distribution curve is shown in Figure 4-9. This curve suggest that 30 percent of the trips will choose to pay the toll and 70 percent will use the no-toll path for an interchange with a travel time savings of one hour with a toll cost of approximately $35.00. As mentioned earlier, both the fixed-toll (toll group numbers 1 and 2) and the variable-toll (toll groups 3 and greater) facilities are included in the preparation of the toll skims in order to correctly evaluate the trade-offs for choosing a toll path. The SOV travel-time savings in the MWCOG region for the year 2020 by origin zone and time period are shown in Figure 4-10 for AM, in Figure 4-11 for PM, in Figure 4-12 for MD and in Figure 4-13 for NT. The AM map shows a logical distribution of home zones with a positive, toll-based time savings, along the region's prominent toll corridors of I-95, I-495 in VA and the Dulles Toll Road/Dulles Greenway.

Figure 4-8: Existing Equivalent Toll Minutes



Source: MWCOG. These value-of-time values are derived from information in the equivalent toll-minutes file (Toll_Minutes.TXT). For details, see page 3-3 of the January 20, 2012 user's guide for the Version 2.3 Travel Model.

Figure 4-9: MWCOG Region Value of Time Distribution

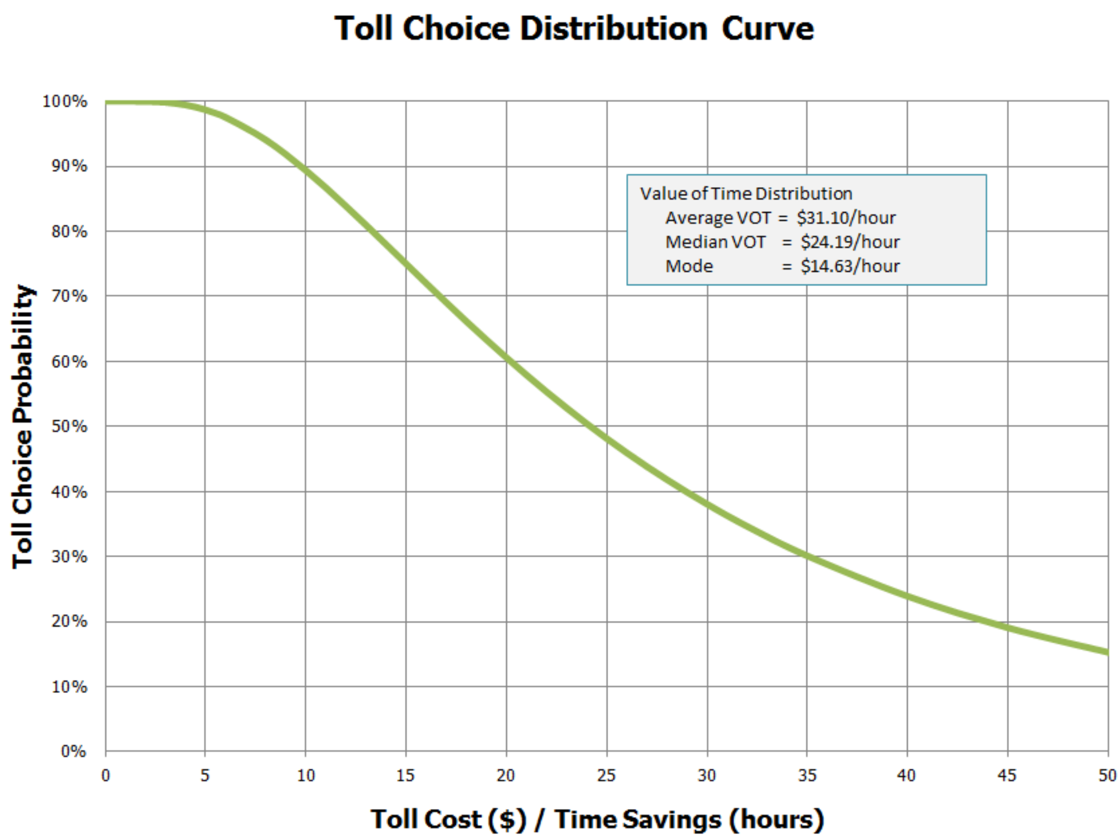


Figure 4-10: SOV AM Origins with Positive Toll Based Time Savings

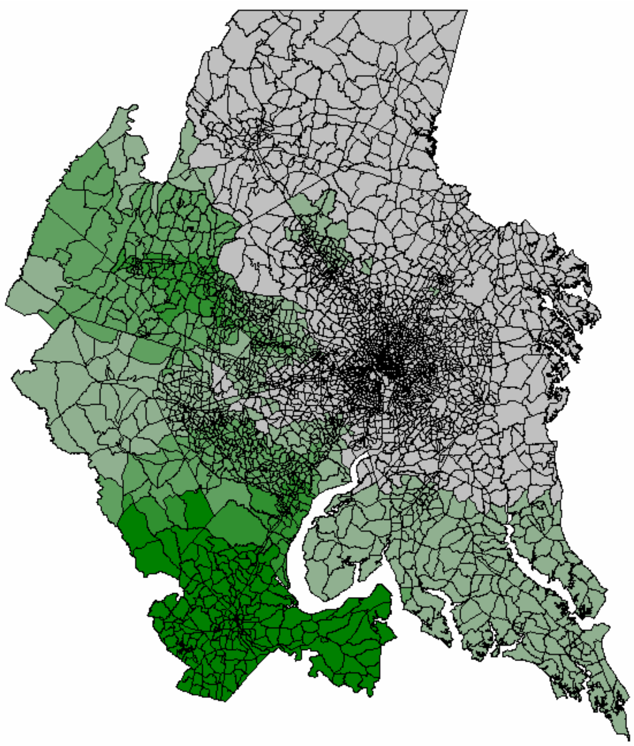


Figure 4-11: SOV PM Origins with Positive Toll Based Time Savings

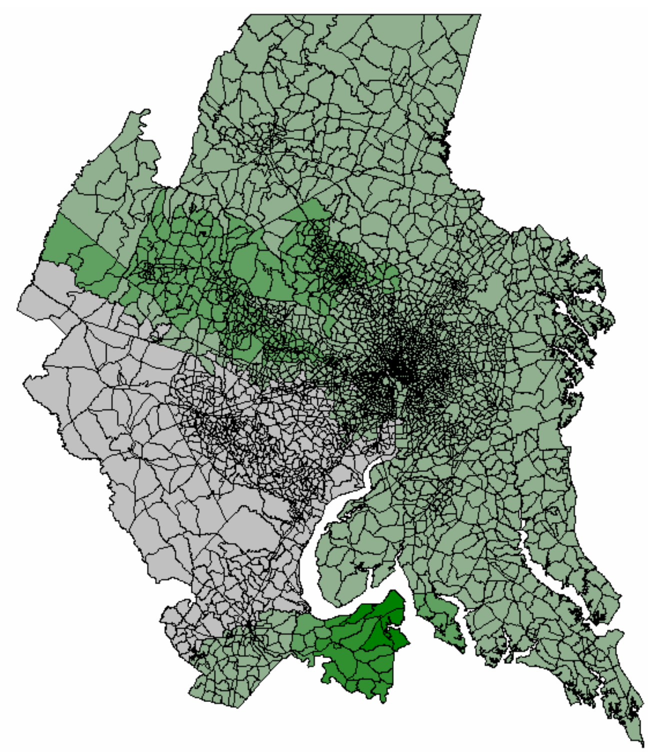


Figure 4-12: SOV MD Origins with Positive Toll Based Time Savings

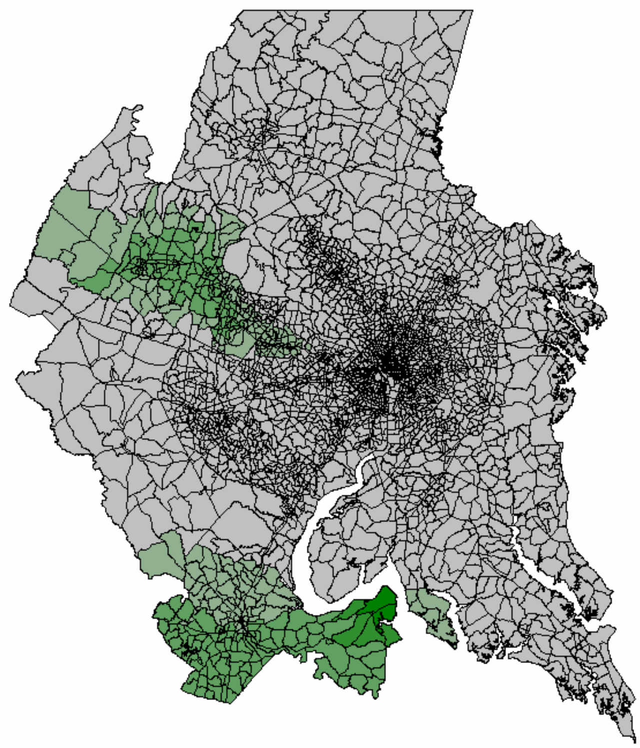
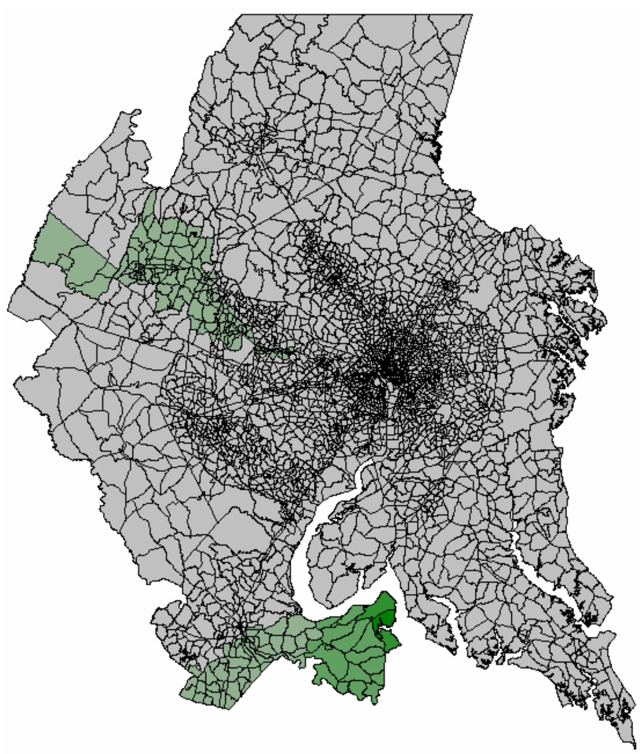


Figure 4-13: SOV NT Origins with Positive Toll Based Time Savings



4.7 Toll Setting

The most important consideration in this integrated toll-setting and highway assignment procedure is the computational efficiency in order to have a reasonable model runtime that is less than the current runtime under the HOV-skim replacement approach (two-step highway assignment) of approximately 48 hours or two days on a modern workstation with 8-cores. MWCOG's current procedure for toll-setting involves manual intermediate steps and takes up to four days of runtime depending on the model year and hardware resources.

A comparison test between the existing toll-setting logic and the modified toll-setting approach is shown in Figure 4-14, Figure 4-15, Figure 4-16 and Figure 4-17 corresponding to the AM, PM, MD and NT time-of-day periods. This test included restricted convergence criteria for testing purposes. In each of these figures, the toll rates (in cents/mile) for each of the 132 variably priced toll groups (#3 to #134) are plotted by the number of toll-setting iterations. Since there are 132 toll groups, each graph could have 132 curves/lines. However, in many cases, the curves lie on top of one another, which we refer to as "grouping," so each graph appears to have many fewer curves/lines than 132. Also, note that, for readability purposes, the legend shows only a subset of the 132 toll groups (toll groups 3-18). The starting toll values were 20 cents/mile for the peak periods and 15 cents/mile for off-peak periods. As noted earlier, the curves representing each toll group exhibit "grouping," i.e., they lie on top of each other. Although there is grouping in both the old/current method and the newer method, the newer method seems to have less grouping (i.e., more toll group curves are visible on the graphs), which means the newer method results in a larger number of distinct toll values. Additionally, in the old/current method, toll values are increased, but never decreased. By contrast, in the new method, tolls are allowed to both increase and decrease. While the newer method required more attempts to converge, the final tolls are quite different between the two approaches. The older method in general tends to end with higher tolls compared to the newer methods, though not in all cases.

Figure 4-14: Toll Setting - AM

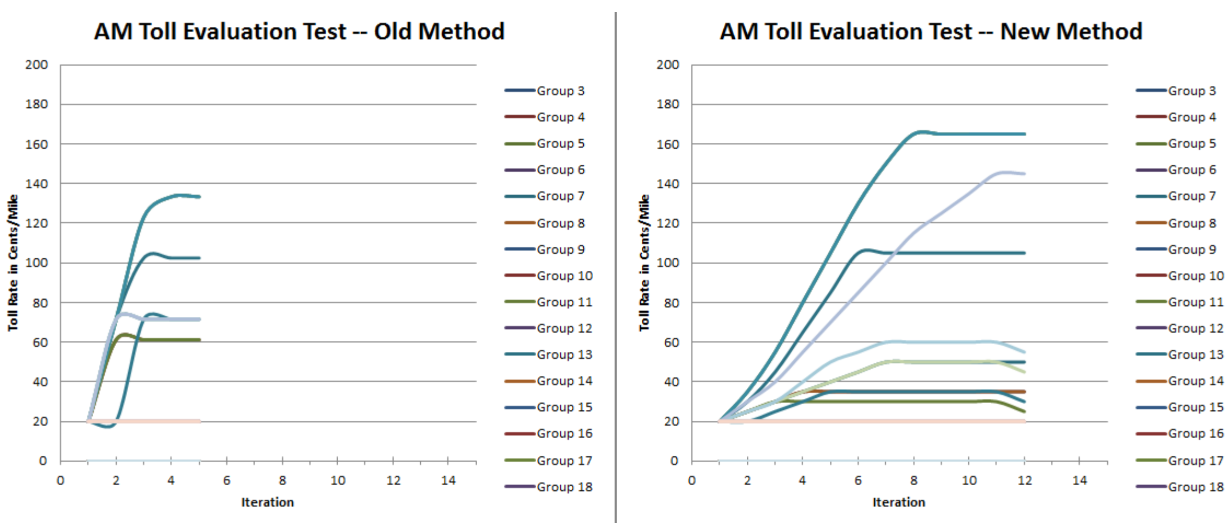


Figure 4-15: Toll Setting - PM

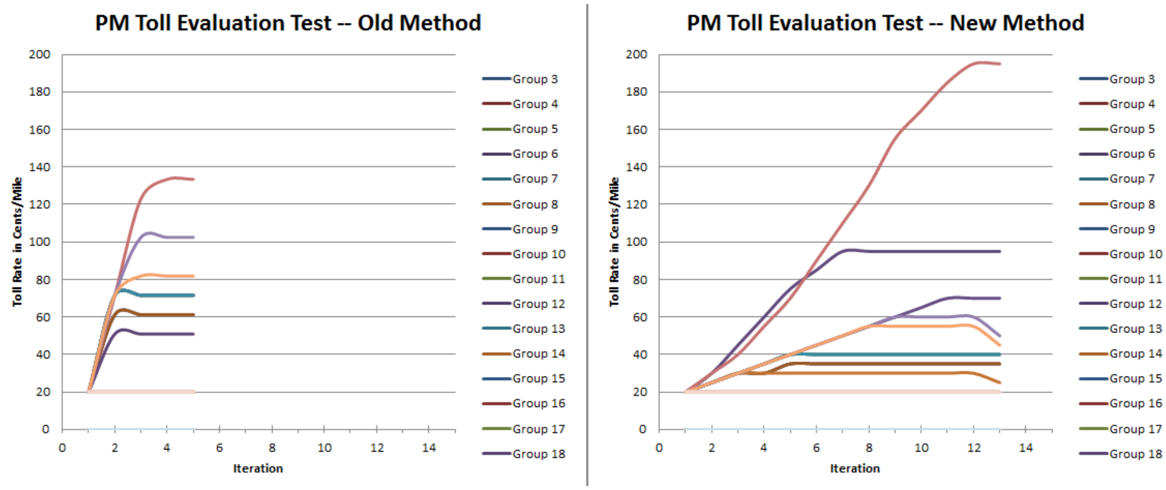


Figure 4-16: Toll Setting - MD

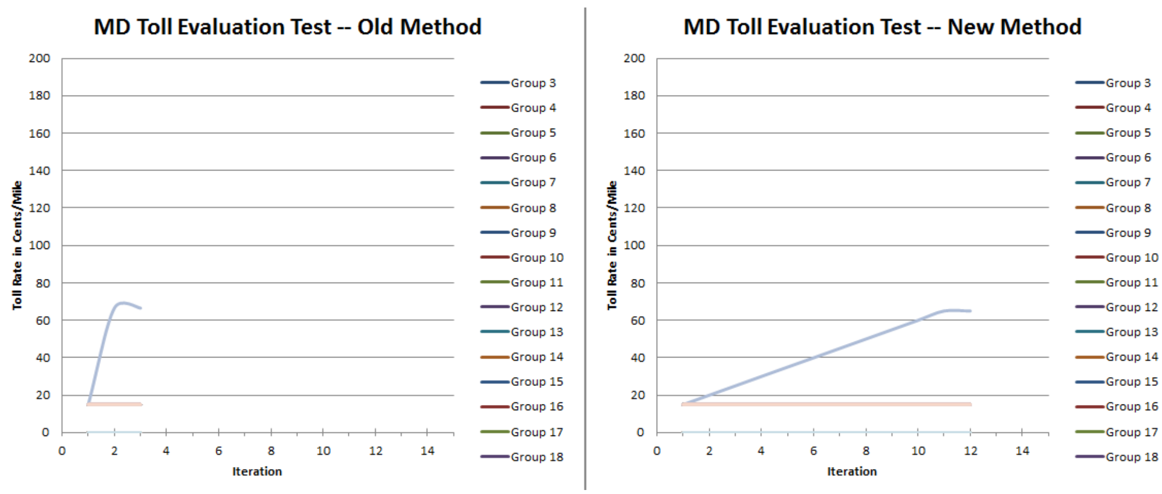
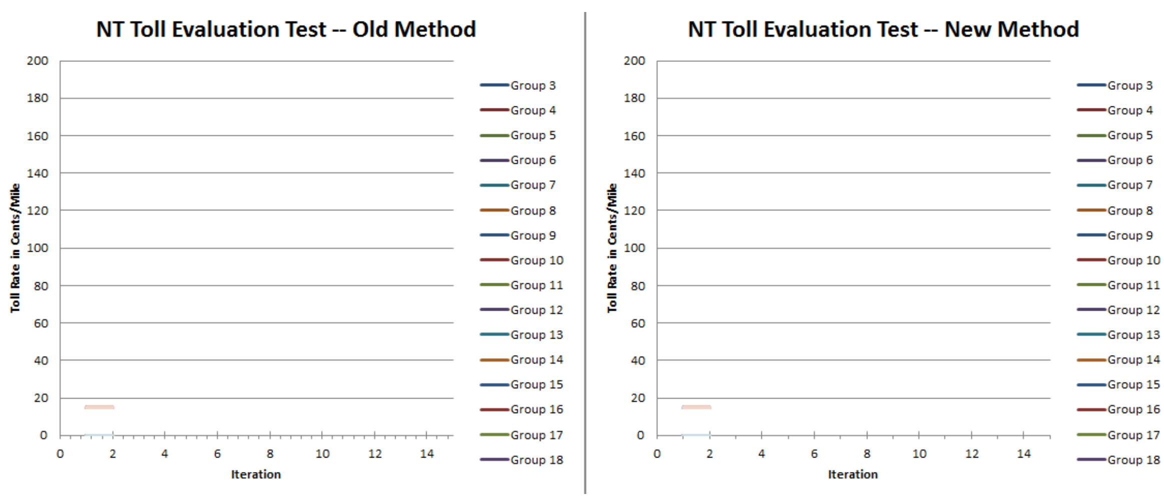


Figure 4-17: Toll Setting - NT



In this regard, the key to computational efficiency lies in minimizing the number of iterations required to estimate the optimal tolls. There are several factors that can be considered to achieve this objective as listed below:

1. Setting a cap on the number of toll estimation attempts;
2. Using "seed" or initial tolls from a previous model run or speed-feedback iteration;
3. Smart logic for upward and downward toll adjustments based on the results of previous assignments;
4. Rounding the V/C ratio threshold values for less stringent evaluation criteria;
5. Adjusting the toll in rounded increments (for e.g.: increments of 5 cents/mile) towards fewer and more realistic tolls;
6. Protecting against erroneous inputs such as network coding errors that might result in unnecessary or infinite toll estimation loops; and
7. Adjusting or consolidating the number of toll groups to reduce the number of combinations for evaluation.

Keeping these factors in mind, the toll setting procedure was created through several rounds of testing and parameterizing most of the factors impacting computational efficiency. For instance, the toll resolution (defining the step size for toll change) is kept at 0.0001 cents, so that the change in toll corresponding to a change in V/C ratio is accurately measured and implemented. For realism, this value can be raised to 1.0 to represent whole cents. This large step size, however, will likely result in a large impact on computational efficiency. Similarly, the V/C ratio is maintained at full precision (double) for calculations, however, for checks against thresholds, it is rounded to the user-specified number of decimal places. For example, with a V/C ratio resolution of "2" (decimals), a computed V/C ratio of 0.9499 would be considered the same as the lower threshold V/C ratio of 0.9500 and would therefore not require any further adjustment.

As part of the toll setting process, the first step is to read a loaded highway network to compute average V/C ratios along with the weighted speed and VMT for each link in a toll group. The next step involves processing every toll group to compare the V/C ratios against the upper threshold V/C ratio of 1.01. For toll groups with V/C ratios above the upper threshold (1.01), the toll is raised, whereas, for the toll groups with V/C ratios below the lower threshold (0.95), the toll is lowered.

To avoid oscillation and reduce the number of toll setting iterations, the toll increase is coded as a non-linear function, whereas the toll decrease is coded as a linear function, as shown below. These toll estimation functions define the relationship between the current tolls and the current V/C ratios used to adjust the tolls for the next iteration.

Non-Linear Toll Increase Function

$$New\ (higher)\ Toll\ Rate = Old\ Toll\ Rate + \left(\lambda_1 * \ln\left(abs\left(VC_{Avg} - 1.01\right)\right) + \lambda_2\right)$$

Where,

$$\lambda_1 = 15.0$$

$$\lambda_2 = 80.0$$

Linear Toll Decrease Function

$$New\ (lower)\ Toll\ Rate = Old\ Toll\ Rate * \ \ VC_{Avg} \ \ 1.01$$

The non-linear toll increase function was formulated by fitting curves to the data points gathered from the initial and preliminary toll setting iterations. During the initial estimation, values of 12.691 and 68.891 for the λ_1 andλ_2, respectively, were found to correspond to the toll-increase function in MWCOG's offline toll setting procedure. These parameters were later updated to their current values to minimize toll setting iterations.

4.8 Assignment Scripts and Batch Files

As part of changes to the Cube Cluster management, the sub-nodes were re-organized to streamline the cluster management and to facilitate performing simultaneous scenario runs. In the wrapper batch file, the 'AMsubnode' and 'MDsubnode' environment variables were consolidated into a single 'HWYsubnode' environment variable and the 'subnode' environment variable was renamed to 'TRNsubnode' for clarity. Additionally, a new environment variable was introduced to uniquely identify the cluster id: 'MWCOGClusterID'. This cluster ID was set equal to the name of the scenario. Additionally, several parameters controlling the configuration of toll setting in the highway assignment and in the HOV choice model were introduced as environment variables in the model-steps batch file. These changes are shown in Figure 4-18 and Figure 4-19.

Figure 4-18: Changes to the Wrapper Batch File

```
set MWCOGClusterID=%scenar%

set HWYsubnode=1-8

set TRNsubnode=1-3
```

Figure 4-19: Changes to Model-Setup Batch File

```
:: Set Toll Setting Parameters

set _perform_toll_setting_=1
set _lower_relgap_for_toll_setting_=0
set _use_last_tolls_=0
set _use_vot_distrib_in_toll_setting_=1

:: Set HOV Binary Choice Lambda Values

set _lambda1_=0.0
set _lambda2_=0.0
set _lambda3_=0.2
set _lambda4_=0.1
set _lambda5_=0.2
set _lambda6_=0.1
```

The highway assignment script underwent the biggest change, as might be expected, due to the integration of the toll choice model and the toll setting procedures. The key change to the script was "off-shoring" portions of the code to external files in order to make it easier to maintain, debug and follow the top-level logic of the assignment script. The final assignment script was thus broken down into one main script and seven sub-scripts stored in external files as shown in the following list:

- Highway_Assignment_Parallel.bat
  - Highway_Assignment_Parallel.s
    - Highway_Assignment_Parallel_0_SummarizeTolls.s
    - Highway_Assignment_Parallel_1_ReadSeedTolls.s
    - Highway_Assignment_Parallel_2a_ApplyVOT_and_Split_intoToll_NonToll.s
    - Highway_Assignment_Parallel_2b_PerformHighwayAssignment.s
    - Highway_Assignment_Parallel_3_CalculateRestrainedFinalVolSpdVC.s
    - Highway_Assignment_Parallel_4_SummarizeAndAdjustTolls.s
    - Highway_Assignment_Parallel_5_TollEvalTerminationCheck.s

The final integrated assignment procedure starts with populating the network with the seed tolls, the latest tolls from last toll-setting iteration or the latest tolls from the last speed feedback iteration, depending on the settings in the model-steps batch file. The next step writes out the recently encoded tolls for testing purposes. The toll/no-toll skims are prepared next and the trips are split into toll and no-toll trips using a toll choice distribution curve. This split is performed only once per toll-setting iteration so as not to affect the highway assignment convergence. Next, two single multi-class assignments are performed using these trip tables. Following the traffic assignment, the restrained speeds are computed and adjusted on the network. After this stage, the loaded network is read to adjust the tolls for each toll group and the adjusted tolls are evaluated against the old tolls and convergence criteria to determine if further iterations are required.

The traffic assignment type remains unchanged as Bi-conjugate Frank-Wolfe, however, based on MWCOG's findings with "progressive" convergence criteria involving testing different convergence thresholds for the traffic assignments, the convergence criteria was modified to include progressively stringent convergence (relative-gap) criteria ranging from a threshold relative-gap/maximum-iterations of 0.01/100 for the pump-prime iteration to 0.0001/1000 for the final 'i4' iteration.

These highway assignment scripts are shown in Figure 4-20, Figure 4-21, Figure 4-22, Figure 4-23, Figure 4-24, Figure 4-25, Figure 4-26, Figure 4-27, and Figure 4-28.

The following is a summary of the controls/variables that the user can specify:

1. DO_TOLL_SETTING = '%_perform_toll_setting_%'
   This parameter is specified in the model-steps batch file and allows the user to turn-off the toll-setting procedure. When turned-off, the seed tolls in the inputs folder are used to populate the tolls on the network links.

2. LOWER_RELGAP_FOR_TOLL_SETTING = '%_lower_relgap_for_toll_setting_%'
   This parameter is specified in the model-steps batch file and allows the user to optionally lower the threshold for assignment convergence during toll-setting iterations. In this mode, after the

toll-setting has finished, one final assignment with the standard convergence threshold is performed using the final tolls.

3. USE_LAST_TOLLS = '%_use_last_tolls_%'
   This parameter is specified in the model-steps batch file and allows for using "evolving" tolls by using the latest tolls from the last speed-feedback iteration as the starting tolls in the current speed-feedback iteration.

4. APPLY_VOT_TO_SPLIT_VTT = '%_use_vot_distrib_in_toll_setting_%'
   This parameter is also specified in the model-steps batch file and allows the user to optionally turn-off the process of splitting trips into toll and no-toll trip tables. When turned off, all trips will have access to toll facilities.

5. ALLOW_EXCLUSIVE_TOLL_REDUCTION_LOOPS = 0
   This parameter is specified only in the main highway assignment script and is recommended to be set to zero for normal applications. Under normal application (set to zero), the toll adjustment functions adjust the tolls both up and down based on the relationship between the current V/C ratios and the threshold V/C ratios. However, when none of the toll groups increase their toll and either all or some of them only lower their toll, the toll-search is considered done when this parameter is set to zero. When it is set to '1', the toll-search will always continue even if the tolls are only decreased. Therefore this key has a major impact on computational efficiency and should only be considered for enhanced toll-search criteria. Note that the criteria for maintaining speeds on HOT lanes assures only a minimum speed corresponding to an upper threshold V/C ratio. Enabling this key would make the toll-search look for a solution in which all toll-groups fall within the threshold V/C ratio range of 0.95 and 1.01 instead of a solution where none of the toll groups exceed the maximum threshold V/C ratio of 1.01. Activating this option will cause the software to maximize the number of vehicles that will use the toll facility during a given time period.

The current MWCOG assignment procedure uses Cube Cluster's Multi-step Distributed Processing (MDP) in addition the Cube Cluster's Intra-step Distributed Processing (IDP). The MDP is designed to run logical blocks of code (model steps) in parallel typically by utilizing one or more similar compute nodes, where each node may or may not have multiple processing cores. The IDP, on the other hand is designed to run a single block of code (model step) efficiently by utilizing multiple cores on a given node. Both MDP and IDP can be used together for efficiency, however, when only one compute node is used (a single physical machine), typically the number of processing cores for IDP is reduced to accommodate MDP on the same node to avoid overloading the node. Hence, when using a single compute node, the most suitable use of MDP is in the situations where multiple blocks of code that take approximately the same processing time (to avoid un-necessary wait) need to be run in parallel.

In theory, the use of IDP should not change modeled results. In real-world tests, however, MWCOG staff found that the use of IDP did, in fact, change the estimated VMT coming out of the travel model.[7] As part of a series of test conducted in 2011, MWCOG staff conducted two model runs: 1) a year-2007 traffic assignment with IDP using 4 cores; and 2) a year-2007 traffic assignment without IDP (i.e., one core). COG/TPB staff then calculated the VMT difference between the two runs at the regional level, the jurisdiction level, and the link level. At the regional level, the use of IDP had almost no effect on modeled results – it resulted in only a 1/100th to 3/100ths of a percent drop in estimated VMT (slide 25). At the jurisdiction level, the use of IDP also resulted in almost no difference in estimated VMT – the difference was as large as 9/100ths of a percent for some jurisdictions (slide 27). At the link level, by contrast, the use of IDP resulted in a number of cases where the VMT difference was above 20% (slide 29). Fortunately, the links with the larger volume differences were lower-class facilities (e.g., not freeways). Both runs were done as part of the full travel model and both were done using Cube Voyager/Cluster version 5.1.2. Newer versions of Cube Voyager/Cluster are now available (e.g., MWCOG is now using 6.0, and version 6.1 now available), but MWCOG has not re-tried the sensitivity test with the newer versions. At any rate, after MWCOG's tests in 2011, Citilabs updated its documentation to note the occurrence of these differences: "Use of Cluster can have a very small effect on volumes generated by the HIGHWAY program. During the ADJUST phase, when iteration volumes are combined, the final assigned volumes might vary slightly over different numbers of cluster nodes"[8]

In Build 16 of the Version 2.3 Travel Model, MWCOG staff added IDP to the highway assignment script. Staff set the travel model up to use four cores, and, based on the findings of the aforementioned tests, staff recommended that users who wanted to replicate MWCOG results also use four cores. In Builds 20 through 24 of the Version 2.3 Travel Model, MWCOG added IDP to other modeling steps, such as MFARE2.s, Time-of-Day.s, and the transit skimming scripts. In 2012, MWCOG asked for AECOM's assistance to further reduce model run times. AECOM suggested model changes that introduced MDP to the modeling chain. Now, in addition to using four cores for IDP traffic assignment, the use of MDP allowed two traffic assignments to run in parallel (thus, 8 cores would be in use, but only 4 in each of the two IDP sessions). MWCOG staff incorporated these AECOM recommendations into Build 40 of the Version 2.3 Travel Model. The latest version of MWCOG's developmental travel model (Build 52) also incorporates these same changes (use of both IDP and MDP in the model). The Version 2.3.52 Travel Model has just been used in the latest air quality conformity determination. Presuming the TPB approves the conformity determination (scheduled for this summer), the Version 2.3.52 model would become the adopted regional model for use in production work.

Using the version of the travel model that was available at the time (Version 2.3.48), AECOM has revised the highway assignment to use only IDP (i.e., the use of MDP was dropped). Since typical model

---

[7] Ronald Milone and Mark S. Moran, "TPB Version 2.3 Travel Model on the 3,722-TAZ Area System: Status Report" (presented at the May 20, 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., May 20, 2011).

[8] p. 1009, Citilabs, Inc. *Cube Voyager Reference Guide, Version 6.0.2*. Citilabs, Inc., July 26, 2012.

applications are performed with a single machine (with multiple cores), this change does not lose any generality and potentially boosts computational efficiency. To illustrate this, consider the current model application on an 8 core machine, running 4 cores each (IDP) for AM and PM assignments in parallel (MDP). In the event that one of these assignments finishes sooner than the other, the other assignment would continue to run with only 4 cores whereas now it could use all 8 cores. The new approach does not do parallel assignments of time periods, but instead processes them in sequence while making all cores available during each assignment. So the AM assignment would use all 8 cores followed by PM assignment using all 8 cores, etc. and thereby always using all available cores to their full potential. The user however, will need to recognize that the performance improvements beyond 10 cores diminish quickly.

Figure 4-20: Modified Portion of Highway_Assignment_Parallel.s

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

IntraStepProcID   = '%MWCOGClusterID%'     ; Cluster Process ID
IntraStepProcList = '%HWYsubnode%'         ; Cluster Process List or Nodes (subnodes)

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

iter        = '%_iter_%'
prev        = '%_prev_%'
;
; Input and Support Files
;
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
;
; Input Network Depends on Previous Speed-Feedback
;
IF (iter   == 'pp')
   INPNET   = 'ZONEHWY.NET'
   NON_PP   = ';;-- no use in pp iter --;;'
ELSE
   INPNET   = '%_prev_%_HWY.NET'
   NON_PP   = ' '
ENDIF
;
; Parameters
;
;
; This is set in the wrapper batch file; a value of '1' enables toll setting, any other value disables it
;
DO_TOLL_SETTING                      = '%_perform_toll_setting_%'
LOWER_RELGAP_FOR_TOLL_SETTING        = '%_lower_relgap_for_toll_setting_%'    ;
USE_LAST_TOLLS                       = '%_use_last_tolls_%'                   ;
APPLY_VOT_TO_SPLIT_VTT               = '%_use_vot_distrib_in_toll_setting_%' ;
ALLOW_EXCLUSIVE_TOLL_REDUCTION_LOOPS = 0
; 1 = Allow; 0 = Don't Allow Exclusive Lowered-Toll Loops.
; For computational efficiency set to '0' or disable exclusive toll-reduction toll-setting loops.
; As part of toll adjustment lowered tolls do not exclusively trigger toll-setting loops.
; Enabling this key would allow toll-setting loops when the tolls only reduce.
; Warning: Enabling this key can significantly increase the model processing time!
;
D_FAC        =    0.92174 ;  Deflation factor converting 2011$ --> 2007$
CONV_VAL     =   20.62    ;  20 cents in current 2010 cents
ST           =    3       ;  First toll group
TGRPS        =  134       ;  Last toll group
TOLL_CAP     = 1032       ;  MAX. $10 TOLL
T_NUM        =  132       ;  Total number of toll groups
```

```
TOLL_FLR_PK  =   20       ; Minimum Toll in dollars/mile by time-period
TOLL_FLR_OP  =   15       ; Minimum Toll in dollars/mile by time-period

Min_VC       =    0.9500  ; Threshold V/Cs for variably priced facilities
Max_VC       =    1.0100  ; Threshold V/Cs for variably priced facilities

TOLL_RES     =    0.0001  ; Resolution of Tolls During Evaluation; Enter in cents/mile; ForRealism keep in
integers
                          ; ForComputationalEfficiency keep at 4 decimals.
VC_RES       =    2       ; Resolution of Tolls During Evaluation; Enter in number of decimals
                          ; ForComputationalEfficiency & Realism Round to 2 decimals

HOT_LAMBDA1  =   15.0     ;  12.691  - muiltiplier
HOT_LAMBDA2  =   80.0     ;  68.891  - additive

; Choose traffic assignment type, using "enhance=" keyword
;   enhance=0 Frank-Wolfe
;   enhance=1 Conjugate Frank-Wolfe
;   enhance=2 Bi-conjugate Frank-Wolfe
assignType   =    2
;
; Set Progressive Convergence Criteria
;
IF     (iter == 'pp')      ; Pump Prime
    rel_gap = 0.01,   mxIters =  100                 ;  Max. number of user equilibrium (UE) iterations,
ELSEIF (iter == 'i1')      ; i1
    rel_gap = 0.01,   mxIters =  100                 ;  Max. number of user equilibrium (UE) iterations,
ELSEIF (iter == 'i2')      ; i2
    rel_gap = 0.01,   mxIters =  100                 ;  Max. number of user equilibrium (UE) iterations,
ELSEIF (iter == 'i3')      ; i3
    rel_gap = 0.0001, mxIters = 1000                 ;  Max. number of user equilibrium (UE) iterations,
ELSEIF (iter == 'i4')      ; i4
    rel_gap = 0.0001, mxIters = 1000                 ;  Max. number of user equilibrium (UE) iterations,
ELSE                      ; Default
    rel_gap = 0.001,  mxIters =  300                 ;  Max. number of user equilibrium (UE) iterations,
ENDIF
;
; Set lower relgap targets for toll-setting if that optin is enabled
;
low_rel_gap = 0.1, low_mxIters = 20
POST_TOLL_SETTING_ASSIGNMENT_ACTIVE = 0
;
; BEGIN Assignment
;
;;;*****************************************************************************
;;; Step A1: Assign All AM/NT Trip Tables
;;;          (SOV, HOV2, HOV3+, CV, TRUCK & AIRPORT PASSENGER TRIPS)
;;;*****************************************************************************

LOOP PERIOD=1,4                     ; Process AM, PM, MD and NT in sequence
    IF     (PERIOD==1)
        PRD = 'AM', PCTADT = 41.7   ; %_AMPF_%  AM PHF (% of traffic in pk hr of period)
    ELSEIF (PERIOD==2)
        PRD = 'PM', PCTADT = 29.4   ; %_PMPF_%  PM PHF (% of traffic in pk hr of period)
    ELSEIF (PERIOD==3)
        PRD = 'MD', PCTADT = 17.7   ; %_MDPF_%  MD PHF (% of traffic in pk hr of period)
    ELSE
        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)
    ;
    ; Set Source Toll File
    ;
    IF (USE_LAST_TOLLS == '1')
        IF (iter == 'pp')
            SourceTollsFile = 'inputs\SEED_TOLLS_' + PRD + '.TXT'
        ELSE
            SourceTollsFile = 'LATEST_TOLLS_'      + PRD + '.TXT'
        ENDIF
    ELSE
            SourceTollsFile = 'inputs\SEED_TOLLS_' + PRD + '.TXT'
    ENDIF

    ;*********************************************************************************
    ;Step A2: Summarize network outputs and adjust toll rate based on HOT lane speed
    ;*********************************************************************************

    LOOP _TSLoop = 0, 99      ; big outer loop for iteration; MAX 100 ITERATION TO FIND TOLL RATES

        TSprv = _TSLoop
```

```
TSitr = _TSLoop + 1

IF( TSprv == 0 )

    ;*********************************************************************************
    ;Step A3: Read Seed Tolls (Seed tolls & final tolls are in base/calibration year dollars;
    ;no deflation applied)
    ;*********************************************************************************

    RUN PGM=HWYNET
        ZONES=3722

        NETI=@INPNET@
        NETO=@iter@_NewTollsToEval_@PRD@_@TSitr@.NET            ; Output network in TP+ format

        ; Code Stored in External File
        READ FILE=..\Scripts\Highway_Assignment_Parallel_1_ReadSeedTolls.s
    ENDRUN

ELSE

    ;********************************************************************
    ;Step A4: Update TOLLs in Highway Network
    ;********************************************************************

    RUN PGM=HWYNET
        ZONES=3722

        NETI=@iter@_NewTollsAssignedMod_@PRD@_@TSprv@.NET
        NETO=@iter@_NewTollsToEval_@PRD@_@TSitr@.NET        ; Output network in TP+ format

        ; READ TOLL RATE BY TOLLGRP
        lookup name = TOLL@PRD@,
               lookup[1]  = 1,result=1,    ; Toll Group
               lookup[2]  = 1,result=2,    ; Num Links
               lookup[3]  = 1,result=3,    ; Average FTYPE
               lookup[4]  = 1,result=4,    ; Old Toll Rate
               lookup[5]  = 1,result=5,    ; New Toll Rate
               lookup[6]  = 1,result=6,    ; Weighted Speed
               lookup[7]  = 1,result=7,    ; Average VC Ratio
               lookup[8]  = 1,result=8,    ; VMT
               lookup[9]  = 1,result=9,    ; Difference in VC Ratio
               lookup[10] = 1,result=10,   ; Toggle Switch (1=Done, 0=NotDone)
               interpolate=N,fail=0,0,0,file=@iter@_OUT@TSprv@@PRD@_TOLLS.TXT
        ;
        ; Replace @PRD@TOLL
        ;
        ; ALL TOLLS ARE UPDATED REGARDLESS OF FLAGS
        ;
        LOOP _IDX=@ST@,@TGRPS@           ;
            IF(LI.1.tollgrp = TOLL@PRD@(1,_IDX))
                @PRD@TOLL    = TOLL@PRD@(5,_IDX) * LI.1.DISTANCE
                @PRD@TOLL_VP = TOLL@PRD@(5,_IDX) * LI.1.DISTANCE
            ELSE
                @PRD@TOLL    = @PRD@TOLL   ; (TOLL)
                @PRD@TOLL_VP = @PRD@TOLL   ; (TOLL_VP)
            ENDIF
        ENDLOOP

    ENDRUN

ENDIF

;********************************************************************
;Step A5.1: Echo coded tolls
;********************************************************************

RUN PGM=HIGHWAY
    ZONES=3722

    NETI =@iter@_NewTollsToEval_@PRD@_@TSitr@.NET

    ; Code Stored in External File
    READ FILE=..\Scripts\Highway_Assignment_Parallel_0_SummarizeTolls.s
ENDRUN

;********************************************************************
;Step A5: Split trip tables into toll & non-toll
;********************************************************************

IF( TSprv == 0 ) ; No need to re-split tables every toll-setting-iteraton
```

```
    RUN PGM=HIGHWAY  ; Split trip tables
        ZONES=3722

        distributeIntrastep processId=@IntraStepProcID@, ProcessList=@IntraStepProcList@

        FILEI NETI     =  @iter@_NewTollsToEval_@PRD@_@TSitr@.NET             ; TP+ Network

        ; Code Stored in External File
        READ FILE=..\Scripts\Highway_Assignment_Parallel_2a_ApplyVOT_and_Split_intoToll_NonToll.s
    ENDRUN

ENDIF
;
; Label to return and repeat assignment when 'LOWER_RELGAP_FOR_TOLL_SETTING' is set to true or '1'
;
:HwyAssignmentA
;
; Set relative gap
;
IF ( LOWER_RELGAP_FOR_TOLL_SETTING  == '1' )
    IF ( POST_TOLL_SETTING_ASSIGNMENT_ACTIVE == 0 )
        rel_gap = low_rel_gap
        mxIters = low_mxIters
    ENDIF
ENDIF

;**********************************************************************
;Step A6: Perform Highway Assignment
;**********************************************************************

RUN PGM=HIGHWAY  ; Combined traffic assignment
    ZONES=3722

    distributeIntrastep processId=@IntraStepProcID@, ProcessList=@IntraStepProcList@

    FILEI NETI     =  @iter@_NewTollsToEval_@PRD@_@TSitr@.NET                  ; TP+ Network

    ; Code Stored in External File
    READ FILE=..\Scripts\Highway_Assignment_Parallel_2b_PerformHighwayAssignment.s
ENDRUN

;**********************************************************************
;Step A7: Calculate restrained final Volumes, speeds, V/Cs (No MSA)
;**********************************************************************

RUN PGM=HWYNET                   ; Calculate restrained speed/perform MSA volume averaging
    ZONES=3722

    FILEI NETI=@iter@_NewTollsAssigned_@PRD@_@TSitr@.NET    ; input network from highway assignment
    FILEO NETO=@iter@_NewTollsAssignedMod_@PRD@_@TSitr@.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,
                                 V11_1, V12_1, V13_1, V14_1, V15_1, V16_1,
                                 V11T_1,V12T_1,V13T_1,V14T_1,V15T_1,V16T_1,
                            CSPD_1,VDT_1,VHT_1,WRSPD,WFFSPD

    ; Code Stored in External File
    READ FILE=..\Scripts\Highway_Assignment_Parallel_3_CalculateRestrainedFinalVolSpdVC.s
ENDRUN

;*********************************************************************************
;Step A8: Make a copy of the Network
;*********************************************************************************

RUN PGM=HWYNET
    ZONES=3722

    NETI = @iter@_NewTollsAssignedMod_@PRD@_@TSitr@.NET
    NETO = @iter@_temp_@PRD@.NET                         ; Copy of Input
ENDRUN

;*********************************************************************************
;Step A9: Summarize network outputs and adjust toll rate based on HOT lane speed
;*********************************************************************************

RUN PGM=HIGHWAY
    ZONES=3722

    neti =@iter@_NewTollsAssignedMod_@PRD@_@TSitr@.NET
```

```
            ; Code Stored in External File
            READ FILE=..\Scripts\Highway_Assignment_Parallel_4_SummarizeAndAdjustTolls.s
        ENDRUN              ; End of HIGHWAY

        ;**********************************************************************
        ;Step A10: Terminate the process upon conditions
        ;**********************************************************************

        RUN PGM=HWYNET
            ZONES=3722

            NETI=@iter@_NewTollsAssignedMod_@PRD@_@TSitr@.NET           ; Input network in TP+ format

            ; Code Stored in External File
            READ FILE=..\Scripts\Highway_Assignment_Parallel_5_TollEvalTerminationCheck.s
        ENDRUN

        ; Check Whether Toll Setting Loop is Enabled
        IF ( DO_TOLL_SETTING != '1' ) BREAK

        ; If post-toll-setting assignment has been performed, exit
        IF ( POST_TOLL_SETTING_ASSIGNMENT_ACTIVE == 1 ) BREAK

        ; Now Check Whether to Continue Toll Setting Loop
        IF     ( PRD == 'AM' )
            IF( TOLL_SETTING_CLOSURE.FLAG_AM == T_NUM )
                IF ( LOWER_RELGAP_FOR_TOLL_SETTING == '0' ) BREAK
            ELSE
                POST_TOLL_SETTING_ASSIGNMENT_ACTIVE = 1
                GOTO HwyAssignmentA
            ENDIF
        ELSEIF ( PRD == 'MD' )
            IF( TOLL_SETTING_CLOSURE.FLAG_MD == T_NUM )
                IF ( LOWER_RELGAP_FOR_TOLL_SETTING == '0' ) BREAK
            ELSE
                POST_TOLL_SETTING_ASSIGNMENT_ACTIVE = 1
                GOTO HwyAssignmentA
            ENDIF
        ELSEIF ( PRD == 'PM' )
            IF( TOLL_SETTING_CLOSURE.FLAG_PM == T_NUM )
                IF ( LOWER_RELGAP_FOR_TOLL_SETTING == '0' ) BREAK
            ELSE
                POST_TOLL_SETTING_ASSIGNMENT_ACTIVE = 1
                GOTO HwyAssignmentA
            ENDIF
        ELSE
            IF( TOLL_SETTING_CLOSURE.FLAG_NT == T_NUM )
                IF ( LOWER_RELGAP_FOR_TOLL_SETTING == '0' ) BREAK
            ELSE
                POST_TOLL_SETTING_ASSIGNMENT_ACTIVE = 1
                GOTO HwyAssignmentA
            ENDIF
        ENDIF
    ENDLOOP
ENDLOOP
;
; END OF ASSIGNMENTS
;
```

Figure 4-21: Highway_Assignment_Parallel_0_SummarizeTolls.s

```
ARRAY TG_@PRD@VMT  = @TGRPS@, ; TollGrp VMT array              for each time prd
      TG_@PRD@WSP  = @TGRPS@, ; TollGrp Speed-VMT product array for each time prd.
      TG_@PRD@SPD  = @TGRPS@, ; TollGrp Weighted Speed array    for each time prd.
      TG_@PRD@BSP  = @TGRPS@, ;
      TG_FTP       = @TGRPS@, ; ROUTE TYPE
      TG_@PRD@AVC  = @TGRPS@, ; Average VC RATIO
      TG_@PRD@SVC  = @TGRPS@, ; SUM OF VC RATIO
      TG_@PRD@VC   = @TGRPS@, ; VC RATIO
      TG_DST       = @TGRPS@, ; Distance
      TG_@PRD@LMT  = @TGRPS@, ; LIMIT CODE
      TG_@PRD@OTL  = @TGRPS@, ; 'OLD' Toll by time period in cents
      TG_@PRD@NTL  = @TGRPS@, ; 'UPDATED' Toll by time period in cents
      D_@PRD@VC    = @TGRPS@, ; VC DIFFERENCE = LOWER BOUND VC RATIO (=0.6) - AVG. WGT. VC
      TG_@PRD@X    = @TGRPS@, ; New Toll
      TG_@PRD@CNT  = @TGRPS@, ; Count of Links in a Toll Group
      TG_@PRD@TOLL = @TGRPS@, ; Total toll in the Toll Group
```

```
      TG_@PRD@DT   = @TGRPS@  ; TOGGLE SWITCH

zones = 1

; Set up arrays
phase = linkread

     lw.tollgrp          = li.tollgrp          ; TOLLGRP IS REPLACED BY TLGP
     lw.ftype            = li.FTYPE            ; ROUTE TYPE
     lw.distance         = li.distance
     lw.@prd@limit       = li.@PRD@LIMIT       ; LIMIT CODE
     lw.@PRD@toll        = li.@PRD@toll
     lw.Tolls@PRD@spd    = li.Tolls@PRD@spd
     lw.Tolls@PRD@vol    = li.Tolls@PRD@vol
     lw.Tolls@prd@vc     = li.Tolls@prd@vc     ; vc ratio

endphase

;************************************************************************************************************
; estimate average V/C ratio in each tollgroup
;
;                  Sum of VC in each toll group
;  average V/C = ---------------------------------
;                  Number of links in a toll group
;
;************************************************************************************************************

PHASE = ILOOP
    ;
    ; Loop through Toll-Groups to initialize Arrays
    ;
    LOOP k=@ST@,@TGRPS@
        TG_@PRD@VMT[k]  = 0
        TG_@PRD@WSP[k]  = 0
        TG_@PRD@SPD[k]  = 0
        TG_@PRD@BSP[k]  = 0
        TG_FTP[k]       = 0
        TG_@PRD@AVC[k]  = 0
        TG_@PRD@SVC[k]  = 0
        TG_@PRD@VC[k]   = 0
        TG_DST[k]       = 0
        TG_@PRD@LMT[k]  = 0
        TG_@PRD@OTL[k]  = 0
        TG_@PRD@NTL[k]  = 0
        D_@PRD@VC[k]    = 0
        TG_@PRD@X[k]    = 0
        TG_@PRD@CNT[k]  = 0
        TG_@PRD@TOLL[k] = 0
        TG_@PRD@DT[k]   = 0
    ENDLOOP
    ;
    ; Loop through links to populate Toll-Group Arrays
    ;
    LINKLOOP
        IF(lw.tollgrp > 1)
            TG_@PRD@CNT[lw.tollgrp]  = TG_@PRD@CNT[lw.tollgrp] + 1         ; Number of Links in a Toll-Group
            TG_DST[lw.tollgrp]       = TG_DST[lw.tollgrp] + lw.distance    ; SUM of Distances (miles)
            TG_FTP[lw.tollgrp]       = TG_FTP[lw.tollgrp] + lw.ftype       ; SUM of FTYPE
            TG_@PRD@SVC[lw.tollgrp]  = TG_@PRD@SVC[lw.tollgrp]+((lw.Tolls@PRD@vol *
                                       lw.distance)*lw.Tolls@PRD@VC)  ; SUM OF VC RATIO
            TG_@PRD@VMT[lw.tollgrp]  = TG_@PRD@VMT[lw.tollgrp]+(lw.Tolls@PRD@vol  * lw.distance)
            TG_@PRD@BSP[lw.tollgrp]  = TG_@PRD@BSP[lw.tollgrp]+((lw.Tolls@PRD@vol *
                                       lw.distance)*lw.Tolls@PRD@spd)
            ; Total toll (NOT rate) for all links
            TG_@PRD@TOLL[lw.tollgrp] = TG_@PRD@TOLL[lw.tollgrp] + lw.@PRD@toll
            ; Toll Rate (cents/mile)  - No need to deflate tolls here as tolls are only deflated from
              input/seed tolls once
            TG_@PRD@OTL[lw.tollgrp]  = TG_@PRD@TOLL[lw.tollgrp]/TG_DST[lw.tollgrp]
        ENDIF
    ENDLINKLOOP
    ;
    ; Loop through Toll-Groups to Estimate Average V/C Ratio & Average Weighted Speed
    ;
    LOOP k=@ST@,@TGRPS@           ; START - loop for estimating average vc ratio & avg wgt speed
        IF(TG_@PRD@DT[K]=0)
            IF (TG_@PRD@VMT[k] == 0.0)
                TG_@PRD@AVC[k] =  0.0
                TG_@PRD@WSP[k] =  0.0
            ELSE
                TG_@PRD@AVC[k] =  TG_@PRD@SVC[k]  / TG_@PRD@VMT[k]  ; average VMT weighted V/C Ratio
                TG_@PRD@WSP[k] =  TG_@PRD@BSP[k]  / TG_@PRD@VMT[k]  ; average VMT weighted speed
```

```
            ENDIF
        ELSE
            TG_@PRD@AVC[k] =  0
            TG_@PRD@WSP[k] =  0
        ENDIF
        D_@PRD@VC[k] = TG_@PRD@AVC[k] - @Max_VC@
        IF(TG_@PRD@CNT[k]>0) TG_FTP[k] = TG_FTP[k] / TG_@PRD@CNT[k]
        ; average FTYPE (for toll-groups with heterogeneous FTYPEs the answer will be a fraction)
    ENDLOOP                       ; END - loop for estimating average vc ratio & avg wgt speed


;****************************************************************************************************
    ; SUMMARIZE TOLL, SPEED, VMT

    LOOP k=@ST@,@TGRPS@
      Print form=13.0 list = k(10),
                             TG_@PRD@CNT[k](10),
                             TG_FTP[k](10.2),
                             TG_@PRD@OTL[k](10.4),
                             TG_@PRD@X[K](10.4),
                             TG_@PRD@WSP[k](10.4),
                             TG_@PRD@AVC[k](10.4),
                             TG_@PRD@VMT[k](10.2),
                             D_@PRD@VC[K](10.4),
                             TG_@PRD@DT[K](10),
                             file=@iter@_IN@TSitr@@PRD@_TOLLS.TXT
    ENDLOOP
ENDPHASE            ; End of Phase
```

Figure 4-22: Highway_Assignment_Parallel_1_ReadSeedTolls.s

```
; READ TOLL RATE BY TOLLGRP
lookup name = TOLL@PRD@,
       lookup[1]  = 1,result=1,    ; Toll Group
       lookup[2]  = 1,result=2,    ; Num Links
       lookup[3]  = 1,result=3,    ; Average FTYPE
       lookup[4]  = 1,result=4,    ; Old Toll Rate
       lookup[5]  = 1,result=5,    ; New Toll Rate
       lookup[6]  = 1,result=6,    ; Weighted Speed
       lookup[7]  = 1,result=7,    ; Average VC Ratio
       lookup[8]  = 1,result=8,    ; VMT
       lookup[9]  = 1,result=9,    ; Difference in VC Ratio
       lookup[10] = 1,result=10,   ; Toggle Switch (1=Done, 0=NotDone)
       interpolate=N,fail=0,0,0,file=@SourceTollsFile@
;
; Replace @PRD@TOLL
;
; ALL TOLLS ARE UPDATED REGARDLESS OF FLAGS
;
LOOP _IDX=@ST@,@TGRPS@          ;
    IF(LI.1.tollgrp = TOLL@PRD@(1,_IDX))
        @PRD@TOLL    = TOLL@PRD@(5,_IDX) * LI.1.DISTANCE ; Toll deflation is not required here
        @PRD@TOLL_VP = TOLL@PRD@(5,_IDX) * LI.1.DISTANCE ; Toll deflation is not required here
    ELSE
        @PRD@TOLL    = @PRD@TOLL    ; (TOLL)
        @PRD@TOLL_VP = @PRD@TOLL    ; (TOLL_VP)
    ENDIF
ENDLOOP
;
; Add and Initialize Toll Variables
;
Tolls@PRD@spd = 0
Tolls@PRD@vol = 0
Tolls@prd@vc  = 0
;
; print debug list
;
PRINT LIST = 'A=', A(10),
             ', B=', B(10),
             ', TOLLGRP=', LI.1.TOLLGRP(10),
             ', @PRD@TOLL=', @PRD@TOLL(10.2),
             ', LI.1.@PRD@TOLL=', LI.1.@PRD@TOLL(10.2),
```

```
             ', DISTANCE=', LI.1.DISTANCE(10.2),
             ', D_FAC=', @D_FAC@(10.2),
             FILE=Debug_@PRD@_SeedTollsEncoding.txt
```

Figure 4-23: Highway_Assignment_Parallel_2a_ApplyVOT_and_Split_intoToll_NonToll.s

```
;  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 MATO[1]=@iter@_@prd@_Split_Toll_NonToll.VTT, MO = 901-906,911-916,701-706,

 NAME = sov_toll_@prd@,    hov2_toll_@prd@,    hov3_toll_@prd@,
        com_toll_@prd@,    trk_toll_@prd@,     apx_toll_@prd@,
        sov_nontoll_@prd@, hov2_nontoll_@prd@, hov3_nontoll_@prd@,
        com_nontoll_@prd@, trk_nontoll_@prd@,  apx_nontoll_@prd@,
        sov_tsave_@prd@,   hov2_tsave_@prd@,   hov3_tsave_@prd@,
        com_tsave_@prd@,   trk_tsave_@prd@,    apx_tsave_@prd@

FILEO MATO[2]=@iter@_@prd@_TollPathSkims.MAT, MO = 101-106,201-206,301-306,

 NAME = sov_dist_@prd@,    hov2_dist_@prd@,    hov3_dist_@prd@,
        com_dist_@prd@,    trk_dist_@prd@,     apx_dist_@prd@,
        sov_tolls_@prd@,   hov2_tolls_@prd@,   hov3_tolls_@prd@,
        com_tolls_@prd@,   trk_tolls_@prd@,    apx_tolls_@prd@,
        sov_time_@prd@,    hov2_time_@prd@,    hov3_time_@prd@,
        com_time_@prd@,    trk_time_@prd@,     apx_time_@prd@

FILEO MATO[3]=@iter@_@prd@_NonTollPathSkims.MAT, MO = 401-406,501-506,601-606,

 NAME = sov_dist_@prd@,    hov2_dist_@prd@,    hov3_dist_@prd@,
        com_dist_@prd@,    trk_dist_@prd@,     apx_dist_@prd@,
        sov_tolls_@prd@,   hov2_tolls_@prd@,   hov3_tolls_@prd@,
        com_tolls_@prd@,   trk_tolls_@prd@,    apx_tolls_@prd@,
        sov_time_@prd@,    hov2_time_@prd@,    hov3_time_@prd@,
        com_time_@prd@,    trk_time_@prd@,     apx_time_@prd@

;------------------------------------------------------$
;    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


; To simulate a VOT distribution - Mean wage rate $31.10/hr,
; Median wage rate $24.19/hr from BLS 2012 for Metropolitan Washington Region
LOOKUP INTERPOLATE=T, LIST=Y, NAME=TOLLCURVE_COG3110,      ;toll choice 2
LOOKUP[1] =1, RESULT=2,
    R= '0 1',
    '0.5 1',
    '1 1',
    '1.5 1',
    '2 1',
    '2.5 1',
    '3 1',
    '3.5 1',
    '4 0.99',
    '4.5 0.99',
    '5 0.99',
    '5.5 0.98',
    '6 0.98',
    '8 0.94',
    '10 0.89',
    '12 0.84',
    '14 0.78',
    '16 0.72',
    '18 0.66',
    '20 0.61',
    '22 0.55',
    '24 0.5',
    '26 0.46',
    '28 0.42',
    '30 0.38',
    '32 0.35',
    '35 0.3',
    '40 0.24',
    '45 0.19',
    '50 0.15',
    '51 0.15',
    '9999 0'

PHASE=LINKREAD
    ;
    ; Use appropriate loaded speeds forskimming
    ;
               SPEED = SPEEDFOR(LI.@PRD@LANE,LI.SPDCLASS) ; Restrained speed (min) in pp iter
    @NON_PP@   SPEED = LI.@prev@@PRD@SPD                 ; Restrained speed (min) in non-pp iter
    ;
    ; Compute loaded times
    ;
    IF (SPEED = 0)
        T1 = 0
    ELSE
        T1 = (LI.DISTANCE / SPEED) * 60.0   ; (miles / mph) * 60 = minutes
    ENDIF
    ;
    ; 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
    ;$
    ;
    ;   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
    ;                 9     TollGrp > 0
    ;
    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.TOLLGRP > 0) ADDTOGROUP=9  ;; NEW   1-2 = static tolls, >3 = dynamic tolls
    ;
    ;
    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
    ;
    ;
    ;-----------------------------------------------------$
    ;    Tolled Network Costs (Including Entire Network)   $
    ;-----------------------------------------------------$
    ;
    ;        --toll paths--                                                 -- skim dist -
                        -- skim tolls --                                   -- skim time --
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  2,3,5,6,7,   MW[101]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[201]=PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS = 0, MW[301]=PATHTRACE(TIME), NOACCESS = 0 ;  SOV veh
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  3,5,6,7,     MW[102]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[202]=PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS = 0, MW[302]=PATHTRACE(TIME), NOACCESS = 0 ;  HOV 2
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  5,6,7,       MW[103]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[203]=PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS = 0, MW[303]=PATHTRACE(TIME), NOACCESS = 0 ;  HOV 3
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  2,3,5,6,7,   MW[104]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[204]=PATHTRACE(LW.CV@PRD@TOLL ), NOACCESS = 0, MW[304]=PATHTRACE(TIME), NOACCESS = 0 ;  CVs
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  2,3,4,5,6,7, MW[105]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[205]=PATHTRACE(LW.TRK@PRD@TOLL), NOACCESS = 0, MW[305]=PATHTRACE(TIME), NOACCESS = 0 ;  Trucks
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=  6,7,         MW[106]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[206]=PATHTRACE(LW.APX@PRD@TOLL), NOACCESS = 0, MW[306]=PATHTRACE(TIME), NOACCESS = 0 ;  Airport
    ;
    ;
    ;-----------------------------------------------------$
    ;    Free Network Costs (Excluding Toll Links)         $
    ;-----------------------------------------------------$
    ;
    ;
    ;        --non-toll paths--                                              -- skim dist -
                         -- skim tolls (should be zero) --                             -- skim time --
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,2,3,5,6,7,   MW[401]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[501]=PATHTRACE(LW.SOV@PRD@TOLL), NOACCESS = 0, MW[601]=PATHTRACE(TIME), NOACCESS = 0 ;  SOV veh
    PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,3,5,6,7,     MW[402]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
         MW[502]=PATHTRACE(LW.HV2@PRD@TOLL), NOACCESS = 0, MW[602]=PATHTRACE(TIME), NOACCESS = 0 ;  HOV 2
```

```
PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,5,6,7,       MW[403]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
     MW[503]=PATHTRACE(LW.HV3@PRD@TOLL), NOACCESS = 0, MW[603]=PATHTRACE(TIME), NOACCESS = 0 ;  HOV 3
PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,2,3,5,6,7,   MW[404]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
     MW[504]=PATHTRACE(LW.CV@PRD@TOLL ), NOACCESS = 0, MW[604]=PATHTRACE(TIME), NOACCESS = 0 ;  CVs
PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,2,3,4,5,6,7, MW[405]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
     MW[505]=PATHTRACE(LW.TRK@PRD@TOLL), NOACCESS = 0, MW[605]=PATHTRACE(TIME), NOACCESS = 0 ;  Trucks
PATHLOAD PATH=TIME,  EXCLUDEGROUP=9,6,7,         MW[406]=PATHTRACE(LI.DISTANCE), NOACCESS = 0,
     MW[506]=PATHTRACE(LW.APX@PRD@TOLL), NOACCESS = 0, MW[606]=PATHTRACE(TIME), NOACCESS = 0 ;  Airport
;
;
;-----------------------------------------------------$
;    Identify trips that have positive tolled time savings & Compute VOT
;-----------------------------------------------------$
;
;
MW[701] = 0
MW[702] = 0
MW[703] = 0
MW[704] = 0
MW[705] = 0
MW[706] = 0

MW[711] = 0
MW[712] = 0
MW[713] = 0
MW[714] = 0
MW[715] = 0
MW[716] = 0

MW[801] = 0
MW[802] = 0
MW[803] = 0
MW[804] = 0
MW[805] = 0
MW[806] = 0

JLOOP
      ; -- time difference
    MW[701] = ROUND((MW[601] - MW[301]) * 100) / 100   ; SOV veh    time-savings rounded to two decimals
    MW[702] = ROUND((MW[602] - MW[302]) * 100) / 100   ; HOV 2      time-savings rounded to two decimals
    MW[703] = ROUND((MW[603] - MW[303]) * 100) / 100   ; HOV 3      time-savings rounded to two decimals
    MW[704] = ROUND((MW[604] - MW[304]) * 100) / 100   ; CVs        time-savings rounded to two decimals
    MW[705] = ROUND((MW[605] - MW[305]) * 100) / 100   ; Trucks     time-savings rounded to two decimals
    MW[706] = ROUND((MW[606] - MW[306]) * 100) / 100   ; Airport    time-savings rounded to two decimals

      ; -+ve save-  -trips-      -$- (cents/100*occupancy equivalent) /      -hr- (minutes/60)
    IF (MW[701] > 0) MW[711] = ((MW[201]/100)*(SV@PRD@EQM/SV@PRD@EQM)) / (MW[701]/60) ; SOV veh
    IF (MW[702] > 0) MW[712] = ((MW[202]/100)*(H2@PRD@EQM/SV@PRD@EQM)) / (MW[702]/60) ; HOV 2
    IF (MW[703] > 0) MW[713] = ((MW[203]/100)*(H3@PRD@EQM/SV@PRD@EQM)) / (MW[703]/60) ; HOV 3
    IF (MW[704] > 0) MW[714] = ((MW[204]/100)*(CV@PRD@EQM/SV@PRD@EQM)) / (MW[704]/60) ; CVs
    IF (MW[705] > 0) MW[715] = ((MW[205]/100)*(TK@PRD@EQM/SV@PRD@EQM)) / (MW[705]/60) ; Trucks
    IF (MW[706] > 0) MW[716] = ((MW[206]/100)*(AP@PRD@EQM/SV@PRD@EQM)) / (MW[706]/60) ; Airport

ENDJLOOP
;
;
;-----------------------------------------------------$
;     Estimate proportion of trips using toll facility
;-----------------------------------------------------$
;
;
IF ((@DO_TOLL_SETTING@ == 1) && (@APPLY_VOT_TO_SPLIT_VTT@ == 1))
     ;; If toll setting is enabled, compute VOT distribution if enabled
    ; Picks toll-choice probability from VOT distribution for each OD ( Toll($)/Time-Savings(hr) )
                    JLOOP ;                                  -- VOT $/hr --
            IF (MW[701] > 0) MW[801] = TOLLCURVE_COG3110(1, MW[711])  ; SOV veh
            IF (MW[702] > 0) MW[802] = TOLLCURVE_COG3110(1, MW[712])  ; HOV 2
            IF (MW[703] > 0) MW[803] = TOLLCURVE_COG3110(1, MW[713])  ; HOV 3
            IF (MW[704] > 0) MW[804] = TOLLCURVE_COG3110(1, MW[714])  ; CVs
            IF (MW[705] > 0) MW[805] = TOLLCURVE_COG3110(1, MW[715])  ; Trucks
            IF (MW[706] > 0) MW[806] = TOLLCURVE_COG3110(1, MW[716])  ; Airport
        ENDJLOOP
ELSEIF ((@DO_TOLL_SETTING@ == 1) && (@APPLY_VOT_TO_SPLIT_VTT@ == 0))
    ;; If toll setting is enabled, & VOT distribution is disabled, include all trips with
    ;; toll-based time savings in toll trips
        JLOOP ; -+ve save-   -trips-
            IF (MW[701] > 0) MW[801] = 1 ; SOV veh
            IF (MW[702] > 0) MW[802] = 1 ; HOV 2
            IF (MW[703] > 0) MW[803] = 1 ; HOV 3
            IF (MW[704] > 0) MW[804] = 1 ; CVs
            IF (MW[705] > 0) MW[805] = 1 ; Trucks
```

```
                IF (MW[706] > 0) MW[806] = 1 ; Airport
            ENDJLOOP
    ELSE
        ;; If toll setting is disabled, allow all trips to take toll paths; ignore VOT distribution
        ;; (=zero non-toll trips)
            JLOOP
                MW[801] = 1
                MW[802] = 1
                MW[803] = 1
                MW[804] = 1
                MW[805] = 1
                MW[806] = 1
            ENDJLOOP
    ENDIF
    ;
    ;
    ;-----------------------------------------------------$
    ;     Calculate toll and non-toll trips
    ;-----------------------------------------------------$
    ;
    ;
    JLOOP
        ; toll trips
        MW[901] = MI.1.1 * MW[801] ; SOV veh    toll trips
        MW[902] = MI.1.2 * MW[802] ; HOV 2      toll trips
        MW[903] = MI.1.3 * MW[803] ; HOV 3      toll trips
        MW[904] = MI.1.4 * MW[804] ; CVs        toll trips
        MW[905] = MI.1.5 * MW[805] ; Trucks     toll trips
        MW[906] = MI.1.6 * MW[806] ; Airport    toll trips
        ;non-toll trips
        MW[911] = MI.1.1 - MW[901] ; SOV veh    non-toll trips
        MW[912] = MI.1.2 - MW[902] ; HOV 2      non-toll trips
        MW[913] = MI.1.3 - MW[903] ; HOV 3      non-toll trips
        MW[914] = MI.1.4 - MW[904] ; CVs        non-toll trips
        MW[915] = MI.1.5 - MW[905] ; Trucks     non-toll trips
        MW[916] = MI.1.6 - MW[906] ; Airport    non-toll trips
    ENDJLOOP
ENDPHASE
```

Figure 4-24: Highway_Assignment_Parallel_2b_PerformHighwayAssignment.s

```
;  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@_Split_Toll_NonToll.VTT ;  ; FILEI MATI=@iter@_@prd@.VTT ;
;
FILEO NETO=@iter@_NewTollsAssigned_@PRD@_@TSitr@.NET         ; Output loaded network of current
iter/time prd.

PARAMETERS COMBINE=EQUI ENHANCE=@assignType@ ; COMBINE=EQUI requires special attention
especially when using Cube Cluster; see Cube's documentation

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@

;
;
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
    ;  available 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:
    ;; This section was moved up from ILOOP phase to LINKREAD phase -
    ;;Check Cube help about COMBINE=EQUI & LW usage
    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
;                 9     TollGrp > 0
;
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.TOLLGRP > 0) ADDTOGROUP=9  ;; NEW   1-2 = static tolls, >3 = dynamic tolls
;
;
```

```
    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
    ;
    ;-----------------------------------------------------$
    ;     Calculate toll and non-toll trips
    ;-----------------------------------------------------$
    ;
    ; toll trips
    FILLMW MW[101] = MI.1.1  ; SOV veh    toll trips
    FILLMW MW[102] = MI.1.2  ; HOV 2      toll trips
    FILLMW MW[103] = MI.1.3  ; HOV 3      toll trips
    FILLMW MW[104] = MI.1.4  ; CVs        toll trips
    FILLMW MW[105] = MI.1.5  ; Trucks     toll trips
    FILLMW MW[106] = MI.1.6  ; Airport    toll trips
    ; non-toll trips
    FILLMW MW[201] = MI.1.7  ; SOV veh    non-toll trips
    FILLMW MW[202] = MI.1.8  ; HOV 2      non-toll trips
    FILLMW MW[203] = MI.1.9  ; HOV 3      non-toll trips
    FILLMW MW[204] = MI.1.10 ; CVs        non-toll trips
    FILLMW MW[205] = MI.1.11 ; Trucks     non-toll trips
    FILLMW MW[206] = MI.1.12 ; Airport    non-toll trips
    ;
    ;-----------------------------------------------------$
    ;  (TOLL) 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]=MW[101]   ;  SOV veh
    PATHLOAD PATH=LW.HV2@PRD@IMP,  EXCLUDEGROUP=  3,5,6,7,     VOL[2]=MW[102]   ;  HOV 2
    PATHLOAD PATH=LW.HV3@PRD@IMP,  EXCLUDEGROUP=  5,6,7,       VOL[3]=MW[103]   ;  HOV 3
    PATHLOAD PATH=LW.CV@PRD@IMP,   EXCLUDEGROUP=  2,3,5,6,7,   VOL[4]=MW[104]   ;  CVs
    PATHLOAD PATH=LW.TRK@PRD@IMP,  EXCLUDEGROUP=  2,3,4,5,6,7,VOL[5]=MW[105]   ;  Trucks
    PATHLOAD PATH=LW.APX@PRD@IMP,  EXCLUDEGROUP=  6,7,         VOL[6]=MW[106]   ;  Airport
    ;
    ;-----------------------------------------------------$
    ;  (Non-TOLL) Multi-user or multiclass assignment implemented through volume sets (vol[#])
    ;-----------------------------------------------------$
    ;
    PATHLOAD PATH=LW.SOV@PRD@IMP,  EXCLUDEGROUP=9,2,3,5,6,7,   VOL[11]=MW[201]   ;  SOV veh
    PATHLOAD PATH=LW.HV2@PRD@IMP,  EXCLUDEGROUP=9,3,5,6,7,     VOL[12]=MW[202]   ;  HOV 2
    PATHLOAD PATH=LW.HV3@PRD@IMP,  EXCLUDEGROUP=9,5,6,7,       VOL[13]=MW[203]   ;  HOV 3
    PATHLOAD PATH=LW.CV@PRD@IMP,   EXCLUDEGROUP=9,2,3,5,6,7,   VOL[14]=MW[204]   ;  CVs
    PATHLOAD PATH=LW.TRK@PRD@IMP,  EXCLUDEGROUP=9,2,3,4,5,6,7,VOL[15]=MW[205]   ;  Trucks
    PATHLOAD PATH=LW.APX@PRD@IMP,  EXCLUDEGROUP=9,6,7,         VOL[16]=MW[206]   ;  Airport
ENDPHASE

PHASE=ADJUST
    ;
    ;-----------------------------------------------------$
    ;    Functions                                        $
    ;-----------------------------------------------------$
    ;
    ;; Moved down to Adjust based on Cube's documentation about Cluster
    FUNCTION {                                    ; Congested Time (TC)specification:
            V = VOL[1] + VOL[2] + VOL[3] + VOL[4] + VOL[5] + VOL[6] + VOL[11] + VOL[12] +
                VOL[13] + VOL[14] + VOL[15] + VOL[16]
```

```
                 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.
    }
    ;
    ; See Cube documentation on the usage of COMBINE=EQUI & LW variables & COST
    ; functions in cluster environments
    ;
    FUNCTION COST = TIME * (V1+V2+V3+V4+V5+V6+V11+V12+V13+V14+V15+V16) /
                    CmpNumRetNum(V,'=',0,1,V) ; cost function (same as COST=TIME)

    ; Initial Iteration LINK IMPEDANCE (HIGHWAY TIME + Equiv.Toll/Time) by vehicle type here:
       ;; This section was moved up from ILOOP phase to ADJUST phase -
       ;; Check Cube help about COMBINE=EQUI & LW usage
    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.ADJUST
    ENDIF
ENDPHASE

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

Figure 4-25: Highway_Assignment_Parallel_3_CalculateRestrainedFinalVolSpdVC.s

```
;------------------------------------------------------$
;    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
; for all iterations
;
```

```
@iter@@PRD@SOV = V1_1 + V11_1  ;; ORIG V1_1
@iter@@PRD@HV2 = V2_1 + V12_1  ;; ORIG V2_1
@iter@@PRD@HV3 = V3_1 + V13_1  ;; ORIG V3_1
@iter@@PRD@CV  = V4_1 + V14_1  ;; ORIG V4_1
@iter@@PRD@TRK = V5_1 + V15_1  ;; ORIG V5_1
@iter@@PRD@APX = V6_1 + V16_1  ;; ORIG V6_1
;
@iter@@prd@VOL = V_1                                              ;  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
;
; -- Duplicating for Toll Setting --
Tolls@prd@VOL = V_1                                               ;  Final AM/PM Link Volume
Tolls@prd@VC=(@iter@@prd@VOL*(@pctadt@/100.0)/@prd@HRLKCAP)       ;  AM/PM VC  ratio
if (@iter@@prd@VDF > 0)  Tolls@prd@SPD = @iter@@prd@FFSPD / @iter@@prd@VDF
                                                                  ;  AM/PM speed (No queuing)
; -- End Duplicating for Toll Setting --
;
; 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@_load_link.asc
;;
```

Figure 4-26: Highway_Assignment_Parallel_4_SummarizeAndAdjustTolls.s

```
ARRAY TG_@PRD@VMT  = @TGRPS@, ; TollGrp VMT array               for each time prd
      TG_@PRD@WSP  = @TGRPS@, ; TollGrp Speed-VMT product array for each time prd.
      TG_@PRD@SPD  = @TGRPS@, ; TollGrp Weighted Speed array    for each time prd.
      TG_@PRD@BSP  = @TGRPS@, ;
      TG_FTP       = @TGRPS@, ; ROUTE TYPE
      TG_@PRD@AVC  = @TGRPS@, ; Average VC RATIO
      TG_@PRD@SVC  = @TGRPS@, ; SUM OF VC RATIO
      TG_@PRD@VC   = @TGRPS@, ; VC RATIO
      TG_DST       = @TGRPS@, ; Distance
      TG_@PRD@LMT  = @TGRPS@, ; LIMIT CODE
      TG_@PRD@OTL  = @TGRPS@, ; 'OLD' Toll by time period in cents
      TG_@PRD@NTL  = @TGRPS@, ; 'UPDATED' Toll by time period in cents
      D_@PRD@VC    = @TGRPS@, ; VC DIFFERENCE = LOWER BOUND VC RATIO (=0.6) - AVG. WGT. VC
      TG_@PRD@X    = @TGRPS@, ; New Toll
      TG_@PRD@CNT  = @TGRPS@, ; Count of Links in a Toll Group
      TG_@PRD@TOLL = @TGRPS@, ; Total toll in the Toll Group
      TG_@PRD@DT   = @TGRPS@  ; TOGGLE SWITCH

zones = 1

; Set up arrays
phase = linkread

     lw.tollgrp          = li.tollgrp          ; TOLLGRP IS REPLACED BY TLGP
     lw.ftype            = li.FTYPE            ; ROUTE TYPE
     lw.distance         = li.distance
     lw.@prd@limit       = li.@PRD@LIMIT       ; LIMIT CODE
     lw.@PRD@toll        = li.@PRD@toll
     lw.Tolls@PRD@spd    = li.Tolls@PRD@spd
     lw.Tolls@PRD@vol    = li.Tolls@PRD@vol
     lw.Tolls@prd@vc     = li.Tolls@prd@vc     ; vc ratio

endphase

;**********************************************************************************************
; estimate average V/C ratio in each tollgroup
;
;                  Sum of VC in each toll group
;  average V/C = ----------------------------------
;                  Number of links in a toll group
;
;**********************************************************************************************

PHASE = ILOOP

    ;
    ; Loop through Toll-Groups to initialize Arrays
    ;
    LOOP k=@ST@,@TGRPS@
        TG_@PRD@VMT[k]   = 0
        TG_@PRD@WSP[k]   = 0
        TG_@PRD@SPD[k]   = 0
        TG_@PRD@BSP[k]   = 0
        TG_FTP[k]        = 0
        TG_@PRD@AVC[k]   = 0
        TG_@PRD@SVC[k]   = 0
        TG_@PRD@VC[k]    = 0
        TG_DST[k]        = 0
        TG_@PRD@LMT[k]   = 0
        TG_@PRD@OTL[k]   = 0
        TG_@PRD@NTL[k]   = 0
        D_@PRD@VC[k]     = 0
        TG_@PRD@X[k]     = 0
        TG_@PRD@CNT[k]   = 0
        TG_@PRD@TOLL[k]  = 0
        TG_@PRD@DT[k]    = 0
    ENDLOOP
    ;
    ; Loop through links to populate Toll-Group Arrays
```

```
;
LINKLOOP
    IF(lw.tollgrp > 1)
        TG_@PRD@CNT[lw.tollgrp]  = TG_@PRD@CNT[lw.tollgrp] + 1
                                   ; Number of Links in a Toll-Group
        TG_DST[lw.tollgrp]       = TG_DST[lw.tollgrp] + lw.distance
                                   ; SUM of Distances (miles)
        TG_FTP[lw.tollgrp]       = TG_FTP[lw.tollgrp] + lw.ftype  ; SUM of FTYPE
        TG_@PRD@SVC[lw.tollgrp]  = TG_@PRD@SVC[lw.tollgrp]+((lw.Tolls@PRD@vol *
                                   lw.distance)*lw.Tolls@PRD@VC)  ; SUM OF VC RATIO
        TG_@PRD@VMT[lw.tollgrp]  = TG_@PRD@VMT[lw.tollgrp]+(lw.Tolls@PRD@vol * lw.distance)
        TG_@PRD@BSP[lw.tollgrp]  = TG_@PRD@BSP[lw.tollgrp]+((lw.Tolls@PRD@vol *
                                   lw.distance)*lw.Tolls@PRD@spd)
        TG_@PRD@TOLL[lw.tollgrp] = TG_@PRD@TOLL[lw.tollgrp] + lw.@PRD@toll
                                   ; Total toll (NOT rate) for all links
        TG_@PRD@OTL[lw.tollgrp]  = TG_@PRD@TOLL[lw.tollgrp]/TG_DST[lw.tollgrp]
                                   ; Toll Rate (cents/mile)  - No need to deflate tolls here
    ENDIF
ENDLINKLOOP
;
; Loop through Toll-Groups to Estimate Average V/C Ratio & Average Weighted Speed
;
LOOP k=@ST@,@TGRPS@           ; START - loop for estimating average vc ratio & avg wgt speed
    IF(TG_@PRD@DT[K]=0)
        IF (TG_@PRD@VMT[k] == 0.0)
            TG_@PRD@AVC[k] =  0.0
            TG_@PRD@WSP[k] =  0.0
        ELSE
            TG_@PRD@AVC[k] =  TG_@PRD@SVC[k] / TG_@PRD@VMT[k]
                              ; average VMT weighted V/C Ratio
            TG_@PRD@WSP[k] =  TG_@PRD@BSP[k] / TG_@PRD@VMT[k]
                              ; average VMT weighted speed
        ENDIF
    ELSE
        TG_@PRD@AVC[k] =  0
        TG_@PRD@WSP[k] =  0
    ENDIF
    D_@PRD@VC[k] = TG_@PRD@AVC[k] - @Max_VC@
    IF(TG_@PRD@CNT[k]>0) TG_FTP[k] = TG_FTP[k] / TG_@PRD@CNT[k]
   ; average FTYPE (for toll-groups with heterogeneous FTYPEs the answer will be a fraction)
ENDLOOP                       ; END - loop for estimating average vc ratio & avg wgt speed
;
; Toll Setting
;
LOOP K=@ST@,@TGRPS@
    ;
    ; Default -- IMPORTANT -- without this, trials will not be efficient and
    ; some un-used toll groups will make the loop run to the max always!!
    ;
    TG_@PRD@DT[K] = 1 ; (0=TryThisGroupAgain, 1=Don'tThisGroupAgain)

    IF ((ROUND( TG_@PRD@AVC[k] * 10^@VC_RES@ ) / (10^@VC_RES@)) > @Max_VC@)
    ; CASE 1: when VC > 1.01, Increase TOLL (use Rounded VCs for comparison only)
        ; This is a Non-Linear Function
        TG_@PRD@X[K] = TG_@PRD@OTL[K] + (@HOT_LAMBDA1@ * LN(ABS(D_@PRD@VC[k])) +
                       @HOT_LAMBDA2@)                       ; Raise tolls
        ; Try again only if the Toll is non-zero
        IF ( TG_@PRD@X[K] > 0 )  TG_@PRD@DT[K] = 0
                                 ; (0=TryThisGroupAgain, 1=Don'tThisGroupAgain)
    ELSEIF (((ROUND( TG_@PRD@AVC[k] * 10^@VC_RES@ ) / (10^@VC_RES@)) < @Min_VC@) &&
           (TG_@PRD@AVC[K] > 0))
        ; CASE 2: when VC < 0.95, Decrease Toll (use Rounded VCs for comparison only)
        ; This is a Linear Function (better to dampen tolls-reduction function to minimize
        ; oscillation)
        TG_@PRD@X[K] = TG_@PRD@OTL[K] * (TG_@PRD@AVC[K] / @Max_VC@) ; Lower tolls
        ; Lowered tolls will trigger trials depending on a user-specified settting
        IF ( @ALLOW_EXCLUSIVE_TOLL_REDUCTION_LOOPS@ == 1 )
            TG_@PRD@DT[K] = 0 ; (0=TryThisGroupAgain, 1=Don'tThisGroupAgain)
        ENDIF
    ELSE
        ; CASE 3: NO TOLL CHANGE WHEN 0.95 <= VC <= 1.01
```

```
            TG_@PRD@X[K]  = TG_@PRD@OTL[K]
        ENDIF
        ; Round Toll Per Toll-Resolution
        TG_@PRD@X[K] = ROUND( TG_@PRD@X[K] / @TOLL_RES@ ) *  @TOLL_RES@
        ; Check Against Toll Cap/Floor
        IF (TG_@PRD@X[K] > @TOLL_CAP@)
            TG_@PRD@X[K] = @TOLL_CAP@
        ELSEIF ( (@PERIOD@ == 1 || @PERIOD@ == 2) && (TG_@PRD@X[K] < @TOLL_FLR_PK@) &&
                (TG_@PRD@OTL[K] > 0) )
            TG_@PRD@X[K] = @TOLL_FLR_PK@
        ELSEIF ( (@PERIOD@ == 3 || @PERIOD@ == 4) && (TG_@PRD@X[K] < @TOLL_FLR_OP@) &&
                (TG_@PRD@OTL[K] > 0) )
            TG_@PRD@X[K] = @TOLL_FLR_OP@
        ELSE
        ENDIF

        ; After rounding & capping if the tolls are not effectively different;
        ; don't try any more
        IF ( TG_@PRD@X[K] == TG_@PRD@OTL[K] )  TG_@PRD@DT[K] = 1
                             ; (0=TryThisGroupAgain, 1=Don'tThisGroupAgain)
    ENDLOOP

    ;****************************************************************************************
    ; SUMMARIZE TOLL, SPEED, VMT

    LOOP k=@ST@,@TGRPS@
      Print form=13.0 list = k(10),                      ; Toll Group
                             TG_@PRD@CNT[k](10),         ; Num Links
                             TG_FTP[k](10.2),            ; Average FTYPE
                             TG_@PRD@OTL[k](10.4),       ; Old Toll Rate
                             TG_@PRD@X[K](10.4),         ; New Toll Rate
                             TG_@PRD@WSP[k](10.4),       ; Weighted Speed
                             TG_@PRD@AVC[k](10.4),       ; Average VC Ratio
                             TG_@PRD@VMT[k](10.2),       ; VMT
                             D_@PRD@VC[K](10.4),         ; Difference in VC Ratio
                             TG_@PRD@DT[K](10),          ; Toggle Switch (1=Done, 0=NotDone)
                             file=@iter@_OUT@TSitr@@PRD@_TOLLS.TXT
    ENDLOOP

ENDPHASE             ; End of Phase
```

Figure 4-27: Highway_Assignment_Parallel_5_TollEvalTerminationCheck.s

```
;
; READ TOLL RATE BY TOLLGRP
;
lookup name = TOLL@PRD@,
       lookup[1] = 1,result=2,    ; Num Links
       lookup[2] = 1,result=3,    ; Average FTYPE
       lookup[3] = 1,result=4,    ; Old Toll Rate
       lookup[4] = 1,result=5,    ; New Toll Rate
       lookup[5] = 1,result=6,    ; Weighted Speed
       lookup[6] = 1,result=7,    ; Average VC Ratio
       lookup[7] = 1,result=8,    ; VMT
       lookup[8] = 1,result=9,    ; Difference in VC Ratio
       lookup[9] = 1,result=10,   ; Toggle Switch (1=Done, 0=NotDone)
       interpolate=N,fail=0,0,0,file=@iter@_OUT@TSitr@@PRD@_TOLLS.TXT

;
; Mark how many toll groups have finished (count of 'ones')
;
PHASE=SUMMARY

    ;
    ; Check to see if design vc raio is achieved
    ;
    FLAG_@PRD@ = 0
```

```
    LOOP _IDX=@ST@,@TGRPS@ ;
        IF (TOLL@PRD@(9,_IDX) = 1)
            FLAG_@PRD@ = FLAG_@PRD@ + 1
        ELSE
            FLAG_@PRD@ = FLAG_@PRD@ + 0
        ENDIF
    ENDLOOP

    ;
    ; Save the number of toll-groups that do not need any further adjustment in the log file
    ;
    LOG PREFIX=TOLL_SETTING_CLOSURE, VAR=FLAG_@PRD@

    ;
    ; Create a copy of the latest tolls with generic name
    ;
    LOOP _IDX=@ST@,@TGRPS@
      Print form=13.0 list = _IDX(10),                    ; Toll Group
                             TOLL@PRD@(1, _IDX)(10),      ; Num Links
                             TOLL@PRD@(2, _IDX)(10.2),    ; Average FTYPE
                             TOLL@PRD@(3, _IDX)(10.4),    ; Old Toll Rate
                             TOLL@PRD@(4, _IDX)(10.4),    ; New Toll Rate
                             TOLL@PRD@(5, _IDX)(10.4),    ; Weighted Speed
                             TOLL@PRD@(6, _IDX)(10.4),    ; Average VC Ratio
                             TOLL@PRD@(7, _IDX)(10.2),    ; VMT
                             TOLL@PRD@(8, _IDX)(10.4),    ; Difference in VC Ratio
                             TOLL@PRD@(9, _IDX)(10),      ; Toggle Switch (1=Done, 0=NotDone)
                             file=LATEST_TOLLS_@PRD@.TXT
    ENDLOOP

    ;
    ; Record number of toll groups finalized
    ;
    Print form=13.0 list = "Number of Toll Groups Finalized in @PRD@ = ",FLAG_@PRD@(10),
               " out of @T_NUM@"
ENDPHASE
```

Figure 4-28: HOV_Binary_Choice_Generic_Pairwise.s

```
purpi             = 'HBW0'
purpo             = 'HBW'

/*
Environment Variables:
_lambda1_
_lambda2_
_lambda3_
_lambda4_
_lambda5_
_lambda6_
*/

;travel time saving thresholds to shift demand among sov, hov2 and hov3
low_ben_sov_hov2  =  0.5 ; lower limit to travel time saving for sov to shift to hov2
sig_ben_sov_hov2  =  1.0
       ; limit for significant travel time saving (different lambda) for sov to shift to hov2
low_ben_sov_hov3  =  0.5
       ; lower limit to travel time saving for sov to shift to hov3
sig_ben_sov_hov3  =  1.0
       ; limit for significant travel time saving (different lambda) for sov to shift to hov3
low_ben_hov2_hov3  =  0.0
       ; lower limit to travel time saving for hov2 to shift to hov3
sig_ben_hov2_hov3  =  0.5
       ; limit for significant travel time saving (different lambda) for hov2 to shift to hov3

RUN PGM = MATRIX

       PARAMETERS ZONES=3722
```

```
        MATI[1] = %_prev_%_am_sov_MC.skm       ; %_iter_%_am_hov3_hovrestrict.skm
        MATI[2] = %_prev_%_am_hov2_MC.skm      ; %_iter_%_am_hov3_hovrestrict.skm
        MATI[3] = %_prev_%_am_hov3_MC.skm      ; %_iter_%_am_hov3_hovallow.skm

        MATI[4] = %_iter_%_@purpi@_NL_MC.MTT

        FILLMW MW[20] = MI.4.4(11)             ; carry forward transit trips

        MATO = @purpo@_NL_MC.MTT, DEC=3*D, MO=17,18,19,20-30, NAME = SOV, HOV2, HOV3,
               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

        JLOOP

               ; working matrices for sov, hov2 and hov3
               MW[1]  = MI.4.1                    ; background SOV
               MW[2]  = MI.4.2                    ; background HOV2
               MW[3]  = MI.4.3                    ; background HOV3

               MW[4]  = MW[1] + MW[2] + MW[3]     ; background SOV  + HOV2 + HOV3
               MW[5]  = MW[1] + MW[2]             ; background SOV  + HOV2
               MW[6]  = MW[1] + MW[3]             ; background SOV  + HOV3
               MW[7]  = MW[2] + MW[3]             ; background HOV2 + HOV3

               MW[8]  = (MI.1.1 - MI.2.1)
                        ; HOV2 travel time benefit over SOV       (+ve implies savings)
               MW[9]  = (MI.1.1 - MI.3.1)
                        ; HOV3+ travel time benefit over SOV      (+ve implies savings)
               MW[10] = (MI.2.1 - MI.3.1)
                        ; HOV3+ travel time benefit over HOV2     (+ve implies savings)

               ; initialize final matrices
               MW[11] = MW[1]
               MW[12] = MW[2]
               MW[13] = MW[3]

               ;shifting SOV to HOV2
               IF ((%_lambda1_% = 0) && (%_lambda2_% = 0))
                       MW[14] = 0
               ELSEIF ((MW[5]>0) && (MW[8]>@sig_ben_sov_hov2@))
                       MW[12] = min(MW[5], MW[5] * MW[2] * exp(%_lambda1_% * MW[8])/(MW[1] +
                        MW[2] * exp(%_lambda1_% * MW[8])))
                       MW[14] = MW[12] - MW[2]
               ELSEIF ((MW[5]>0) && (MW[8]<@sig_ben_sov_hov2@) && (MW[8] > @low_ben_sov_hov2@))
                       MW[12] = min(MW[5], MW[5] * MW[2] * exp(%_lambda2_% * MW[8])/(MW[1] +
                      MW[2] * exp(%_lambda2_% * MW[8])))
                       MW[14] = MW[12] - MW[2]
               ENDIF

;              MW[1]  = MW[11]
;              MW[2]  = MW[12]


               ;shifting SOV to HOV3
               IF ((%_lambda3_% = 0) && (%_lambda4_% = 0))
                       MW[15] = 0
               ELSEIF ((MW[6]>0) && (MW[9]>@sig_ben_sov_hov3@))
                       MW[13] = min(MW[6], MW[6] * MW[3] * exp(%_lambda3_% * MW[9])/(MW[1] +
                                MW[3] * exp(%_lambda3_% * MW[9])))
                       MW[15] = MW[13] - MW[3]
               ELSEIF ((MW[6]>0) && (MW[9] <@sig_ben_sov_hov3@) && (MW[9] > @low_ben_sov_hov3@))
                       MW[13] = min(MW[6], MW[6] * MW[3] * exp(%_lambda4_% * MW[9])/(MW[1] +
                        MW[3] * exp(%_lambda4_% * MW[9])))
                       MW[15] = MW[13] - MW[3]
               ENDIF
;              MW[1]  = MW[11]
;              MW[3]  = MW[13]

               ;shifting HOV2 to HOV3
               IF ((%_lambda5_% = 0) && (%_lambda6_% = 0))
                       MW[16] = 0
```

```
                ELSEIF ((MW[7]>0) && (MW[10]>@sig_ben_hov2_hov3@))
                       MW[13] = min(MW[7], MW[7] * MW[3] * exp(%_lambda5_% * MW[10])/(MW[2] +
                        MW[3] * exp(%_lambda5_% * MW[10])))
                       MW[16] = MW[13] - MW[3]
                ELSEIF ((MW[7]>0) && (MW[10]<@sig_ben_hov2_hov3@) && (MW[10] >
                           @low_ben_hov2_hov3@))
                       MW[13] = min(MW[7], MW[7] * MW[3] * exp(%_lambda6_% * MW[10])/(MW[2] +
                                MW[3] * exp(%_lambda6_% * MW[10])))
                       MW[16] = MW[13] - MW[3]
                ENDIF

                MW[17] =  MW[1] - MW[14] - MW[15]
                MW[18] =  MW[2] + MW[14] - MW[16]
                MW[19] = MW[3] + MW[15] + MW[16]
;               MW[2]  = MW[12]
;               MW[3]  = MW[13]
        ENDJLOOP
ENDRUN
```

4.9 Performance Tests

As part of this research effort, four performance tests were implemented using the configurations designed primarily to measure the computational efficiency of each. The four tests were grouped into two sets: ‘A’ and ‘B’. The two tests under set A were labeled ‘A-1’ and ‘A-2’ and similarly, the two tests under set ‘B’ were labeled ‘B-1’ and ‘B-2’. Primarily, the set ‘A’ (A-1 & A-2) differed from set ‘B’ (B-1 & B-2) in the level of convergence prior to the iterative toll evaluation and resetting stage. Furthermore, the first test in each set (A-1/B-1) is different from the second test (A-2/B-2) in the type of “seed” or initial tolls used in the speed-feedback loops. A-1/B-1 always used the same starting tolls in each speed-feedback, whereas A-2/B-2 used the latest tolls from the last speed-feedback. Table 4-5 summarizes the key differences between each of the four tests. All of these options were based on an integrated toll-setting and highway assignment approach using single multi-class (bi-conjugate Frank Wolfe) highway assignments. These tests were implemented by customizing the user-settings in the model-steps batch file. The highway assignment process flow charts for test A-1, A-2, B-1 and B-2 are shown in Figure 4-29, Figure 4-30, Figure 4-31 and Figure 4-32, respectively.

These tests were performed with the latest release of Cube – 6.1.0 SP1 which includes enhancements to the Bi-conjugate Frank-Wolfe assignment procedure to ensure a smooth convergence.

Table 4-5 Description of performance tests A-1, A-2, B-1, and B-2

Stopping criterion for UE traffic assignment (relative gap)		**Seed tolls used at the start of the toll setting loop**	
Within toll-setting loop	**Following toll-setting loop**	**Use same seed tolls for each SFB iteration**	**Use seed tolls from previous SFB iteration**
Normal RG threshold	Not used/needed	Test A-1	Test A-2
Relaxed RG threshold	Normal RG threshold	Test B-1	Test B-2

Figure 4-29: Highway Assignment Process in Performance Test A-1

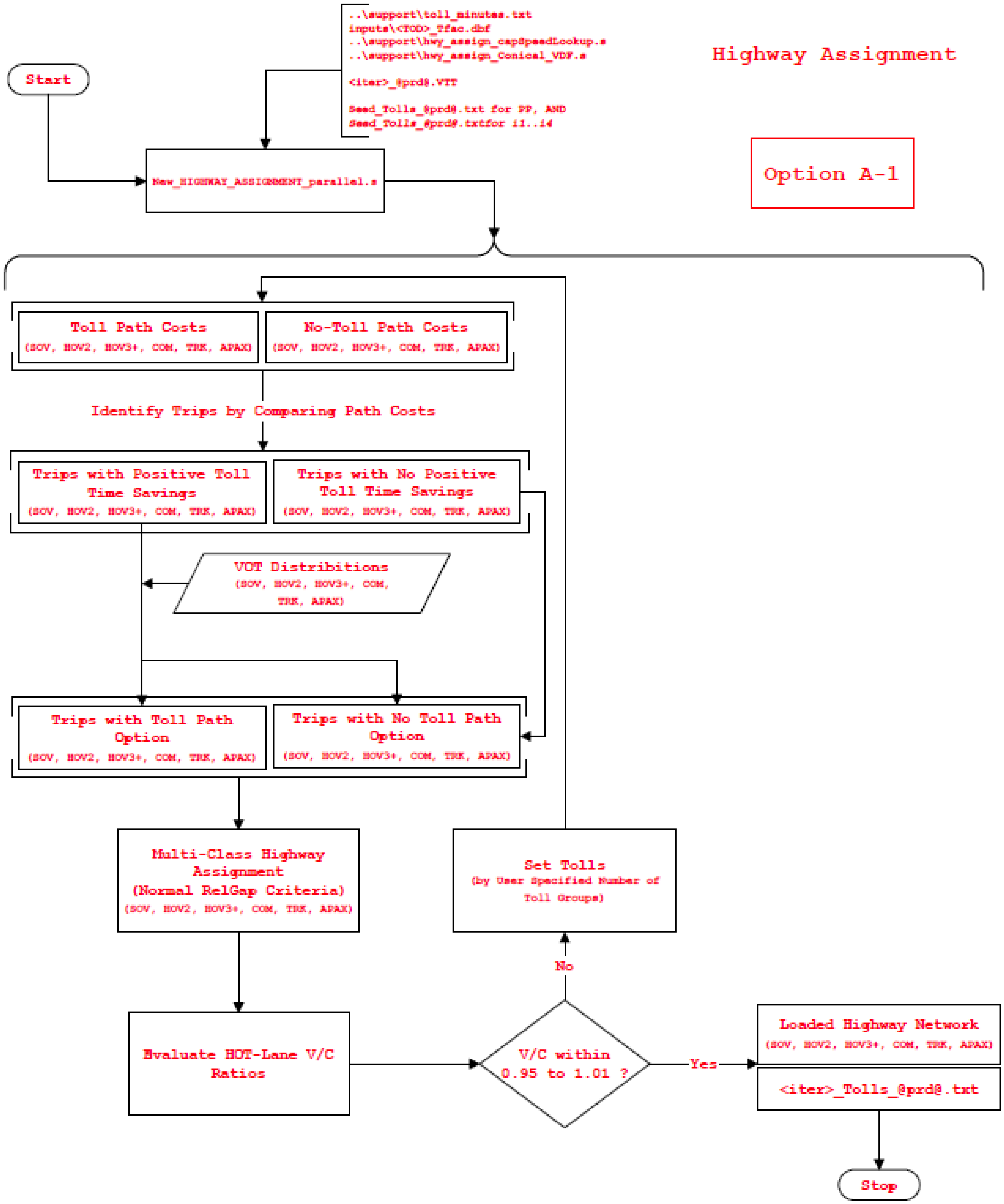


Figure 4-30: Highway Assignment Process in Performance Test A-2

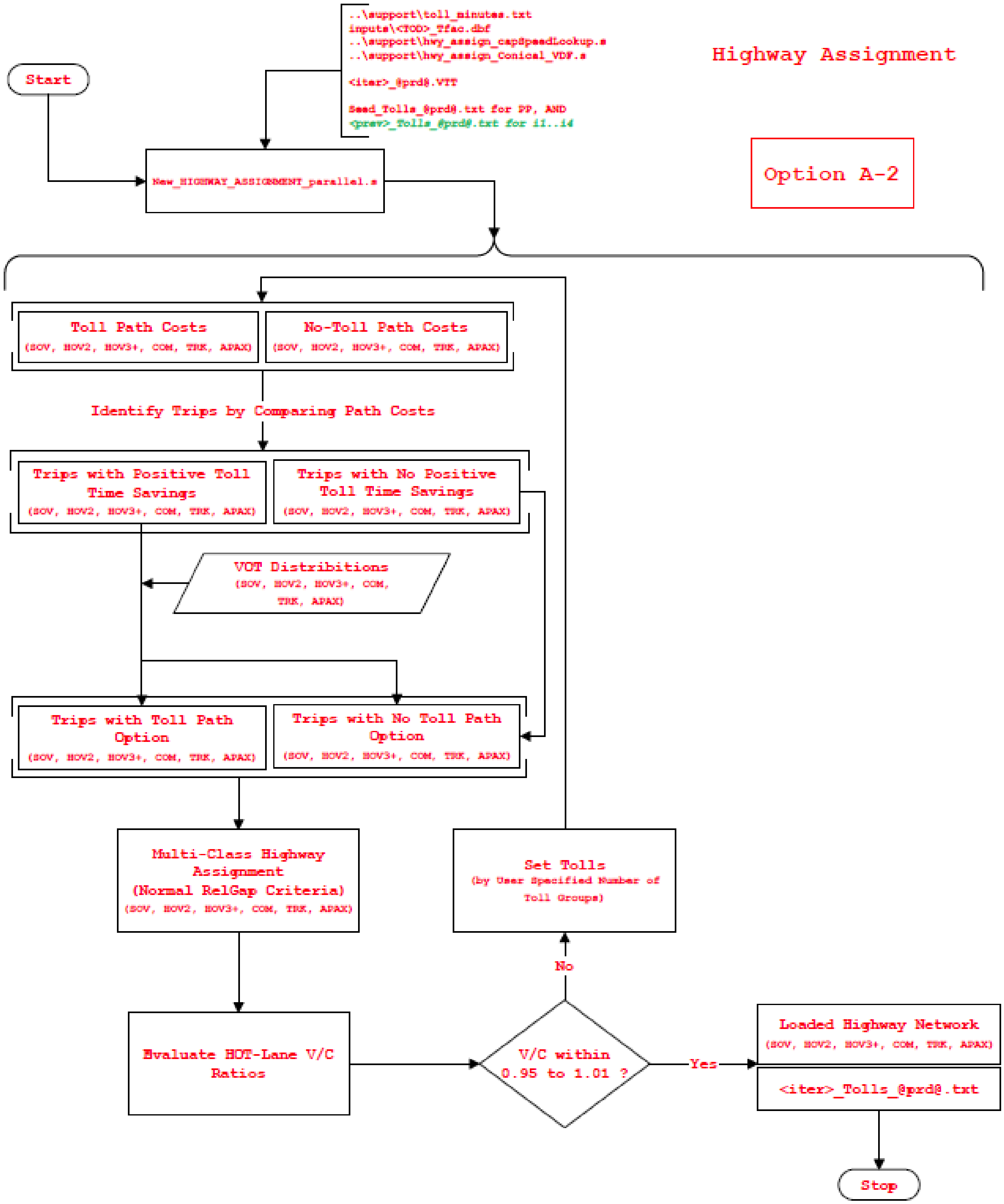


Figure 4-31: Highway Assignment Process in Performance Test B-1

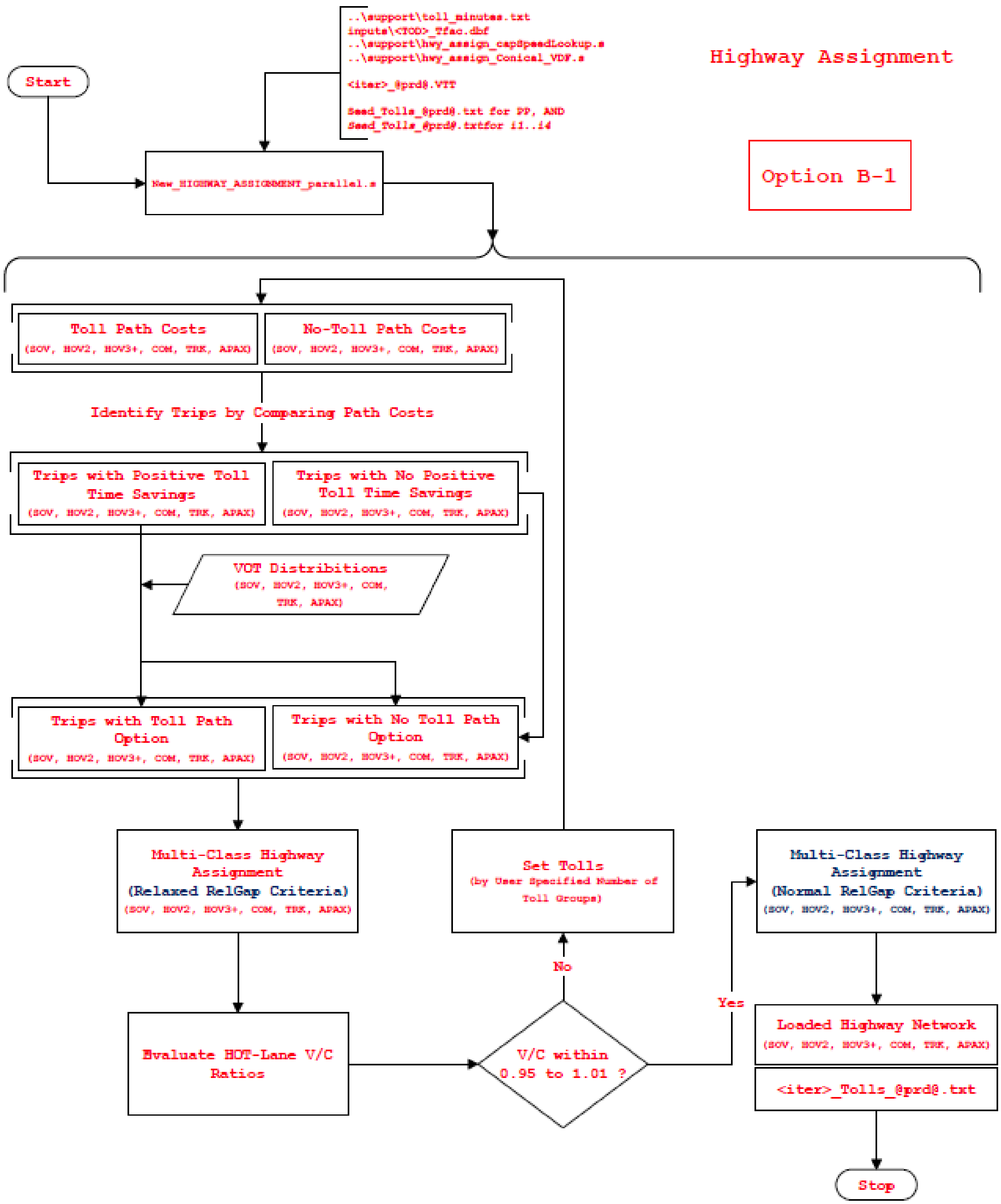
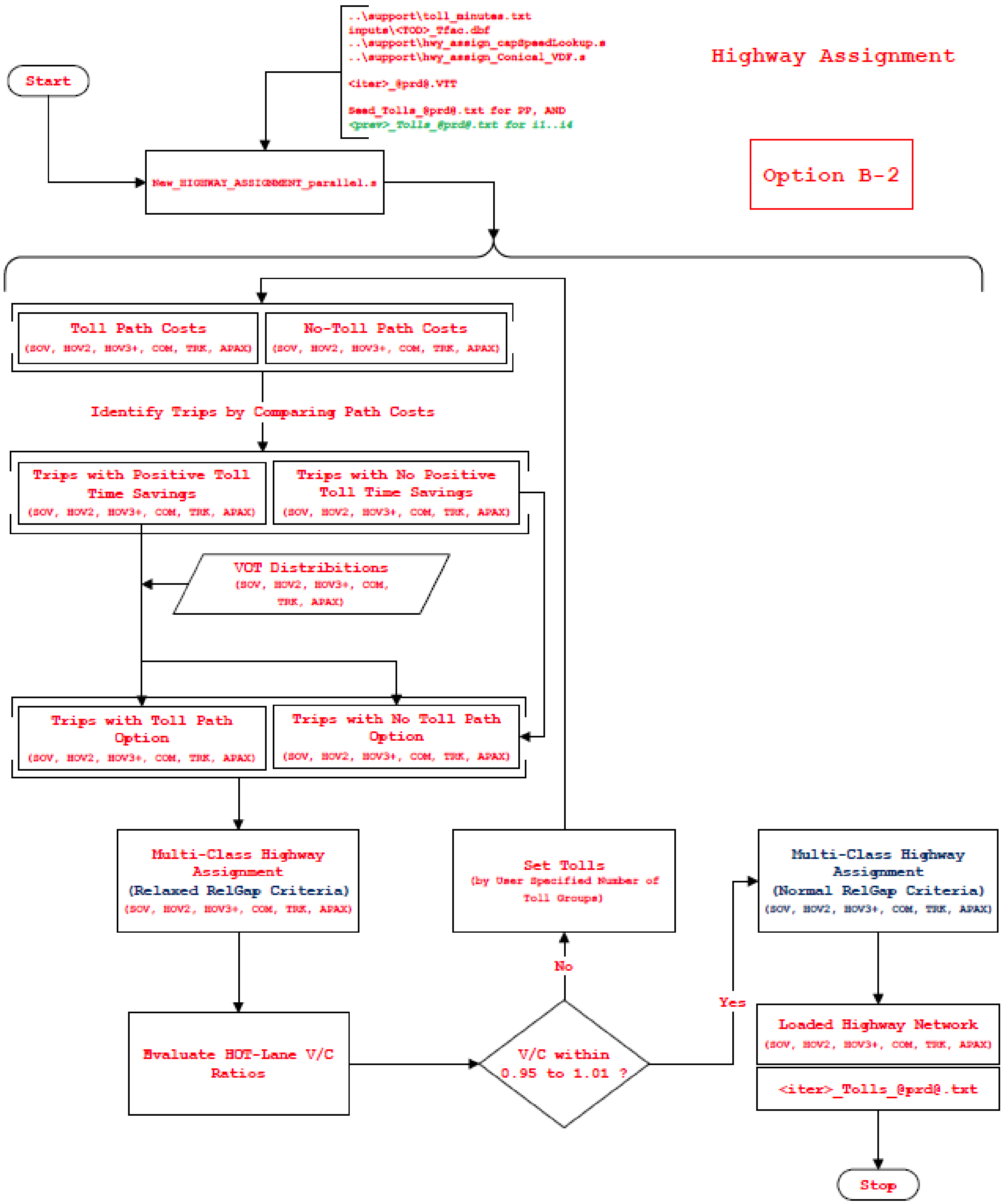


Figure 4-32: Highway Assignment Process in Performance Test B-2



4.10 Tests Results

The performance tests were made simultaneously with progressive convergence criteria and in the same root folder using the changes to Cube Cluster management described earlier on a large 32-core (64-core with Hyper-Threading) machine with Cube 6.1.0 SP1 with 8-cores assigned for each test. Table 4-6 presents the runtime and number of toll iterations in each of the four tests. As can be seen, test B-1 ran for more than a week and had to be terminated in its 10th day of execution. The summaries from B-1 were based on the last successful assignment. In terms of runtime, A-1 and B-2 configurations took the least amount of time and came in under the current MWCOG's HOV-skim replacement (with two-step assignments) model runtime of nearly 48 hours. Note that all these runs include toll-setting. Also note that the set 'A' did not need to adjust tolls whereas in set 'B' the toll adjustments maxed out several times.

Table 4-6: Runtimes from Performance Tests

	A-1	A-2	B-1	B-2
Total Runtime	43 hrs. 49 min.	71 hrs. 30 min.	218 hrs. 00 min.	46 hrs. 27 min.
	AM | MD | PM | NT | (Max = 99)			
pp toll settings	- | - | - | -	- | - | - | -	1 | - | 1 | -	1 | - | 1 | -
I1 toll settings	- | - | - | -	- | - | - | -	99 | 99 | 99 | 99	99 | 99 | 99 | 99
I2 toll settings	- | - | - | -	- | - | - | -	1 | - | 7 | -	- | - | - | -
I3 toll settings	- | - | - | -	- | - | - | -	99 | 99 | 99 | 99	- | - | - | -
I4 toll settings	- | - | - | -	- | - | - | -	99 | 99 | 99 | 99	1 | - | 3 | -
Total toll settings	- | - | - | -	- | - | - | -	299 | 297 | 305 | 297	100 | 99 | 103 | 99

Figure 4-33, Figure 4-34 and Figure 4-35 present all the one hundred and ninety five metrics gathered as part of the "View_From_Space" summary for the four test runs A-1, A-2, B-1 and B-2, along with the "out of the box" runs from MWCOG model Version 2.3.48 for 2010 and 2020_final. A comparison column is also included, which shows the percent difference between test A-1 and the 2020_final model run (the base case). These summaries suggest that, for many of the summary metrics, the values from the four tests bracketed the base case (2020_final).). For example, for the last variable in Figure 4-35 (total average speed or Total_SPD), tests A-1 and A-2 resulted in slightly higher regional speeds than the base (34 mph vs. 32 mph), whereas tests B-1 and B-2 resulted in slightly lower regional speeds (31.6 mph and 31.9 mph vs. 32.0 mph). As for vehicle hours of delay (Total_VHD), tests A-1 and A-2 resulted in about 17 percent fewer hours of delay, whereas tests B-1 and B-2 resulted in slightly higher VHD than the base case (2020_final). Generally, A-1 and A-2 resulted in similar performance across most measures; however, A-2 had slightly lower congestion. Similarly, B-2 had slightly lower congestion when compared to B-1. In all of these tests, the HBW occupancy (HWB_OCC) increased from 1.11 in 2020_final to 1.16 in each of the performance tests, this may be attributed to the combination of the HOV choice and new HOT-lane model assignment process.

Given A-1 performed with the least runtime and resulted in lower congestion when compared B-2, a comparison of the final assigned daily and time-period link volumes was created for A-1 and 2020_final as maps shown in Figure 4-36, Figure 4-37, Figure 4-38, Figure 4-39, and Figure 4-40. Red indicates lower volumes in A-1 and green indicates higher link volumes in A-1 when compare to 2020_final. These maps suggest higher utilization of limited access facilities in A-1 closer to the Beltway and lower away from it.

Figure 4-33: Performance Test Results - View from Space part 1 of 3

	2,010	2020 final	A1 vs 2020 final	A1	A2	B1	B2
1 Households	2,474,631.00	2,824,938.00	0%	2,824,938.00	2,824,938.00	2,824,938.00	2,824,938.00
2 Jobs	3,902,756.00	4,523,477.00	0%	4,523,477.00	4,523,477.00	4,523,477.00	4,523,477.00
3 HH Population	6,486,098.00	7,270,757.00	0%	7,270,757.00	7,270,757.00	7,270,757.00	7,270,757.00
4 HH & GQ Population	6,625,374.00	7,421,032.00	0%	7,421,032.00	7,421,032.00	7,421,032.00	7,421,032.00
5 HH_Inc1	674,023.00	775,754.00	0%	775,754.00	775,754.00	775,753.00	775,753.00
6 HH_Inc2	767,267.00	879,456.00	0%	879,455.00	879,455.00	879,455.00	879,455.00
7 HH_Inc3	510,812.00	582,686.00	0%	582,687.00	582,686.00	582,687.00	582,687.00
8 HH_Inc4	522,557.00	587,078.00	0%	587,078.00	587,078.00	587,078.00	587,077.00
9 HH_All_Incs	2,474,658.00	2,824,973.00	0%	2,824,973.00	2,824,973.00	2,824,973.00	2,824,973.00
10 HH_Siz1	632,604.00	755,554.00	0%	755,555.00	755,555.00	755,554.00	755,554.00
11 HH_Siz2	755,227.00	868,233.00	0%	868,233.00	868,233.00	868,234.00	868,233.00
12 HH_Siz3	432,317.00	485,764.00	0%	485,764.00	485,764.00	485,764.00	485,764.00
13 HH_Siz4	654,510.00	715,422.00	0%	715,421.00	715,422.00	715,421.00	715,421.00
14 HH_ALL_Sizs	2,474,658.00	2,824,973.00	0%	2,824,973.00	2,824,973.00	2,824,973.00	2,824,973.00
15 HH_VA1	214,400.00	278,377.00	1%	281,303.00	281,363.00	279,555.00	278,979.00
16 HH_VA2	748,384.00	883,782.00	0%	885,892.00	885,827.00	885,310.00	885,006.00
17 HH_VA3	943,229.00	1,049,568.00	0%	1,046,944.00	1,046,904.00	1,048,230.00	1,048,751.00
18 HH_VA4	568,645.00	613,246.00	0%	610,834.00	610,879.00	611,878.00	612,237.00
19 HH_All_VAs	2,474,658.00	2,824,973.00	0%	2,824,973.00	2,824,973.00	2,824,973.00	2,824,973.00
20 Inp_HBWAutoDrXI	263,338.00	305,997.00	0%	305,997.00	305,997.00	305,997.00	305,997.00
21 Inp_HBSAutoDrXI	47,086.00	54,632.00	0%	54,632.00	54,632.00	54,632.00	54,632.00
22 Inp_HBOAutoDrXI	133,491.00	156,524.00	0%	156,524.00	156,524.00	156,524.00	156,524.00
23 Inp_NHBAutoDrXI	62,873.00	71,745.00	0%	71,745.00	71,745.00	71,745.00	71,745.00
24 Inp_CVXI	37,214.00	43,703.00	0%	43,703.00	43,703.00	43,703.00	43,703.00
25 Inp_TruckXI	26,360.00	30,195.00	0%	30,195.00	30,195.00	30,195.00	30,195.00
26 Inp_AutoDrXI	544,002.00	632,601.00	0%	632,601.00	632,601.00	632,601.00	632,601.00
27 Inp_HBWAutoDrIX	165,874.00	190,788.00	0%	190,788.00	190,788.00	190,788.00	190,788.00
28 Inp_HBSAutoDrIX	42,363.00	49,124.00	0%	49,124.00	49,124.00	49,124.00	49,124.00
29 Inp_HBOAutoDrIX	189,467.00	222,202.00	0%	222,202.00	222,202.00	222,202.00	222,202.00
30 Inp_NHBAutoDrIX	62,865.00	71,734.00	0%	71,734.00	71,734.00	71,734.00	71,734.00
31 Inp_CVIX	37,209.00	43,701.00	0%	43,701.00	43,701.00	43,701.00	43,701.00
32 Inp_TruckIX	26,360.00	30,195.00	0%	30,195.00	30,195.00	30,195.00	30,195.00
33 Inp_AutoDrIX	497,778.00	577,549.00	0%	577,549.00	577,549.00	577,549.00	577,549.00
34 HBWAutoPsnXI	279,138.00	324,357.00	0%	324,357.00	324,357.00	324,357.00	324,357.00
35 HBSAutoPsnXI	68,275.00	79,216.00	0%	79,216.00	79,216.00	79,216.00	79,216.00
36 HBOAutoPsnXI	217,590.00	255,134.00	0%	255,134.00	255,134.00	255,134.00	255,134.00
37 NHWAutoPsnXI	28,614.00	32,651.00	0%	32,651.00	32,651.00	32,651.00	32,651.00
38 NHOAutoPsnXI	55,643.00	63,494.00	0%	63,494.00	63,494.00	63,494.00	63,494.00
39 AutoPsnXI	649,259.00	754,853.00	0%	754,853.00	754,853.00	754,853.00	754,853.00
40 HBWAutoPsnIX	175,826.00	202,235.00	0%	202,235.00	202,235.00	202,235.00	202,235.00
41 HBSAutoPsnIX	61,426.00	71,230.00	0%	71,230.00	71,230.00	71,230.00	71,230.00
42 HBOAutoPsnIX	308,831.00	362,189.00	0%	362,189.00	362,189.00	362,189.00	362,189.00
43 NHWAutoPsnIX	28,610.00	32,646.00	0%	32,646.00	32,646.00	32,646.00	32,646.00
44 NHOAutoPsnIX	55,636.00	63,485.00	0%	63,485.00	63,485.00	63,485.00	63,485.00
45 AutoPsnIX	630,329.00	731,785.00	0%	731,785.00	731,785.00	731,785.00	731,785.00
46 NonMotr_HBW Trips	136,169.00	173,231.00	0%	173,163.00	173,161.00	173,198.00	173,209.00
47 NonMotr_HBS_Trips	232,498.00	299,787.00	0%	299,657.00	299,652.00	299,729.00	299,753.00
48 NonMotr_HBO_Trips	717,514.00	847,256.00	0%	846,848.00	846,838.00	847,117.00	847,200.00
49 NonMotr_NHW_Trips	417,284.00	470,955.00	0%	470,779.00	470,777.00	470,873.00	470,905.00
50 NonMotr_NHO_Trips	324,600.00	364,006.00	0%	363,800.00	363,798.00	363,910.00	363,947.00
51 NonMotr_ALL_Trips	1,828,064.00	2,155,235.00	0%	2,154,247.00	2,154,225.00	2,154,828.00	2,155,014.00
52 TD_HBWPsn_Trips	4,268,952.00	4,833,433.00	0%	4,830,983.00	4,830,930.00	4,831,909.00	4,832,374.00
53 TD_HBSPsn_Trips	3,204,099.00	3,575,621.00	0%	3,574,423.00	3,574,419.00	3,574,963.00	3,575,211.00
54 TD_HBOPsn_Trips	7,663,023.00	8,531,225.00	0%	8,527,953.00	8,527,919.00	8,529,869.00	8,530,509.00
55 TD_NHWPsn_Trips	1,716,541.00	1,938,723.00	0%	1,938,025.00	1,938,025.00	1,938,404.00	1,938,529.00
56 TD_NHOPsn_Trips	3,265,662.00	3,664,922.00	0%	3,662,905.00	3,662,890.00	3,663,968.00	3,664,334.00
57 TD_ALLPsn_Trips	20,118,277.00	22,543,924.00	0%	22,534,290.00	22,534,183.00	22,539,113.00	22,540,957.00
58 TD_Ext_HBWPsn_Trips	454,941.00	526,573.00	0%	526,574.00	526,575.00	526,576.00	526,573.00
59 TD_Ext_HBSPsn_Trips	129,638.00	150,389.00	0%	150,388.00	150,389.00	150,385.00	150,386.00
60 TD_Ext_HBOPsn_Trips	526,432.00	617,338.00	0%	617,335.00	617,336.00	617,337.00	617,338.00
61 TD_Ext_NHWPsn_Trips	57,105.00	65,187.00	0%	65,185.00	65,184.00	65,186.00	65,186.00
62 TD_Ext_NHOPsn_Trips	111,157.00	126,873.00	0%	126,875.00	126,874.00	126,873.00	126,874.00
63 TD_Ext_ALLPsn_Trips	1,279,274.00	1,486,360.00	0%	1,486,356.00	1,486,358.00	1,486,358.00	1,486,358.00
64 TD_Int_HBWPsn_TD	3,814,011.00	4,306,860.00	0%	4,304,410.00	4,304,355.00	4,305,333.00	4,305,801.00
65 TD_Int_HBSPsn_TD	3,074,461.00	3,425,232.00	0%	3,424,035.00	3,424,030.00	3,424,578.00	3,424,825.00
66 TD_Int_HBOPsn_TD	7,136,591.00	7,913,888.00	0%	7,910,618.00	7,910,582.00	7,912,531.00	7,913,172.00

Figure 4-34: Performance Test Results - View from Space part 2 of 3

	2,010	2020 final	A1 vs 2020 final	A1	A2	B1	B2
67 TD_Int_NHWPsn_TD	1,659,436.00	1,873,535.00	0%	1,872,840.00	1,872,841.00	1,873,218.00	1,873,343.00
68 TD_Int_NHOPsn_TD	3,154,504.00	3,538,049.00	0%	3,536,031.00	3,536,016.00	3,537,095.00	3,537,460.00
69 TD_Int_ALLPsn_TD	18,839,003.00	21,057,564.00	0%	21,047,934.00	21,047,824.00	21,052,754.00	21,054,600.00
70 Ext_HBWAdr	395,632.00	457,918.00	0%	457,918.00	457,915.00	457,925.00	457,917.00
71 Ext_HBSAdr	78,796.00	91,439.00	0%	91,440.00	91,440.00	91,440.00	91,440.00
72 Ext_HBOAdr	326,904.00	383,394.00	0%	383,391.00	383,393.00	383,396.00	383,395.00
73 Ext_NHWAdr	44,559.00	50,874.00	0%	50,872.00	50,873.00	50,873.00	50,872.00
74 Ext_NHOAdr	86,810.00	99,082.00	0%	99,083.00	99,083.00	99,080.00	99,082.00
75 Ext_ALLAdr	932,701.00	1,082,707.00	0%	1,082,703.00	1,082,704.00	1,082,713.00	1,082,706.00
76 Ext_ComVeh	74,416.00	87,405.00	0%	87,406.00	87,405.00	87,405.00	87,406.00
77 Ext_Medium_Trk	23,986.00	27,333.00	0%	27,331.00	27,331.00	27,334.00	27,333.00
78 Ext_Heavy_Trk	28,511.00	32,865.00	0%	32,866.00	32,865.00	32,865.00	32,865.00
79 MC_HBWPsn	3,802,468.00	4,294,186.00	0%	4,291,755.00	4,291,701.00	4,292,622.00	4,293,166.00
80 MC_HBSPsn	3,070,838.00	3,421,343.00	0%	3,420,152.00	3,420,149.00	3,420,646.00	3,420,903.00
81 MC_HBOPsn	7,129,772.00	7,906,967.00	0%	7,903,526.00	7,903,478.00	7,905,407.00	7,906,063.00
82 MC_NHWPsn	1,652,688.00	1,866,467.00	0%	1,865,625.00	1,865,630.00	1,865,990.00	1,866,110.00
83 MC_NHOPsn	3,149,402.00	3,532,800.00	0%	3,530,779.00	3,530,779.00	3,531,831.00	3,532,216.00
84 MC_ALLPsn	18,805,168.00	21,021,763.00	0%	21,011,836.00	21,011,737.00	21,016,495.00	21,018,458.00
85 MC_HBW_Trn	783,806.00	946,815.00	-3%	917,466.00	917,037.00	918,254.00	927,205.00
86 MC_HBS_Trn	24,813.00	28,757.00	0%	28,725.00	28,728.00	28,732.00	28,721.00
87 MC_HBO_Trn	201,016.00	237,286.00	0%	236,876.00	236,848.00	237,094.00	237,113.00
88 MC_NHW_Trn	108,643.00	156,087.00	0%	156,338.00	156,367.00	156,603.00	156,695.00
89 MC_NHO_Trn	41,999.00	56,794.00	0%	56,763.00	56,797.00	56,885.00	56,902.00
90 MC_All_Trn	1,160,278.00	1,425,740.00	-2%	1,396,168.00	1,395,778.00	1,397,568.00	1,406,636.00
91 HBW_TransitPct	20.61	22.05	-3%	21.38	21.37	21.39	21.60
92 HBS_TransitPct	0.81	0.84	0%	0.84	0.84	0.84	0.84
93 HBO_TransitPct	2.82	3.00	0%	3.00	3.00	3.00	3.00
94 NHW_TransitPct	6.57	8.36	0%	8.38	8.38	8.39	8.40
95 NHO_TransitPct	1.33	1.61	0%	1.61	1.61	1.61	1.61
96 ALL_TransitPct	6.17	6.78	-2%	6.64	6.64	6.65	6.69
97 MC_MetroOnly_HBW	370,046.00	472,506.00	-1%	465,955.00	466,904.00	467,507.00	468,232.00
98 MC_MetroOnly_HBS	5,194.00	6,676.00	0%	6,671.00	6,671.00	6,695.00	6,679.00
99 MC_MetroOnly_HBO	72,133.00	92,110.00	0%	91,963.00	91,942.00	92,062.00	92,041.00
100 MC_MetroOnly_NHW	65,139.00	99,911.00	0%	100,063.00	100,065.00	100,208.00	100,250.00
101 MC_MetroOnly_NHO	17,441.00	26,319.00	0%	26,315.00	26,332.00	26,371.00	26,380.00
102 MC_MetroOnly_ALL	529,954.00	697,521.00	-1%	690,967.00	691,914.00	692,844.00	693,581.00
103 MC_Bus_Metro_HBW	174,769.00	182,906.00	-5%	174,391.00	173,855.00	174,080.00	177,526.00
104 MC_Bus_Metro_HBS	2,328.00	2,437.00	0%	2,438.00	2,437.00	2,434.00	2,435.00
105 MC_Bus_Metro_HBO	28,226.00	30,483.00	0%	30,482.00	30,502.00	30,535.00	30,547.00
106 MC_Bus_Metro_NHW	12,617.00	14,869.00	0%	14,925.00	14,951.00	14,987.00	15,015.00
107 MC_Bus_Metro_NHO	4,282.00	5,182.00	0%	5,197.00	5,208.00	5,206.00	5,212.00
108 MC_Bus_Metro_ALL	222,222.00	235,876.00	-4%	227,433.00	226,953.00	227,243.00	230,735.00
109 MC_Comm_Rail_HBW	18,240.00	34,620.00	-12%	30,427.00	30,043.00	29,706.00	31,072.00
110 MC_Comm_Rail_HBS	19.00	14.00	0%	14.00	14.00	14.00	14.00
111 MC_Comm_Rail_HBO	427.00	197.00	1%	198.00	197.00	195.00	194.00
112 MC_Comm_Rail_NHW	289.00	216.00	0%	216.00	216.00	216.00	218.00
113 MC_Comm_Rail_NHO	557.00	292.00	0%	292.00	293.00	289.00	287.00
114 MC_Comm_Rail_ALL	19,532.00	35,339.00	-12%	31,147.00	30,764.00	30,420.00	31,785.00
115 MC_Bus_Only_HBW	220,751.00	256,783.00	-4%	246,693.00	246,235.00	246,961.00	250,375.00
116 MC_Bus_Only_HBS	17,272.00	19,630.00	0%	19,602.00	19,606.00	19,588.00	19,593.00
117 MC_Bus_Only_HBO	100,230.00	114,497.00	0%	114,233.00	114,207.00	114,302.00	114,331.00
118 MC_Bus_Only_NHW	30,597.00	41,091.00	0%	41,134.00	41,135.00	41,191.00	41,213.00
119 MC_Bus_Only_NHO	19,719.00	25,002.00	0%	24,959.00	24,964.00	25,018.00	25,023.00
120 MC_Bus_Only_ALL	388,569.00	457,004.00	-2%	446,621.00	446,148.00	447,060.00	450,535.00
121 MC_HBW_AutoPsn	3,018,662.00	3,347,371.00	1%	3,374,289.00	3,374,663.00	3,374,368.00	3,365,961.00
122 MC_HBS_AutoPsn	3,046,024.00	3,392,586.00	0%	3,391,427.00	3,391,420.00	3,391,914.00	3,392,182.00
123 MC_HBO_AutoPsn	6,928,756.00	7,669,680.00	0%	7,666,650.00	7,666,630.00	7,668,313.00	7,668,950.00
124 MC_NHW_AutoPsn	1,544,045.00	1,710,379.00	0%	1,709,287.00	1,709,263.00	1,709,387.00	1,709,415.00
125 MC_NHO_AutoPsn	3,107,403.00	3,476,006.00	0%	3,474,016.00	3,473,982.00	3,474,946.00	3,475,314.00
126 MC_ALL_AutoPsn	17,644,890.00	19,596,023.00	0%	19,615,668.00	19,615,959.00	19,618,928.00	19,611,821.00
127 Int_HBWAutoDrv	2,763,856.00	3,027,696.00	-4%	2,904,926.00	2,913,309.00	2,921,148.00	2,905,093.00
128 Int_HBSAutoDrv	2,019,979.00	2,244,009.00	0%	2,243,788.00	2,243,779.00	2,244,034.00	2,244,288.00
129 Int_HBOAutoDrv	4,413,900.00	4,871,229.00	0%	4,872,131.00	4,872,126.00	4,872,806.00	4,873,333.00
130 Int_NHWAutoDrv	1,270,930.00	1,373,011.00	0%	1,372,231.00	1,372,152.00	1,371,029.00	1,371,406.00
131 Int_NHOAutoDrv	2,083,742.00	2,279,219.00	0%	2,277,072.00	2,277,044.00	2,277,600.00	2,278,171.00
132 Int_ALLAutoDrv	12,552,406.00	13,795,164.00	-1%	13,670,148.00	13,678,410.00	13,686,616.00	13,672,291.00

Figure 4-35: Performance Test Results - View from Space part 3 of 3

	2,010	2020 final	A1 vs 2020 final	A1	A2	B1	B2
133 HBW_OCC	1.09	1.11	5%	1.16	1.16	1.16	1.16
134 HBS_OCC	1.51	1.51	0%	1.51	1.51	1.51	1.51
135 HBO_OCC	1.57	1.57	0%	1.57	1.57	1.57	1.57
136 NHW_OCC	1.21	1.25	0%	1.25	1.25	1.25	1.25
137 NHO_OCC	1.49	1.53	0%	1.53	1.53	1.53	1.53
138 ALL_OCC	1.41	1.42	1%	1.43	1.43	1.43	1.43
139 Int_CommVeh	1,072,830.00	1,235,954.00	0%	1,235,938.00	1,235,939.00	1,235,908.00	1,235,924.00
140 Int_Med_Truck	467,511.00	530,492.00	0%	530,471.00	530,481.00	530,458.00	530,465.00
141 Int_Hvy_Truck	131,672.00	150,490.00	0%	150,488.00	150,488.00	150,495.00	150,499.00
142 ALL_HBWAdr	3,159,488.00	3,485,614.00	-4%	3,362,844.00	3,371,224.00	3,379,072.00	3,363,010.00
143 ALL_HBSAdr	2,098,774.00	2,335,448.00	0%	2,335,228.00	2,335,219.00	2,335,474.00	2,335,728.00
144 ALL_HBOAdr	4,740,804.00	5,254,623.00	0%	5,255,522.00	5,255,519.00	5,256,202.00	5,256,729.00
145 ALL_NHWAdr	1,315,490.00	1,423,885.00	0%	1,423,102.00	1,423,025.00	1,421,902.00	1,422,278.00
146 ALL_NHOAdr	2,170,552.00	2,378,302.00	0%	2,376,155.00	2,376,127.00	2,376,679.00	2,377,253.00
147 ALL_ALLAdr_MC	13,485,107.00	14,877,871.00	-1%	14,752,851.00	14,761,114.00	14,769,329.00	14,754,997.00
148 ALL_ALLAdr_TOD	13,485,546.00	14,878,196.00	-1%	14,753,344.00	14,761,595.00	14,769,801.00	14,755,489.00
149 ALL_CV	1,147,247.00	1,323,359.00	0%	1,323,343.00	1,323,344.00	1,323,313.00	1,323,330.00
150 ALL_Mtk	491,497.00	557,826.00	0%	557,802.00	557,812.00	557,792.00	557,799.00
151 ALL_Htk	160,183.00	183,355.00	0%	183,355.00	183,353.00	183,360.00	183,365.00
152 THRU_Truck	32,457.00	38,093.00	0%	38,093.00	38,093.00	38,093.00	38,093.00
153 THRU_Auto&CV	41,142.00	48,211.00	0%	48,211.00	48,211.00	48,211.00	48,211.00
154 Taxi_AutoDrv	127,348.00	147,667.00	0%	147,667.00	147,667.00	147,667.00	147,667.00
155 Visitor/Tourist Adr	258,928.00	300,142.00	0%	300,142.00	300,142.00	300,142.00	300,142.00
156 School AutroDrv	289,754.00	330,798.00	0%	330,798.00	330,798.00	330,798.00	330,798.00
157 Final_Medium_Truck	532,147.00	593,271.00	0%	593,173.00	593,190.00	593,160.00	593,231.00
158 Final_Heavy_Truck	165,232.00	183,894.00	0%	183,877.00	183,873.00	183,879.00	183,891.00
159 AirPax_AutoDrv	58,770.00	79,198.00	0%	79,198.00	79,198.00	79,198.00	79,198.00
160 Final_Comm_Veh	1,321,568.00	1,488,447.00	0%	1,488,205.00	1,488,166.00	1,488,314.00	1,488,379.00
161 All_Veh_Trips_MC	16,312,452.00	18,087,593.00	-1%	17,962,216.00	17,970,453.00	17,978,791.00	17,964,607.00
162 All_Veh_Trips_TOD	16,312,891.00	18,087,917.00	-1%	17,962,708.00	17,970,934.00	17,979,263.00	17,965,098.00
163 All_Vehs_Assigned	16,312,891.00	18,087,917.00	-1%	17,962,708.00	17,970,934.00	17,979,263.00	17,965,098.00
164 TRIPS_per_HH	7.61	7.45	0%	7.45	7.45	7.45	7.45
165 TRIPS_per_Pop	2.84	2.84	0%	2.84	2.84	2.84	2.84
166 Fwy_VMT	66,392,238.00	73,673,137.00	-2%	71,941,593.00	72,209,447.00	72,710,863.00	72,266,874.00
167 Maj_VMT	42,696,352.00	61,674,471.00	-2%	60,575,366.00	60,571,872.00	62,537,667.00	62,075,125.00
168 Min_VMT	33,829,392.00	23,362,835.00	-2%	22,877,120.00	22,871,593.00	24,040,864.00	23,838,875.00
169 Col_VMT	9,159,040.00	12,332,766.00	-2%	12,030,586.00	12,005,854.00	13,007,868.00	12,862,084.00
170 Exp_VMT	6,881,434.00	7,424,490.00	-2%	7,286,584.00	7,298,662.00	7,360,819.00	7,311,347.00
171 Rmp_VMT	1,599,687.00	1,686,037.00	-3%	1,630,567.00	1,636,441.00	1,848,760.00	1,824,392.00
172 Total_VMT	160,558,143.00	180,153,736.00	-2%	176,341,816.00	176,593,870.00	181,506,840.00	180,178,697.00
173 VMTperCapita	24.23	24.28	-2%	23.76	23.80	24.46	24.28
174 VMTperHH	64.88	63.77	-2%	62.42	62.51	64.25	63.78
175 VMTperTrip	9.84	9.96	-1%	9.82	9.83	10.10	10.03
176 AM_VMT	30,492,722.92	34,188,277.17	-3%	33,014,133.74	33,089,888.48	34,645,020.19	34,269,460.20
177 MD_VMT	49,245,015.65	55,288,501.52	-1%	54,628,410.93	54,667,392.47	55,594,004.99	55,403,080.10
178 PM_VMT	45,191,183.69	50,812,027.95	-2%	49,864,504.42	49,934,120.14	52,103,129.82	51,593,886.90
179 NT_VMT	35,629,220.63	39,864,929.19	-3%	38,834,766.69	38,902,468.54	39,164,685.42	38,912,270.17
180 Total_VMT	160,558,142.89	180,153,735.84	-2%	176,341,815.79	176,593,869.64	181,506,840.42	180,178,697.37
181 AM_VHT	1,181,380.32	1,393,050.36	-13%	1,214,387.99	1,210,761.95	1,389,143.33	1,353,960.49
182 MD_VHT	1,184,523.74	1,355,087.54	-1%	1,336,421.67	1,337,543.98	1,430,017.04	1,415,756.64
183 PM_VHT	1,710,526.68	2,004,601.47	-9%	1,825,400.62	1,824,687.77	2,066,545.56	2,010,540.51
184 NT_VHT	776,528.53	874,102.19	-3%	850,598.78	852,098.27	866,021.71	860,539.51
185 Total_VHT	4,852,959.27	5,626,841.56	-7%	5,226,809.05	5,225,091.96	5,751,727.65	5,640,797.15
186 AM_VHD	537,407.29	668,596.33	-23%	516,152.83	511,602.48	651,208.17	625,116.01
187 MD_VHD	160,178.33	201,683.89	-3%	196,129.36	196,545.18	262,695.72	252,738.36
188 PM_VHD	752,803.80	924,533.07	-17%	765,295.51	763,787.46	951,239.14	907,053.64
189 NT_VHD	42,506.34	50,538.95	-6%	47,375.38	47,604.56	55,439.48	54,633.14
190 Total_VHD	1,492,895.76	1,845,352.24	-17%	1,524,953.07	1,519,539.68	1,920,582.51	1,839,541.14
191 AM_SPD(VMT/VHT)	25.81	24.54	11%	27.19	27.33	24.94	25.31
192 MD_SPD(VMT/VHT)	41.57	40.80	0%	40.88	40.87	38.88	39.13
193 PM_SPD(VMT/VHT)	26.42	25.35	8%	27.32	27.37	25.21	25.66
194 NT_SPD(VMT/VHT)	45.88	45.61	0%	45.66	45.65	45.22	45.22
195 Total_SPD(VMT/VHT)	33.08	32.02	5%	33.74	33.80	31.56	31.94

Figure 4-36: Highway Volume Difference (Daily)

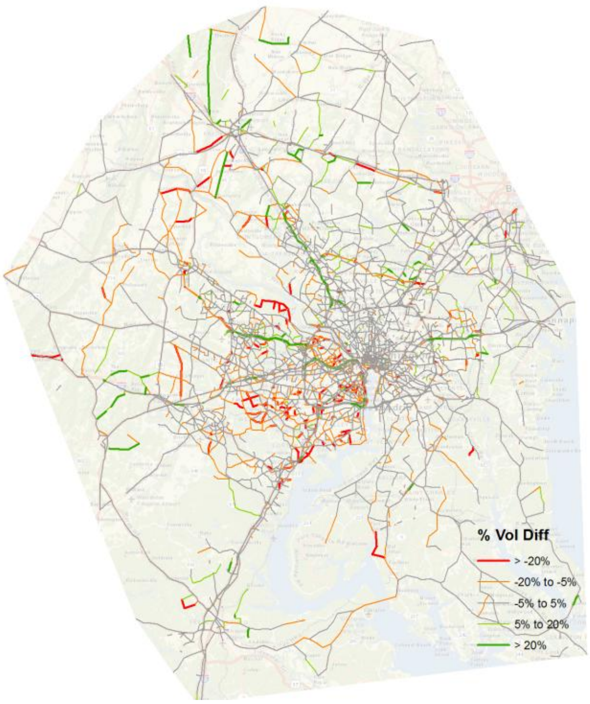


Figure 4-37: Highway Volume Difference (AM)

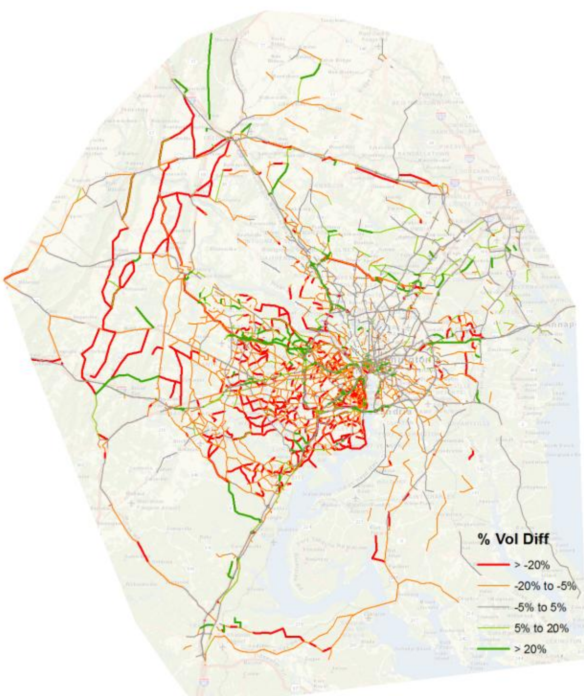


Figure 4-38: Highway Volume Difference (PM)

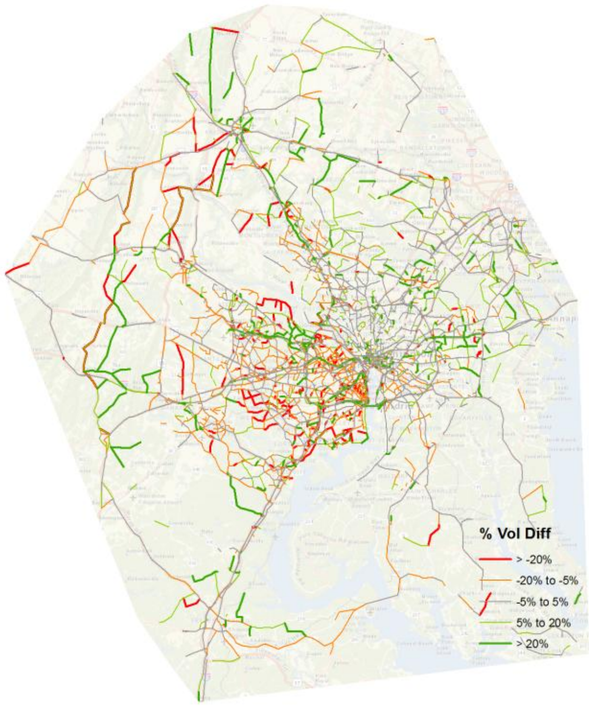


Figure 4-39: Highway Volume Difference (MD)

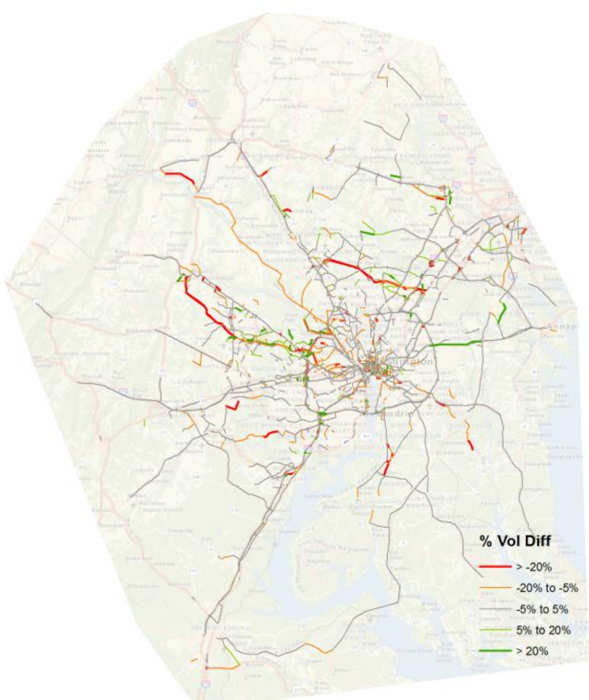
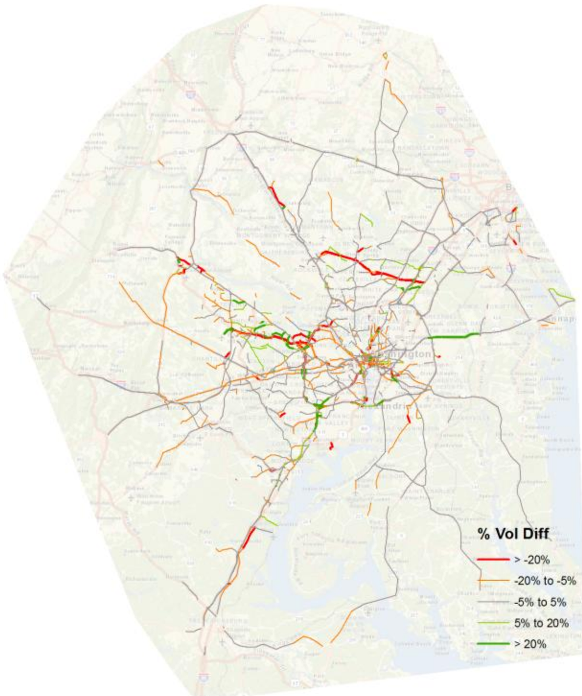


Figure 4-40: Highway Volume Difference (NT)



4.11 Latest Tolls

In order to further develop the seed tolls, another test run similar to A-1 was performed with the setting ALLOW_EXCLUSIVE_TOLL_REDUCTION_LOOPS set to 1. This test allowed for exclusive reduction in tolls. The final tolls from this test are presented in the “latest” columns of Table 4-7 and Table 4-8. These tables also present the original (deflated) tolls from MWCOG’s offline toll setting process in the columns labeled “Orig” and the seed tolls used for the performance tests in the columns labeled “Seed.” A “sparkline” chart (a very small bar or line chart) is presented next to each comparison to graphically represent the tolls across these sources. Each sparkline chart has three bars, representing the values of “Orig,” “Seed,” and “Latest.”

In general, for the peak periods the original tolls are considerably lower than the seed tolls and the latest tolls are even lower than the original tolls. However, there are a few toll groups (for example toll groups 19 and 85 in AM and 30 and 84 in PM) where the original tolls are slightly higher than the seed tolls.

The off-peak periods (MD and NT) behave the same in all three sources and are therefore shown in a single column for each source. The toll values are also almost the same across all sources.

Initial testing and these comparisons indicate that the ending tolls are dependent on the starting point and the configuration of the toll setting process, suggesting that the solution for toll search may not be unique. It bears mentioning that the solutions to MWCOG’s current toll setting process are also not unique. It is generally understood that the user equilibrium (UE) traffic assignment, when adequately converged, results in unique link flows, but not unique route flows, and not unique multi-class link flows. According to Boyce, "consistent forecasts of multiple-class link flows are needed for the evaluation of proposed facilities for which vehicle classes are treated differently, such as toll roads, bridges, HOV lanes and HOT lanes" (p. 1).[9] Similarly, "consistent forecasts of route flows by OD pair are needed to apply methods such as select link analysis, select zone analysis, [and] subarea analysis …, which depend upon route flows" (p. 1). So, it is not expected that either the current MWCOG toll setting process, or the revised toll-setting process described in this chapter, will result in unique solutions. Despite this shortcoming, it is felt that both the current and revised toll setting processes are better than having no model at all.

[9] See, for example, David E. Boyce et al., *Field Test of a Method for Finding Consistent Route Flows and Multiple-Class Link Flows in Road Traffic Assignments*, Travel Model Improvement Program (TMIP) (Washington, D.C.: U.S. Department of Transportation, Federal Highway Administration, March 8, 2010), http://www.transportation.northwestern.edu/docs/research/Boyce_FieldTestConsistentRouteFlows.pdf.

Table 4-7: Toll Comparison, Part 1 of 2

	AM			PM			MD/NT		
Tgroup	Orig	Seed	Latest	Orig	Seed	Latest	Orig	Seed	Latest
3	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0
6	133	225	61	247	346	122	14	15	15
7	19	20	20	19	20	20	14	15	15
8	133	225	61	247	346	122	14	15	15
9	19	20	20	19	20	20	14	15	15
10	19	20	20	19	20	20	14	15	15
11	19	20	20	19	20	20	14	15	15
12	19	20	20	19	20	20	14	15	15
13	86	44	21	19	20	20	14	15	15
14	19	20	20	19	20	20	14	15	15
15	19	20	20	19	20	20	14	15	15
16	19	20	20	19	20	20	14	15	15
17	19	20	20	19	20	20	14	15	15
18	19	20	20	19	20	20	14	15	15
19	143	127	67	19	20	20	14	15	15
20	19	20	20	19	20	20	14	15	15
21	19	20	20	19	20	20	14	15	15
22	19	20	20	19	20	20	14	15	15
23	19	20	20	19	20	20	14	15	15
24	19	20	20	162	152	93	14	15	15
25	19	20	20	19	20	20	14	15	15
26	19	20	20	19	20	20	14	15	15
27	19	20	20	19	20	20	14	15	15
28	19	20	20	19	20	20	14	15	15
29	19	20	20	19	20	20	14	15	15
30	19	20	20	57	20	20	14	15	15
31	19	20	20	19	20	20	14	15	15
32	19	20	20	19	20	20	14	15	15
33	276	463	116	162	254	107	14	15	15
34	19	20	20	19	20	20	14	15	15
35	276	463	116	162	254	107	14	15	15
36	19	20	20	19	20	20	14	15	15
37	276	463	116	162	254	107	14	15	15
38	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0
61	19	20	20	19	20	20	14	15	15
62	0	0	0	0	0	0	0	0	0
63	19	20	20	19	20	20	14	15	15
64	0	0	0	0	0	0	0	0	0
65	19	20	20	19	20	20	14	15	15
66	0	0	0	0	0	0	0	0	0
67	19	20	20	19	20	20	14	15	15
68	0	0	0	0	0	0	0	0	0
69	19	20	20	19	20	20	14	15	15

Table 4-8: Toll Comparison, Part 2 of 2

Tgroup	AM Orig	AM Seed	AM Latest	PM Orig	PM Seed	PM Latest	MD/NT Orig	MD/NT Seed	MD/NT Latest
70	0	0	0	0	0	0	0	0	0
71	19	20	20	19	20	20	14	15	15
72	0	0	0	0	0	0	0	0	0
73	19	20	20	19	20	20	14	15	15
74	0	0	0	0	0	0	0	0	0
75	19	20	20	19	20	20	14	15	15
76	19	20	20	19	20	20	14	15	15
77	19	20	20	19	20	20	14	15	15
78	19	20	20	19	20	20	14	15	15
79	19	20	20	19	20	20	14	15	15
80	19	20	20	19	20	20	14	15	15
81	19	20	20	19	20	20	14	15	15
82	19	20	20	437	623	329	14	15	15
83	19	20	20	19	20	20	14	15	15
84	19	20	20	133	86	45	14	15	15
85	114	20	20	19	20	20	14	15	15
86	19	20	20	86	65	39	14	15	15
87	570	604	291	19	20	20	14	15	15
88	19	20	20	19	20	20	14	15	15
89	133	132	37	19	20	20	14	15	15
90	19	20	20	19	20	20	14	15	15
91	114	61	28	19	20	20	14	15	15
92	19	20	20	19	20	20	14	15	15
93	133	192	48	19	20	20	14	15	15
94	19	20	20	48	45	23	14	15	15
95	133	193	49	19	20	20	14	15	15
96	19	20	20	48	45	23	14	15	15
97	133	233	59	19	20	20	14	15	15
98	19	20	20	48	45	23	14	15	15
99	133	232	54	19	20	20	14	15	15
100	19	20	20	48	45	23	14	15	15
101	67	58	22	19	20	20	14	15	15
102	19	20	20	48	45	23	14	15	15
103	19	20	20	19	20	20	14	15	15
104	19	20	20	19	63	26	14	15	15
105	19	20	20	19	20	20	14	15	15
106	19	20	20	95	145	50	14	15	15
107	133	163	51	19	20	20	14	15	15
108	19	20	20	86	113	46	14	15	15
109	143	171	57	19	20	20	14	15	15
110	19	20	20	19	20	20	14	15	15
111	380	519	193	19	20	20	14	15	15
112	19	20	20	76	23	31	14	15	15
113	19	20	20	19	20	20	14	15	15
114	19	20	20	19	20	20	14	15	15
115	19	20	20	19	20	20	14	15	15
116	19	20	20	19	20	20	14	15	15
117	0	0	0	0	0	0	0	0	0
118	19	20	20	19	20	20	14	15	15
119	0	0	0	0	0	0	0	0	0
120	19	20	20	19	20	20	14	15	15
121	0	0	0	0	0	0	0	0	0
122	19	20	20	19	20	20	14	15	15
123	0	0	0	0	0	0	0	0	0
124	19	20	20	19	20	20	14	15	15
125	0	0	0	0	0	0	0	0	0
126	19	20	20	19	20	20	14	15	15
127	0	0	0	0	0	0	0	0	0
128	19	20	20	19	20	20	14	15	15
129	0	0	0	0	0	0	0	0	0
130	19	20	20	19	20	20	14	15	15
131	0	0	0	0	0	0	0	0	0
132	19	20	20	19	20	20	14	15	15
133	0	0	0	0	0	0	0	0	0
134	19	20	20	19	20	20	14	15	15

4.12 Recommendations

The primary objective of the HOT lane modeling process (preventing degradation of HOV speeds, while using a full multi-class assignment) was achieved along with other objectives such as incorporation of toll-setting and toll-choice in the standard highway assignment process, while moderately reducing the overall model runtime. The following are the recommendations from the HOT lane modeling exercise:

1. Based on the performance tests, the integrated highway assignment configuration A-1 performs the best amongst all configurations and results in the shortest model runtime due to the fewest toll-setting iterations. This configuration also results in relatively higher regional average roadway speeds and correspondingly lower vehicle hours of delay in the region. This configuration is recommended for further evaluation with varied seed tolls and enhanced precision for the toll search parameters.
2. The seed tolls were found to have a significant impact on model runtimes. The seed tolls used in the performance tests were based on preliminary runs. It would be desirable to generate more "seasoned" seed tolls from model runs with enhanced toll search precision. Coding and retaining tolls from the calibration year network (2007 in these tests) in all future year networks is also recommended to simplify the comparisons and avoid unnecessary toll deflation.
3. The final assigned volumes are quite sensitive to the input value-of-time (VOT) distributions, since they define the toll market. These tests used a common VOT distribution for each of the six assignment classes (SOV, HOV2, HOV3+, CV, TRK and APAX) based on an average regional wage distribution. Since the airport passengers and trucks typically have a much higher value of time, compared to other automobile classes, it would be desirable to derive and incorporate distinct VOT distributions for each assignment class based on observed data.
4. Citilabs recommends taking precaution while using bi-conjugate Frank-Wolfe assignment algorithms and the COMBINE = EQUI method of assignment because, in this specification, the link-based variables are not properly updated and therefore may impact the convergence. To address this potential issue, the highway assignment script was reorganized. In addition the latest release from Citilabs: Cube 6.1.0 provides additional features in the highway assignment procedures to smooth convergence while providing optional backwards compatibility with the version of Cube currently used by MWCOG (ver. 6.0.2).
5. MWCOG currently utilizes a total of 134 toll groups (including 2 toll groups for static tolls) covering all the priced roadways. These toll groups are coded to separate continuous links from those connecting to or from ramps. The performance tests indicate that many toll groups seem to behave similarly or appear to be clustered. Figure 4-41 displays the current toll groups by their AM toll value. It suggests that a few toll groups on I-95 and I-495 require a high threshold toll. Consolidating the toll-groups to better conform to the I-95 and I-495 access maps as shown in Figure 4-42 and Figure 4-43 is recommended to enhance the representation and to reduce the number of toll setting iterations and model runtime.

Figure 4-41: AM Toll Value of Existing Toll Groups

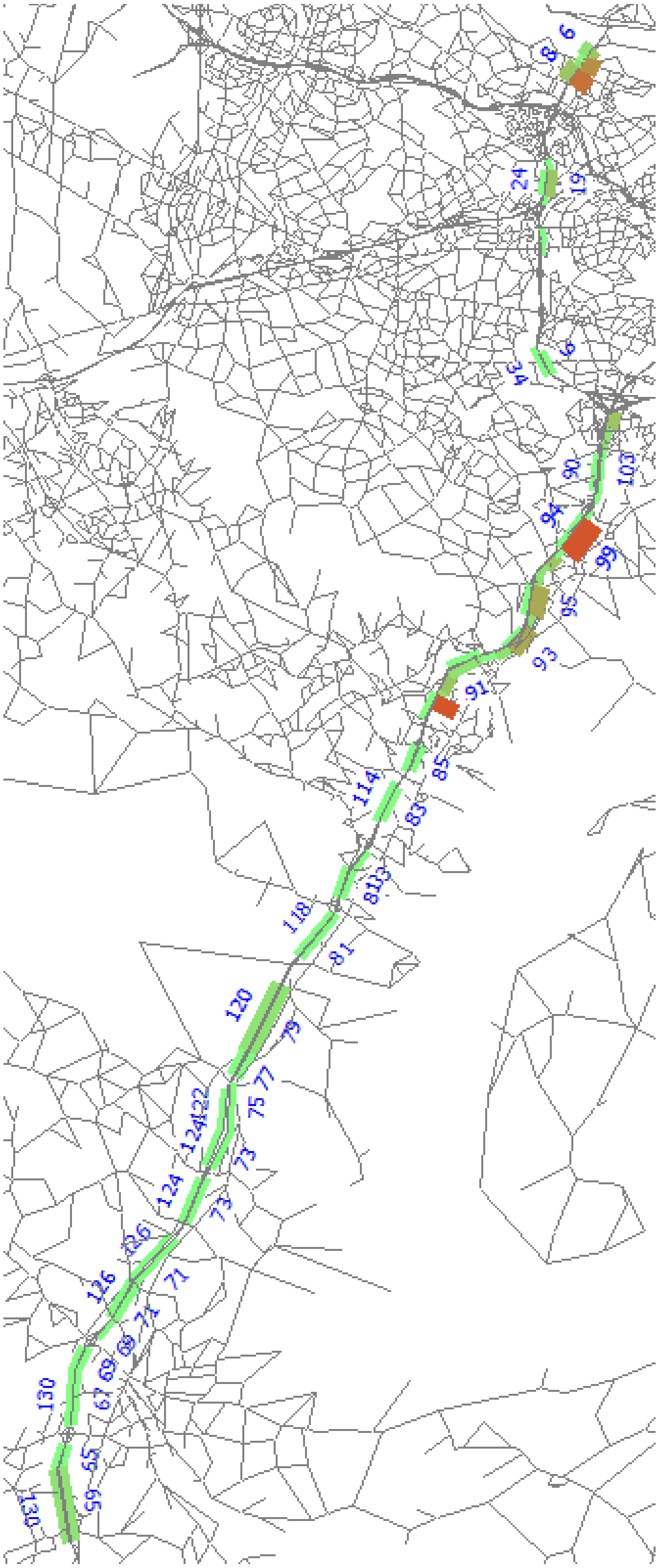
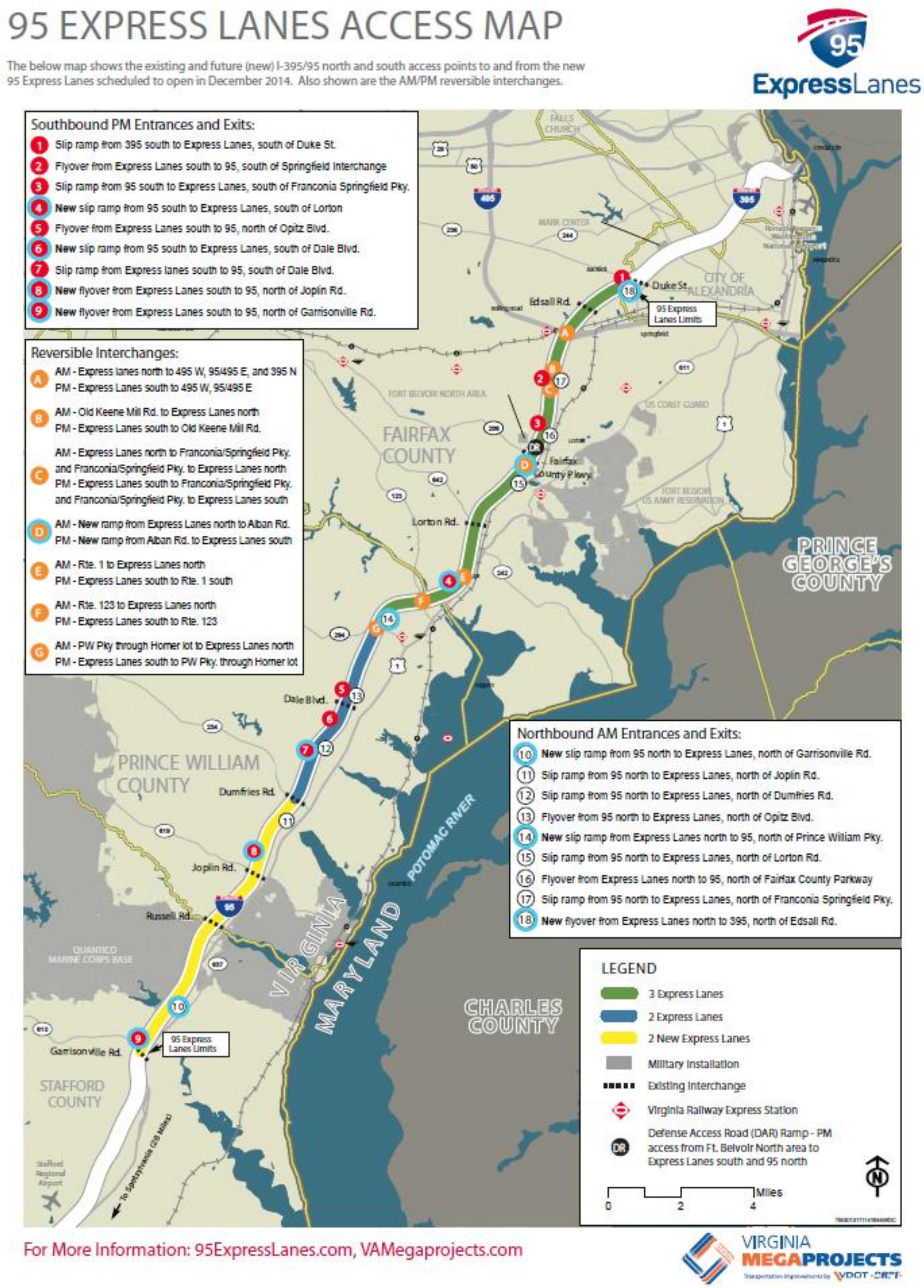


Figure 4-42: I-95 Express Lanes Access Map



Source: 95expresslanes.com (VDOT)

Figure 4-43: I-495 Express Lanes Access Map

Source: 495expresslanes.com (VDOT)

5 Speed Validation – INRIX Data (Task Order 8)

This chapter documents the findings of a literature search on the topic of using observed speed data to make improvements to a regional travel demand model. The work was conducted using a variety of Internet searches and posting a question on the US DOT Travel Model Improvement Program (TMIP) e-mail distribution list (Listserv).

It appears that INRIX is still in a "ramp-up" period with respect to getting their name and services more widely known among transportation planners. They are actively promoting their data products and as more people learn about their products, their data are likely to have more widespread use. Another factor that will limit the use of their products is price. Their pricing schedule appears to be expensive enough so that, at the moment, they are working mainly with state DOTs and the larger MPOs. However, they do have a contract to provide data to the I-95 Corridor Coalition and that is being shared with several agencies. INRIX is also actively marketing their data to the news media, which may well be a more fruitful source of revenue than local planning agencies.

Another issue is that in travel modeling, validating estimated network speeds and travel times has often been considered a "luxury" until recently – something that would be nice to have, but is not necessary. In the past, it has been relatively expensive for agencies to collect their own observed speed data. But it looks like as more of this data becomes available (and the price comes down), more modelers will start using it for network speed validation. This will also be driven by the continued importance of speed in the air quality conformity process, particularly so in the transition from MOBILE6 to MOVES. A related issue is that cost pressures have affected the quantity and quality of the data, reducing its usefulness to travel forecasters.

As a result, there are very few references in the literature of agencies using observed speed data (or INRIX data in particular) to validate travel demand models. There were some references to using observed speed data in general and several references on improving volume-delay functions (VDFs). There is definitely an increased level of recent interest in this topic, but there was very little direct information on the questions that MWCOG is asking. The next section describes the most relevant documented experience.

5.1 Specific Experience in Validation

Atlanta[10]

In the past two months, the Atlanta Regional Council (ARC) revalidated their traffic assignment process with the primary focus on matching traffic counts, but a secondary objective looked at matching freeway and arterial speeds as well. This involved 63 links, for which speed data was collected in the fall of 2001 using floating car techniques with GPS data capture.

[10] *ARC: Assignment Validation Work*, Internal Memorandum from W. Allen to P. Schropp (Atkins Global), 4 January 2013.

Initially, the model was over-estimating the higher facility types (FTs) and under-estimating the lower FTs. This suggested that the higher FT speeds were too high and the lower FT speeds were too low. Freeway, expressway, and parkway speeds were reduced by 1-9 mph; arterial and collector speeds were increased by 1-2 mph, and on-ramp speeds were reduced by 20 mph to be equal to the off-ramp speeds and in order to reduce volume on the freeways.

In the end, the freeway speeds were under-estimated in the aggregate by 6.9%, but the AM peak period percent root mean squared error (RMSE) was only 14.5%. The arterial speeds were over-estimated by 10.5% in the peak and by 8.2% in the off-peak with RMSEs of 18.2% and 16.7%, respectively. Any further changes to the speed look-up table to improve the observed vs. estimated speed values would have made the volume vs. count comparison worse and so the speed accuracy was judged to be close enough.

Indiana[11]

Bernardin, Lochmueller & Associates and Cambridge Systematics validated the Indiana statewide model in 2004 for Indiana DOT. Speed data collected at 64 locations were used to revise the speed look-up table and to develop relationships between free-flow speed and posted speed limit by facility type. They used a separate post-processor to estimate the average speed by hour based on the estimated hourly volume and the VDF, then calculated the daily average speed as the VMT-weighted average of the hourly speeds. When they compared the estimated and observed daily average speed, the RMSE was 23.8%.

San Francisco[12]

MTC did a detailed analysis of speed data using the Caltrans Performance Measurement System (PeMS) in 2012. This system collects statewide volume and speed data continuously via loop detectors, mostly on freeways, reported at 5-minute and 60-minute intervals. This report looked at changes to the speed and capacity look-up tables (they use a conventional facility type/area type stratification) and the VDF in order to improve validation. Several issues emerged, including:

- Numerous examples were found where the peak speed exceeded the midday speed, in the off-peak direction.
- They discovered that “micro-effects” such as lane width, shoulder width, horizontal/vertical alignment, pavement condition, truck percentage, adjacent distractions, weaving sections, and sun glare had a significant impact on actual freeway capacity. These could not be accounted for by a simple look-up table.

---

[11] *Indiana Statewide Travel Demand Model Upgrade, Technical Memorandum: Model Update and Validation*, by Bernardin, Lochmueller & Associates and Cambridge Systematics for Indiana DOT, September 2004.

[12] *Initial Examination of Volume Delay Functions Using PeMS Data*, Internal Memorandum to Travel Model One Development File from David Ory, Rupinder Singh, MTC, 6 March 2012.

- They found that observed speed and volume data are available at a greater level of geographic and temporal detail than other elements of the model (such as area type, and thus capacity).
- Their initial analyses found discrepancies between look-up table capacity values and maximum flow rates found in the field.
- They found higher observed hourly flow rates in the AM (vs. PM) for the same section of roadway, possibly suggesting the use of higher capacities for AM assignments than PM assignments.
- Capacity is not directly observable from count data. Actual flow rates cannot exceed capacity but modeled demand can. This complicates evaluation and adjustment of VDFs.
- Their freeway VDF applies a factor of 1.333 to V/C before exponentiation:

  $$Tc = T0 * [1 + 0.2*(1.333*V/C)^6]$$

  (the report cites an additional reference for this equation: http://mtc.ca.gov/maps_and_data/datamart/research/boston1.htm but does not explain the derivation or source of the parameters further)

ORLANDO[13]

At a 2012 conference, Kittelson & Associates described using Bluetooth readers to collect speed data on various corridors near Orlando. This involved 15-minute intervals mostly on arterial, collector, and local roads. Counts were obtained using tube counters. The data were used to develop new VDFs for a super-regional model of central Florida. The general VDF curve fitted from the data was

$$Tc = T0 * [1 + 0.56*(V/C)^{4.59}]$$

Different curves were used by corridor and direction (but, interestingly, not by facility type). The authors chose to base this equation on "practical capacity" (LoS C/D) instead of LoS E capacity, but this was not explained further.

HOUSTON[14]

In 2012, Cambridge Systematics reported on the comparison of survey reported O/D travel time to network time in Houston. They accounted for the fact that survey respondents tend to report "clock face" times (nearest 5 to 10 minutes) and the fact that reported times implicitly include terminal times, which must be added to the network times for comparison. They found that the reported times were lower than the network times, suggesting that the modeled speeds were too low. Most of the outliers (time difference > 30 minutes) were due to geocoding errors, unreported stops, and highway incidents.

[13] *Collection and Use of MAC Address Data for the Travel Demand Model*, Li Jin, Matt Wiesenfeld, Karl Passetti, and Jon Weiss, presented at the 4th TRB Conference on Innovations in Travel Modeling, April-May 2012.

[14] *Reducing Garbage-In…for Discrete Choice Model Estimation,* David Kurth, Marty Milkovits, Jason Lemp, and Pat Coleman, presented at the 4th TRB Conference on Innovations in Travel Modeling, April-May 2012.

Northern New Jersey[15]

AECOM revalidated the Northern New Jersey model in 2011 for purposes of evaluating strategies for I-78. They used INRIX speed data obtained from the I-95 Corridor Coalition. This involved speed by 5-minute period for 270 freeway segments from 2008 to 2011. The purpose of the speed analysis was to stratify the 30-minute periods in a year and identify the 30 minute period to use for analysis and reporting. They found a high degree of variability in freeway speeds – there were a few cases of extremely high peak period congestion, but then much less congestion in the off-peak.

TMIP Workshop[16]

In 2009, FHWA sponsored a series of web-based workshops on travel model development. Session 8 (June 2009) on model validation discussed the question of why it is so difficult for a regional model to match both observed volumes and speeds. They found that:

- Most commonly used VDFs are too simplistic.
- Link speeds are computed independently of adjacent links.
- Some factors affecting route choice are not included in impedance, such as travel time reliability, road familiarity, perceived safety, etc.
- Observed speed data has been relatively low in quantity and quality, but technology is addressing this. (Note that this document pre-dates INRIX data.)

5.2 Evaluating INRIX Data

Several agencies reported that they are currently in the process of evaluating INRIX speed data. Ohio DOT just started comparing INRIX speeds to speeds from their statewide model; a presentation was made at the 2013 TRB Planning Applications Conference[17]. They are looking at more sophisticated capacity algorithms in order to include effects such as traffic control devices, horizontal/vertical alignment, and railroad crossings.

Florida DOT is looking at INRIX data collected on freeways and arterials statewide at 5-minute intervals[18][19]. This will be separately supplemented with count data so that the MPOs can evaluate/adjust their VDFs. A report on the value of INRIX data is due by December 2013.

---

15 *Task Order No. 15, Route I-78 Ramp Metering Evaluation Diversion and Design Analysis, Hunterdon and Somerset Counties,* by AECOM for New Jersey DOT, 16 August 2011.

16 *Travel Modeling Workshop Session 8: Evaluation of Model Validation Results,* prepared by Cambridge Systematics for Travel Model Improvement Program (TMIP) Webinar Series, 9 June 2009.

17 Challenge 5: Transportation Economics: Development of an Enhanced User Benefit and Cost Calculator for Ohio, by C. Beard, Presented at the 14th TRB National Transportation Planning Applications Conference, May 5-9, 2013, Columbus, Ohio.

18 Personal communication from Frank Tabatabaee, Florida DOT, 13 December 2012.

19 *Travel Time Data*, presented to Florida Model Task Force by Vidya Mysore, 7 March 2012.

INRIX made a 2012 presentation to the Washington (state) Transportation Commission[20]. This focused mainly on real-time system monitoring, identifying traffic incidents, and providing trend data. In 2009, the Puget Sound Regional Council looked at truck GPS data and analyzed the differences between observed truck speeds and modeled speeds[21]. However, they did not report adjusting their model based on that work.

For several years, Baltimore Metropolitan Council performed GPS-based speed data collection (2001, 2004, and 2006)[22]. This was used for Congestion Management System (CMS) reporting and travel model validation, although the details were not documented. They are currently working with the I-95 Corridor Coalition to evaluate INRIX speed data (at least for freeways), which appears to be of higher quality than their earlier data, but is also expensive[23].

5.3 System Performance Monitoring

Much of the literature on INRIX and other sources of observed speed focused on present-day highway performance monitoring. This broad term includes a variety of techniques to collect and display data describing the congestion status of the highway system, sometimes in real time (or nearly so). In addition to showing current conditions (usually 15, 30, or 60 minutes ago), some applications attempt to do a short-term projection (next 30 minutes) of conditions. In addition to speed and delay, some applications display a variety of other travel statistics, such as travel time reliability. This information is compiled and provided to the public for a variety of time frames – everything from real time to annually.

This is similar to MWCOG's current "Congestion Dashboard", except that the Dashboard is updated only each quarter. A number of other MPOs have found this effort to be not only popular with the public, but a useful input to their FHWA-mandated Congestion Management work. Some specific examples:

- *Atlanta:* ARC does not currently use INRIX data; they use the GDOT "NaviGAtor", an extensive video camera system to monitor freeway conditions[24]. Speed and volume data are captured at 15-minute intervals during weekday peak periods. These data are used mainly for performance reporting and is merged with a variety of other data sources to produce the annual Metropolitan Atlanta Performance (MAP) Report.
- *Hampton Roads:* In 2010, VDOT purchased INRIX data and provided it to the MPOs. This included 5-minute and 15-minute data for freeways and selected arterials statewide. The

[20] *INRIX Data Services – Update*, presented to Washington Transportation Commission, 2012.

[21] *Planning for Freight in the Central Puget Sound Region, Travel Model Improvements for the Congestion Management Process and Long Range Transportation Plan Update,* by the Puget Sound Regional Council, July 2009.

[22] Personal communication from Charles Baber, BMC, 14 December 2012.

[23] *I-95 Corridor Coalition – Vehicle Probe Project (VPP), INRIX Data Evaluation*, presentation by Baltimore Metropolitan Council and BRTB, 3 January 2012.

[24] *Appendix, 2010 Transportation MAP Report,* by Georgia Regional Transportation Authority.

Hampton Roads Transportation Planning Organization is using these data to report/describe existing congestion and delay[25]. They plan to keep using these data to monitor trends and for CMS reporting. The INRIX data replaced HRTPO's limited GPS-based speed data collection effort every five years from 1986 to 2005. However, the INRIX data was not used in a 2011-12 VDOT project to develop a new regional model for Hampton Roads and there are no documented plans to use observed speed data in model validation.

- *Nashville:* In its annual CMS report of 2007, the MPO switched from using level of service to speed as the main indicator of congestion level[26]. A congested route is one where the peak time/free flow time ratio exceeds 1.30.
- *I-95 Corridor:* The I-95 Corridor Coalition has budgeted nearly $2 million annually to fund its Vehicle Probe project, which among other things includes purchasing from INRIX continuous travel time data for a wide area in the states which carry I-95[27]. The data are used for real-time monitoring of traffic conditions and incidents, improved emergency response, and travel time reporting for traveler information services. The data are shared with many agencies and it seems that most of them are so busy using the data for current performance monitoring that they have not yet documented any efforts to use INRIX data for long-term model improvements.

5.4 Possible Use in Travel Models

Several of the references mentioned above describe possible uses of observed speed data for model validation. These include:

- Improving speed estimation from an initial look-up table (mainly free-flow speeds).
- Revise the VDFs for more accurate congested speed estimation.
- Fine-tune the speed feedback process.
- Post-process speeds (outside of the travel model), mainly for air quality analysis purposes.
- Analyze peak spreading.

It would seem that updating/validating a speed look-up table from observed data would be pretty straightforward, but it isn't. A key problem is that observed speed data has so much variability that any sort of mean values are often not very meaningful. As is noted elsewhere in this memo, micro-effects have enough of an impact on speed so as to render the normal two-dimensional look-up table ineffective. As a result, observed speed data stratified by facility type and area type often do not obey the conventional wisdom that speed must always increase (or stay the same) as one goes from lower to higher facility types and from more developed to less developed areas.

---

[25] *Hampton Roads Regional Travel Time/Speed Study,* Hampton Roads Transportation Planning Organization, April 2012.

[26] *Congestion Management Process,* Nashville Area MPO, 19 September 2007.

[27] *I-95 Corridor Coalition – Vehicle Probe Project (VPP), INRIX Data Evaluation*, presentation by Baltimore Metropolitan Council and BRTB, 3 January 2012.

Some references noted the difficulty of using a simplistic algorithm to estimate the speed of every link in a region. The emerging practice is to go beyond the conventional facility type/area type look-up table, using additional procedures that better account for intersections, terrain, etc. Although regional travel models have mainly avoided modeling intersection delay, this is increasingly seen as necessary in order to properly represent arterials and collectors. A better source of observed speed data (like INRIX) is needed to support such improvements (and perhaps to reveal their necessity).

The high variability in observed speeds was noted by several observers and it is clear that this confounds efforts to reliably compare estimated and observed speeds. One way to address this is to look at the average speed of a contiguous string of links rather than individual isolated links (either for a corridor or by O/D movement), to smooth out the variability. Although there is significant interest in this, not much experience has been documented to date.

NCHRP Report 716, Travel Demand Forecasting: Parameters and Techniques (2012, Cambridge Systematics) is the update to NCHRP Report 365, a widely used reference in travel forecasting. Here's what it has to say about model validation and speed data:

> *Although regional travel demand forecasting validation generally focuses on volume and trip length-related measures, there is often a desire to look at loaded link speeds and travel times. The analyst should be cognizant that "model time" may differ from real-world time due to the many network simplifications present in the modeled world, among other reasons.* (p. 20)

5.5 Revising the Volume Delay Functions

Adjusting speed feedback opens the question of whether the model really intends to replicate observed speeds. For example, during speed feedback (or as part of the VDF), some modelers set a "floor" (lower bound) on the speed for each link, in order to prevent the model from creating so-called death links on any iteration of the assignment process. Death links are those for which the estimated congested speed is a tiny fraction of 1 mph, which is never experienced in the real world, even under crawling-along LOS F conditions. However, the developers of modeling software contend that this kind of limitation renders the equilibrium volume averaging algorithm built into the assignment program inaccurate and should thus not be used. The presence of death links usually indicates a hidden problem with network topology or specific unusual O/D movements. So this forces a difficult decision about whether the assignment itself must produce reasonable speeds on all links or whether this can/should be accomplished through post-processing.

Speed post-processing remains somewhat controversial. It seems that some agencies examine both model-estimated speeds and post-processed speeds (based on observed data) and use whichever set gives the better results for conformity. In some conformity documents, those agencies that use observed speed data aren't clear about how it's being used. Although it is clear that it is important to match observed speeds for air quality, it is not always obvious how useful that is otherwise. Some analysts believe that regional travel models might actually be better served by using "perceived" speeds that produce better paths. Others question whether it's really possible and logical for "model travel time" to match "real travel time".

Using observed speed data to identify peak spreading makes considerable sense. However, peak spreading is a phenomenon that changes very slowly. Therefore, in order to observe any usable trends, one would have to examine many years' worth of speed data. It would probably require 5-10 years of data to be able to identify a pattern in such a way that it could be incorporated into a model.

5.6 Volume Delay Research

Considerable research has been conducted in recent years on the topic of volume-delay functions (VDFs). One conclusion that is widely supported is that a single VDF is inappropriate. VDFs need to be stratified by facility type or at least facility type group, such as freeway/arterial/collector. Another point of agreement is that capacity should be expressed as true capacity, representing LOS E (instead of LOS C "practical capacity"). Beyond those points, there is little agreement. Much of the research focuses on identifying the most appropriate mathematical form of the function and the coefficient values. The "BPR function", dating to the 1960's, is the most commonly used formulation, but some agencies have used the conical function developed by Heinz Spiess in the 1990's. Others have investigated more detailed models of traffic delay, such as from Akcelik and Webster. Most of the literature[28] [29] [30] [31] suggests that the BPR function still does a pretty good job and is not clearly overshadowed by the other functions.

Many agencies have made a number of changes to the basic BPR function, including:

- Set a floor on the computed speed (see above).
- Use different curves (equations) for V/C <= 1.0 and V/C > 1.0.
- Modify the basic equation. For example, the basic BPR equation is:

  $Tc = T0 * (1 + a*[(V/C)^b])$.

  One variant is

  $Tc = T0 * (1 + c*(V/C) + a*[(V/C)^b])$.

  This lowers the constrained speed at low V/C's. Another is to factor the capacity before exponentiating the V/C ratio:

---

[28] *Initial Examination of Volume Delay Functions Using PeMS Data*, Internal Memorandum to Travel Model One Development File from David Ory, Rupinder Singh, MTC, 6 March 2012.

[29] *Collection and Use of MAC Address Data for the Travel Demand Model*, Li Jin, et al., presented at the 4th TRB Conference on Innovations in Travel Modeling, April-May 2012.

[30] *Volume Delay Function Research and Development* (Project Number: 26661-07-20), prepared by The Corradino Group for VDOT, 2009.

[31] *Calibration of Volume-Delay Functions for Traffic Assignment in Travel Demand Models*, by Old Dominion University (M. Cetin, et al.) for VDOT, submitted to TRB 2012 annual meeting, August 2011.

$$Tc = T0 * (1 + a*[(V/\{d*C\})^b])$$

- Use different curves for links with and without traffic control devices (stop signs or signals).

Some of these changes have been validated using observed speed data, but most have not.

Since INRIX (and methods using GPS and/or Bluetooth detecting technology in general) provides improved observed speed data, this has encouraged many agencies to re-examine their VDFs. However, in addition to observed speed and volume, VDF analysis also requires information on observed capacity. As it turns out, this provides the greatest challenge in studying VDFs.

Speed and volume are relatively straightforward concepts, but capacity is not easy to define objectively or measure in the field. First, the measured capacity of any particular link is likely to not equal that estimated by the model, mainly due to the micro-effects noted above. Second, it is obvious that V/C ratios above 1.0 cannot be observed in the real world (assuming one uses a literal definition of capacity). If a road is subject to over-saturated conditions, it is easier to identify the capacity by plotting different flow rates and vehicle densities, but this is feasible only where the counted volume approaches a value that can be confirmed to represent its maximum observed value. On other roads, true capacity cannot be observed, only inferred from the available data. Some studies have resorted to doing a more through estimation of capacity on the subject links, but it's still just an estimate (compared to the speed and volume, which are observed). This complicates the evaluation of VDFs.

In 2009, VDOT sponsored an extensive project to examine VDFs[32]. This report confirmed the difficulty of estimating capacity in under-saturated conditions. It also examined many areas that were not part of any regional travel model. For those areas, count data and synthetic HCM procedures were used to compute capacity. More recently, VDOT contracted with Old Dominion University to conduct research into optimizing VDF parameters[33]. This paper found that using observed data to identify the "correct" VDF for a model or to derive the "right" values is generally not feasible. This research suggests that the VDF should be determined based on minimizing overall network RMSE and reports on an innovative optimization procedure. However, this process sometimes leads to parameters that are inconsistent with experience in other areas, such as parameters that don't vary in a logical way by facility type. In addition, since numerous factors contribute to the RMSE value, it may not be advisable to make one set of parameters (the VDFs) dependent on RMSE.

Several reports documented efforts to derive the optimal mathematical form of a VDF and the coefficients from field data, but this was usually not very successful. It would seem to be straightforward in theory, but it turns out to be significantly more difficult in practice. As noted above,

[32] *Volume Delay Function Research and Development* (Project Number: 26661-07-20), prepared by The Corradino Group for VDOT, 2009.

[33] *Calibration of Volume-Delay Functions for Traffic Assignment in Travel Demand Models*, by Old Dominion University (M. Cetin, et al.) for VDOT, submitted to TRB 2012 annual meeting, August 2011.

the available documentation mostly suggests that a modified version of the original BPR equation still works fairly well.

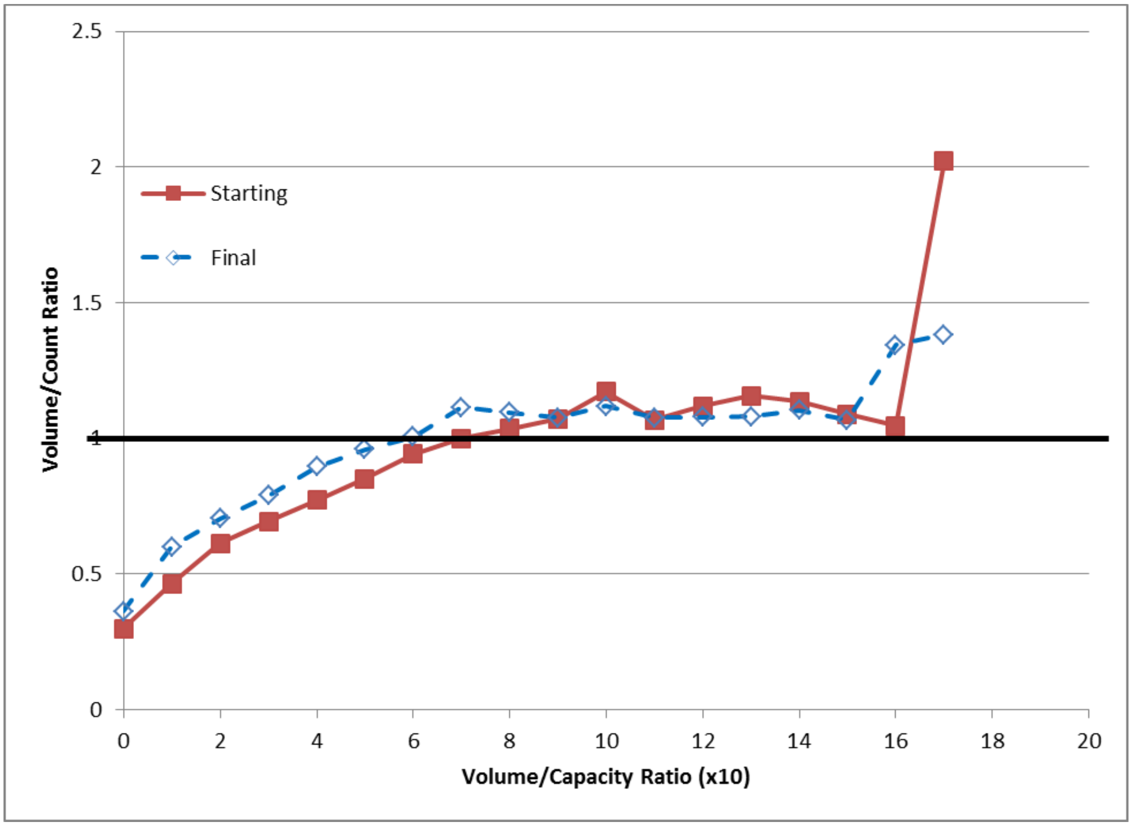
VDFs are mainly theoretical constructs and it is difficult to know whether a particular VDF is "good". One useful way to evaluate VDFs is to tabulate the volume/count ratio by ranges of volume/capacity ratio. Ideally, this should be a horizontal line (see Figure 5-1, from Atlanta[34]). That would indicate that the model's error is consistent across the range of volume/capacity values, which suggests that the VDFs are doing their job properly. In practice, this ideal is rarely achieved – it is the authors' experience that most models under-estimate the count on low volume/capacity links and over-estimate the count on high volume/capacity links.

In January 2009, TMIP issued a Technical Synthesis of comments received through the web on Speed Adjustments Using Volume-Delay Functions[35]. They found that the use of different VDFs by facility type is now accepted as good practice, but that the use of minimum speeds (floors) remains controversial. One contributor asked *"…whether it is possible to produce a model that can calibrate to observed speed conditions (and observed flow conditions) and yet remain viable for forecast scenarios where those conditions will most likely be different. (p. 3)".*

[34] *ARC: Assignment Validation Work*, Internal Memorandum from W. Allen to P. Schropp (Atkins Global), 4 January 2013.

[35] *Speed Adjustments Using Volume-Delay Functions,* Technical Synthesis, TMIP, January 2009.

Figure 5-1: Volume/Count by Volume/Capacity (Atlanta)



5.7 Observations

This literature review identified a number of findings and concerns about using observed speed data to improve regional travel models.

- In the past, observed speed data has been of relatively poor quantity/quality. INRIX may solve this problem, but the literature does not yet demonstrate that better data will by itself automatically make observed speed data more relevant to regional travel demand model validation or produce better models.
- There is not much doubt that INRIX data are extremely useful as a highway performance monitoring tool. Current performance reporting is something that many agencies are adding to their repertoire and it seems to be popular with the public.
- It is not uncommon for observed peak speeds to exceed off-peak speeds in particular locations, especially in the off-peak direction. This is likely due to high variation in speeds, more aggressive driving during the peak period, and capacity micro-effects that affect congested speeds at specific locations. It is unclear how a regional model can deal effectively with this phenomenon.
- INRIX speed data (collected via millions of GPS-enabled vehicles and mobile devices) are generally regarded as being of higher quality and quantity than floating car or other techniques commonly used in the past. Still, the literature suggests using caution regarding INRIX speed data on non-freeway links. Some problems have been noted in measuring speeds under stop-

and-go conditions. Also, with GPS and Bluetooth technology (and the greater quantity of available data) it is apparently difficult to identify sporadic equipment malfunctions which can lead to extremely high or low speed values.

- There is a growing interest in developing speed algorithms that are more sophisticated than the present facility type/area type look-up tables. It seems that observed speed data should be useful in supporting such analysis.
- Even today, there are a surprising number of models that have not yet switched to assignments by time of day. The literature suggests that that should be done (which MWCOG has) before an agency should try to match observed speeds. A number of references (such as 1, 3, 8, 20) opined that matching speeds is secondary to matching counts. Since it is so difficult to match both, the latter should take precedence.
- Most regional travel models do not reflect roadway queuing or intersection delay, since these phenomena cannot be handled by “static” traffic assignments. In this context, what do the estimated speeds from a static model actually represent? The authors believe that it is not reasonable to compare them to observed space-mean speeds -- this is an example of “apples vs. oranges.” Some concern has been expressed that a direct comparison of travel model-estimated speeds with observed data may be technically inappropriate. In other words, the availability of better-quality observed speed data such as provided by INRIX has served only to further highlight the fact that static models do not estimate the same speed information. Thus, it is inadvisable to be overly ambitious to try to get the model to match the observed speed data. This has the danger of making the model less suitable to produce reasonable volume forecasts.

6 Using INRIX Data for Travel Models (Task Order 8)

In this chapter, INRIX data is used to evaluate MWCOG model results. INRIX, Inc. provides real-time traffic flow data such as speed and travel time over major freeways, highways, and arterials. The data provided by INRIX have been used in many studies and research projects, and are considered to be reliable. This section, therefore, compares link speeds in different time periods obtained by the MWCOG travel model with INRIX speeds. As stated in the previous chapter, comparing speeds from a static assignment model to observed speeds ignores the primary purpose of volume-delay functions in the static assignment algorithm – to estimate reasonable link volumes. That being said, there is still much that can be learned about how to interpret model results and assess model sensitivities by comparing model speeds to INRIX speeds.

Access to INRIX data: The I-95 Corridor Coalition (I95CC) is "an alliance of transportation agencies, toll authorities, and related organizations, including public safety, from the State of Maine to the State of Florida, with affiliate members in Canada."[36] The Vehicle Probe Project (VPP) is an initiative and collaborative effort among the I95CC, the University of Maryland, and INRIX, Inc. to provide comprehensive and continuous real-time travel information for member agencies.[37] As an affiliate member of the I95CC, the TPB has access to the INRIX data from the VPP. This access is governed by a data use agreement (DUA) between with the University of Maryland and MWCOG/TPB. This DUA allows access to the INRIX data for both the "agency" (MWCOG/TPB) and "agents and employees" of the agency. Since AECOM has a contract with MWCOG, AECOM can be considered an agent of COG. AECOM has agreed to be in full compliance with the MWCOG/UMD DUA and must limit its use of the INRIX data to projects specified by MWCOG/TPB.[38]

6.1 Combining of the MWCOG Model Outputs with INRIX Speed Data

The INRIX data provided by MWCOG for this analysis covers Maryland, Virginia, and District of Columbia and were collected by INRIX, Inc. in October 2010 and provided to MWCOG in 5-minute intervals. MWCOG/TPB staff aggregated the data to hourly periods and combined selected records with hourly traffic counts to generate hourly counts and speeds for an average weekday in 2010.

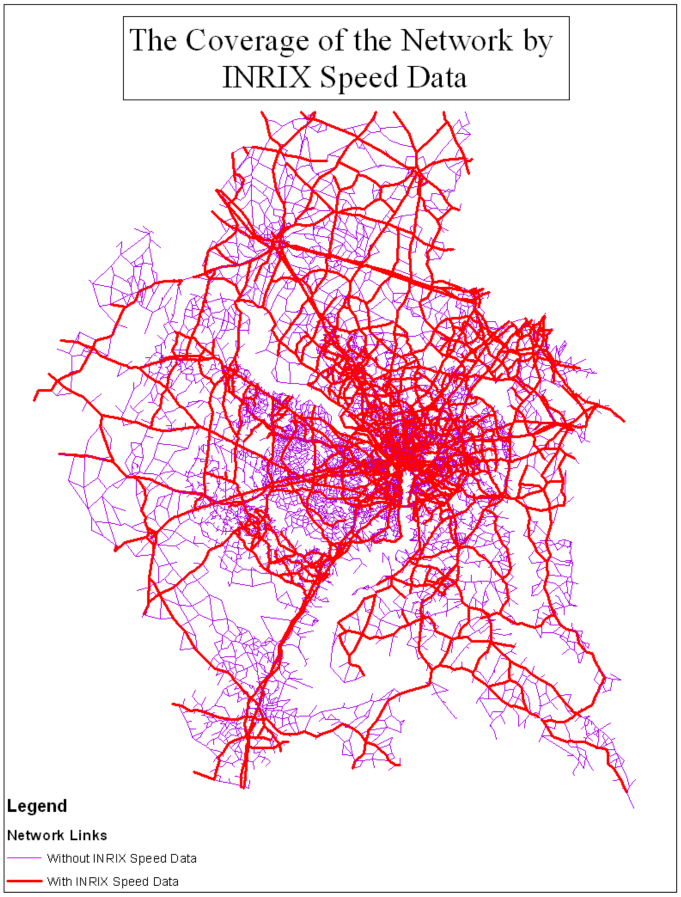
As shown in Figure 6-1, the year-2010 INRIX dataset includes a large number of the major freeways, expressways, and arterials in the region. As a result, 17,012 links out of 48,692 links in the MWCOG regional network have speed values for every hour of an average day.

[36] "What Is the Coalition?," *I-95 Corridor Coalition*, 2013, http://www.i95coalition.org/i95/Home/WhatistheCoalition/tabid/112/Default.aspx.

[37] "Vehicle Probe Project," *I-95 Corridor Coalition*, 2013, http://www.i95coalition.org/i95/VehicleProbe/tabid/219/Default.aspx.

[38] Mark S. Moran to David Roden et al., "Data Use Agreement (DUA) for Using INRIX Traffic Speed Data," Memorandum, February 5, 2013.

Figure 6-1: The Network Coverage by INRIX Data



The MWCOG traffic assignment output is reported for four time periods as follows:

1. AM peak period (AM): from 6 AM to 9 AM,
2. Midday period (MD): from 9 AM to 3 PM,
3. PM peak period (PM): from 3 PM to 7 PM, and
4. Nighttime/early morning period (NT): from 7 PM to 6 AM

Using the hourly, average-link-speed data set supplied by MWCOG, AECOM aggregated INRIX speed data to the four time-of-day periods used in the traffic assignment. As stated earlier, the hourly average speeds were developed from the 5-minute observations, using a harmonic mean. However, traffic volume varies over different hours of each time period, which affects speeds. In fact, speeds can be very different for hours of one time period due to variation in traffic volume and congestion. Thus, a simple average of speeds would not be a reliable representative of the speeds during each multi-hour time period. Consequently, AECOM chose to use a weighted average as a good measure to show the tendency of speed for each time period. The best weights are traffic volumes; however, they are not available for all links.[39] For each of the four time-of-day periods, MWCOG staff has calculated, based on trips in motion from the 2007/2008 Household Travel Survey, a peak-hour factor, which is the proportion of traffic in a given time period that occurs in the peak hour of the period (Table 6-1). The peaking factor is necessary in the travel model for converting hourly lane capacities into "period lane capacities," from which V/C ratios are computed.[40] A peaking factor of 33.3% for a three-hour period implies that there is little to no peaking, with traffic evenly spread over the three-hour period.

Table 6-1: Peak-Hour Factors from the MWCOG 2007/2008 Household Travel Survey

Time Period	AM	PM	MD	NT
Highest Hour Share	41.7%	29.4%	17.7%	15.0%

AECOM used these peaking factors to calculate the weighted average of INRIX speeds for each time period. Then the weighted averages were added to the output of the MWCOG model to provide a comprehensive database, including MWCOG model results and observed speeds for every link. This database was used for the MWCOG model output evaluation, which is described in the next section.

6.2 Analysis of MWCOG Model Speeds

Traffic volumes, the output of traffic assignment, were used in volume-delay functions (VDF) to calculate travel time and speed for each link of the network. The MWCOG model starts with free flow speeds and at every user equilibrium iteration and every global speed feedback iteration, it updates speeds based on traffic volumes and VDFs. Four global iterations are used to balance the travel times used to estimate demand with the assigned speeds. Table 6-2 presents the number and percent of links with INRIX speed data for each facility type, and Table 6-3 compares the MWCOG model speeds and INRIX speeds for different categories of facility type.

Table 6-2: The Number of Links with INRIX Speed Data in Different Facility Types

Facility Type	Number of Links	Number of links with INRIX Data	Coverage (%)
Freeway	2,662	1,955	73.44
Expressway	426	380	89.20
Major Art.	6,910	6,237	90.26
Minor Art.	11,522	6,539	56.75
Collector	10,488	1,710	16.30

Table 6-2 confirms that the INRIX data covers a major portion of freeways, expressways, and major arterials.

---

[39] About 1,500 links contain both INRIX link speed data and hourly traffic count data.

[40] Ronald Milone et al., *Calibration Report for the TPB Travel Forecasting Model, Version 2.3, on the 3,722-Zone Area System*, Final Report (Washington, D.C.: National Capital Region Transportation Planning Board, January 20, 2012), page 8–18, http://www.mwcog.org/transportation/activities/models/documentation.asp.

Table 6-3: The Comparison between the MWCOG Model and INRIX speed in Different Facility Types

Facility Type	Average Difference (mph)				Average Absolute Difference (mph)				RMSE (mph)			
	AM	MD	PM	NT	AM	MD	PM	NT	AM	MD	PM	NT
Freeways	-3.5	1.1	-3.4	1.0	12.5	7.6	13.5	5.3	16.8	10.6	17.5	7.9
Expressway	-7.3	-2.2	-8.3	-3.3	13.2	10.0	14.1	7.9	17.7	12.9	18.3	9.9
Major Art.	0.4	6.7	0.3	4.9	10.1	9.8	9.9	8.3	14.5	11.4	12.1	9.9
Minor Art.	1.4	6.1	0.7	4.5	9.3	8.6	9.0	7.5	12.6	10.7	11.5	9.3
Collector	2.4	4.3	1.7	2.2	8.1	8.1	8.1	7.8	10.0	10.0	10.0	9.5

The comparison shown in Table 6-3 is based on link-by-link speed differences between the MWCOG model and INRIX. In the first set of columns, labeled "Average Difference," the difference is defined as the MWCOG model speed minus the INRIX speed. Positive numbers, therefore, mean that the speeds in the MWCOG model are faster than those in the INRIX data. Since, in every facility type, some links have positive difference and some links have negative difference, in the calculation of "Average Difference," they cancel each other out, meaning the average difference does not show the true values, but it shows the correct sign. In other words, Table 6-3 shows that freeways and expressways in the MWCOG model are slower than INRIX in peak periods. On the other hand, arterials and collectors are faster in the MWCOG model. It also shows that freeways are faster in the MWCOG model than INRIX during night and mid-day.

The average of absolute difference, the middle set of columns in Table 6-3, removes the error of positive and negative signs in the average calculation by taking the average of absolute value of the difference, instead of the real value of difference. The output numbers, therefore, are higher than "Average Difference" and show the true difference between the MWCOG model and INRIX. This measure has higher values in AM and PM peak periods on freeways and expressways, implying that the MWCOG model performs better in mid-day and night periods.

Root-mean-squared error (RMSE), shown in the last set of columns in Table 6-3, is a statistical way of comparing two datasets especially when positive and negative signs exist. Equation 6-1 indicates how this measure is calculated.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(S_m - S_I)^2}{n}} \qquad (6\text{-}1)$$

Where:

S_m : The MWCOG model speed for link i

S_I: The INRIX speed for link I, and

n: Number of links in the facility type.

As was the case for the "Average Absolute Difference," the RMSE shows that the MWCOG model performs better in mid-day and night periods, although this difference is the least pronounced for collectors.

Table 6-4 presents the average travel speed of major corridors in the MWCOG model, INRIX, and the difference between them. The average speed has been calculated by direction to more accurately compare the data.

Table 6-4: The Comparison between the MWCOG Model and INRIX speed in Major Corridors

Corridor	MWCOG Speed (mph)				INRIX Speed(mph)				Difference (mph)			
	AM	MD	PM	NT	AM	MD	PM	NT	AM	MD	PM	NT
I-395 NB	38.1	47.8	48.3	53.7	37.2	52.2	55.4	60.5	0.9	-4.5	-7.0	-6.8
I-395 SB	53.5	52.4	38.7	53.5	57.2	58.4	46.0	58.5	-3.6	-6.0	-7.3	-5.0
I-66 EB	41.5	54.7	52.1	58.7	44.7	56.9	55.3	60.5	-3.2	-2.2	-3.3	-1.9
I-66 WB	55.9	56.4	43.4	57.8	58.6	57.3	51.8	59.1	-2.6	-0.8	-8.4	-1.3
I-495 Inner Loop (VA)	31.4	52.0	33.4	56.4	42.6	48.2	42.4	57.7	-11.2	3.8	-9.0	-1.4
I-495 Outer Loop (VA)	37.3	52.5	26.3	55.8	54.5	54.0	42.6	56.5	-17.2	-1.5	-16.3	-0.6
I-95 NB (VA)	42.5	55.3	50.3	60.3	51.1	57.2	56.5	62.8	-8.6	-1.9	-6.1	-2.5
I-95 SB (VA)	53.1	56.6	42.6	60.2	60.7	56.5	43.2	60.6	-7.6	0.2	-0.6	-0.5
I-270 NB	56.4	54.9	31.0	56.5	60.0	59.3	42.2	59.0	-3.5	-4.4	-11.2	-2.6
I-270 SB	25.3	50.5	49.7	56.8	31.4	55.5	61.0	60.6	-6.1	-5.0	-11.3	-3.8

According to Table 6-4, the MWCOG model performs well for I-395, I-66, and I-270 especially during AM peak period. The difference between the MWCOG model and INRIX on I-495 is significant, showing the model is slower than INRIX. Most of the differences in this table are negative, which confirms that the MWCOG model is slower than the (observed) INRIX data on freeways. This conclusion is consistent with the numbers shown in Table 6-3.

6.3 The Impact of INRIX Speeds on Skims

Skims are the matrices of time and distance between every pair of zones, using the minimum-impedance paths developed for each zone pair. The skims used in a given speed feedback iteration are based on the traffic assignment from the previous speed feedback iteration. Highway skims contain matrices for auto modes including SOV, HOV2, and HOV3+. AECOM developed a new set of skims where the model speed was replaced by INRIX speed wherever INRIX data existed (the modeled speed was retained in cases where there was no INRIX speed). In this section of the report, the new skims based on INRIX speeds are compared to those based on the MWCOG model speeds. Since skims are calculated for peak and off-peak periods, the results are presented for these two time periods as well. Table 6-5 and Table 6-6 present the mean, maximum and minimum of travel time and distance for different modes in different skims.

Table 6-5: Statistics of Travel Time Skim in the MWCOG Model with and without INRIX Data

Speed Reference	Mode	Peak Period			Off-Peak Period		
		Min	Max	Mean	Min	Max	Mean
MWCOG	SOV	0.4	322.4	81.5	0.4	215.0	52.9
	HOV2	0.4	322.4	79.6	0.4	215.0	53.0
	HOV3+	0.4	322.4	75.7	0.4	215.0	53.0
INRIX	SOV	0.4	215.3	57.0	0.4	195.3	51.5
	HOV2	0.4	215.3	56.7	0.4	195.3	51.6
	HOV3+	0.4	214.9	55.8	0.4	195.3	51.6

Table 6-6: Statistics of Distance Skim in the MWCOG Model with and without INRIX Data

Speed Reference	Mode	Peak Period			Off-Peak Period		
		Min	Max	Mean	Min	Max	Mean
MWCOG	SOV	0.1	185.8	39.8	0.1	161.6	38.6
	HOV2	0.1	185.8	40.0	0.1	161.6	38.6
	HOV3+	0.1	185.8	40.7	0.1	161.6	38.6
INRIX	SOV	0.1	177.2	39.7	0.1	161.9	39.5
	HOV2	0.1	177.2	39.9	0.1	161.9	39.6
	HOV3+	0.1	177.2	40.0	0.1	161.9	39.6

According to Table 6-5 and Table 6-6, peak-period paths in the MWCOG model with INRIX speeds are shorter and travel times are shorter than the model without INRIX speeds, implying that the estimated speeds in the travel model are too low. The improvements in travel time in the peak period are significant; for instance, the average travel time for SOV in the model with INRIX speeds is 30 percent smaller than the other model. However, distances (as opposed to speeds) are very close, with a difference of less than 5 percent. Off-peak skims do not change significantly; however, paths in the model with INRIX data are longer than the other one; while the travel time is shorter. Overall, it can be concluded that using INRIX speeds in the MWCOG model affects the peak-period skims significantly.

The travel time and distance skims were aggregated by jurisdictions and differences calculated. Table 6-7 describes the jurisdiction codes and the results of this aggregation are presented in Table 6-8 through Table 6-17 for travel time, distance, and different modes during peak and off-peak periods. Cells of the tables are also highlighted with colors that are proportional to the cell values. The darker cells present the higher differences between the two skims. As explained, distances are not significantly different; however, the travel time tables show considerable differences between the two models for some pair of jurisdictions.   According to these tables, INRIX speeds have a significant impact on paths to the regional core, Montgomery County, Prince George's County, Arlington core, Alexandria, and Fairfax County. Paths from the regional core to other jurisdictions are also affected by INRIX speeds.

Table 6-7: Description of Jurisdiction Codes Used in this Report

Jurisdiction Code	Name
1	Arlington Non-Core
2	Alexandria
3	Fairfax County
4	Loudoun County
5	Prince William County
6	Fauquier County
7	Stafford County
8	Clarke County and Jefferson County
9	Spotsylvania County and Fredericksburg
10	King George County
11	District of Columbia Core
12	Arlington Core
13	District of Columbia Non-Core
14	Montgomery County
15	Prince George County
16	Frederick County
17	Carroll County
18	Howard County
19	Anne Arundel County
20	Calvert County
21	St. Mary's County
22	Charles County
23	External Zones

Table 6-8: Percent Travel Time Difference for Peak Period SOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	13.2	5.3	7.4	8.3	6.8	7.8	19.2	10.5	21.5	15.7	32.2	27.2	24.4	17.3	17.2	16.0	24.0	22.3	23.4	15.4	10.7	5.8	20.5
2	21.3	7.3	10.7	10.6	7.2	8.4	19.9	11.8	22.2	14.2	34.6	32.4	27.7	18.7	21.6	17.0	25.6	26.2	25.3	15.6	10.8	5.8	21.4
3	37.3	33.8	22.3	14.3	8.3	8.1	20.8	12.2	23.1	20.1	40.7	40.9	36.3	34.6	31.9	26.4	30.7	34.1	31.8	23.6	18.7	19.6	25.9
4	38.8	36.6	36.3	22.6	20.8	11.5	17.8	6.7	18.3	17.8	42.0	41.4	39.6	39.8	36.3	24.3	28.4	32.3	34.3	29.5	25.3	27.7	23.7
5	41.9	41.2	36.3	19.8	10.1	6.8	16.8	9.7	20.0	18.1	42.6	43.4	40.1	37.9	37.8	25.4	29.4	35.5	35.8	31.8	26.4	30.3	26.3
6	38.4	36.6	34.5	20.7	19.1	8.6	10.2	9.4	11.6	13.4	38.9	39.3	37.0	35.4	34.8	17.4	21.8	28.5	32.8	27.1	21.6	25.2	21.5
7	43.8	44.9	39.0	22.5	26.1	10.4	4.7	12.3	7.3	12.3	44.7	45.5	42.5	39.6	39.6	23.5	29.1	37.7	37.4	24.2	23.2	23.6	26.7
8	37.9	35.5	36.3	31.8	25.0	22.3	20.9	7.6	20.5	20.3	41.0	41.0	39.6	39.0	36.6	12.3	20.4	24.5	30.0	32.4	28.7	30.1	20.2
9	42.3	43.2	38.0	21.7	26.2	11.6	10.1	12.5	0.2	15.2	43.3	43.9	41.5	38.9	37.3	21.0	27.1	37.1	33.9	21.0	24.1	23.1	24.9
10	33.8	35.4	32.6	20.1	21.9	11.2	9.1	12.2	7.0	2.6	31.7	33.7	29.0	27.9	26.0	21.4	29.7	28.6	23.9	19.0	25.5	24.2	22.0
11	1.0	-5.7	5.5	7.4	5.2	7.7	16.3	6.7	19.1	7.7	-16.8	-0.4	-22.1	-4.1	-10.9	5.9	18.2	9.9	11.2	5.4	5.6	-5.6	14.9
12	1.0	-0.7	8.0	8.8	6.9	8.7	18.8	9.3	21.2	14.6	24.8	7.9	16.2	7.8	11.3	10.7	21.1	18.5	20.6	13.2	10.5	3.7	18.5
13	24.6	22.4	20.1	14.9	15.3	14.5	22.0	10.9	23.5	9.8	27.2	28.3	8.8	5.9	-1.4	10.7	22.8	16.2	13.4	6.6	7.9	0.6	18.5
14	33.1	32.5	36.6	25.1	30.1	24.9	32.0	9.3	32.0	23.9	33.2	35.1	28.8	14.3	23.4	6.5	22.1	14.6	20.8	18.4	19.2	20.2	21.4
15	35.5	39.4	30.4	22.0	24.7	21.2	28.3	15.6	27.8	12.2	35.1	37.5	27.2	16.8	18.6	17.0	29.6	24.7	20.4	10.4	12.2	8.7	22.9
16	43.8	43.5	46.2	42.2	36.7	25.0	29.9	10.1	27.9	27.8	44.1	44.9	43.5	46.0	41.4	13.1	21.7	30.8	35.2	35.7	33.1	36.1	23.8
17	48.1	48.3	48.7	41.0	40.3	28.3	38.3	20.8	36.8	36.6	50.3	49.1	50.3	51.2	49.0	24.9	26.6	47.0	45.5	42.4	38.8	42.5	31.8
18	47.5	48.1	47.1	34.9	40.6	31.1	39.8	9.5	39.3	30.6	49.1	48.8	48.3	42.4	45.7	11.2	30.5	28.9	35.2	34.2	31.4	35.8	27.6
19	50.8	52.5	46.5	37.0	41.5	36.0	42.0	24.5	41.3	29.8	51.9	52.6	49.9	40.2	50.4	30.0	39.4	42.0	28.2	34.7	31.1	36.1	32.8
20	40.0	42.9	36.6	30.4	32.6	26.3	25.6	27.0	21.4	22.4	40.3	41.0	39.0	33.9	41.1	30.9	38.3	38.7	36.6	9.4	19.6	25.6	30.3
21	37.0	39.5	34.0	27.9	29.6	18.2	18.1	23.2	17.7	20.8	36.7	37.4	33.7	29.3	32.7	27.1	33.6	32.6	30.6	15.5	13.0	24.4	26.7
22	32.8	36.3	28.6	22.3	24.6	17.5	15.4	19.6	12.0	13.0	31.8	33.1	27.6	23.7	23.1	22.5	30.5	28.1	21.3	6.3	11.1	5.7	22.6
23	44.0	44.0	43.1	33.6	35.8	27.6	33.0	16.4	32.2	28.6	44.8	45.2	43.6	41.4	41.3	20.6	26.4	34.4	34.6	33.9	32.0	34.7	26.4

Table 6-9: Percent Travel Time Difference for Peak Period HOV2 Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	12.5	5.2	7.4	3.9	6.8	7.8	19.2	6.0	21.5	15.7	29.6	24.9	22.8	17.2	16.2	16.0	24.0	22.1	22.5	14.7	10.7	5.7	20.1
2	21.2	7.2	10.7	7.9	7.2	8.4	19.9	8.4	22.2	14.3	34.6	32.4	27.7	18.7	21.6	17.0	25.6	26.2	25.2	15.6	10.8	5.8	21.1
3	29.0	30.1	21.1	11.7	8.3	8.1	20.8	9.2	23.1	20.1	30.6	31.0	29.1	33.3	29.4	25.5	30.2	33.3	29.7	23.1	18.4	19.1	25.2
4	33.8	32.6	34.5	22.8	21.5	11.5	18.3	6.6	18.8	18.3	31.8	32.8	32.3	38.3	33.4	23.8	28.1	31.4	31.7	28.3	24.4	26.6	23.3
5	31.6	36.7	31.5	19.2	10.0	6.8	16.8	9.7	20.0	18.1	32.6	33.3	31.7	33.4	33.5	25.4	29.4	33.3	32.1	29.5	24.5	27.7	25.4
6	24.4	29.5	27.9	20.2	18.8	8.6	10.1	9.4	11.6	13.4	23.9	24.3	24.6	29.6	28.1	17.4	21.8	26.5	27.5	24.1	21.0	22.3	20.5
7	42.5	44.8	38.1	22.0	26.1	10.4	4.7	12.3	7.3	12.3	44.4	45.3	42.2	39.1	39.5	23.5	29.1	37.7	37.4	24.3	23.2	23.6	26.7
8	36.6	35.8	37.6	32.1	25.3	22.3	20.9	7.6	20.5	20.3	34.2	35.4	35.6	39.1	36.0	12.3	20.4	24.5	29.9	32.4	28.1	30.5	20.2
9	41.4	43.1	37.3	21.3	26.2	11.6	10.1	12.5	0.2	15.2	43.3	43.9	41.3	38.7	37.3	21.0	27.1	37.0	33.9	21.1	24.1	23.1	24.9
10	33.7	35.5	32.3	19.8	21.9	11.2	9.1	12.2	7.0	2.6	31.7	33.8	29.0	28.0	26.0	21.5	29.7	28.7	23.9	19.0	25.5	24.2	22.1
11	1.0	-5.7	5.5	4.3	5.2	7.7	16.3	5.2	19.1	7.7	-16.8	-0.4	-22.1	-4.1	-10.9	5.9	18.2	9.9	10.9	5.4	5.6	-5.6	14.8
12	1.0	-0.7	8.0	4.7	6.9	8.7	18.8	6.7	21.2	14.6	24.8	7.9	16.2	7.8	11.3	10.7	21.1	18.5	20.4	13.2	10.5	3.7	18.3
13	24.6	22.4	20.1	13.8	15.3	14.5	22.0	10.1	23.5	9.9	27.2	28.3	8.8	5.9	-1.5	10.7	22.8	16.2	13.2	6.6	7.9	0.6	18.4
14	35.5	33.0	36.8	26.0	30.2	25.1	32.1	9.7	32.1	23.9	33.4	35.3	28.9	14.4	23.5	6.5	22.1	14.6	20.8	18.4	19.3	20.2	21.5
15	35.5	39.3	30.4	21.6	24.7	21.2	28.3	15.1	27.8	12.2	35.0	37.5	27.2	16.8	18.6	17.0	29.6	24.7	20.3	10.4	12.2	8.7	22.9
16	45.1	43.9	45.1	42.4	36.7	25.0	30.0	10.1	28.0	27.9	44.3	45.1	43.6	46.2	41.4	13.1	21.7	30.8	35.2	35.7	33.1	36.1	23.9
17	49.0	48.4	48.3	41.1	40.2	28.3	38.4	20.8	36.9	36.7	50.3	49.1	50.3	51.2	49.0	24.9	26.6	47.0	45.5	42.4	38.8	42.5	31.9
18	48.1	48.2	47.1	35.1	40.6	31.1	39.8	9.5	39.3	30.6	49.1	48.8	48.3	42.4	45.7	11.2	30.5	28.9	35.2	34.2	31.4	35.8	27.6
19	50.6	52.4	46.4	36.9	41.4	35.9	41.9	24.4	41.2	29.9	51.6	52.3	49.6	40.1	50.3	30.0	39.4	42.0	28.2	34.7	31.1	36.1	32.8
20	40.0	42.9	36.6	30.1	32.6	26.3	25.6	26.5	21.4	22.4	40.3	41.0	39.0	33.9	41.1	30.9	38.3	38.7	36.6	9.4	19.6	25.6	30.3
21	37.0	39.5	34.0	27.9	29.5	18.3	18.1	23.5	17.7	20.8	36.7	37.4	33.7	29.3	32.7	27.1	33.6	32.6	30.6	15.5	13.0	24.4	26.7
22	32.8	36.3	28.6	21.9	24.6	17.5	15.5	19.0	12.1	13.0	31.8	33.1	27.6	23.7	23.1	22.5	30.5	28.1	21.3	6.3	11.1	5.7	22.5
23	43.2	43.6	42.6	33.6	35.7	27.6	33.0	16.4	32.2	28.7	43.0	43.6	42.3	41.0	40.7	20.7	26.4	34.2	33.9	33.6	31.7	34.4	26.3

Table 6-10: Percent Travel Time Difference for Peak Period HOV3+ Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	11.7	5.1	7.4	6.4	6.8	7.8	19.2	7.4	21.5	15.7	27.0	22.3	20.4	15.9	15.0	15.3	23.6	21.4	21.5	14.6	10.7	5.7	19.7
2	16.7	7.1	10.6	9.9	7.2	8.4	19.9	9.6	22.2	14.3	23.9	17.7	18.9	10.8	18.3	11.9	22.7	23.1	23.1	15.6	10.8	5.8	19.4
3	22.0	26.3	20.3	13.0	8.3	8.1	20.8	10.6	23.1	20.1	23.2	21.0	22.3	27.7	25.6	21.5	27.5	29.6	25.8	22.3	18.2	18.8	23.2
4	34.2	33.0	34.9	22.9	22.1	11.6	19.0	6.6	19.5	18.9	32.9	33.3	32.9	38.3	33.5	24.0	28.2	31.4	31.5	28.4	24.6	26.9	23.4
5	12.8	17.1	22.2	17.9	10.0	6.8	16.8	9.8	20.0	18.1	12.5	10.0	12.1	17.9	17.0	18.1	23.0	20.6	17.0	18.6	18.3	18.7	19.1
6	19.5	21.7	25.6	20.2	18.8	8.6	10.1	9.4	11.6	13.4	18.1	17.1	18.1	25.7	21.1	17.4	21.8	22.9	20.2	21.0	20.9	21.2	18.4
7	18.1	22.9	23.1	19.2	26.1	10.4	4.7	12.3	7.3	12.3	19.5	18.1	18.2	14.3	21.2	14.8	20.8	20.1	20.6	20.8	23.1	22.5	18.7
8	36.7	35.8	37.4	32.1	25.4	22.3	20.9	7.6	20.5	20.3	35.4	36.1	36.0	39.1	35.8	12.3	20.4	24.5	29.7	32.1	28.2	30.6	20.2
9	20.7	24.6	24.5	18.8	26.2	11.6	10.1	12.5	0.2	15.2	21.7	20.8	20.5	16.8	22.8	16.2	21.8	21.5	22.0	21.2	24.1	23.1	19.6
10	18.6	23.6	21.4	17.4	21.9	11.2	9.1	12.2	7.0	2.6	21.0	18.6	21.2	17.0	24.1	15.9	25.4	26.0	23.4	19.0	25.5	24.2	20.0
11	1.0	-5.7	5.4	5.9	5.2	7.7	16.3	6.4	19.1	7.7	-16.8	-0.4	-22.1	-4.1	-10.9	5.9	18.2	9.9	10.9	5.4	5.6	-5.6	14.9
12	1.0	-0.7	7.9	7.0	6.9	8.7	18.8	8.1	21.2	14.6	24.8	7.9	16.2	7.8	11.3	10.7	21.1	18.5	20.4	13.2	10.5	3.7	18.4
13	24.6	22.4	20.1	14.8	15.3	14.5	22.0	10.8	23.5	9.9	27.2	28.3	8.8	5.9	-1.5	10.7	22.8	16.2	13.2	6.6	7.9	0.6	18.5
14	35.5	33.0	36.8	26.5	30.2	25.1	32.1	9.8	32.1	23.9	33.4	35.3	28.9	14.4	23.5	6.5	22.1	14.6	20.8	18.4	19.3	20.2	21.5
15	35.5	39.3	30.4	22.5	24.7	21.2	28.3	15.8	27.8	12.2	35.0	37.5	27.2	16.8	18.6	17.0	29.6	24.7	20.3	10.4	12.2	8.7	22.9
16	45.1	43.9	46.1	42.5	38.1	25.0	31.2	10.1	29.4	28.9	44.3	45.2	43.6	46.2	41.5	13.1	21.7	30.8	35.2	35.7	33.1	36.2	24.4
17	49.0	48.4	48.5	41.2	40.5	28.3	38.4	20.8	36.9	36.7	50.3	49.1	50.3	51.2	49.0	24.9	26.6	47.0	45.5	42.4	38.8	42.5	31.9
18	48.1	48.2	47.1	35.4	40.6	31.1	39.8	9.5	39.3	30.7	49.1	48.8	48.3	42.4	45.7	11.2	30.5	28.9	35.2	34.2	31.4	35.8	27.6
19	50.6	52.4	46.4	37.2	41.4	35.9	41.9	24.5	41.2	29.9	51.6	52.3	49.6	40.1	50.3	30.0	39.4	42.0	28.2	34.7	31.1	36.1	32.8
20	40.0	42.9	36.6	30.8	32.6	26.3	25.6	27.0	21.4	22.4	40.3	41.0	39.0	33.9	41.1	30.9	38.3	38.7	36.6	9.4	19.6	25.6	30.3
21	37.0	39.5	34.0	28.6	29.5	18.3	18.1	24.0	17.7	20.8	36.7	37.4	33.7	29.3	32.7	27.1	33.6	32.6	30.6	15.5	13.0	24.4	26.8
22	32.8	36.3	28.6	22.8	24.6	17.5	15.5	19.6	12.1	13.0	31.8	33.1	27.6	23.7	23.1	22.5	30.5	28.1	21.3	6.3	11.1	5.7	22.6
23	41.1	41.4	41.2	33.3	36.1	27.6	33.3	16.4	32.6	29.0	40.8	41.3	39.9	37.7	38.8	19.1	24.7	30.6	31.4	33.5	31.8	34.4	25.4

Table 6-11: Percent Travel Time Difference for Off-Peak SOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-8.1	-7.9	-3.6	-0.4	0.8	2.2	1.7	3.3	0.5	1.4	-15.9	-4.1	-11.1	-7.8	-4.5	1.9	7.6	4.8	6.6	4.3	0.6	-3.8	6.3
2	-8.7	-7.5	-4.2	-1.6	0.0	0.9	1.5	2.9	0.3	0.8	-20.2	-5.7	-11.7	-6.4	-0.2	2.5	8.9	7.6	7.7	7.4	2.0	-1.9	6.7
3	-0.3	0.2	-1.1	3.8	0.7	2.9	1.0	5.4	-0.1	2.5	-2.5	4.5	-2.7	-1.6	1.2	4.7	8.8	7.0	7.2	6.0	1.8	-1.1	7.1
4	2.4	2.9	6.5	4.7	5.0	4.1	5.2	4.5	4.1	5.2	2.6	6.3	2.0	2.1	4.0	5.9	9.4	6.7	7.9	6.2	3.4	2.2	6.9
5	3.8	4.5	1.6	4.0	-1.6	1.2	1.6	3.3	0.1	4.0	2.1	6.5	1.7	0.7	3.1	4.7	7.9	6.7	6.6	6.6	2.9	1.2	6.7
6	4.3	4.6	3.3	3.5	1.3	4.3	3.6	5.1	3.1	5.3	3.3	6.1	2.7	2.6	3.6	3.1	6.1	5.6	6.2	6.1	3.6	2.2	6.1
7	6.7	8.0	4.5	5.4	3.6	3.8	-2.0	5.3	-6.7	4.1	4.3	8.4	4.4	1.4	5.7	4.0	7.6	8.3	8.2	7.1	5.5	3.8	7.6
8	5.9	6.4	7.9	5.4	4.2	4.8	5.4	3.3	5.4	6.3	5.6	8.0	5.4	5.8	6.5	2.7	7.2	6.1	9.1	7.1	5.5	4.4	6.3
9	6.7	7.7	4.9	5.5	3.9	3.6	-2.3	5.8	-6.7	3.6	4.7	8.1	4.7	2.1	5.8	4.2	7.5	8.1	8.0	5.8	5.4	3.6	7.7
10	5.3	5.4	5.8	6.1	6.5	5.8	5.6	6.6	2.3	1.7	1.2	5.6	0.4	2.5	0.5	5.0	8.2	5.3	2.0	2.7	5.3	1.3	6.6
11	-16.3	-15.8	-5.7	-0.9	-0.9	0.5	0.6	2.7	-0.3	-1.4	-46.2	-22.8	-27.2	-10.8	-12.4	1.5	7.5	1.8	3.5	2.8	-0.7	-6.6	5.2
12	-6.1	-5.0	0.1	1.1	2.8	3.1	3.3	4.6	1.9	1.6	-26.4	-6.4	-14.9	-5.0	-6.4	4.5	9.0	4.8	6.7	3.9	0.2	-4.7	7.4
13	-11.1	-9.3	-4.6	-0.7	-0.9	0.6	0.3	3.1	-0.5	-2.3	-22.0	-12.9	-18.3	-7.1	-10.1	2.8	10.6	6.2	4.7	2.0	-1.0	-6.8	6.0
14	-7.4	-4.3	0.6	2.0	1.7	3.9	0.0	3.8	-0.7	0.4	-9.8	-4.6	-7.7	-6.9	-1.6	3.2	10.5	3.1	5.2	3.4	2.3	-1.7	5.4
15	-3.1	1.2	0.5	2.1	0.9	2.1	1.0	4.7	0.2	-1.6	-8.8	-3.3	-8.4	-0.4	-3.3	5.6	13.6	8.5	5.7	3.6	2.3	-2.4	7.2
16	3.8	5.2	7.4	6.9	6.1	3.8	4.1	2.2	3.2	4.3	3.6	6.3	4.1	5.7	6.8	0.7	7.5	7.2	11.6	7.3	6.0	4.0	5.2
17	9.1	11.7	10.7	9.7	9.2	6.9	7.5	6.6	6.6	7.9	10.1	10.7	12.7	11.7	15.7	7.4	9.9	18.2	19.0	13.4	11.1	11.4	8.5
18	8.1	11.4	9.9	6.8	8.6	7.7	7.4	4.2	6.2	4.1	6.6	8.9	8.9	4.1	10.3	4.6	16.6	9.4	12.4	8.8	7.0	6.4	8.0
19	7.4	9.1	8.7	8.4	6.6	6.9	5.3	8.2	4.2	1.7	6.0	8.4	6.3	7.7	7.0	10.7	18.7	14.8	4.2	7.7	6.1	3.8	9.5
20	5.2	6.9	4.4	3.8	3.7	3.5	3.9	5.9	2.9	2.1	4.5	5.8	3.1	5.1	4.9	7.4	13.1	9.4	8.5	3.3	4.3	2.9	8.0
21	4.6	6.9	4.5	3.5	4.3	4.1	5.2	6.1	3.7	3.6	3.8	4.8	3.1	5.7	5.6	7.7	11.9	9.3	6.4	4.1	2.7	5.9	7.9
22	-2.0	1.5	-1.0	-0.3	0.0	1.0	2.3	3.2	1.6	0.0	-4.4	-2.2	-4.8	0.1	-0.6	3.9	10.8	7.0	3.0	1.0	4.6	-0.5	5.9
23	8.4	9.7	9.2	7.3	7.9	6.9	7.2	5.8	6.3	6.1	7.7	9.9	7.9	6.6	8.9	4.9	8.4	8.9	10.2	8.7	7.3	6.5	7.5

Table 6-12: Percent Travel Time Difference for Off-Peak HOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-8.1	-7.9	-3.7	0.4	0.8	2.2	1.7	3.5	0.5	1.5	-15.9	-4.1	-11.1	-7.8	-4.5	1.9	7.6	4.8	6.4	4.3	0.6	-3.8	6.3
2	-8.7	-7.5	-3.9	-1.3	0.0	0.9	1.5	3.0	0.3	0.9	-20.2	-5.7	-11.7	-6.4	-0.2	2.5	8.9	7.6	7.5	7.4	2.0	-1.9	6.7
3	-0.7	0.8	-0.4	3.9	0.6	2.9	1.0	5.4	-0.1	2.6	-2.4	4.1	-2.6	-1.8	1.4	4.7	8.7	6.8	7.0	6.4	2.1	-0.6	7.0
4	2.9	4.2	7.1	4.8	5.0	4.1	5.2	4.6	4.1	5.3	2.8	5.4	2.1	2.3	4.4	5.9	9.4	6.8	7.9	6.6	3.8	2.8	6.9
5	3.8	4.5	1.7	4.0	-1.6	1.2	1.6	3.3	0.1	4.0	2.1	6.5	1.7	0.7	3.1	4.7	7.9	6.7	6.5	6.6	2.9	1.2	6.7
6	4.3	4.6	3.3	3.5	1.3	4.3	3.6	5.1	3.1	5.3	3.3	6.1	2.7	2.7	3.6	3.1	6.1	5.7	6.2	6.1	3.6	2.2	6.2
7	6.7	8.0	4.7	5.4	3.6	3.8	-2.0	5.3	-6.7	4.1	4.3	8.4	4.4	1.4	5.7	4.0	7.6	8.3	8.1	7.1	5.5	3.7	7.6
8	6.3	6.5	7.9	5.5	4.2	4.8	5.4	3.3	5.4	6.3	5.6	7.7	5.4	5.8	6.5	2.7	7.2	6.1	9.1	7.1	5.5	4.4	6.3
9	6.7	7.7	5.0	5.5	3.9	3.6	-2.3	5.8	-6.7	3.6	4.7	8.1	4.7	2.1	5.8	4.2	7.5	8.1	8.0	5.5	5.4	3.7	7.7
10	5.6	5.3	6.1	6.2	6.5	5.8	5.6	6.6	2.3	1.7	1.1	5.7	0.3	2.4	0.3	4.9	8.1	5.1	1.9	2.7	5.3	1.3	6.4
11	-16.3	-15.8	-5.5	-1.0	-0.9	0.5	0.6	2.7	-0.3	-1.5	-46.2	-22.8	-27.2	-10.8	-12.4	1.5	7.5	1.8	3.3	2.8	-0.7	-6.6	5.2
12	-6.1	-5.0	-0.1	1.7	2.8	3.1	3.3	4.8	1.9	1.9	-26.4	-6.4	-14.9	-5.0	-6.4	4.5	9.0	4.8	6.5	3.9	0.2	-4.7	7.5
13	-11.1	-9.3	-4.5	-0.9	-0.9	0.6	0.3	3.1	-0.5	-2.3	-22.0	-12.9	-18.3	-7.1	-10.1	2.8	10.6	6.2	4.5	2.0	-1.0	-6.8	6.0
14	-7.4	-4.3	0.5	1.6	1.7	3.9	0.0	3.8	-0.7	0.3	-9.8	-4.6	-7.7	-6.9	-1.6	3.2	10.5	3.1	5.1	3.4	2.3	-1.7	5.4
15	-3.1	1.2	0.6	1.9	0.9	2.1	1.0	4.7	0.2	-1.7	-8.8	-3.3	-8.4	-0.4	-3.3	5.6	13.6	8.5	5.6	3.6	2.3	-2.4	7.1
16	3.8	5.2	7.4	7.0	6.1	3.8	4.1	2.2	3.2	4.2	3.6	6.3	4.1	5.7	6.8	0.7	7.5	7.2	11.6	7.3	6.0	4.0	5.2
17	9.1	11.7	10.6	9.8	9.2	6.9	7.5	6.6	6.6	7.9	10.1	10.7	12.7	11.7	15.7	7.4	9.9	18.2	19.0	13.4	11.1	11.4	8.5
18	8.1	11.4	9.8	6.6	8.6	7.7	7.4	4.2	6.2	4.0	6.6	8.9	8.9	4.1	10.3	4.6	16.6	9.4	12.4	8.8	7.0	6.4	8.0
19	6.9	8.8	8.3	8.1	6.4	6.7	5.1	8.1	4.0	1.6	5.4	7.9	5.7	7.5	6.7	10.7	18.7	14.8	4.2	7.7	6.1	3.7	9.4
20	5.2	6.9	4.4	3.2	3.7	3.6	4.0	5.9	2.9	2.0	4.5	5.8	3.1	5.1	4.9	7.4	13.1	9.4	8.5	3.3	4.3	2.9	8.0
21	4.6	6.9	4.7	3.4	4.3	4.1	5.2	6.1	3.7	3.6	3.8	4.8	3.1	5.7	5.6	7.7	11.9	9.3	6.4	4.1	2.7	5.9	7.9
22	-2.0	1.5	-0.7	-0.4	0.0	1.0	2.3	3.2	1.7	0.0	-4.4	-2.2	-4.8	0.1	-0.6	3.9	10.8	7.0	3.0	1.0	4.6	-0.5	5.9
23	8.5	9.6	9.2	7.3	7.9	6.9	7.2	5.8	6.3	6.1	7.7	9.9	7.9	6.6	8.9	4.9	8.4	8.9	10.2	8.7	7.3	6.5	7.5

Table 6-13: Percent Distance Difference for Peak Period SOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-1.0	-5.4	-3.7	-1.7	-1.7	0.7	-2.7	-1.7	-2.4	-8.9	1.4	0.6	-1.4	-3.0	-8.3	2.2	0.8	-17.6	-0.7	-1.5	-2.5	-3.1	1.1
2	-8.4	-3.3	-3.0	-1.6	-3.8	-0.4	-1.1	-1.4	-1.1	-7.3	1.5	-1.9	0.7	0.3	-1.8	3.5	-1.6	-10.7	0.6	0.6	-3.8	-3.4	1.6
3	-0.5	0.2	-2.2	-2.2	-3.2	-1.2	-2.1	-3.2	-2.1	-4.4	-3.1	-2.3	-1.2	2.1	0.1	1.6	4.7	0.6	0.0	0.8	-2.8	-1.9	1.9
4	3.8	2.0	1.2	-0.3	3.1	-1.5	-1.2	-1.2	-1.8	-2.9	2.1	3.3	3.0	0.3	2.8	0.8	2.9	4.7	3.5	1.8	0.1	0.9	1.0
5	-1.1	2.5	-3.6	-2.3	-1.0	-0.7	-2.3	-0.5	-2.1	-4.1	-0.2	-0.3	0.8	2.6	-0.1	7.4	8.9	8.0	0.0	0.7	-1.2	-1.4	4.3
6	3.7	6.0	1.3	0.0	-0.8	-1.0	-3.6	-2.0	-2.0	-2.7	2.0	3.9	3.3	3.9	3.7	2.4	3.1	12.5	7.4	3.9	-0.6	2.8	5.9
7	-4.0	-0.5	-2.2	0.1	-1.7	-3.6	-1.9	-2.1	-3.7	-4.7	5.0	-2.2	1.6	0.2	0.4	10.9	17.1	7.6	-0.3	-2.2	-6.4	-4.4	6.0
8	0.0	-0.9	-2.1	-5.4	-0.9	1.6	0.1	0.0	1.2	0.4	-0.6	-0.8	-1.0	1.2	1.0	-0.1	1.3	1.6	1.8	1.1	1.9	-1.1	0.0
9	-3.1	-0.3	-1.3	1.0	-0.9	-1.6	-2.4	-1.2	-0.8	-1.3	4.1	-1.7	0.7	0.2	4.1	9.3	16.3	10.8	7.0	-0.2	-0.4	-0.5	8.2
10	-1.7	-6.3	-4.3	-1.2	-3.1	-2.2	-3.7	-1.6	-1.6	-0.3	1.7	0.3	2.0	3.4	-0.5	6.0	2.1	-0.5	-1.6	0.0	-0.5	-0.8	1.4
11	0.2	-1.3	-1.3	-3.2	-0.4	0.9	-0.6	-1.4	-0.7	-6.5	-3.6	2.5	-7.1	-12.5	-10.4	-6.6	-13.9	-14.1	-1.9	-4.0	-3.3	0.6	-3.5
12	-3.7	-0.1	-1.2	-2.0	-0.4	0.9	-0.6	1.7	-0.7	-7.9	1.5	0.6	-3.2	0.7	-9.8	2.9	-0.2	-18.0	-0.3	-3.6	-1.2	-1.3	1.8
13	-2.4	-2.3	-2.8	-4.1	-1.6	-0.2	-0.9	-0.3	-0.9	-5.4	-3.8	-1.1	-2.1	-7.5	-6.0	-5.5	-9.8	-5.5	-0.8	-3.8	-2.8	-0.7	-1.6
14	3.7	-3.0	4.4	-1.1	3.2	3.0	0.7	0.3	0.3	-1.0	-3.2	2.7	-5.7	-1.5	-2.8	-2.6	4.8	-3.2	-4.5	-2.6	-0.6	1.1	1.0
15	-1.5	-5.0	-1.2	-2.1	-2.1	0.2	-1.8	-1.1	-1.3	-1.6	-2.1	-0.5	0.7	-2.3	-3.3	-7.9	-7.2	-2.7	-2.7	-1.9	-1.7	-1.0	-1.8
16	11.5	8.1	11.3	8.1	11.1	6.0	11.6	-0.2	10.1	8.2	3.2	11.7	-4.9	1.9	-4.5	-2.5	1.8	-0.5	-1.0	-0.8	1.9	1.6	3.2
17	9.9	-1.9	12.5	7.3	16.3	6.7	23.5	-0.1	22.4	15.4	2.7	7.0	0.4	9.9	1.7	0.6	1.1	5.7	2.9	1.0	3.9	4.9	8.6
18	-8.2	-7.2	5.2	5.8	10.0	15.4	8.5	0.3	8.8	3.6	-6.1	-11.4	1.0	-1.8	2.5	-1.9	3.0	-2.7	-0.7	-5.9	1.4	4.6	4.4
19	-3.2	-6.9	-2.3	-1.3	-2.9	3.4	-3.1	-0.6	-2.2	-3.5	-3.1	-3.0	-2.6	-2.9	-4.7	-3.9	-1.1	-3.5	-1.4	5.6	1.7	-3.5	-0.8
20	1.7	-4.5	-1.7	-2.0	-2.8	-0.4	-6.2	-7.8	-2.4	0.0	0.8	3.0	1.6	-0.8	0.0	-11.1	-2.2	-5.7	1.7	0.1	0.0	1.8	-3.9
21	1.3	-1.7	-0.2	4.2	-1.2	-1.8	-6.3	4.2	0.3	-0.4	2.6	2.4	2.8	0.8	-0.8	-1.3	-3.2	-4.7	0.0	3.1	-1.3	4.3	0.3
22	2.9	-5.0	-1.7	1.7	-2.9	0.2	-5.4	2.9	0.7	0.9	5.7	5.2	5.3	3.8	-0.9	0.7	-5.3	-1.0	-1.5	0.4	-2.5	-0.1	0.7
23	4.5	1.8	5.4	3.6	6.5	8.0	7.6	-1.1	7.6	4.9	3.3	3.8	1.0	2.8	0.7	2.2	5.1	5.2	2.6	1.0	2.0	1.3	4.1

Table 6-14: Percent Distance Difference for Peak Period HOV2 Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0.2	-4.8	-3.5	-3.8	-1.7	0.7	-2.7	-2.9	-2.4	-8.3	5.6	4.8	1.0	-2.9	-7.6	2.2	0.9	-17.5	0.0	-1.5	-2.6	-3.2	0.9
2	-7.5	-3.2	-2.5	-3.5	-3.8	-0.4	-1.1	-1.6	-1.1	-6.9	1.5	-1.8	0.7	0.5	-1.8	3.6	-1.5	-10.7	2.0	0.6	-3.8	-3.4	1.9
3	3.0	-0.9	-1.3	-2.9	-3.2	-1.2	-2.1	-3.2	-2.1	-4.4	0.6	3.5	-0.5	3.4	-3.4	3.4	5.6	-0.7	-1.6	0.6	-2.5	-1.5	2.4
4	2.8	2.1	4.7	-0.2	3.5	-1.6	-1.0	-1.3	-1.7	-2.7	1.4	3.8	1.4	0.4	-0.4	0.7	2.9	3.7	0.4	3.3	0.7	2.5	0.5
5	3.7	5.1	3.6	-1.0	-0.8	-0.7	-2.1	-0.5	-2.1	-4.1	2.5	3.9	1.6	6.4	-0.3	7.5	9.0	4.5	-1.0	3.0	-0.4	1.2	4.2
6	3.0	7.2	5.0	0.5	-0.5	-1.0	-3.5	-2.0	-2.0	-2.7	1.8	4.0	1.0	5.3	1.8	2.4	3.1	6.6	0.4	4.0	-2.1	2.1	2.6
7	2.8	-2.1	3.8	1.5	-1.6	-3.6	-1.9	-2.1	-3.7	-4.7	4.0	1.8	3.7	4.8	1.1	10.9	17.1	7.7	1.0	-1.9	-6.4	-4.3	6.4
8	-2.0	-2.8	2.8	-5.4	-0.8	1.6	0.1	0.0	1.2	0.4	-1.2	-1.8	-3.8	0.7	-2.1	-0.1	1.3	1.7	0.8	0.7	-4.2	-2.7	0.0
9	1.9	-1.5	3.6	2.3	-0.9	-1.6	-2.4	-1.2	-0.8	-1.3	0.2	-1.2	1.2	4.4	5.2	9.3	16.3	12.1	7.5	0.0	-0.4	-0.5	8.2
10	-1.2	-6.2	-1.0	0.2	-3.1	-2.2	-3.7	-1.6	-1.6	-0.3	1.8	0.9	2.1	4.3	-0.4	5.7	1.6	-0.5	-1.6	0.0	-0.5	-0.8	1.1
11	0.2	-1.3	-1.5	-5.8	-0.4	0.9	-0.6	-2.3	-0.7	-5.7	-3.6	2.5	-7.1	-12.5	-10.3	-6.6	-13.9	-14.1	-1.8	-4.0	-3.3	0.6	-3.4
12	-3.7	-0.1	-1.2	-4.1	-0.4	0.9	-0.6	0.6	-0.7	-7.3	1.5	0.6	-3.2	0.7	-9.7	2.9	-0.2	-18.0	-0.2	-3.6	-1.2	-1.3	1.8
13	-2.5	-2.3	-3.0	-8.0	-1.6	-0.1	-0.9	-1.7	-0.9	-4.7	-3.8	-1.1	-2.1	-7.5	-6.0	-5.5	-9.8	-5.5	-0.7	-3.8	-2.8	-0.7	-1.6
14	1.2	-2.3	4.5	-4.0	3.5	3.6	0.8	-0.4	0.5	-0.3	-3.5	3.1	-5.6	-1.4	-2.5	-2.6	4.8	-3.3	-4.3	-2.4	-0.4	1.4	1.0
15	-1.6	-4.9	-1.1	-5.0	-2.1	0.4	-1.7	-1.7	-1.1	-1.4	-2.2	-0.5	0.8	-2.3	-3.3	-7.9	-7.2	-2.7	-2.2	-1.9	-1.7	-0.9	-1.7
16	10.5	8.6	11.8	8.1	11.2	6.0	12.0	-0.2	10.1	8.1	2.0	11.7	-4.6	2.0	-4.3	-2.5	1.8	-0.5	-1.0	-0.9	1.9	1.4	3.2
17	10.0	-0.4	14.6	7.3	16.7	6.7	23.7	-0.1	22.4	15.0	2.8	7.9	0.5	10.0	1.7	0.6	1.1	5.7	2.9	1.0	3.9	4.9	8.6
18	-8.1	-7.1	6.0	5.6	10.3	15.5	8.7	0.3	8.9	3.6	-6.1	-11.4	1.0	-1.8	2.5	-1.9	3.0	-2.7	-0.7	-5.9	1.4	4.6	4.4
19	1.6	-0.8	-0.4	-2.8	-1.0	3.9	-0.4	-0.7	0.4	-2.9	-0.4	2.4	-0.5	-3.3	-3.3	-3.9	-1.1	-3.5	-1.4	5.6	1.7	-3.5	-0.4
20	1.7	-4.5	-1.7	-3.8	-2.8	-0.1	-5.6	-7.6	-1.9	0.0	0.8	3.0	1.6	-0.8	0.0	-11.1	-2.2	-5.7	1.7	0.1	0.0	1.8	-3.8
21	1.3	-1.7	0.0	2.2	-0.9	-1.8	-6.1	3.4	0.3	-0.4	2.6	2.4	2.8	0.8	-0.8	-1.3	-3.2	-4.7	0.0	3.1	-1.3	4.3	0.2
22	3.0	-5.0	-1.3	0.0	-2.9	0.4	-5.0	3.5	1.0	0.9	5.7	5.2	5.3	3.8	-0.9	0.7	-5.3	-1.0	-1.5	0.4	-2.5	-0.1	0.8
23	6.5	3.5	8.1	3.7	7.2	8.1	8.6	-1.1	8.5	5.0	4.6	6.0	1.6	3.9	0.6	2.2	5.1	4.5	1.5	1.0	0.9	1.1	3.8

Table 6-15: Percent Distance Difference for Peak Period HOV3+ Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	2.0	-4.7	-3.5	-1.1	-1.7	0.7	-2.7	-2.2	-2.4	-8.4	8.1	9.1	3.3	-2.0	-7.0	2.4	1.1	-16.9	0.6	-1.9	-2.6	-3.2	1.0
2	6.5	-2.1	-1.8	-1.1	-3.8	-0.4	-1.1	1.5	-1.1	-6.3	10.0	13.1	6.7	3.8	-2.2	4.8	-1.4	-15.4	1.8	0.6	-3.8	-3.4	2.3
3	10.5	13.4	2.7	2.0	-3.2	-1.2	-2.1	0.6	-2.1	-4.4	8.3	10.6	6.8	8.2	1.7	6.8	6.6	-1.7	2.5	2.6	-0.8	1.1	4.2
4	4.3	5.8	4.4	0.0	3.1	-1.6	-1.4	-1.2	-2.1	-2.8	8.8	7.3	6.4	0.5	2.2	1.0	3.1	3.8	2.0	4.4	0.5	2.3	0.6
5	9.7	21.9	13.4	2.9	-0.8	-0.7	-2.1	0.1	-2.1	-4.1	6.9	9.3	8.3	13.7	9.9	6.3	3.0	4.0	6.3	9.2	9.0	13.4	4.3
6	9.4	19.4	8.0	0.5	-0.5	-1.0	-3.5	-2.0	-2.0	-2.7	9.1	12.0	10.2	12.9	11.0	2.5	2.8	9.0	8.1	9.5	-1.3	5.3	3.9
7	-0.5	9.3	9.1	6.0	-1.6	-3.6	-1.9	-1.7	-3.7	-4.7	-1.0	-0.8	-0.7	-1.2	1.4	1.8	-1.6	-9.2	0.6	-0.5	-4.9	-1.4	0.9
8	2.2	3.4	3.7	-5.4	-0.5	1.6	0.1	0.0	1.2	0.4	4.3	5.8	5.5	1.3	2.4	-0.1	1.3	1.7	1.4	2.3	-4.2	-2.0	0.0
9	-0.2	7.8	7.8	5.1	-0.9	-1.6	-2.4	-1.2	-0.8	-1.3	-0.6	-0.5	-0.4	-0.9	1.2	0.9	-1.4	-7.9	0.7	0.3	-0.4	-0.7	1.2
10	1.8	9.4	5.8	3.9	-3.1	-2.2	-3.7	-1.6	-1.6	-0.3	4.7	1.9	5.3	2.3	1.1	2.1	2.6	-1.2	0.2	-0.1	-0.5	-0.8	1.8
11	0.2	-1.3	-1.5	-3.2	-0.4	0.9	-0.6	-2.1	-0.7	-5.3	-3.6	2.5	-7.1	-12.5	-10.3	-6.6	-13.9	-14.1	-1.8	-4.0	-3.3	0.6	-3.4
12	-3.7	-0.1	-1.2	-1.5	-0.4	0.9	-0.6	0.6	-0.7	-7.4	1.5	0.6	-3.2	0.7	-9.7	2.9	-0.2	-18.0	-0.2	-3.6	-1.2	-1.3	1.7
13	-2.2	-2.3	-3.0	-5.3	-1.6	0.0	-0.8	-1.5	-0.9	-4.4	-3.8	-0.6	-2.1	-7.4	-6.0	-5.4	-9.8	-5.5	-0.7	-3.8	-2.8	-0.7	-1.6
14	1.2	-2.3	4.5	-2.4	3.5	3.6	0.8	-0.1	0.5	-0.1	-3.5	3.1	-5.6	-1.4	-2.5	-2.6	4.8	-3.3	-4.3	-2.4	-0.4	1.4	1.0
15	-1.5	-4.9	-1.2	-3.1	-2.1	0.4	-1.6	-1.5	-1.0	-1.1	-2.2	-0.3	0.8	-2.2	-3.3	-7.8	-7.2	-2.7	-2.2	-1.9	-1.7	-0.9	-1.7
16	10.7	8.8	11.5	8.1	10.6	6.0	11.4	-0.2	9.2	7.6	1.9	11.7	-4.6	2.0	-4.2	-2.5	1.8	-0.5	-1.0	-0.9	1.9	1.4	2.9
17	10.0	-0.4	14.7	7.3	16.5	6.7	23.7	-0.1	22.4	14.8	2.8	7.9	0.5	10.0	1.7	0.6	1.1	5.7	2.9	1.0	3.9	4.9	8.6
18	-8.1	-7.1	6.1	5.7	10.3	15.6	8.7	0.3	8.9	4.0	-6.1	-11.4	1.0	-1.8	2.5	-1.9	3.0	-2.7	-0.7	-5.9	1.4	4.6	4.4
19	1.6	-0.8	-0.4	-2.5	-1.0	3.9	-0.4	-0.8	0.5	-2.7	-0.4	2.4	-0.5	-3.3	-3.3	-3.9	-1.1	-3.5	-1.4	5.6	1.7	-3.5	-0.4
20	1.7	-4.5	-1.8	-2.7	-2.8	0.0	-5.3	-7.4	-1.7	0.0	0.8	3.0	1.6	-0.8	0.0	-11.1	-2.2	-5.7	1.7	0.1	0.0	1.8	-3.7
21	1.3	-1.7	0.0	2.8	-0.7	-1.8	-6.0	3.5	0.3	-0.4	2.6	2.4	2.8	0.8	-0.8	-1.3	-3.2	-4.7	0.0	3.1	-1.3	4.3	0.2
22	3.0	-5.0	-1.3	1.2	-2.9	0.5	-4.8	3.1	1.0	0.9	5.7	5.2	5.3	3.8	-0.9	0.7	-5.3	-1.0	-1.5	0.4	-2.5	-0.1	0.9
23	7.5	7.7	9.1	4.4	7.1	8.1	8.5	-1.1	8.3	4.7	6.3	8.4	4.8	3.5	2.0	-0.3	0.5	0.3	1.3	2.1	1.0	1.5	2.9

Table 6-16: Percent Distance Difference for Off-Peak SOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-3.9	-6.8	-3.3	-4.1	-1.4	-2.1	-0.9	-1.2	-0.8	-4.0	-5.2	-5.4	-6.2	-9.9	-7.8	-3.5	-6.0	-15.3	-1.7	-2.9	-1.0	-0.8	-3.1
2	-5.6	-4.6	-3.5	-5.8	-0.5	-1.0	-0.7	-1.5	-0.7	-4.3	-2.8	2.3	-3.8	-7.6	-5.1	-0.9	-8.2	-14.4	0.4	0.9	-2.3	-2.2	-2.3
3	-3.0	-7.8	-2.6	-5.6	-2.6	-3.0	-1.1	-3.0	-0.9	-2.3	-1.6	-0.9	-2.5	-2.1	-3.6	-0.1	-1.5	-3.6	-0.8	-0.7	-2.7	-2.8	-0.8
4	0.7	-5.8	-3.5	-1.9	-2.9	-2.1	-1.8	-3.2	-1.1	-1.8	-2.4	-0.6	-3.0	-2.9	-4.2	-1.2	-1.4	-1.2	-1.6	-2.2	-3.5	-3.7	-1.3
5	-1.9	-3.5	-3.0	-3.0	-1.2	-1.0	-0.8	-4.1	-0.2	-1.8	-1.0	-0.6	-1.6	-1.6	-2.4	-0.3	-1.7	-3.5	-0.5	-0.3	-2.3	-2.2	-0.7
6	-2.3	-4.6	-3.3	-2.3	-1.4	-0.3	-0.9	-1.3	-0.5	-1.0	-1.2	-1.2	-2.0	-2.0	-2.9	-1.7	-1.9	-2.4	-1.8	-2.0	-3.3	-3.2	-1.6
7	-0.7	-1.0	-1.3	-1.9	-0.6	-1.0	-0.5	-1.4	-1.7	-2.5	-0.9	-0.2	-1.0	-1.7	-1.8	-0.3	-2.0	-5.6	0.5	-2.9	-5.2	-3.0	-1.1
8	1.0	-1.4	-1.4	-1.7	-3.7	-1.2	-2.4	-0.1	-0.5	-0.6	0.0	1.4	-0.4	-0.6	-0.9	-1.3	-1.1	-0.3	-0.1	-1.7	-2.5	-2.5	-1.5
9	-0.6	-0.8	-1.0	-1.6	-0.4	-0.9	-2.5	-1.0	-0.6	-2.1	-0.7	-0.1	-0.8	-1.5	-1.6	-0.5	-1.8	-5.0	0.5	-1.6	-0.8	-1.8	-0.9
10	-5.0	-5.6	-2.7	-2.6	-1.6	-1.3	-2.5	-1.0	-1.9	-0.1	-6.5	-5.7	-6.4	-2.8	-3.3	-1.7	-1.0	-1.3	-2.9	-0.8	-0.2	0.4	-1.4
11	-4.2	-4.2	-2.3	-4.1	-0.9	-1.8	-0.9	-2.0	-0.8	-5.2	-10.5	-3.7	-10.8	-11.5	-10.7	-4.2	-14.0	-17.8	-2.6	-3.8	-1.1	-1.2	-4.8
12	-5.9	1.0	-1.9	-4.1	-0.9	-2.0	-0.5	-0.7	-0.5	-4.6	-4.3	1.4	-6.0	-8.5	-8.1	-1.2	-8.0	-21.9	-0.8	-3.3	-0.7	1.2	-3.1
13	-4.5	-3.0	-2.6	-4.9	-1.2	-2.2	-0.4	-1.5	-0.5	-4.9	-8.9	-4.0	-5.5	-8.6	-6.8	-3.0	-8.5	-5.1	-1.5	-3.8	-1.2	-1.7	-2.4
14	-12.2	-8.5	-2.5	-4.9	-1.4	-2.2	-1.5	-1.1	-1.3	-2.7	-8.5	-11.5	-8.1	-3.8	-5.3	-2.3	-0.7	-4.2	-4.4	-2.8	-1.6	-4.7	-1.8
15	-5.6	-3.4	-2.2	-4.9	-1.1	-2.4	-0.5	-1.2	-0.4	-1.9	-8.6	-5.5	-5.9	-5.3	-3.7	-2.8	-2.4	-0.9	-2.4	-1.6	-0.8	-2.1	-0.8
16	-3.9	-0.9	0.0	-1.3	-0.4	0.4	-0.1	-1.1	-0.5	-1.7	-3.2	-1.5	-3.2	-2.0	-3.4	-1.7	-1.1	-1.7	-0.4	-2.6	-1.3	-3.7	-1.1
17	-6.9	-6.8	-2.0	-1.5	-1.8	-1.4	-0.9	-0.9	-0.9	-2.3	-16.3	-8.3	-11.0	-1.4	-5.4	-1.4	-0.5	2.2	2.4	-5.0	-3.3	-4.8	-1.0
18	-15.6	-7.5	-3.8	-2.4	-1.7	-3.7	-1.6	-0.4	-1.4	-0.7	-18.4	-21.1	-6.7	-6.3	-2.0	-2.1	1.7	-3.2	-1.3	-2.9	0.0	-0.5	-1.6
19	2.1	4.5	0.7	-2.3	0.2	-2.8	3.2	-0.7	2.7	-1.5	2.1	3.3	2.5	-3.6	0.8	-1.5	0.2	-4.2	-2.4	-0.9	-1.5	-1.2	-0.8
20	-3.1	-0.9	-0.4	-3.2	-0.4	-1.2	-2.1	-0.6	-1.1	-0.7	-3.5	-3.2	-2.6	-1.3	-0.5	-1.3	-0.2	-0.6	-0.2	-0.3	-0.3	-0.2	-0.3
21	-0.9	-2.1	-1.5	-3.6	-1.3	-2.5	-4.5	-2.2	-1.0	-0.6	-0.5	-0.4	-1.3	-2.2	-1.6	-1.4	-0.7	0.3	-1.7	-0.2	-0.2	-0.2	-0.9
22	-1.3	-2.6	-1.6	-4.0	-1.2	-1.7	-2.0	-3.3	-0.3	0.1	-0.7	-0.4	-1.7	-4.4	-2.7	-3.0	-1.6	0.2	-1.9	0.0	-0.1	0.4	-1.0
23	-2.9	-1.9	-1.2	-1.3	-1.1	-1.1	-1.0	-0.9	-0.6	-1.1	-4.7	-2.6	-2.6	-1.8	-0.8	-1.2	-1.1	-2.1	-0.4	-1.3	-0.8	-1.4	-1.2

Table 6-17: Percent Distance Difference for Off-Peak HOV Skims with and without INRIX Data

Jurisdiction	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	-3.9	-6.8	-3.3	-4.0	-1.4	-2.1	-0.9	-1.0	-0.8	-4.0	-5.2	-5.4	-6.2	-9.9	-7.7	-3.5	-6.0	-15.3	-1.6	-2.9	-1.0	-0.8	-3.0
2	-5.6	-4.6	-3.8	-6.8	-0.5	-1.0	-0.7	-1.4	-0.7	-4.4	-2.8	2.3	-3.8	-7.6	-5.1	-0.9	-8.2	-14.4	0.5	0.9	-2.3	-2.2	-2.3
3	-3.1	-8.3	-3.0	-6.1	-2.6	-3.0	-1.1	-3.1	-0.9	-2.5	-1.8	-1.0	-2.7	-2.3	-3.8	-0.2	-1.6	-3.7	-1.0	-0.9	-2.8	-3.0	-0.9
4	-2.7	-8.6	-4.5	-1.8	-2.9	-2.1	-1.8	-3.2	-1.1	-1.8	-2.6	-1.1	-3.5	-3.5	-5.0	-1.2	-1.3	-1.4	-1.9	-3.2	-4.4	-4.9	-1.4
5	-1.9	-3.5	-3.1	-3.0	-1.2	-1.0	-0.8	-4.1	-0.2	-1.8	-1.0	-0.6	-1.6	-1.6	-2.4	-0.3	-1.7	-3.5	-0.5	-0.3	-2.3	-2.2	-0.7
6	-2.3	-4.6	-3.3	-2.3	-1.4	-0.3	-0.9	-1.3	-0.5	-1.0	-1.2	-1.2	-2.0	-1.7	-2.9	-1.7	-1.9	-2.3	-1.7	-1.9	-3.3	-3.2	-1.6
7	-0.7	-1.0	-1.4	-2.0	-0.6	-1.0	-0.5	-1.4	-1.7	-2.5	-0.9	-0.2	-1.0	-1.7	-1.8	-0.3	-2.0	-5.6	0.5	-2.8	-4.9	-3.0	-1.1
8	-0.1	-1.6	-1.4	-1.7	-3.7	-1.2	-2.4	-0.1	-0.5	-0.6	0.0	1.1	-0.4	-0.6	-0.9	-1.3	-1.1	-0.3	-0.1	-1.7	-2.5	-2.5	-1.5
9	-0.6	-0.8	-1.1	-1.6	-0.4	-0.9	-2.5	-1.0	-0.6	-2.1	-0.7	-0.1	-0.8	-1.5	-1.6	-0.5	-1.8	-5.0	0.5	-3.1	-0.8	-1.5	-0.9
10	-4.7	-6.2	-2.6	-2.7	-1.6	-1.3	-2.5	-1.0	-1.9	-0.1	-7.3	-5.6	-7.1	-3.1	-4.0	-1.9	-1.2	-2.2	-3.3	-0.8	-0.2	0.4	-1.8
11	-4.2	-4.2	-2.4	-4.1	-0.9	-1.8	-0.9	-2.0	-0.8	-5.7	-10.5	-3.7	-10.8	-11.5	-10.7	-4.2	-14.0	-17.8	-2.4	-3.8	-1.1	-1.2	-4.8
12	-5.9	1.0	-1.9	-4.0	-0.9	-2.0	-0.5	-0.3	-0.5	-4.1	-4.3	1.4	-6.0	-8.5	-8.0	-1.2	-8.0	-21.9	-0.7	-3.3	-0.7	1.2	-3.1
13	-4.5	-3.0	-2.6	-5.0	-1.2	-2.2	-0.4	-1.5	-0.5	-5.3	-8.9	-4.0	-5.5	-8.6	-6.8	-3.0	-8.5	-5.1	-1.3	-3.8	-1.2	-1.7	-2.5
14	-12.2	-8.5	-2.6	-5.1	-1.4	-2.2	-1.5	-1.1	-1.3	-2.9	-8.5	-11.5	-8.1	-3.8	-5.3	-2.3	-0.7	-4.2	-4.3	-2.8	-1.6	-4.7	-1.8
15	-5.6	-3.4	-2.4	-5.6	-1.1	-2.4	-0.4	-1.2	-0.4	-2.2	-8.6	-5.5	-5.9	-5.3	-3.7	-2.8	-2.4	-0.9	-2.2	-1.6	-0.8	-2.1	-0.8
16	-3.9	-0.9	-0.1	-1.3	-0.4	0.4	-0.1	-1.1	-0.5	-1.9	-3.2	-1.5	-3.2	-2.0	-3.4	-1.7	-1.1	-1.7	-0.4	-2.6	-1.3	-3.7	-1.1
17	-6.9	-6.8	-2.1	-1.4	-1.8	-1.4	-0.9	-0.9	-0.9	-2.1	-16.3	-8.3	-11.0	-1.4	-5.4	-1.4	-0.5	2.2	2.4	-5.0	-3.3	-4.8	-0.9
18	-15.6	-7.5	-3.8	-2.5	-1.7	-3.7	-1.6	-0.4	-1.4	-1.2	-18.4	-21.1	-6.7	-6.3	-2.0	-2.1	1.7	-3.2	-1.3	-2.9	0.0	-0.5	-1.6
19	2.3	4.7	1.1	-1.9	0.3	-2.8	3.5	-0.5	3.0	-2.1	2.4	3.3	2.7	-2.6	1.1	-1.3	0.2	-4.2	-2.4	-0.9	-1.5	-1.1	-0.7
20	-3.1	-0.9	-0.7	-4.7	-0.4	-1.1	-1.6	-0.6	-1.0	-0.8	-3.5	-3.2	-2.6	-1.3	-0.5	-1.3	-0.2	-0.6	-0.2	-0.3	-0.3	-0.2	-0.3
21	-0.9	-2.1	-1.6	-4.5	-1.3	-2.5	-4.1	-2.2	-1.0	-0.6	-0.5	-0.4	-1.3	-2.2	-1.6	-1.4	-0.7	0.3	-1.7	-0.2	-0.2	-0.2	-0.9
22	-1.3	-2.6	-1.8	-5.3	-1.1	-1.6	-1.5	-3.3	0.0	0.1	-0.7	-0.4	-1.7	-4.4	-2.7	-3.0	-1.6	0.2	-1.9	0.0	-0.1	0.4	-1.0
23	-2.9	-1.7	-0.9	-1.3	-0.6	-1.0	-0.1	-0.9	0.2	-1.3	-4.5	-2.5	-2.7	-1.8	-0.8	-1.2	-1.1	-2.1	-0.3	-1.4	-0.8	-1.2	-1.1

In the next step, we calculated the average path travel time from each zone to all other zones, using both skims. Similarly, we calculated the average path travel time from all zones to each zone, using both skims. The difference between the average travel time based on the origin or destination zone is presented into Figure 6-3 through Figure 6-6 for different modes and different time periods.

Figure 6-2 to Figure 6-4 show the results for peak period. According to these figures, trips from outer zones have higher travel time differences because these trips are more likely to use freeways and model speeds for freeways are lower than INRIX speeds. As a result, the model travel time is much longer than the INRIX travel time. In many respects, this is the natural consequence of a static assignment that permits volume-to-capacity ratios to exceed 1.0.

Figure 6-5 and Figure 6-6 show the results for the off-peak period. Comparing these figures with the previous figures indicates that off-peak travel times are very close in both skims. The highest difference during the off-peak period is less than 20 minutes while during the peak period the difference is greater than 30 minutes for a significant number of zones. Similar to the peak period, trips from outer zones have higher travel time differences than other zones. As was the case for the peak period, these trips are more likely to use freeways, and model speeds for freeways are lower than INRIX speeds.

Figure 6-2: Peak Period SOV Travel Time Differences

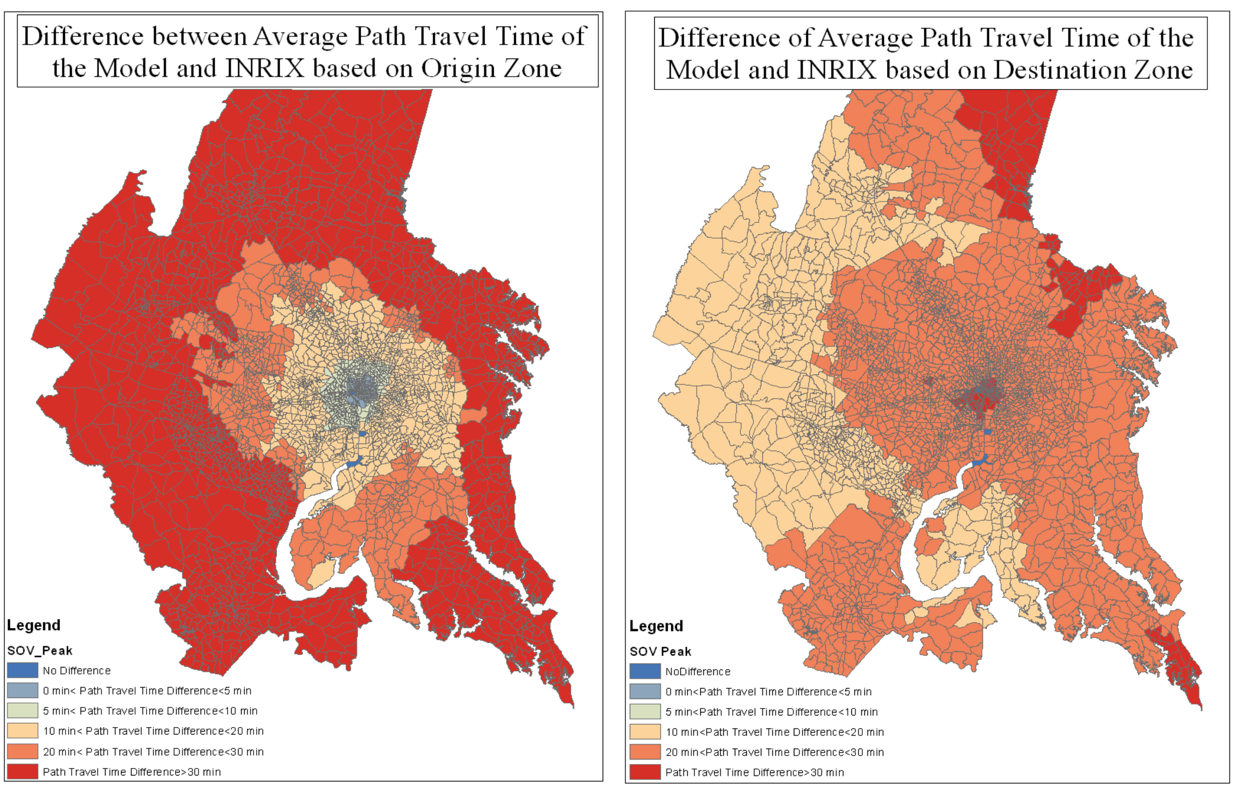


Figure 6-3: Peak Period HOV2 Travel Time Differences

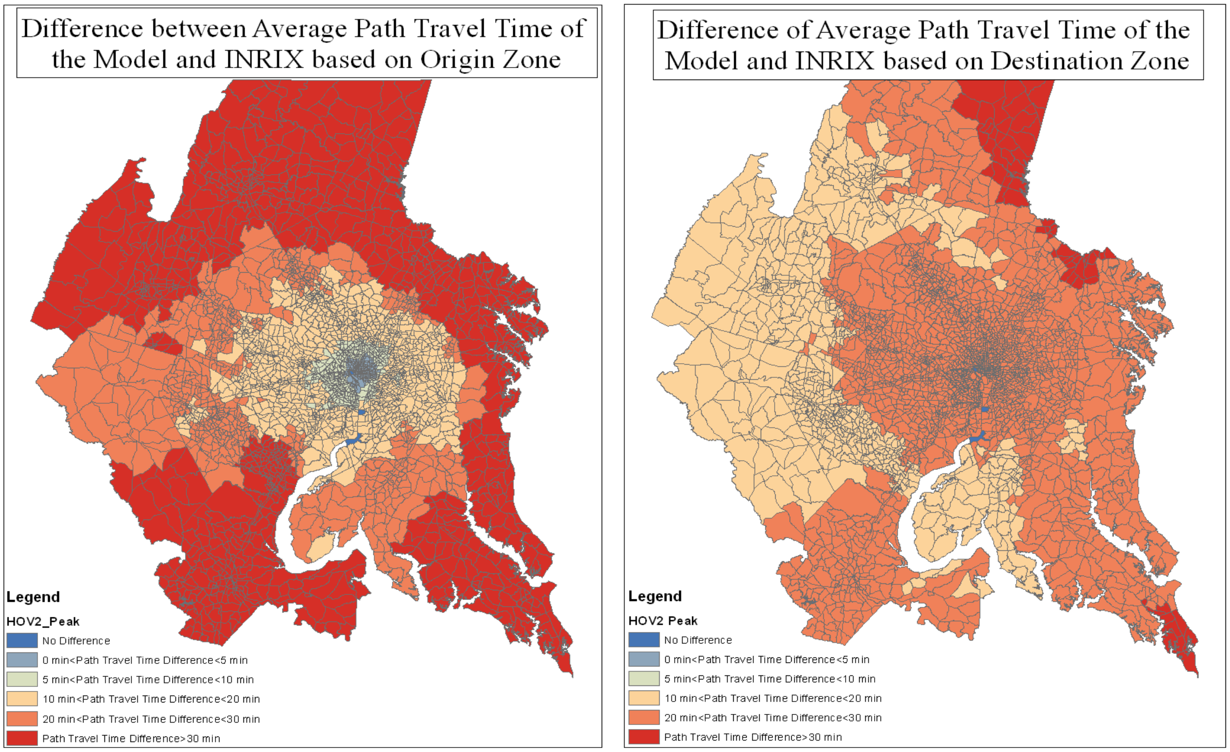


Figure 6-4: Peak Period HOV3+ Travel Time Differences

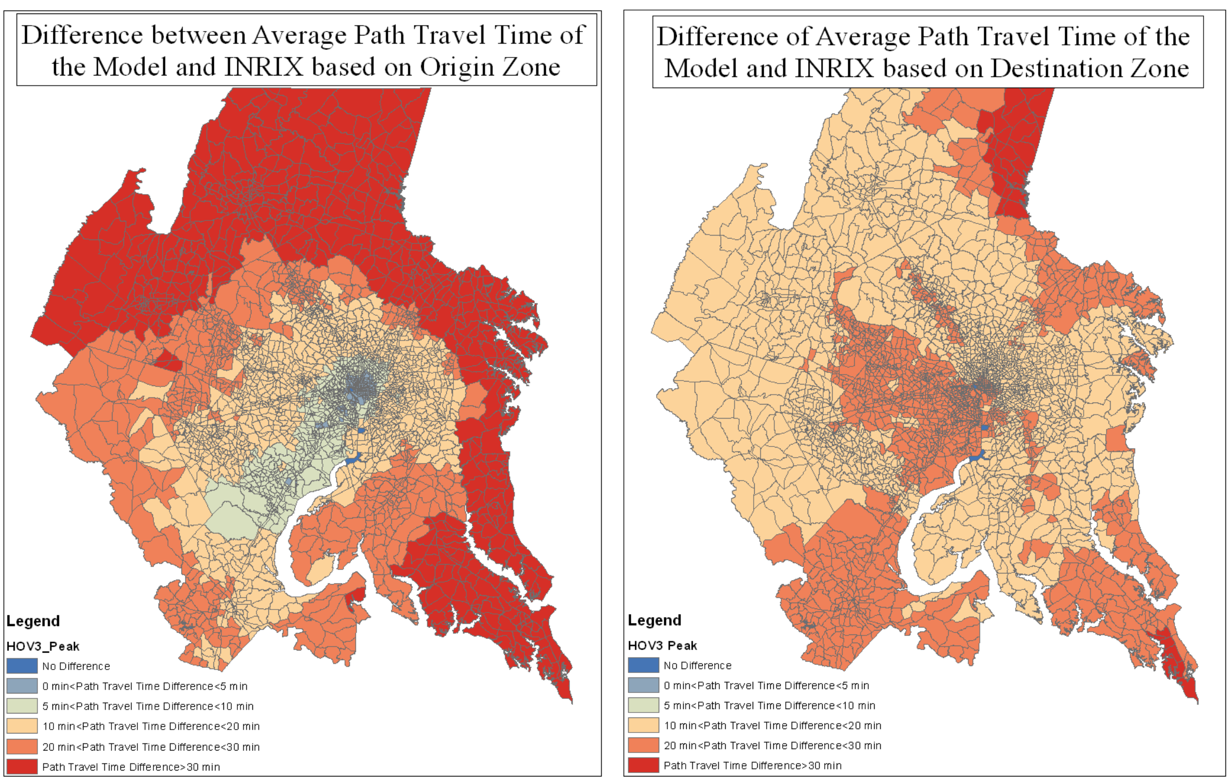


Figure 6-5: Off Peak Period SOV Travel Time Differences

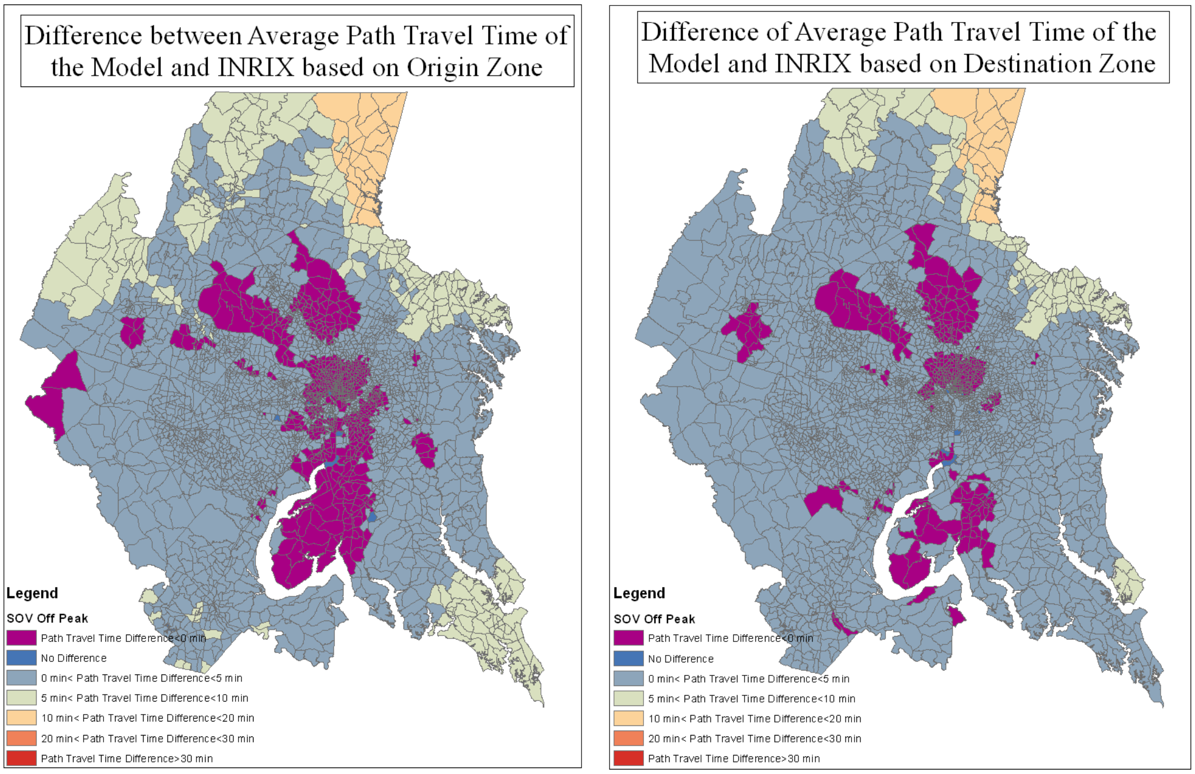
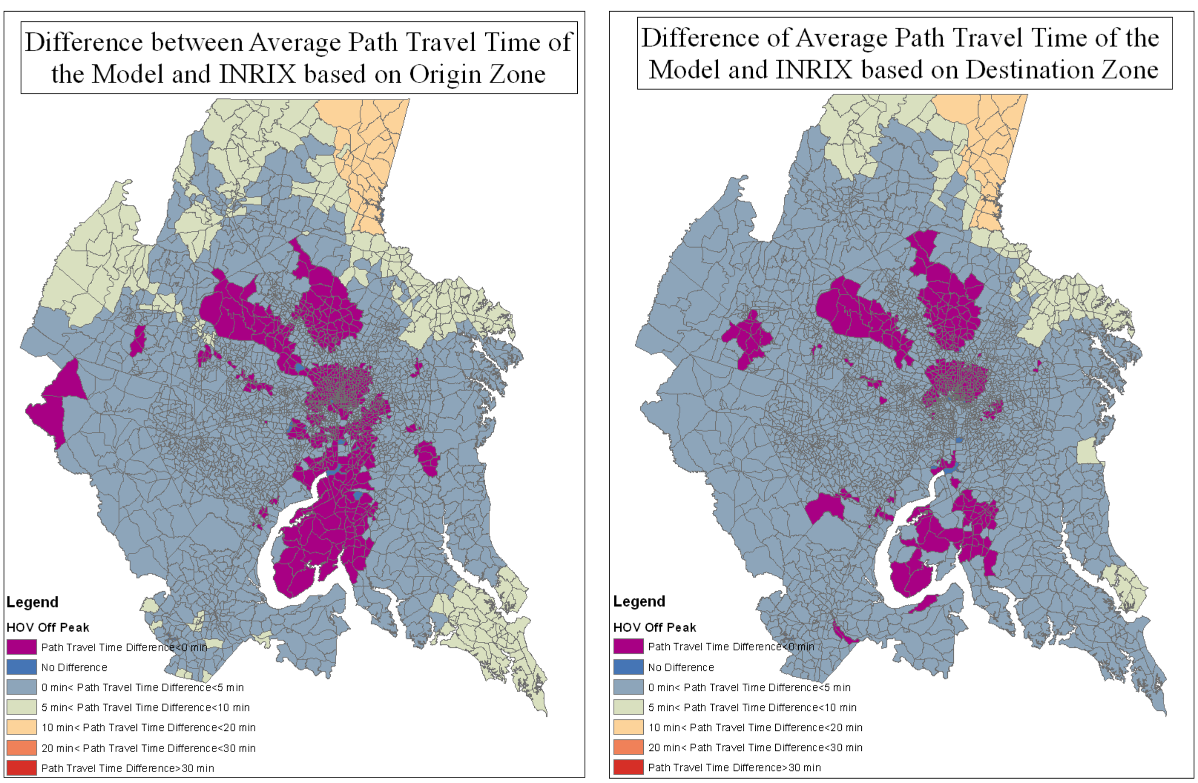


Figure 6-6: Off Peak HOV Travel Time Differences



6.4 INRIX Speed Profile over Major Freeways

In this section, the INRIX speed profile of major freeways in the region is presented. Figure 6-7 through Figure 6-16 show the hourly speed for I-66, I-95 in Virginia, I-270, I-395 in Virginia, and I-495 in Virginia by direction. The average speed by direction for the four time-of-day periods included in the MWCOG traffic assignment is also shown in these figures. The estimated average speed by direction from the model is shown as a vertical bar. Instead of making the vertical bars the width of the time-of-day period (e.g., three hours for the AM peak), each vertical bar has been made one hour wide and is placed in the approximate middle of the time-of-day period.

An analysis of Figure 6-7 through Figure 6-16 suggests that the MWCOG model speeds follow the same pattern as INRIX speeds and for off-peak periods the difference between the MWCOG speeds and the INRIX speeds are negligible. However, most of the freeways have a significant difference for the AM peak or PM peak periods. The model performance on I-66 during the AM and PM peak periods is better than other freeways.

Figure 6-7: Speed Profile of I-66 Eastbound based on INRIX Data

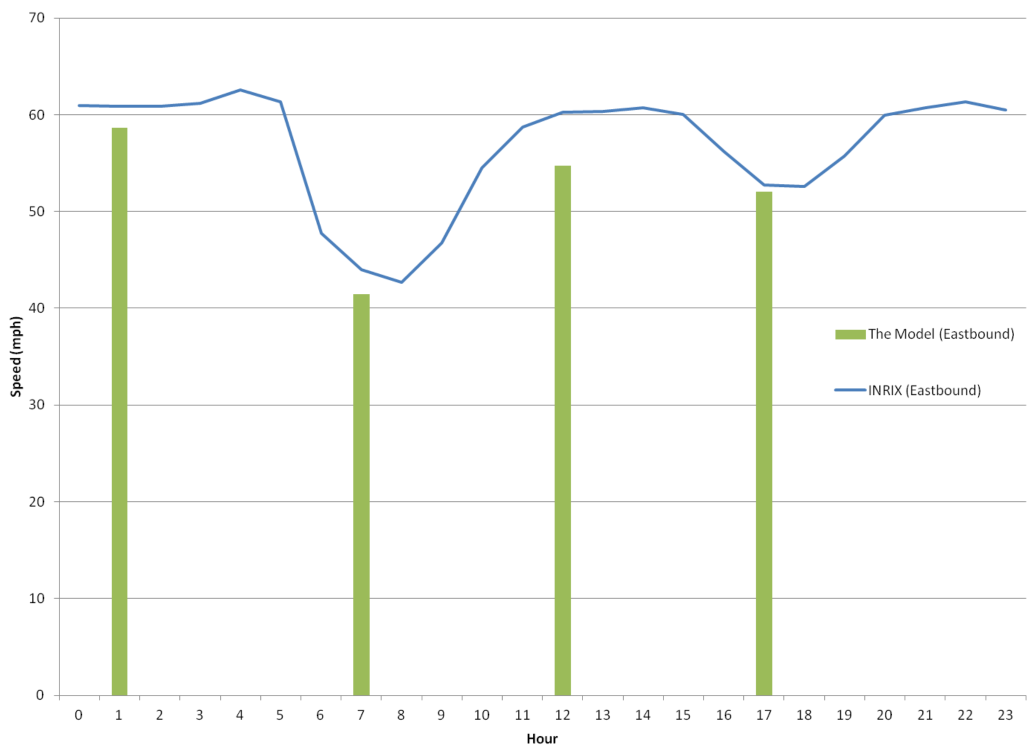


Figure 6-8: Speed Profile of I-66 Westbound based on INRIX Data

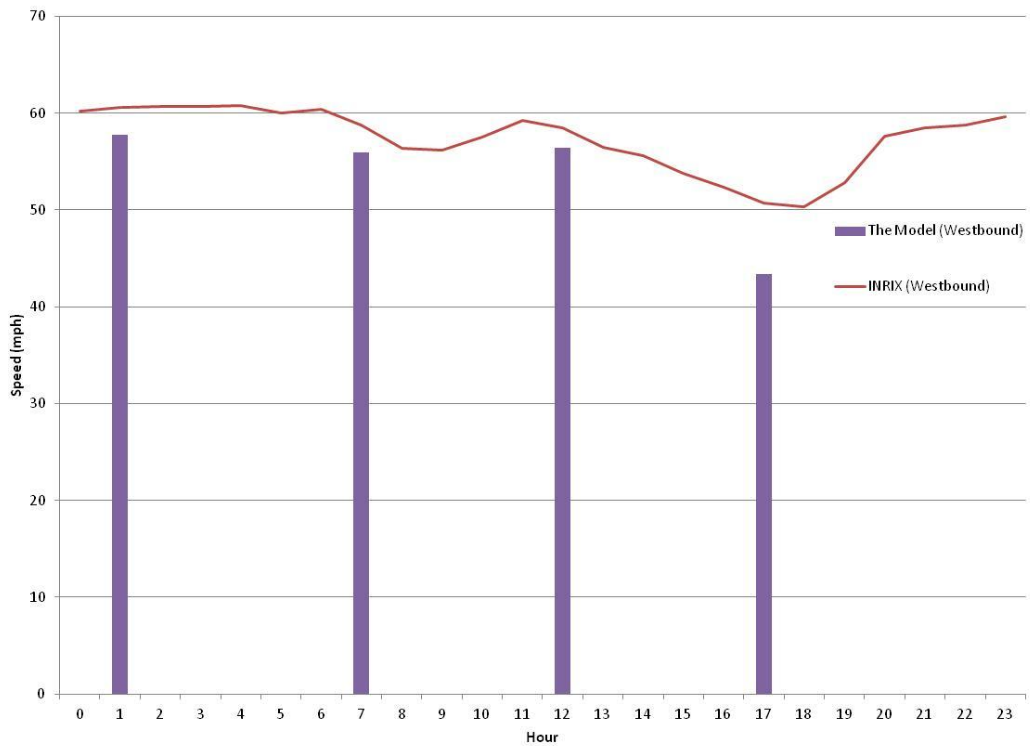


Figure 6-9: Speed Profile of I-95 Northbound in Virginia based on INRIX Data

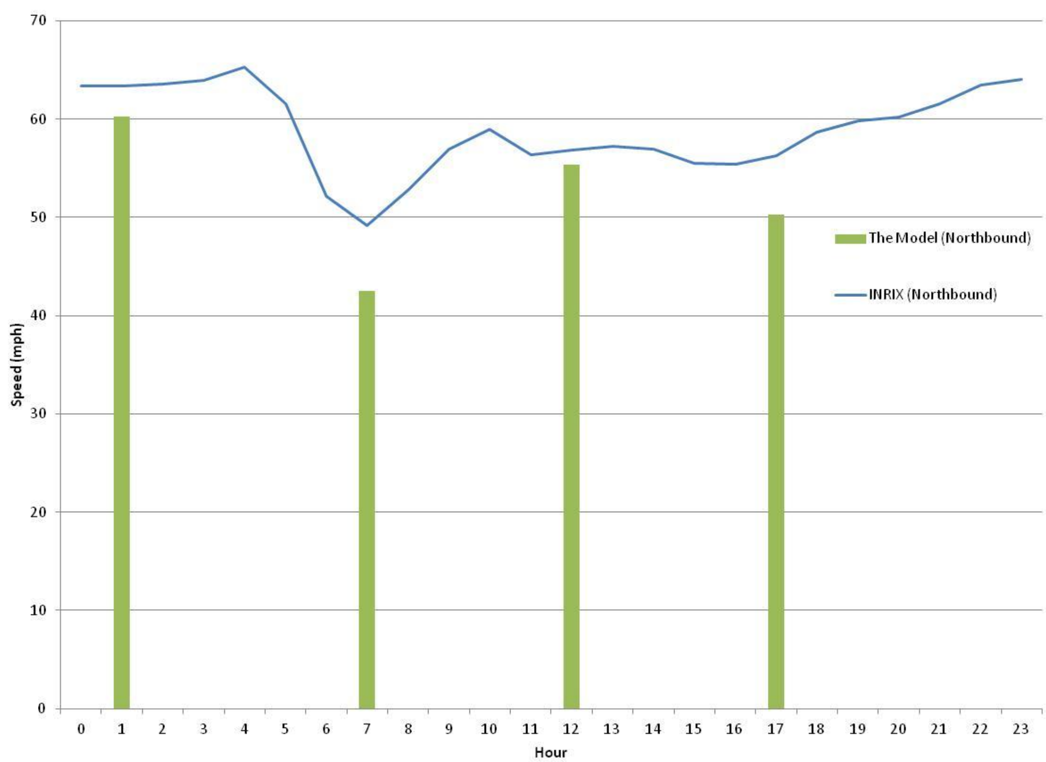


Figure 6-10: Speed Profile of I-95 Southbound in Virginia based on INRIX Data

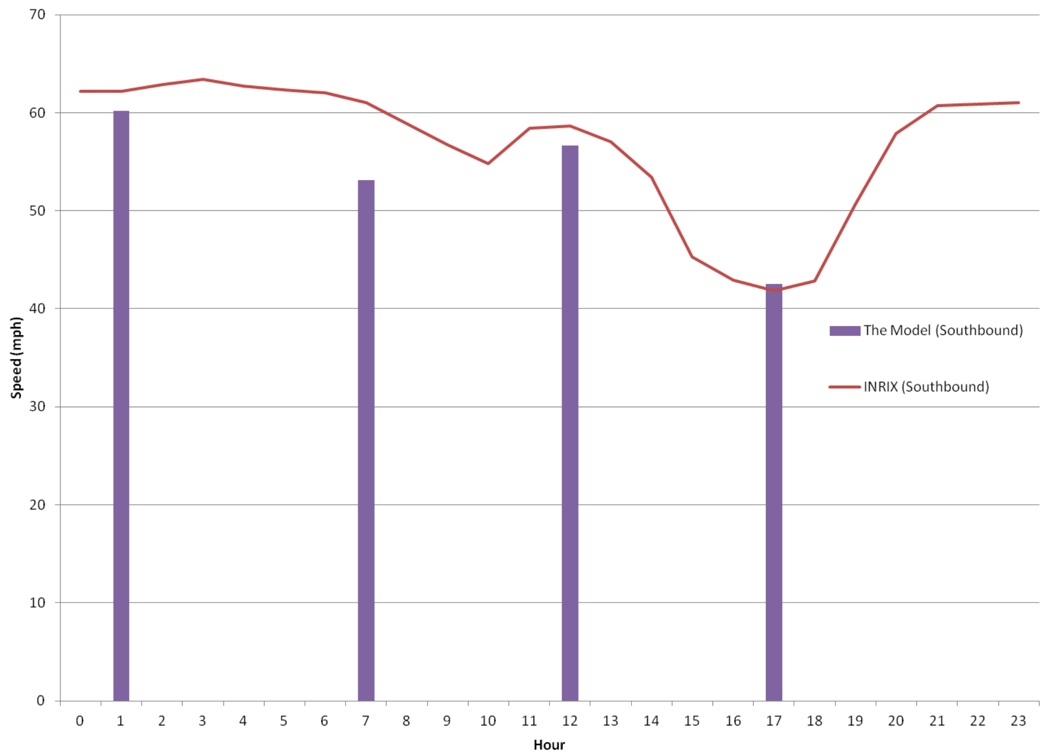


Figure 6-11: Speed Profile of I-270 Northbound based on INRIX Data

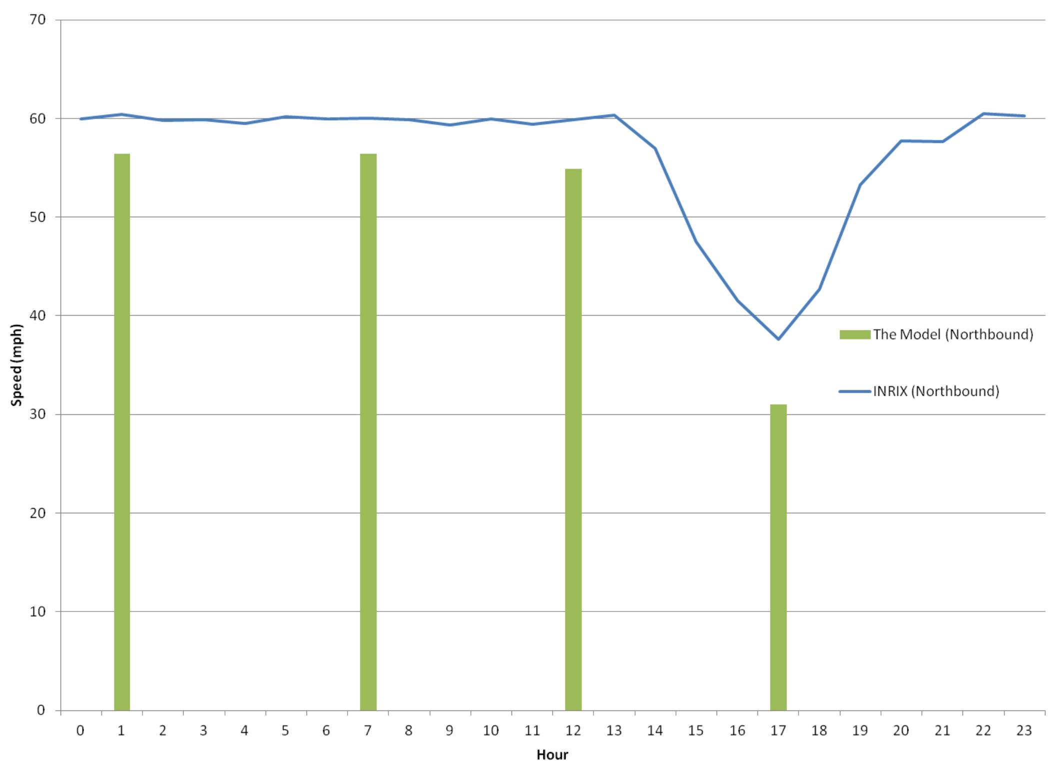


Figure 6-12: Speed Profile of I-270 Southbound based on INRIX Data

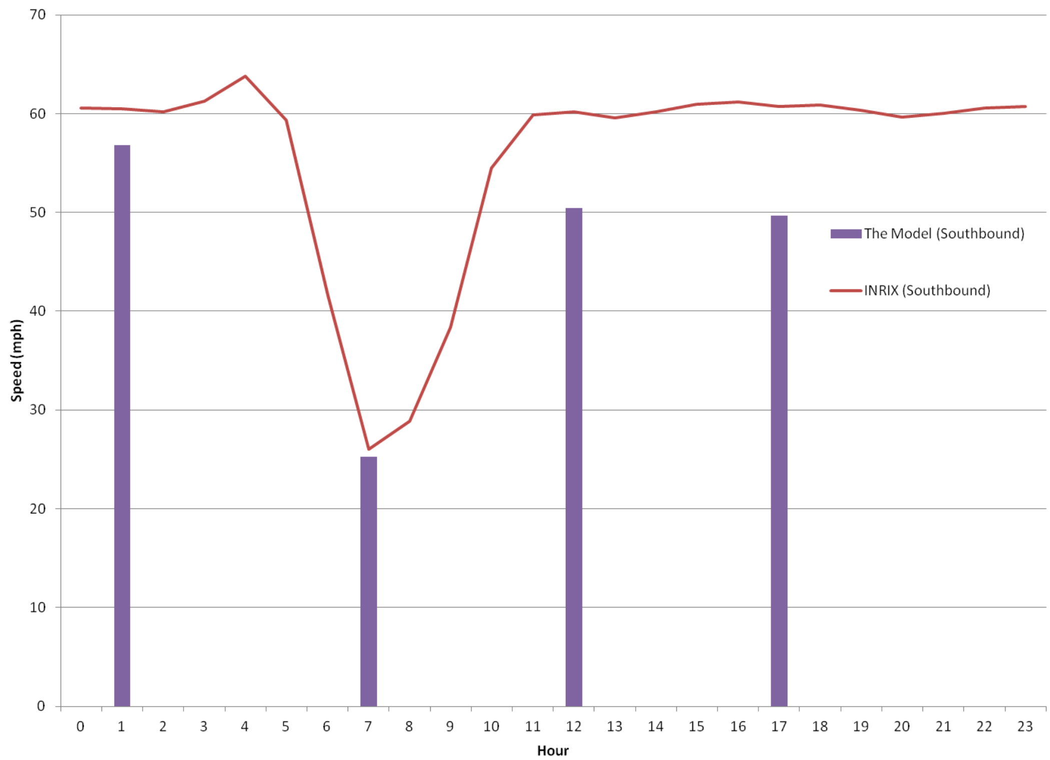


Figure 6-13: Speed Profile of I-395 Northbound in Virginia based on INRIX Data

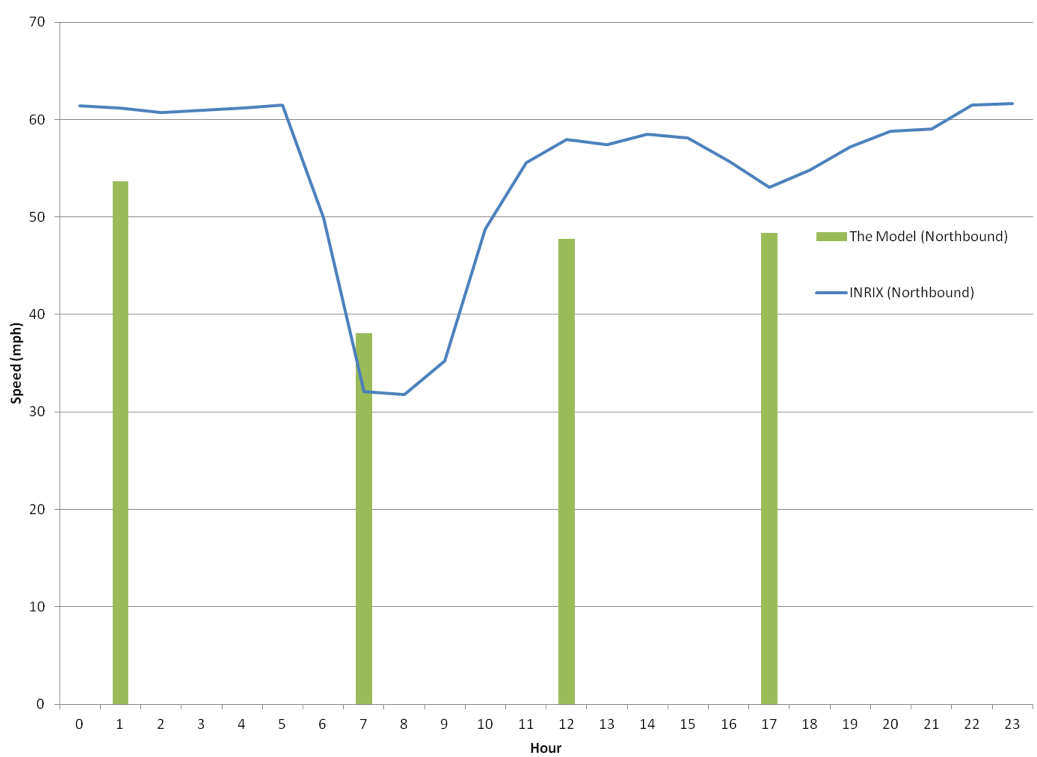


Figure 6-14: Speed Profile of I-395 Southbound in Virginia based on INRIX Data

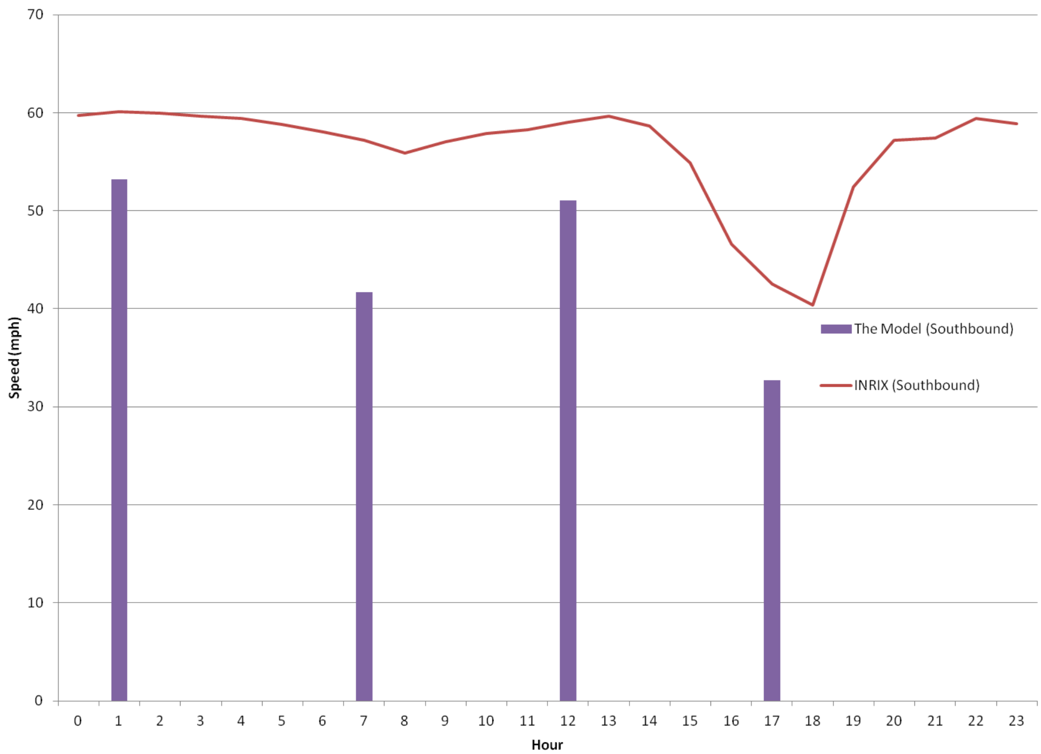


Figure 6-15: Speed Profile of I-495 Inner Loop in Virginia based on INRIX Data

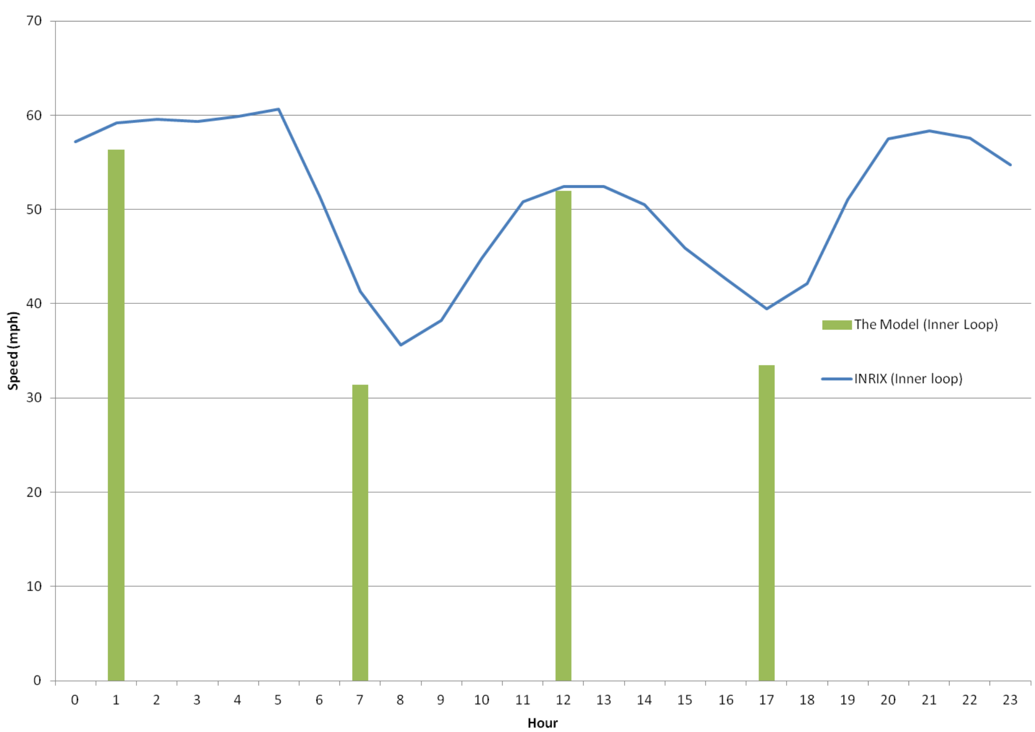
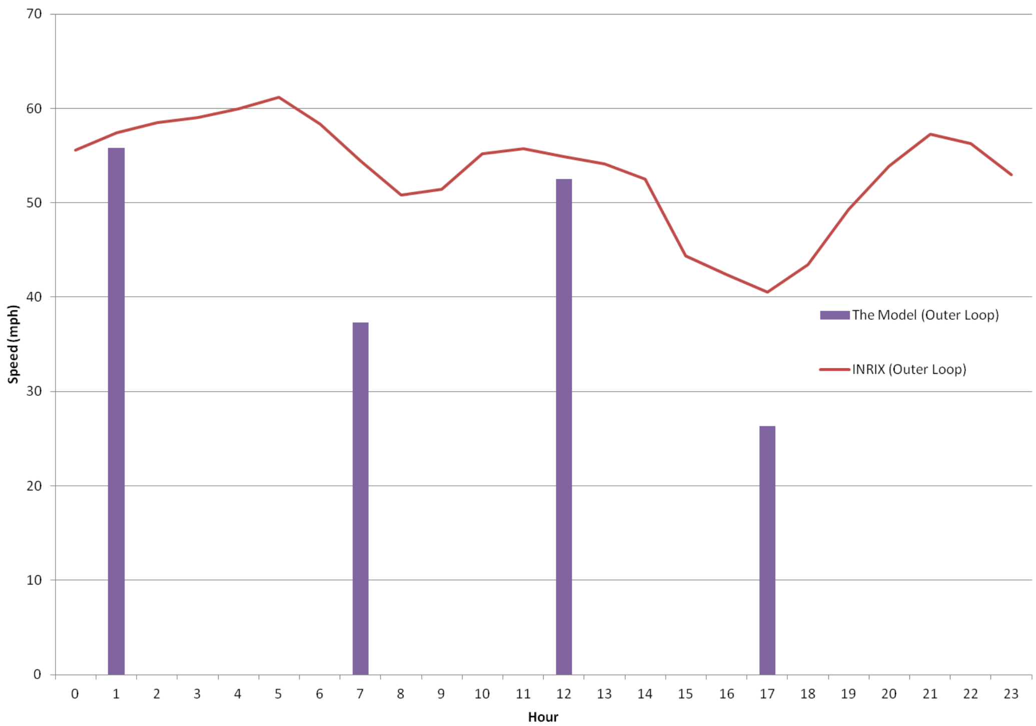


Figure 6-16: Speed Profile of I-495 Outer Loop in Virginia based on INRIX Data



6.5 Conclusion and Summary

In this chapter, INRIX speeds were used to evaluate the MWCOG model results. The INRIX dataset covers around 17,000, which is a subset of the highway links, but includes most of the freeways, expressways, and major arterials. Comparing the MWCOG speeds with INRIX speeds showed that the MWCOG model has lower speeds than INRIX on freeways and expressways. Since many trips in the region will make use of freeways for some portion of their path, the model's apparent underestimation of speeds on freeways will result in an unrealistically high travel time for many trips in the region. Performance of the model during off-peak periods is better than peak periods. Free-flow speed, therefore, looks reasonable; however, the volume-delay functions used in traffic assignment are reducing the speed on congested facilities more than is observed in the field.

Based on the difference between the MWCOG model speeds and INRIX speeds for different facility types, free-flow speeds are somewhat high on arterials and collectors, and low on freeways. This partially explains why arterial speeds are higher in the model than INRIX. On the other hand, the volume-delay functions used in the model reduce speeds in the peak periods more than observed in the INRIX data. The combination of lower free-flow speeds and greater capacity impacts create a more pronounced difference between modeled speeds and INRIX data on freeways than arterials.

Analyzing different volume-delay functions and adjusting function parameters could improve the loaded speeds estimated by the MWCOG model. At this point, these analyses cannot be performed due to lack of volume data. Only 1,400 links have volume data among more than 48,000 links, which makes the results of analyses unreliable. Given the experience and findings referenced in Chapter 5, it is not likely that improved volume-delay functions will generate better static assignments. The model can generate reasonable volumes or reasonable speeds, but is not likely to reproduce both reasonable volumes and reasonable speeds at the same time.

7 Converting TRNBUILD Networks to PT (Task Order 9)

Moving from the Citilabs Cube TRNBUILD transit path builder to the Citilabs Cube Public Transport (PT) transit path builder offers a number of advantages and challenges. It will, however, take time and require a considerable amount of testing and validation work. AECOM proposed a five-phase conversion process to keep the tasks manageable and demonstrate incremental progress. MWCOG is also interested in considering some of the new C++ tools, such as ModeChoice and LineSum, developed by AECOM for WMATA. The overall effort was shared between MWCOG staff and the consultant team to maximize understanding, to proceed at a reasonable pace, and to keep the costs down.

7.1 Network Preparation

This task focused on developing the PT network, routes, and non-transit legs.[41] It established the rules and procedures for integrating transit-only links and nodes into the roadway network and consolidating the transit routes into logical path-building groups that avoid the possibility of a blank line file.[42] The bulk of the work, however, involved generating the non-transit legs. The new process utilizes PT Generate statements to some extent. The PT Generate process was used to generate drive-access non-transit legs for PNR (park-and-ride) and KNR (kiss-and-ride) modes. The overall goal of this phase of the analysis is not to replicate the access links generated by the existing TRNBUILD process, but to generate a reasonable set of access legs that capture the full range of analysis needs with minimal network coding and computer processing requirements. The new process, for example, is designed to summarize all mode-of-access combinations at Metrorail stations.

AECOM prototyped a number of conversion and implementation options, but MWCOG had primary responsibility for converting and enhancing the transit route files and integrating the transit-only links into the highway network. AECOM had primary responsibility for developing the transit access components and path building scripts.

7.2 Transit Access

A concern about the current process of generating walk access links from zones to stations is that the links only connect zones to nodes on the sidewalk network. Figure 7-1 attempts to depict the current and proposed changes to the access link coding. One proposed solution is to expand the zone access logic to permit walk connections from zones to Metrorail and commuter rail stations. Such a connection is represented by the thick dotted green line in the Figure 7-1. This solution can be implemented in the PT procedure using the PT Generate statement. However, the current network setup does not have the Metrorail and Commuter rail station nodes connected to any nodes on the highway network. The station centroids,[43] and not the station nodes, are part of the highway network and hence are

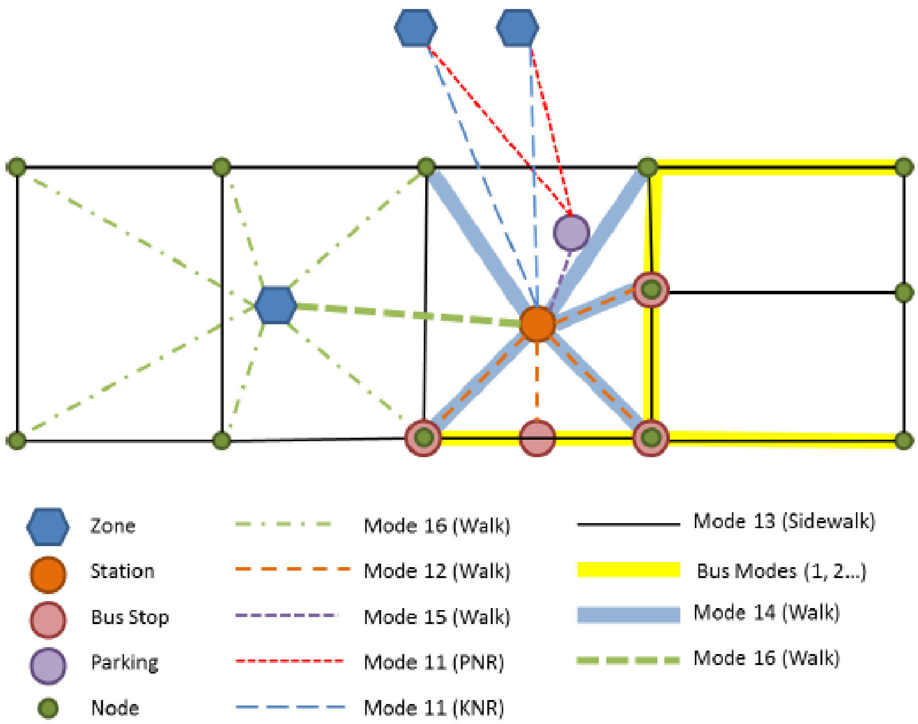
[41] In PT, a trip can be viewed as composed of transit and non-transit segments (called "legs"). A non-transit leg is used to represent transfers between transit lines and also access to and egress from transit lines.

[42] Line files are the text files used by Cube to store information about transit lines/routes.

[43] A station centroid represents the location of the station's PNR lot and is used for building drive-access paths to each station.

connected to the rest of the highway network. This current configuration would prevent the PT Generate procedure from generating direct walk access legs from zones to station nodes.

Figure 7-1: Transit Access Coding Concepts



Four possible options to address this issue were explored:

1. Manually connect stations to the nearest highway node

   This is primarily a one-time task that will need to be updated whenever new stations are added or new highway network links are built near stations. This option would result in the most accurate walk connections from the highway network to the station nodes. However, it would entail more manual work compared to the options below.

2. Connect stations to nodes that are currently connected to the station centroids.

   This is a reasonable approach if the currently coded links between station centroids and highway nodes are acceptable connections from the nearest highway nodes to station nodes.

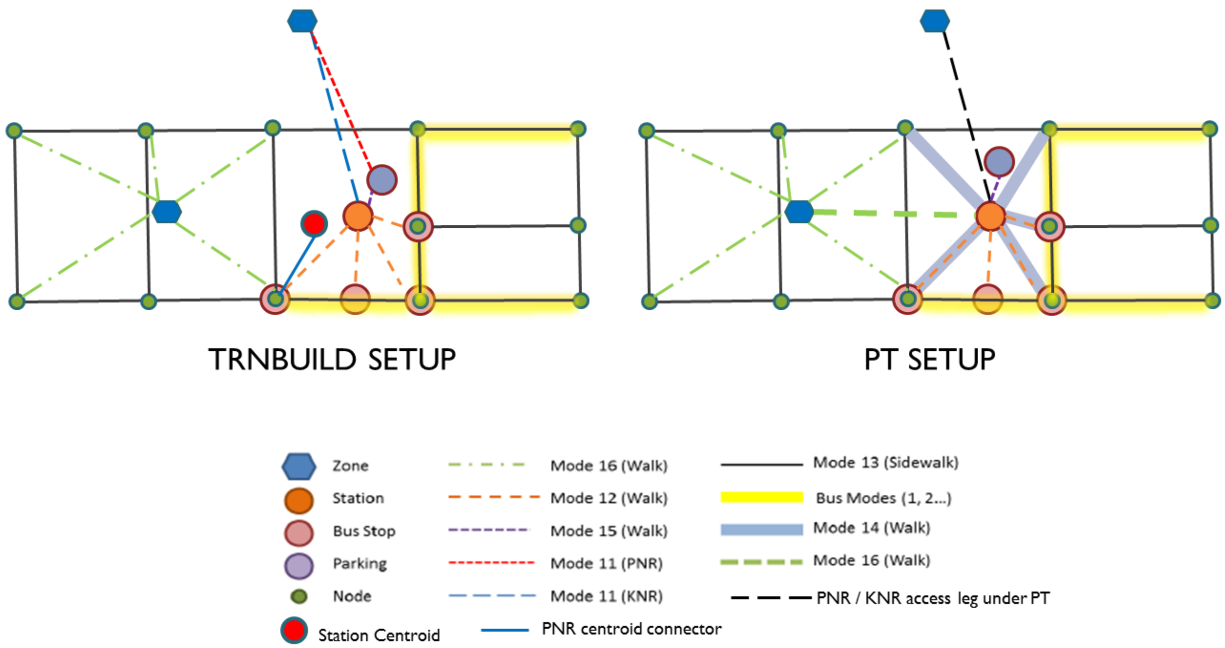
3. Find the nearest “N” nodes to the stations

   A program in Cube could be developed to generate the nearest “N” nodes to the station nodes. This approach disregards any physical constraints between one or more of the nearest highway nodes and station node. However, it could establish possible links between the nearest highway node(s) and station nodes that were not included in approach 2.

4. Develop links between station centroids and station nodes, and have the PT Generate procedure use these links in generating direct walk access links from zones to station nodes.

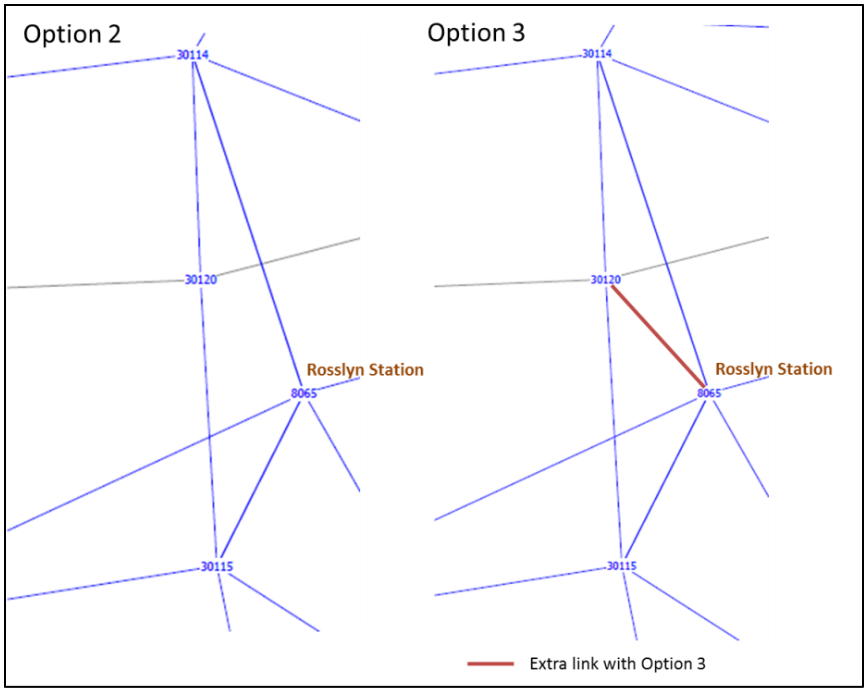
Each of the above approaches has pros and cons associated with it. One consideration should be the eventual conversion from TRNBUILD to PT procedures. Under the PT setup, the non-transit legs would need to connect the TAZs to stations. Hence, the station centroid nodes would no longer be needed. The station nodes need to be part of the overall PT network that has both highway and transit network elements, which are separated under the current setup. Figure 7-2, a revision of Figure 7-1, shows the differences in the configuration of access legs under TRNBUILD and PT systems.

Figure 7-2: Transit Access Coding Concepts under TRNBUILD and PT



Considering the above, options 2 and 3 were tested to generate the links between station nodes and highway nodes. A sample difference between the two options is illustrated in Figure 7-3. Under option 2, the station node is connected to all highway nodes that are currently connected to station centroids. With this option, node 30120 is not connected to the station node. However, with a maximum distance threshold of 0.2 miles, node 30120 is connected to the station node with option 3. In this case, the link seems to be a feasible connection to the Rosslyn station. However, this is not guaranteed with all stations. Hence, if this method were to be employed, a manual check must be conducted to ensure that the connections generated are physically feasible.

Figure 7-3: Sample Links between Station Nodes and Nearby Highway Nodes



With either of the above two options, the script to generate walk access links was tested. The script that was provided to MWCOG for testing was slightly modified. The script to merge highway and transit nodes and links (Mrg_TrnHwyNet.s) was modified to include the above discussed "station node – highway node" connector links. The relevant part of the script is highlighted in Figure 7-4, below, where the file represented by "LINKI[6]" has the station-node links necessary to generate walk and drive access non-transit legs to stations.

Figure 7-4: PT Script to Merge Highway and Transit Networks

```
RUN PGM  = NETWORK
   NETI = "ZONEHWY.NET" ;---- input highway network ----
   NETO = "PT.NET"       ;---- output transit network ----

 ;---- transit only links ----

 LINKI[2]  = "Met_Link.TB",
      VAR = A,12-17,  B,20-25,  MODES,33-34,  DISTANCE,43-45,  SPEED,64-69,  REV=2
 LINKI[3]  = "Com_Link.TB",
      VAR = A,12-17,  B,20-25,  MODES,33-34,  DISTANCE,43-45,  SPEED,64-69,  REV=2
;LINKI[4]  = "Inputs\LRT_Link.TB",
;     VAR = A,12-17,  B,20-26,  MODES,34-35,  DISTANCE,42-47,  SPEED,66-71,  REV=2
; LINKI[5]  = "Inputs\New_Link.TB",
;      VAR = A,12-17,  B,20-26,  MODES,34-35,  DISTANCE,42-47,  SPEED,66-71,  REV=2
;Added by NKJ 02/05/2013 to facilitate generation of busstop-station transfers from the
PT_Walk_Acc_Links script
;option 2 of connecting station nodes to highway nodes
LINKI[6]  = "STN_NODE_LINK.TXT",
     VAR = A,  B,  DISTANCE

  ;---- transit only nodes ----
 NODEI[2]  = "Met_Node.TB",
      VAR = N,9-14,  X,18-27,  Y,31-40
 NODEI[3]  = "Com_Node.TB",
      VAR = N,9-14,  X,18-27,  Y,31-40
```

```
;NODEI[4] = "Inputs\LRT_Node.TB",
;     VAR = N,9-14, X,18-27, Y,31-40
; NODEI[5] = "Inputs\New_Node.TB",
;      VAR = N,9-14, X,18-27, Y,31-40
;added by NKJ 02/05/13 to include station.dbf data
 NODEI[6] = "STATION.TB",
      VAR = N,79-83, X,146-155,Y,158-166,PNR,87-93
```

The resulting walk access links from zones to stations and bus stop nodes are marginally different with different ways of creating station node-highway node links. With either of the methods, the direct walk links between zones and stations (the thick dotted green line in Figure 7-1) are successfully generated.

7.3 Transfers from Bus to Rail

Another proposed short-term measure to improve access link coding and development was a new mode 14 to distinguish between walk access to the station from nearby nodes (MODE 14) and walk access from nearby bus routes (MODE 12). In generating MODE 14 links, option 3 outlined above can be used with one modification. Only the nodes that are on the sidewalk network are used to generate the links between the nearest highway nodes and stations.

The script to generate the transfer links for Metrorail (MODE 3) and express Metrobus (MODE 2) is modified such that modes 2 and 3 are excluded in generating transfer links (MODE 12) between stops and stations. The change is highlighted in the Figure 7-5 code.

Figure 7-5: PT Generate Logic for Transfer Legs

```
;GENERATE 2: Transfer legs for the PT Network - 10 minutes or 1/2 mile
RUN PGM=PUBLIC TRANSPORT

  FILEI LINEI[1] = "Mode2AM.lin"
  FILEI LINEI[2] = "Mode3AM.lin"
  FILEI NETI = "PT.NET"
  FILEI FACTORI[1] = "MR_AM_WK2.FAC"

  FILEO NTLEGO = "M_XFER.LEG", XN=N
  PARAMETERS TRANTIME = GENTRSTIME

  PROCESS PHASE = LINKREAD
    GENTRSTIME = li.DISTANCE*20   ;60/3
  ENDPROCESS

  PROCESS PHASE = DATAPREP
    GENERATE,
      NTLEGMODE=12,
      COST=LI.DISTANCE*20,
      MAXCOST=10*10.0,
      FROMNODE=5001-60000,
      TONODE=5001-60000,
      DIRECTION=3,
      INCLUDELINK=(li.FTYPE <> 1-2),
      EXCLUDELINK=(li.MODES = 2-3),
      MAXNTLEGS=10
  ENDPROCESS
ENDRUN
```

With complete conversion of TRNBUILD to PT, MODE 14 will not be useful in path building since PT does not consider successive non-transit legs to build paths. MODE 14 could to be used to generate different types of access links corresponding to transfers directly from the nearest bus stop.

7.4 Drive Access

Prior to running the script to generate drive access links, the script to generate PT.NET (Mrg_TrnHwyNet.s) was modified so as to include a station node file that lists the PNR node for stations with a park-n-ride lot. The script to generate PNR and KNR drive access links was modified to properly associate the PNR nodes with station nodes. This change is highlighted in the script in Figure 7-4. The script successfully generates the KNR access links. The drive access script outputs a PNR link which generates PNR drive access legs.

The park-n-ride drive access links are connected to the PNR lot, but PNR access links under the PT process need a complete path from the zone to the station. This will require PT.NET to include "PNR lot to station" links (MODE 15). These links need to be included in generating the drive access links. The PNR lot nodes are currently not connected to the highway network. The station centroid zones could be used as PNR lot nodes, where applicable. The parking cost associated with the PNR lot needs to be included in generating the drive PNR access skims.

Based on the initial PT network preparation effort, AECOM recommended using the following coding techniques to improve the access processing performed by the PT Generate statement:

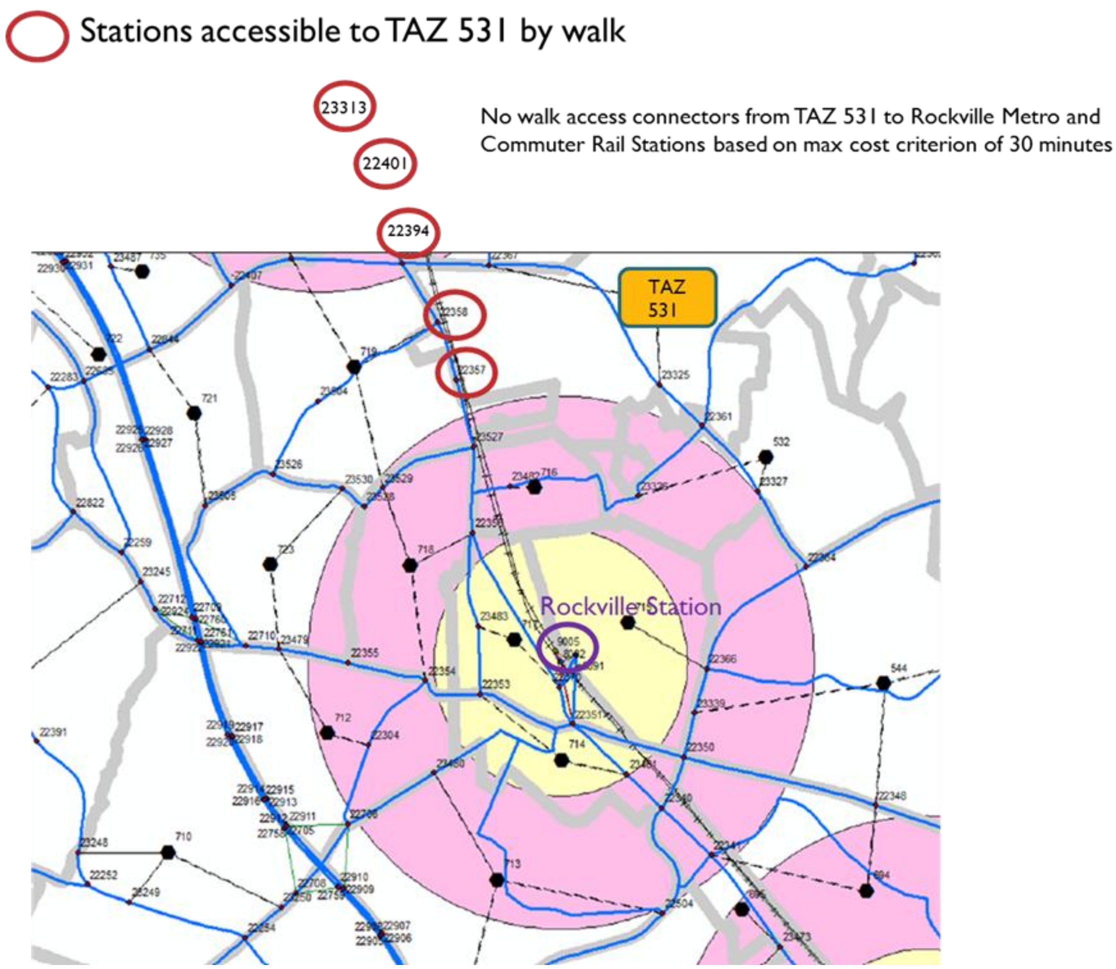
1. The station-to-highway node connections that are needed to generate any kind of access directly to stations (currently only walk access, but eventually drive access to completely convert to PT) should be coded manually. As discussed under the Transit Access section (section 7.2), options 2 and 3 can be used to automate the process of generating these connection links. However, since it is a one-time task, manual coding would provide more accurate connection links. The links generated by options 2 and 3 can be used as reference.
2. Drive access links to stations are needed for complete conversion to PT. This will require connecting the PNR lot to both the station and highway network.

7.5 Recommended Coding Methodology

MWCOG expressed concern about generating walk access links to stations using the PT Generate process due to the sparse street network around most stations in the model[44]. Another aspect of the concern was stations with local development around them. A particular example of the Rockville station was discussed. The memorandum noted that development nearby the station is not limited to the town center, and that it includes dense residential development to the east. The model has relatively large TAZs and sparse highway/sidewalk network around the station. Figure 7-6 shows that the PT Generate process creates walk access legs from zone 531 to a few bus stops along the rail line, but did not generate a walk access leg to the Rockville Metrorail and commuter rail stations (8002 and 9005, respectively). An aerial inspection indicates that zone 531 should have access to the rail stations.

[44] *Notes and Questions Regarding the Public Transport (PT) Conversion,* Memorandum prepared by Ronald Milone, COG/TPB, April 17, 2013

Figure 7-6: Walk Access to Rockville Metrorail and Commuter Rail Stations



Given these limitations of the PT Generate process, a better approach for walk access generation would involve an automated search-radius approach supplemented by a manual review. Since the mode choice model considers the relationship between walk access connections and the long/short walkshed market areas, it was suggested that any TAZ with some walk potential to the station should be connected to the station.

Based on recommendations from the initial network preparation efforts and a subsequent meeting with MWCOG to discuss details and work assignments, MWCOG agreed to update the ESRI geodatabase code used to manage multi-year networks to export highway and transit elements together. The transit links will contain attributes that are common to highway links (for example, distance), but will also contain dissimilar attributes (for example MODE and SPEED). All transit-related links (fixed guideway, transfer links, etc.) will also include AMLIMIT, PMLIMIT and OPLIMIT attributes that will be assigned values of "9" which excludes all such links from a highway assignment.

For drive access paths, MWCOG agreed to add links connecting the highway network to park-n-ride lots. Similarly, links connecting the highway network to rail stations were added to enable kiss-n-ride paths.

For walk access paths, MWCOG agreed to add direct zone-to-rail station links. MWCOG generated these links using an automated process that assesses the distance between stations to TAZs and also considers the zonal walkshed file. The automated link-generation process will be followed by a manual review of the connections to consider possible physical barriers affecting walk access that were not accounted for in the automated process. In addition, the rail station centroids and associated connector links will be removed from the highway network. The elimination of station centroids will obviate the need to: 1) generate zone-to-PNR lot skims (HighwaySkims_mod.s) and 2) create PNR/KNR access links (AutoAcc4.s). Instead, the PT program will be used to generate zone-to-PNR-to-station legs.

The PT scripts were appropriately modified to incorporate the above support files that MWCOG provided[45]. The support files provided by MWCOG (relevant to bus and Metrorail) are:

- Non-Transit Legs for Walk access: from TAZs to bus and Metrorail
    - Walk access to Bus: Walkacc.nt
    - Transfer from Bus to Metrorail stations: Met_Bus.nt
    - Walk access to Metrorail: Met_Wlk.nt
- Non-Transit Legs for Drive Access: from highway nodes to bus and metro
    - Met_PNR_Dr.nt: Highway node to PNR node
    - Met_KNR_Dr.nt: Highway node to station

The revised PT scripts are detailed in Chapter 8 in the discussion of the PT path building process.

[45] *Transmittal of 2010 Highway and Transit Networks in Public Transport (PT) Format,* Memorandum prepared by Ron Milone and Meseret Seifu (COG/TPB), June 14, 2013.

8 PT Path Building and Assignments (Task Order 9)

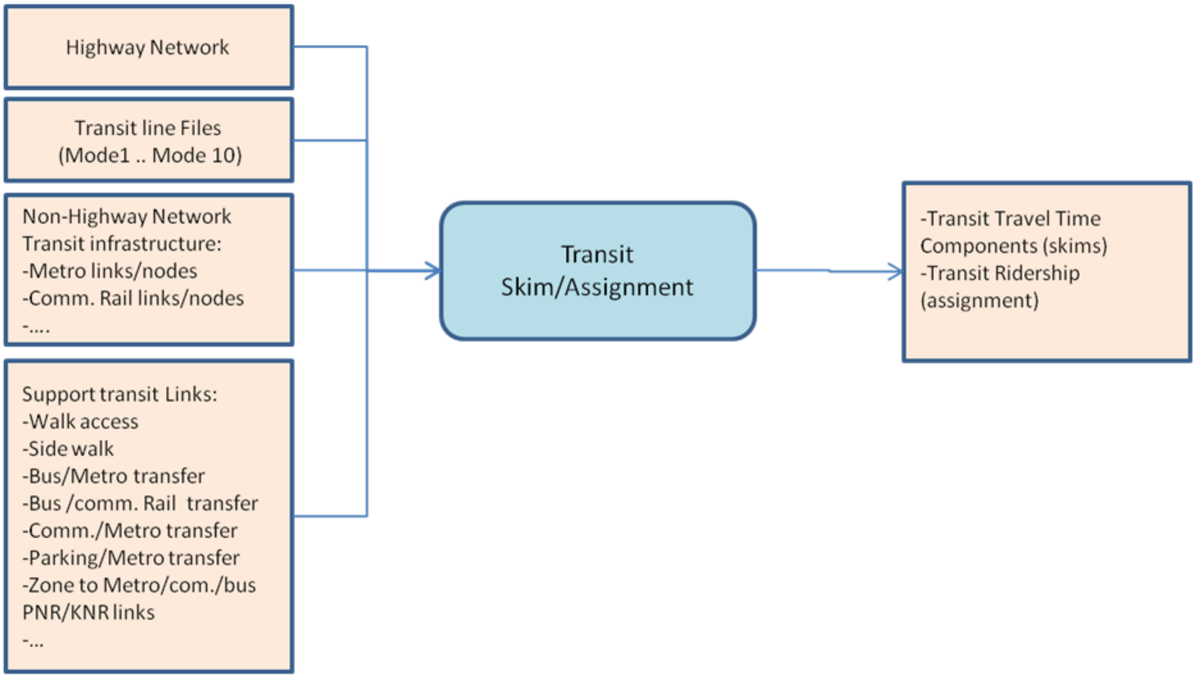
The path building options included within PT are sufficiently different from the methods and controls provided by TRNBUILD to warrant a rigorous assessment of the strengths and weaknesses of various approaches in reproducing observed behavior. Once again, the goal of this task is not so much to reproduce the paths generated by TRNBUILD, but to generate a set of reasonable paths using the PT path building process. This effort assessed the processing methods used to generate the path combinations included in the current MWCOG process. AECOM had primary responsibility for developing the path building procedures and evaluating the strengths and weaknesses of the implementation options. MWCOG will have primary responsibility for validating the transit paths against observed data and testing and evaluating the modeling scripts in a variety of configurations.

8.1 Transit Path Building in the MWCOG Travel Model

The MWCOG Version 2.3 travel model utilizes Cube's TRNBUILD program to generate transit skims and perform transit assignment. The TRNBUILD transit skim and assignment scripts are fairly similar, but the transit assignment script includes a transit demand matrix and saves the transit path leg information. Transit skims and assignment are executed for 22 set of transit paths. There are two time periods (peak and off-peak), three access modes (walk, kiss-n-ride and park-n-ride), and four line-haul modes (bus-only, Metrorail-only, bus and Metrorail, and commuter rail). It should be noted that park-n-ride and kiss-n-ride access modes are combined to a single drive access mode for commuter rail. Bus-only transit paths do not include any Metrorail legs. Similarly, Metrorail-only transit paths do not include any bus legs. However, bus and Metrorail transit paths must include both bus and Metrorail legs. Commuter rail transit paths may include bus or Metrorail legs as part of the paths.

The input files to transit skim/assignment scripts are defined based on the time period, access modes and line-haul modes. Figure 8-1 shows the input and output files of a typical transit skim/assignment process in the MWCOG Version 2.3 model. The script itself includes multiple lines defining characteristics of each transit mode such as weighting factors that convert actual travel time into perceived travel time, boarding and transfer penalties, initial and transfer wait penalties, transfer prohibitions, and maximum transit path costs. TRNBUILD does not consider transit fares while evaluating transit paths. However, it can report the first boarding and last alighting stations for each transit mode using the *NODEO[mode]* command. This information is used to calculate the transit fare for the best transit path found for each origin-destination used in the mode choice model.

Figure 8-1: Input and Output Components of the MWCOG Transit Skim/Assignment Process



8.2 PT Transit Path Building

Public Transport (PT) is the Cube Voyager program that prepares transit access legs and builds transit paths. This section provides an example of how PT can be configured to create transit skims and perform transit assignment for peak period bus-Metrorail paths using three access modes (walk, kiss-n-ride, and park-n-ride). This example can be easily expanded to other line-haul modes.

Network preparation is the first step in building transit paths using PT. One of the necessary inputs to the PT program is a transportation network that contains characteristics of zones, nodes and links (that is, node coordinates, walk and transit link times, distances, and so forth), over which the public transport system operates. This network usually is prepared by the Network program. Figure 8-2 shows a script that can be used to create a network for the PT program using MWCOG Version 2.3 data structures. The script combines the MWCOG highway network with the transit links required to support transit routes with exclusive guideways or special access requirements. The MWCOG model defines such links for Metrorail, LRT, and commuter rail routes.

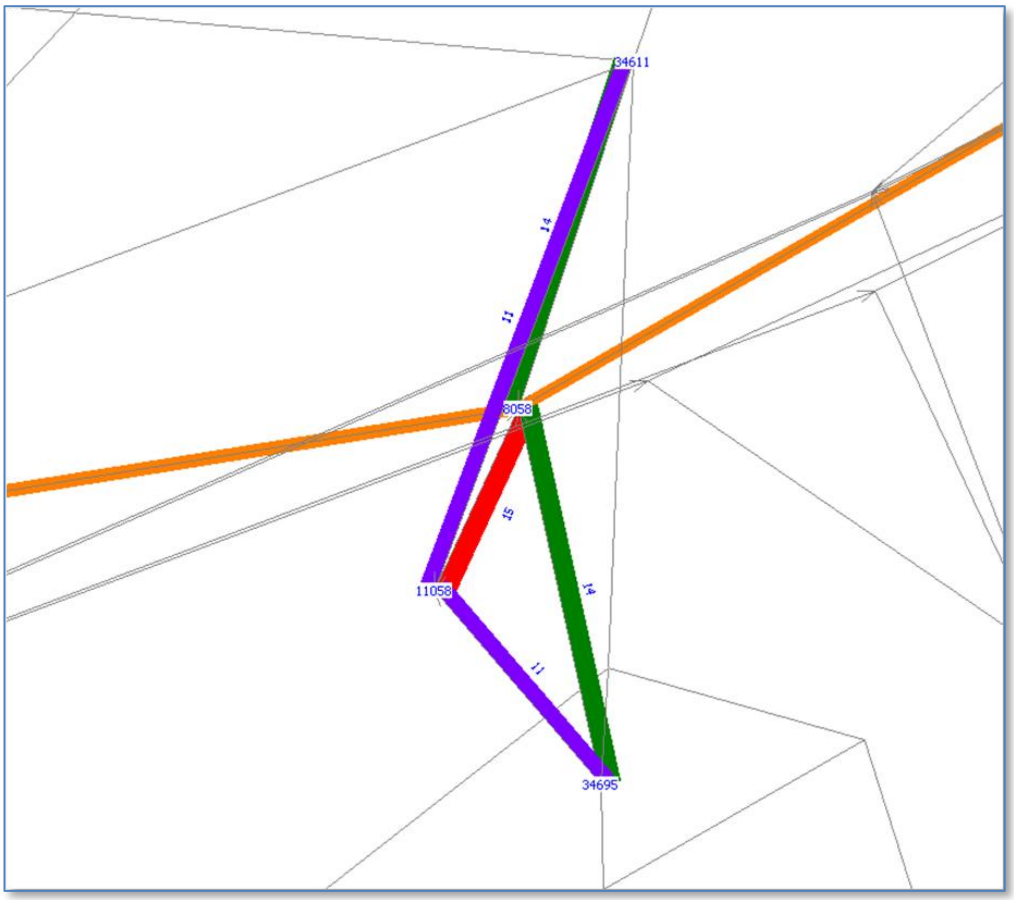
The script also reads the connection links between these networks (Met_KNR_DR.TB, etc.) and adds them to the output (PT.NET) network. It should be noted that in the MWCOG model, parking lots are introduced to the model using nodes that are different than bus stops and Metrorail stations. Therefore, it is necessary to add those nodes (Met_PNRN.TB, etc.) and their connecting links to the transit network (metampnr.tb, etc.) and the PT network. *NETI* reads the highway network. *LINKI* and *NODEI* commands read transit links and nodes. It should also be noted that *LINKI*[ ] and *NODEI*[ ] should start from [2], since *LINKI*[1] and *NODEI*[1] have been automatically assigned to the network file introduced by *NETI*.

Figure 8-2: Sample Data Preparation Script – Combine Highway and Transit Networks

```
;Prepare PT Network using MWCOG Version 2.3 data
RUN PGM  = NETWORK
   NETI = "Zonehwy_rev.NET"        ;---- input highway network ----
   NETO = "PT.NET"                 ;---- output transit network ----

;---- transit only links ----
LINKI[2] = "Inputs\Met_Link.TB",
     VAR = A,12-17, B,20-24, MODES,33-34, DISTANCE,42-45, SPEED,64-69, REV=2 ; Metrorail  links
LINKI[3] = "Inputs\Com_Link.TB",
     VAR = A,12-17, B,20-24, MODES,33-34, DISTANCE,42-45, SPEED,64-69, REV=2 ; Com. rail links
LINKI[4] = "Inputs\metampnr.tb",
     VAR = A,12-16, B,18-22, MODES,38-39, DISTANCE,46-51   ;Transfer Metro st and parking lots
LINKI[5] = "Inputs\comampnr.tb",
     VAR = A,12-16, B,18-22, MODES,38-39, DISTANCE,46-51  ;Transfer Com. rail st and parking lots
LINKI[6] = "Inputs\busampnr.tb",
     VAR = A,12-16, B,18-22, MODES,38-39, DISTANCE,46-51   ;Transfer Bus st and parking lots
LINKI[7] = "Inputs\Met_KNR_DR.TB",
     VAR = A,12-17, B,19-24, MODES,31-33, DISTANCE,65-69, SPEED,42-46  ;KNR highway/Met. St
LINKI[8] = "Inputs\Met_PNR_DR.TB",
     VAR = A,12-17, B,19-24, MODES,31-33, DISTANCE,65-69, SPEED,42-46  ;PNR highway/Met. St

 ;---- transit only nodes ----
 NODEI[2] = "Inputs\Met_Node.TB", VAR = N,9-14, X,18-27, Y,31-40    ; Metro station nodes
 NODEI[3] = "Inputs\Com_Node.TB", VAR = N,9-14, X,18-27, Y,31-40    ; Commuter rail station nodes

;---- park-n-ride nodes ----
  NODEI[4] = "Inputs\Met_PNRN.TB", VAR = N,9-14, X,18-27, Y,31-40  ; Metrorail parking lot nodes
  NODEI[5] = "Inputs\Com_PNRN.TB", VAR = N,9-14, X,18-27, Y,31-40  ; Comm. rail parking lot nodes
  NODEI[6] = "Inputs\Bus_PNRN.TB",  VAR = N,9-14, X,18-27, Y,31-40 ; Bus parking lot nodes

;---- link processing ----
MERGE RECORD=TRUE
PHASE = LINKMERGE
    IF (SPEED >0)
       DISTANCE = DISTANCE / 100.0
       TRANTIME = DISTANCE * 60.0 / SPEED
       AMHTIME = TRANTIME
    ELSEIF (MODES=15)                   ;Metrorail station-PNR connector links
       DISTANCE = DISTANCE / 5280
       SPEED = 3
       TRANTIME = DISTANCE * 60.0 / SPEED
       AMHTIME = TRANTIME
    ELSE                                ;highway links and station-highway connector links
       DISTANCE = DISTANCE
       SPEED = 3                        ;Walk link speeds
       TRANTIME = DISTANCE * 60.0 / SPEED
       AMHTIME = TRANTIME
    ENDIF
ENDPHASE
ENDRUN
```

Figure 8-3 shows the resulting PT network structure at the Dunn Loring-Merrifield Metrorail station. The numbers posted on the links are mode numbers. Mode 11 (purple lines) shows the station parking lot connections to the highway network. Mode 14 (green lines) shows the station kiss-n-ride connection to the highway network. Mode 15 (red line) shows the transfer link between the station parking lot and station. These links will later be used by the PT program to estimate travel time from different zones to Metrorail stations for park-n-ride and kiss-n-ride access modes. It is important to know that PT does not use these links during transit skimming and assignment. The PT program generates origin to park-n-ride station (mode 11) and origin to kiss-n-ride station (mode 14) access legs for use in path building. The user has the option of defining a maximum cost to limit the number of drive (PNR/KNR) access legs. The user will also have the option to treat the stations differently, for example, end of the line station versus intermediate stations with parking lot locations. The current MWCOG's TRNBUILD program creates a direct connection between zone centroids to parking lots for park-n-ride links (mode 11) and then uses mode 15 to connect the parking lots to stations. Kiss-n-ride access links (mode 14), on the other hand, directly connect zone centroids to Metrorail stations. It should be noted that the way PT treats walk access is slightly different from TRNBUILD and those links do not need to be added to the highway network during this step.

Figure 8-3: Highway and Transit Network for the Dunn Loring-Merrifield Metrorail Station



The second step of the PT data preparation is to generate non-transit legs. These links present opportunities to access the public transport system, egress from it and transfer between services during the course of a trip. Non-transit legs may be determined externally and/or generated by PT under user control. Non-transit legs for kiss-n-ride and park-n-ride access modes are generated in a separate step, assigning them modes 14 and 11, respectively. (By contrast, the current MWCOG TRNBUILD process uses mode 11 for both drive/PNR-access links and drive/KNR-access links.) Since MWCOG uses custom GIS database tools to generate walk access/egress links, those links will be read directly during the transit skim/assignment step. Figure 8-4 shows a sample script that uses PT's *GENERATE* statement to create drive access links. The sample script treats all the Metro stations equally while creating drive access links. However, PT is capable of creating different set of drive access links for Metro station depending on the station characteristics such as end of the line stations or special type of stations, for example Pentagon station.

Figure 8-4: Sample Script to Create Drive Access Links

```
;Prepare PT Network using MWCOG Version 2.3 data
;; *********************************
;; *****1- Create kiss-n-Ride links
;; *****2- Create Park-n-Ride links
;; *********************************

;; *************PNR Links***************
RUN PGM=PUBLIC TRANSPORT

  FILEI NETI = "PT.NET"
  FILEI LINEI[1] = "Inputs\Mode1AM.lin"
  FILEI LINEI[2] = "Inputs\Mode2AM.lin"
  FILEI LINEI[3] = "Inputs\Mode9AM.lin"
  FILEI LINEI[4] = "Inputs\Mode3AM.lin"
  FILEI FACTORI[1] = "Inputs\AM_TRN.FAC"
  FILEI SYSTEMI = "Inputs\TSYSD.PTS"

  FILEO NTLEGO = "BM_PNR.LEG", XN=N

  PARAMETERS TRANTIME = GENTRSTIME

  PROCESS PHASE = LINKREAD
       GENTRSTIME = li.AMHTIME
  ENDPROCESS

  PROCESS PHASE = DATAPREP
    GENERATE,
      NTLEGMODE = 11,
      COST=li.DISTANCE,
      MAXCOST=10*5.0,
      FROMNODE=1-3675,
      TONODE=8001-8098,13000-13140,
      DIRECTION = 1,
      EXCLUDELINK = (li.MODES = 1-10, 12, 14, 16),
      MAXNTLEGS = 10,
      LIST = Y
  ENDPROCESS

ENDRUN

;; *************KNR Links***************
RUN PGM=PUBLIC TRANSPORT

  FILEI NETI = "PT.NET"
  FILEI LINEI[1] = "Inputs\Mode1AM.lin"
  FILEI LINEI[2] = "Inputs\Mode2AM.lin"
  FILEI LINEI[3] = "Inputs\Mode9AM.lin"
  FILEI LINEI[4] = "Inputs\Mode3AM.lin"
  FILEI FACTORI[1] = "Inputs\AM_TRN.FAC"
  FILEI SYSTEMI = "Inputs\TSYSD.PTS"

  FILEO NTLEGO = "BM_KNR.LEG", XN=N

  PARAMETERS TRANTIME = GENTRSTIME

  PROCESS PHASE = LINKREAD
       GENTRSTIME = li.AMHTIME
  ENDPROCESS

  PROCESS PHASE = DATAPREP
    GENERATE,
      NTLEGMODE = 14,
      COST=li.DISTANCE,
      MAXCOST=10*3.0,
```

```
      FROMNODE=1-3675,
      ;TONODE=8001-8098,
      DIRECTION = 1,
      EXCLUDELINK = (li.MODES = 1-10, 11, 12, 16),
      MAXNTLEGS = 10,
      LIST = Y
  ENDPROCESS

ENDRUN
```

Basically, PT builds a minimum impedance path between zones and Metrorail stations through parking lots. If the estimated travel time for a zone is less than the maximum travel time defined in the script, PT create a non-transit leg from the zone to the station and uses that leg to create transit skims or perform a transit assignment. This generates the home-to-destination paths needed for modeling home-based travel demand. Although PT is capable of generating auto egress legs, AECOM assumes MWCOG will not need to build destination-to-home transit paths in the near future, since the MWCOG transit path building and transit assignment are conducted in production/attraction format. This capability is primarily needed to support activity-based models or origin-destination transit assignments for the PM peak period.

Figure 8-5 shows a sample PT transit skim script for peak period, walk access paths to bus-Metrorail line-haul modes. As stated, MWCOG creates walk access and egress links using a GIS process. Since the files are prepared in PT format, the scripts reads those files. The *GENERATE* statement is used to read these files and add them to the legs database needed for path building.

Figure 8-5: Sample PT Transit Skim Script for Peak Period, Walk Access, Bus-Metrorail Paths

```
RUN PGM = PUBLIC TRANSPORT; MSG = 'Bus_Metro AM WK transit paths and skims'
;----Input PT network ----
FILEI NETI = "PT.NET"

;----Output files----
FILEO MATO[1] = "AM_BM_WK.SKM", MO=1-20,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, INWT, XFERWT, WACCT, OWLKT, XFERP, BRDS,
  DACCT, DACCD, PRKT, PRKC, MRD, LRTD
FILEO ROUTEO[1] = "AM_BM_WK.RTE"
FILEO REPORTO = "PT_Skims_AM_BM_WK.PRN"
FILEO NETO = "AM_BM_WK.NET"
FILEO NTLEGO ="NTLEG_AM_BM_WK.NT"

;----Input Mode specific factors----
FILEI FACTORI[1] = "Inputs\AM_WK_TRN.FAC"
FILEI SYSTEMI = "Inputs\TSYSD.PTS"

;---- Transit Line Cards (modes 1-10) ----
FILEI LINEI[1] = "Inputs\Mode1AM.lin"
FILEI LINEI[2] = "Inputs\Mode2AM.lin"
FILEI LINEI[3] = "Inputs\Mode9AM.lin"
FILEI LINEI[4] = "Inputs\Mode3AM.lin"

;---- Access Links (modes 11, 12 and 16) ----
FILEI NTLEGI[1] = "Inputs\walkacc.nt"    ; Walk Access/Egress links to bus stops
FILEI NTLEGI[2] = "Inputs\Met_Wlk.nt"    ; Walk Access/Egress links to Metro stops
FILEI NTLEGI[3] = "Inputs\Met_Bus.nt"    ; Walk transfer links between bus and Metro stops

; ----- OVERALL PARAMETERS OF RUN ------
PARAMETERS FARE=N, HDWAYPERIOD=1,NOROUTEERRS=999999,
           NOROUTEMSGS=999999,SKIPBADLINES=Y,
          TRANTIME=LI.TRANTIME

REPORT LINES=T
```

```
PROCESS PHASE=LINKREAD
 LW.TRANTIME=LI.TRANTIME
 LW.WALKTIME=(LI.DISTANCE/3*60)
 LW.WALKDISTANCE=LI.DISTANCE
 LW.DISTANCE=LI.DISTANCE
ENDPROCESS

PROCESS PHASE=DATAPREP
       GENERATE READNTLEGI=1
       GENERATE READNTLEGI=2
       GENERATE READNTLEGI=3
ENDPROCESS

PROCESS PHASE=SKIMIJ
;---- specify output skims ----

 MW[1] = TIMEA(0,1,6,8),                         ;---- ivt-local bus    (0.01 min)
 MW[2] = TIMEA(0,2,7,9),                         ;---- ivt-exp bus      (0.01 min)
 MW[3] = TIMEA(0,3),                             ;---- ivt-metrorail    (0.01 min)
 MW[4] = TIMEA(0,4),                             ;---- ivt-commuter rail(0.01 min)
 MW[5] = TIMEA(0,5),                             ;---- ivt-new rail mode(0.01 min)
 MW[6] = TIMEA(0,10),                            ;---- ivt-new bus mode (0.01 min)
 MW[7] = IWAITA(0),                              ;---- ini.wait time    (0.01 min)
 MW[8] = XWAITA(0),                              ;---- xfr wait time    (0.01 min)
 MW[9] = TIMEA(0,14) + TIMEA(0,16)/2.0,          ;---- walk acc time    (0.01 min)
 MW[10] = TIMEA(0,12) + TIMEA(0,13)/2.0,         ;---- other walk time  (0.01 min)
 MW[11] = XFERPENA(0),                           ;---- added xfer time  (0.01 min)
 MW[12] = BRDINGS(0),                            ;---- boardings        (1+)
 MW[13] = TIMEA(0,11),                           ;---- drv acc time     (0.01 min)
 MW[14] = DIST(0,11),                            ;---- drv acc distance (0.01 mile)
 MW[15] = TIMEA(0,15),                           ;---- pnr impedance    (0.01 min)
 MW[16] = DIST(0,15),                            ;---- pnr cost         (cents)
 MW[19] = DIST(0,3),                             ;---- metrorail distance
 MW[20] = DIST(0,5)                              ;---- light rail distance

ENDPROCESS

ENDRUN
```

Figure 8-6 and Figure 8-7 show sample PT transit skim scripts for park-n-ride and kiss-n-ride access modes respectively. Since the process of saving output data is the same for all the scripts regardless of access mode, that part has not been shown. These scripts read the drive access non-transit legs (BM_PNR.LEG and BM_KNR.LEG) generated in the previous step. It should be noted that although walk access/egress non-transit walk legs to bus and Metrorail stops are read in these scripts, PT's Factor File excludes walk access links and only uses walk links as egress options for drive access trips. For example, a park-n-ride access Metrorail trip can involve driving from the origin TAZ to the park-n-ride lot (mode 11), walk from park-n-ride lot to the station (mode 15) take Metrorail (mode 3) and walk from the station to the destination TAZ (mode 16). The access leg of the trip cannot use walk links (mode 16) because the condition in the factor file prevents mode 16 from being used for access.

Figure 8-6: Sample PT Transit Skim Script for Peak Period, PNR Access, Bus-Metrorail Paths

```
RUN PGM = PUBLIC TRANSPORT; MSG = 'Bus_Metro AM PNR transit paths and skims'
;----Input PT network ----
FILEI NETI = "PT.NET"

;----Output files----
FILEO MATO[1] = "AM_BM_PNR.SKM", MO=1-20,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, INWT, XFERWT, WACCT, OWLKT, XFERP, BRDS,
  DACCT, DACCD, PRKT, PRKC, MRD, LRTD
FILEO ROUTEO[1] = "AM_BM_PNR.RTE"
FILEO REPORTO = "PT_Skims_AM_BM_PNR.PRN"
FILEO NETO = "AM_BM_PNR.NET"
```

```
FILEO NTLEGO ="NTLEG_AM_BM_PNR.NT"

;----Input Mode specific factors----
FILEI FACTORI[1] = "Inputs\AM_PNR_TRN.FAC"
FILEI SYSTEMI = "Inputs\TSYSD.PTS"

;---- Transit Line Cards (modes 1-10) ----
FILEI LINEI[1] = "Inputs\Mode1AM.lin"
FILEI LINEI[2] = "Inputs\Mode2AM.lin"
FILEI LINEI[3] = "Inputs\Mode9AM.lin"
FILEI LINEI[4] = "Inputs\Mode3AM.lin"

;---- Access Links (modes 11, 12 and 16) ----
FILEI NTLEGI[1] = "Inputs\walkacc.nt"   ; Walk Access/Egress links to bus stops
FILEI NTLEGI[2] = "Inputs\Met_Wlk.nt"   ; Walk Access/Egress links to Metro stops
FILEI NTLEGI[3] = "Inputs\Met_Bus.nt"   ; Walk transfer links between bus and Metro stops
FILEI NTLEGI[4] = "BM_PNR.LEG"          ; Drive access links to Metro stops

zonemsg=50
; OVERALL PARAMETERS OF RUN
PARAMETERS FARE=N, HDWAYPERIOD=1,NOROUTEERRS=999999,
           NOROUTEMSGS=999999,SKIPBADLINES=Y,
          TRANTIME=LI.TRANTIME
REPORT LINES=T
PROCESS PHASE=LINKREAD
 LW.TRANTIME=LI.TRANTIME
 LW.WALKTIME=(LI.DISTANCE/3*60)
 LW.WALKDISTANCE=LI.DISTANCE
 LW.DISTANCE=LI.DISTANCE
ENDPROCESS

PROCESS PHASE=DATAPREP

        GENERATE READNTLEGI=1
        GENERATE READNTLEGI=2
        GENERATE READNTLEGI=3
        GENERATE READNTLEGI=4

ENDPROCESS
```

Figure 8-7: Sample PT Transit Skim Script for Peak Period, KNR Access, Bus-Metrorail Paths

```
RUN PGM = PUBLIC TRANSPORT; MSG = 'Bus_Metro AM KNR transit paths and skims'
;----Input PT network ----
FILEI NETI = "PT.NET"

;----Output files----
FILEO MATO[1] = "AM_BM_KNR.SKM", MO=1-20,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNRM, IVNBM, INWT, XFERWT, WACCT, OWLKT, XFERP, BRDS,
  DACCT, DACCD, PRKT, PRKC, MRD, LRTD
FILEO ROUTEO[1] = "AM_BM_KNR.RTE"
FILEO REPORTO = "PT_Skims_AM_BM_KNR.PRN"
FILEO NETO = "AM_BM_PNR.NET"
FILEO NTLEGO ="NTLEG_AM_BM_KNR.NT"

;----Input Mode specific factors----
FILEI FACTORI[1] = "Inputs\AM_KNR_TRN.FAC"
FILEI SYSTEMI = "Inputs\TSYSD.PTS"

;---- Transit Line Cards (modes 1-10) ----
FILEI LINEI[1] = "Inputs\Mode1AM.lin"
FILEI LINEI[2] = "Inputs\Mode2AM.lin"
FILEI LINEI[3] = "Inputs\Mode9AM.lin"
FILEI LINEI[4] = "Inputs\Mode3AM.lin"

;---- Access Links (modes 11, 12 and 16) ----
FILEI NTLEGI[1] = "Inputs\walkacc.nt"   ; Walk Access/Egress links to bus stops
FILEI NTLEGI[2] = "Inputs\Met_Wlk.nt"   ; Walk Access/Egress links to Metro stops
FILEI NTLEGI[3] = "Inputs\Met_Bus.nt"   ; Walk transfer links between bus and Metro stops
FILEI NTLEGI[4] = "BM_KNR.LEG"          ; Drive access links to Metro stops
```

```
; OVERALL PARAMETERS OF RUN
PARAMETERS FARE=N, HDWAYPERIOD=1,NOROUTEERRS=999999,
           NOROUTEMSGS=999999,SKIPBADLINES=Y,
          TRANTIME=LI.TRANTIME
REPORT LINES=T
PROCESS PHASE=LINKREAD
 LW.TRANTIME=LI.TRANTIME
 LW.WALKTIME=(LI.DISTANCE/3*60)
 LW.WALKDISTANCE=LI.DISTANCE
 LW.DISTANCE=LI.DISTANCE
ENDPROCESS

PROCESS PHASE=DATAPREP
        GENERATE READNTLEGI=1
        GENERATE READNTLEGI=2
        GENERATE READNTLEGI=3
        GENERATE READNTLEGI=4
ENDPROCESS
```

Figure 8-8 shows a sample PT factor file that excludes the walk access mode for drive access trips.

Figure 8-8: Sample Factor File for Peak Period, Drive (PNR_KNR) Access Mode

```
;PT Factors File for Drive access transit trips

;Global Settings
 BESTPATHONLY=T
 MAXFERS=3
 SERVICEMODEL=FREQUENCY
 RECOSTMAX=650.0
 FREQBYMODE=T

;Control access/egress modes
DELACCESSMODE = 16        ; Exclude walk access links.

;Fare and Wait Times
 FARESYSTEM=1, MODE=1
 FARESYSTEM=2, MODE=2
 FARESYSTEM=3, MODE=3
 FARESYSTEM=4, MODE=4
 FARESYSTEM=5, MODE=5
 FARESYSTEM=6, MODE=6
```

Figure 8-9 shows an example of the graphical display of a park-n-ride bus-Metrorail path generated by the PT program.

Figure 8-9: Sample PNR Access Bus-Metrorail Transit Path Built by PT

9 PT Transit Fare Options (Task Order 9)

The MWCOG fare-building process calculates fares after transit paths are constructed. It includes a relatively complicated process of modeling station-to-station movements on the Metrorail system to calculate the constrained, distance-based fare and transfer rules included in the WMATA fare policy. The PT program includes a number of new and sophisticated methods of integrating fare calculations into the transit modeling process. It supports fare calculations after the path is built, but it also includes methods of integrating the fare into the path selection process. In addition, the WMATA fare policy is not simply service-based, but also varies by traveler type and payment method (i.e., cash or card). The regional transit authority also includes a number of employer-based transit discount programs that make the actual cost born by some travelers significantly different from other travelers. PT can model these types of differences using multiple traveler classes. This task analyzed how well the region's fare policies can be reproduced using the PT fare calculation tools and how paths and path weights are affected by considering fares in the path impedance.

AECOM had primary responsibility for developing the fare calculations and evaluating the strengths and weaknesses of the implementation options. MWCOG will have primary responsibility for validating the transit fares for various path configurations and testing and evaluating the modeling scripts.

AECOM reviewed the PT capabilities of calculating transit fares and incorporating fares into the path building and transit assignment process. It was found that PT benefits from a flexible transit fare processing method which is capable of considering transit fares to evaluate transit paths.

9.1 Transit Fare Calculation in the MWCOG Travel Model

The MWCOG travel model as well as WMATA's transit model[46] does not consider transit fares during the transit path building process. Both models use the TRNBUILD program which is not capable of considering transit fares while evaluating transit paths. Transit fares are calculated for the best transit path between zones using the MFARE1 and MFARE2 scripts. The main goal of estimating fares in the MWCOG model is to provide inputs to the mode choice model. The calculated fares are used to determine transit mode shares by line-haul mode. MFARE1 determines the transit fare for the Metrorail segment of a transit path. MFARE2 calculates transit fares for the "bus" (i.e., non-Metrorail transit) segments of a transit path and combines the calculated fare with the Metrorail fare if the path includes both Metrorail and "bus" segments. The MWCOG model does not consider transit fares for commuter rail paths.

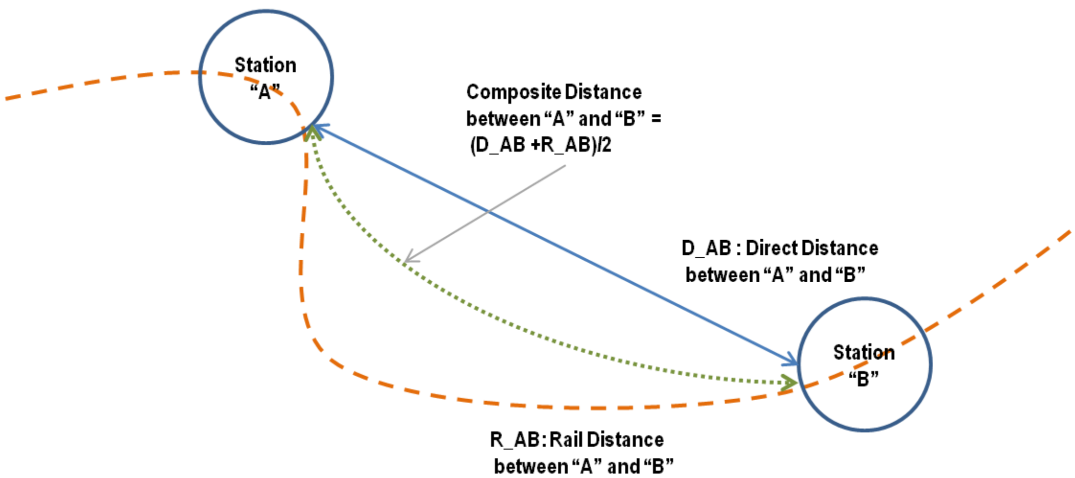
As per WMATA policy,[47] MFARE1 determines the Metrorail fares based on the "composite" distance traveled between the first boarding station "A" and the last alighting station "B" on the Metrorail

[46] AECOM Consult, Inc. (2005). Post MWCOG – AECOM Transit Component of Washington Regional Demand Forecasting Model: User's Guide. AECOM Consult, Inc. (Updated to Version 2.3 in 2011).

[47] See, for example, footnote #5 of page 4 of Washington Metropolitan Area Transit Authority, "Tariff of The Washington Metropolitan Area Transit Authority on Metro Operations Within the Washington Metropolitan Area, Tariff Number 32, Effective July 1, 2012," July 1, 2012, http://www.wmata.com/about_metro/docs/tariff.PDF.

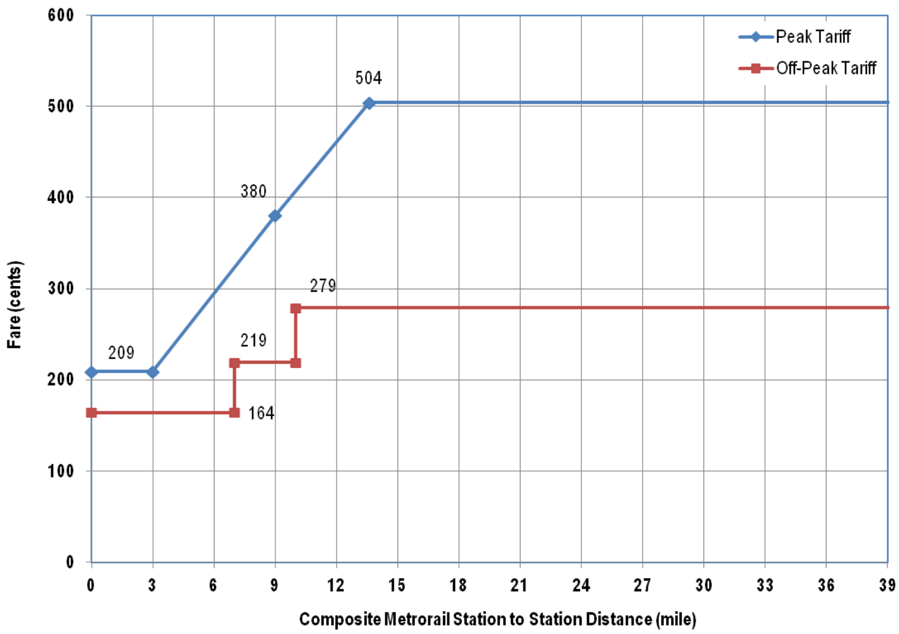
system. The composite distance is the average of the straight-line distance and the over-the-rail distance between station "A" and "B". Figure 9-1 shows the straight-line ("direct"), over-the-rail, and composite distance between station "A" and "B" on the Metrorail. The Metrorail fare is then determined using the composite distance and input tariffs. Figure 9-2 shows the tariff that the MWCOG model uses to determine Metrorail fare. Note: The current WMATA tariff is Tariff #32, but Figure 9-2 reflects an earlier WMATA tariff. Additionally, the fares shown in Figure 9-2 have been adjusted by MWCOG staff to create one composite fare for both the peak and off-peak periods. The composite fare is the average of the SmarTrip rate, the paper fare card rate, and the peak-of-the-peak surcharge.[48]

Figure 9-1: WMATA Method of Calculating Composite Distance between Metrorail Stations



[48] See, for example, Jane A. Posey to Files, "Metrorail and Metrobus Fares for the 2010 CLRP and FY 2011-2016 TIP Air Quality Conformity Analysis," Memorandum, July 12, 2010.

Figure 9-2: MWCOG Input Tariff for Calculating Metrorail Fare



Note: Although the current WMATA tariff is Tariff #32, the two lines in this figure reflect an earlier WMATA tariff. Additionally, the fares shown in this figure have been adjusted by MWCOG staff to create one composite fare for both the peak and off-peak periods. The composite fare is the average of the SmarTrip rate, the paper fare card rate, and the peak-of-the-peak surcharge.

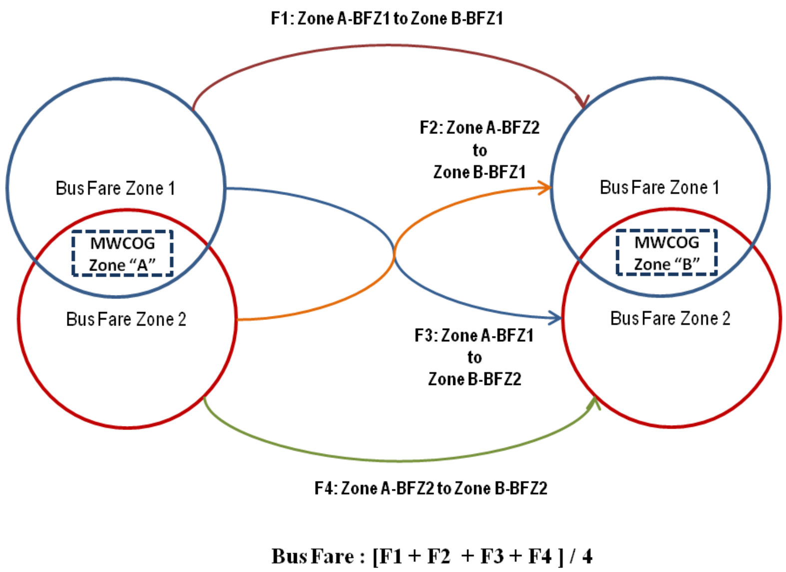
MFARE2 determines the fare for non-Metrorail transit (so called “bus”) segments of a transit path and later combines that with the calculated Metrorail fare if the transit path includes a Metrorail segment. The MWCOG model uses a zone-base fare system to estimate bus fares. A predetermined bus fare is assigned to travel between bus fare zones. Also, a bus fare zone is assigned to each MWCOG zone. MFARE2 uses these two fare values to determine non-Metrorail transit fares. The program also applies fare discounts to transfers between the bus and Metrorail systems. Access and egress bus fares are calculated separately for transit paths with a Metrorail segment.

The MWCOG model assigns a primary and secondary fare zone to zones with multiple services and fare structures[49]. For example, S.E. Fairfax County is served by Fairfax Connector (bus fare zone 1) and VRE commuter rail service (bus fare zone 18). Therefore in this area, each TAZ would have two bus fare zones (a primary and a secondary) listed in the TAZ/bus fare equivalence file. MFARE2 would calculate the cost of a trip from a TAZ in this area to downtown D.C. (bus fare zone 1) by averaging the cost of a trip from bus fare zone 1 to bus fare zone 1 with the cost of a trip from bus fare zone 18 to bus fare zone 1. It is possible that for some origin and destination combinations the average fare of four paths is considered to be the final transit fare for those interchanges. Figure 9-3 depicts MWCOG’s approach for calculating bus fare for traveling between two zones using a bus only path. For most of the zones bus

[49] See, for example, 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.

fare zone 1 and 2 are equal which means the fare is the same for all four paths. However, for approximately 700 of the zones bus fare zone 1 and 2 are different.

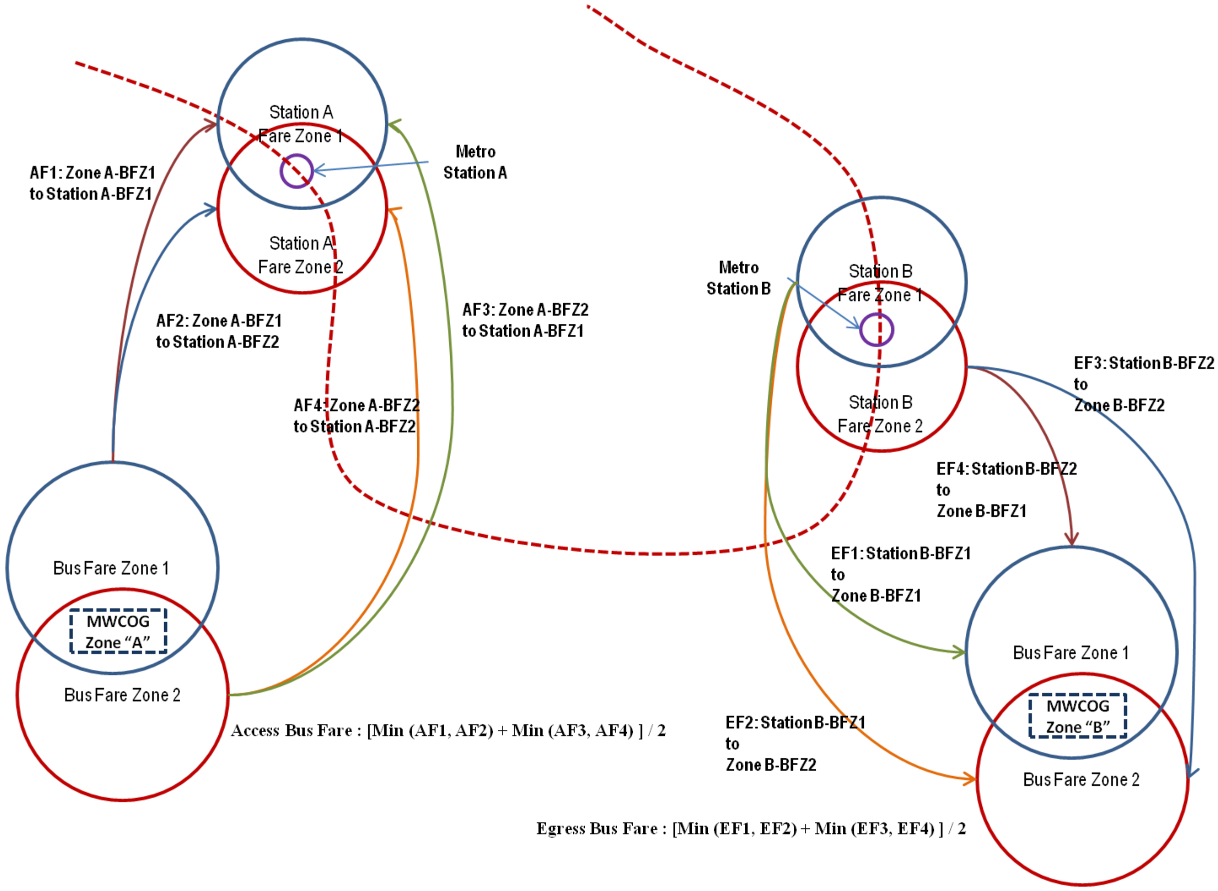
Figure 9-3: MWCOG's Approach to Calculate Bus Fare for Bus-Only Transit Paths



As mentioned above, access and egress bus fares to/from Metrorail stations are calculated for transit paths with a Metrorail segment. Basically, a bus fare zone is assigned to each Metrorail station and bus fares for accessing the first Metrorail station are estimated using the input bus fare zone matrix. The same procedure is applied between the last alighting Metrorail station and the destination zone.

The MWCOG model assigns a primary and secondary fare zone to each Metrorail station, similar to the methodology used for the bus-only paths. Therefore, four bus fares are estimated for the access and egress legs of a Metrorail trip. Figure 9-4 shows MWCOG's approach to calculate access and egress bus fares to/from Metrorail station. Transfer discounts are applied equally to calculated access and egress bus fares.

Figure 9-4: MWCOG's Approach to Calculate Bus Access/Egress Fares to Metrorail



9.2 PT Transit Fare Processing

The PT program is not only capable of calculating transit fares, but also can incorporate the calculated transit fares in the transit path building process. This section describes the limitation of the PT program to replicate MWCOG's fare estimation process and provides recommendations to replace the current process with a more accurate methodology.

Unfortunately, PT does not include the first node ("NODE0") and last node ("NODEL") path attributes. These values are used by the MWCOG process to identify the first and last stations used on the Metrorail system for fare calculations. Specifically, MWCOG uses this information to build a minimum-distance path between stations using Metrorail links as part of the composite distance calculate. The Metrorail fare is estimated based on the input tariff table and the calculated composite distance.

PT, on the other hand, is capable of reading fare matrices such as a station-to-station fare matrix for Metrorail trips. Therefore, PT can accurately calculate the fare for transit paths with/without a Metrorail segment.

In order to implement fare calculations in PT and use that information to evaluate transit paths, the following steps should be taken:

- The PT node attribute table should be modified to include a fare zone code for transit stops. Fare zones are either groups of nodes or single nodes. A Public Transport network might have multiple fare-zone schemes, but a fare system must have only one zone scheme. Therefore, a node can be in only one fare zone. Figure 9-5 shows the modified PT network node attribute table. The “MR_FAREZ” includes the Metrorail fare zone. AECOM recommends assigning a unique fare zone number to each Metrorail station. However, a unique fare zone number can be assigned to a group of bus stop to represent MWCOG’s 21 bus fare zones.

Figure 9-5: PT Network Node Attribute Table with Fare Zones Assigned to Metrorail Stations

17644 Nodes

N	X	Y	PNR	MR_FAREZ	BUS_FAREZ
8038	1307577	440592	0	38	0
8039	1310669	440605	0	39	0
8040	1313614	435895	11040	40	0
8041	1315875	429106	0	41	0
8042	1319340	427615	11042	42	0
8043	1324683	431302	11043	43	0
8044	1331647	428855	11044	44	0
8045	1337334	422607	11045	45	0
8046	1275089	412373	11046	46	0

- Peak and off-peak fare matrices could then be prepared and placed in the input folder. The MWCOG input folder currently includes fare matrices representing bus fare zones for peak and off-peak time period. Those matrices can be used unchanged. However, Metrorail fare matrices should be prepared. Since Metrorail has a set fares for traveling between stations by time of day, preparation of these matrices should be relatively easy.
- A PT fare file could then be prepared that includes multiple FARESYSTEM statements defining the structure of each fare system. Figure 9-6 shows a sample PT fare file defining the structure of the fare systems for bus and Metrorail modes. Although PT can include different fare structures, the “FROMTO” structure is best suited for the MWCOG model. This structure is also capable of considering penalties as well as discounts for transferring between fare systems. Fare systems are assigned to transit modes using the PT factor file. For the Metrorail path, the node variables, NI.MR_FAREZ, defines the fare zones for Metrorail transit stops, and the program extracts the fare from fare matrix number 2, input on FAREMATI[2], row NI. MR_FAREZ (first boarding station) and column NI. MR_FAREZ (last alighting station). There are no boarding costs or costs for transferring within the same fare system. However, there is 50-cent discount for transferring between fare system 1 and 2.

Figure 9-6: Sample PT Fare File

```
;PT Fare File;MWCOG Version 2.3
FARESYSTEM NUMBER=1, NAME="Bus Fare system", STRUCTURE="FROMTO", FAREMATRIX=FMI.1.1,
FAREZONES=NI.Bus_FAREZ, FAREFROMFS[1]=0,-0.5
FARESYSTEM NUMBER=2, NAME="Metro Fare System", STRUCTURE="FROMTO", FAREMATRIX=FMI.2.1,
FAREZONES=NI.MR_FAREZ, FAREFROMFS[1]=-0.5,0
```

- The PT factor file is used to link fare systems to transit modes and include the required statement, "ENUMFARE=T", to ask PT to include the fare as one of the factors to find the best transit path. Figure 9-7 shows a part of a PT factor file that needs to be modified in order to consider fare as part of path building process. The following script assigns fare system 1 to bus modes and fare system 2 to Metrorail and LRT.

Figure 9-7: Sample PT Factor File to include Fare in Path Building Process

```
;PT Factors File for ******** Metrorail AM Drive Access ********
;MWCOG Version 2.3

;Global Settings
 BESTPATHONLY=T, ENUMFARE=T
 MAXFERS=3
 SERVICEMODEL=FREQUENCY
 RECOSTMAX=650.0
 FREQBYMODE=T


;Fare System
 FARESYSTEM=1,  MODE=1,2,6,7,8,9,10
 FARESYSTEM=2,  MODE=3,5
```

- PT transit skim and transit assignment scripts should also be modified. Basically, it is possible to ask PT to calculate transit fares and evaluate transit paths based on fares or just calculate and skim the fare for the best path. Figure 9-8 shows a sample PT transit skim script where highlighted items show required changes to incorporate transit fares in the transit path building process and report the fare.

Figure 9-8: Public Transport Skim Script

```
RUN PGM = PUBLIC TRANSPORT; MSG = 'Metrorail AM WK transit paths and skims'

FILEI NETI = "PT.NET"

FILEO MATO[1] = "MR_AM_WK.SKM", MO=1-22,
  NAME = IVLB, IVXB, IVMT, IVCR, IVNM, INIT, XFERT, WACCT, WLKT, BRDS,
  DACCT, DACCD, PRKT, PRKC, ISTOS, JSTOS, DMR, DLRT, MRFAREA, MRFAREP

FILEO ROUTEO[1] = "MR_AM_WK.RTE"

FILEO REPORTO = "PT_Skims_MR_AM_WK.PRN"

FILEO NETO = "MR_AM_WK.NET"
FILEO NTLEGO ="NTLEG_MR_AM_WK.NT"

;Input ModeAccess specific factors
FILEI FACTORI[1] = "..\Parameters\MR_AM_WK.FAC"
FILEI SYSTEMI = "TSYSD.PTS"

;---- Transit Line Cards (modes 1-10) ----

FILEI LINEI[1] = "MODE3AM.LIN"  ;---- M3- metrorail
;FILEI LINEI[2] = "MODE5AM.LIN"  ;---- M5- other rail (future)

;---- Access Links (modes 11, 12 and 16) ----

FILEI NTLEGI[1] = met_bus.nt   ;--- bus-metro links&xfer cards
;FILEI NTLEGI[2] = lrt_bus.nt   ;--- bus-LRT links&xfer cards
```

```
FILEI NTLEGI[3] = walkacc.nt  ;--- walk to local transit

;---- Sidewalk Network (mode 13) ----

FILEI NTLEGI[4] = sidewalk.nt  ;--- walk network for transfers
; READ FILE = inputs\pedtunnels.asc ;--- pedestrian tunnels

FILEI FAREMATI[1] = "..\Fare\AM Bus Fares.MTX"
FILEI FAREMATI[2] = "..\Fare\AM Metrorail Fares.MTX"
FILEI FAREI="..\Fare\PT PK Fare.far"

zonemsg=50

; OVERALL PARAMETERS OF RUN
PARAMETERS EFARE=T, HDWAYPERIOD=1,NOROUTEERRS=999999,
           NOROUTEMSGS=999999,SKIPBADLINES=Y,
          TRANTIME=LI.TRANTIME

REPORT LINES=T
PROCESS PHASE=LINKREAD
 LW.TRANTIME=LI.TRANTIME
 LW.WALKTIME=(LI.DISTANCE/3*60)
 LW.WALKDISTANCE=LI.DISTANCE
 LW.DISTANCE=LI.DISTANCE
ENDPROCESS

PROCESS PHASE=DATAPREP
    ; Generate xfer-access links
        GENERATE READNTLEGI=1
;       GENERATE READNTLEGI=2
        GENERATE READNTLEGI=3
        GENERATE READNTLEGI=4
ENDPROCESS

PROCESS PHASE=SKIMIJ

;---- specify output skims ----

 MW[1] = TIMEA(0,1,6,8),                        ;---- ivt-local bus    (0.01 min)
 MW[2] = TIMEA(0,2,7,9),                        ;---- ivt-exp bus      (0.01 min)
 MW[3] = TIMEA(0,3),                            ;---- ivt-metrorail    (0.01 min)
 MW[4] = TIMEA(0,4),                            ;---- ivt-commuter rail(0.01 min)
 MW[5] = TIMEA(0,5),                            ;---- ivt-new rail mode(0.01 min)
 MW[6] = TIMEA(0,10),                           ;---- ivt-new bus mode (0.01 min)
 MW[7] = IWAITA(0),                             ;---- ini.wait time    (0.01 min)
 MW[8] = XWAITA(0),                             ;---- xfr wait time    (0.01 min)
 MW[9] = TIMEA(0,14) + TIMEA(0,16)/2.0,         ;---- walk acc time    (0.01 min)
 MW[10] = TIMEA(0,12) + TIMEA(0,13)/2.0,        ;---- other walk time  (0.01 min)
 MW[11] = XFERPENA(0),                          ;---- added xfer time  (0.01 min)
 MW[12] = BRDINGS(0),                           ;---- boardings        (1+)
 MW[13] = TIMEA(0,11),                          ;---- drv acc time     (0.01 min)
 MW[14] = DIST(0,11),                           ;---- drv acc distance (0.01 mile)
 MW[15] = TIMEA(0,15),                          ;---- pnr impedance    (0.01 min)
 MW[16] = DIST(0,15),                           ;---- pnr cost         (cents)
; MW[17] = NODE0(0,3) - 8000.0,                 ;---- metro board  sta (1-150)
; MW[18] = NODEL(0,3) - 8000.0,                 ;---- metro alight sta (1-150)
 MW[19] = DIST(0,3),                            ;---- metrorail distance
 MW[20] = DIST(0,5)                             ;---- light rail distance
 MW[21] = FAREA(0,3)                            ;---- Metro Fare in monetary units
 MW[22] = FAREP(0,3)                  ;---- Metro generalized cost with mode specific VALUEOFTIME

ENDPROCESS

ENDRUN
```

10 ModeChoice Software (Task Order 9)

As an addition to the original scope of work for Task Order 9, MWCOG asked AECOM to prepare a memorandum outlining the advantages of migrating from the AEMS software to the ModeChoice software for mode choice modeling. In response, AECOM wrote a memo, dated February 1, 2013, which is summarized in this section of the report. In addition, AECOM prepared a prototype conversion of the MWCOG HBW mode choice model to the ModeChoice software for evaluation purposes.

10.1 Background

The AEMS program was original developed by Bill Woodford at KPMG in the early 1990s. It is a console-based FORTRAN program that provides a control file interface for implementing a wide variety of mode choice model structures. It reads and writes trip tables and path skim matrix files in the native format of several travel demand software vendors, including MinUTP, TPPlus, Cube, TransCAD, and Emme2. The program is compiled separately for each software interface. As such it is often integrated into regional modeling tools using different names for different clients (e.g., MINMS, TPPMS, TCMS, EM2MS, etc.). AEMS is the name assigned to the program when the transportation group at KPMG was purchased by AECOM in 2000. It is shorthand for AECOM Mode Split. The software includes a companion program called CALIBMS for assisting with mode choice calibration efforts. The software is still in use in numerous travel demand forecasting systems throughout the country.

David Roden joined KPMG shortly before the transportation group was purchased by AECOM. Prior to joining KPMG, he developed the System II travel demand software package at JHK and applied the software for many clients in the Washington region. Shortly after joining AECOM, David became the project manager for the FHWA research project to implement the TRANSIMS software in Portland, Oregon. As part of this project and several subsequent projects sponsored by FHWA, Mr. Roden re-wrote the TRANSIMS software and made it available free of charge through the TRANSIMS Open-Source website. When Bill Woodford left AECOM to join another firm in 2010, Mr. Roden used the software libraries developed for TRANSIMS to convert the AEMS program to the TRANSIMS Open-Source platform under the name ModeChoice. The program has been used to implement mode choice models for WMATA and CFRPM (Central Florida Regional Planning Model) in Orlando, Florida.

10.2 Advantages of Migrating from AEMS to ModeChoice

The ModeChoice program is designed to replicate most of the capabilities of the AEMS program within a user interface and software maintenance framework that should ensure its continued viability. The following bullets describe many of benefits of migrating from AEMS to ModeChoice in more detail.

- AEMS is an old FORTRAN program that is not easily compiled using currently available FORTRAN compilers. The ModeChoice program is a C++ program that is easily compiled using the latest C++ compliers for 32-bit and 64-bit Windows and Linux operating systems.

- The AEMS code is also poorly organized and not written with the expectation that a programmer other than Bill Woodford will be able to understand or update it. The ModeChoice code is specifically designed for distribution through the TRANSIMS Open-Source website and, as such, is structured and written to be understood and potentially improved by other programmers.

- There is no active maintenance or support for the AEMS program. The ModeChoice program is supported and maintained through the TRANSIMS e-mail distribution list and SourceForge version control system. David Roden continues to improve and expand the capabilities of the software as new applications, compiler options and 64-bit versions of TransCAD or Cube are released.

- AEMS is compiled separately for Cube, TransCAD, and EMME2. ModeChoice can be compiled to work with Cube and TransCAD interchangeably, plus a generic TRANSIMS matrix format.

- In order to improve processing time, AEMS internally converts the log functions to a piece-wise lookup table with straight-line interpolation between data points. The ModeChoice program uses the log functions directly. This makes ModeChoice slightly more accurate and generates small differences in the resulting mode shares.

- The user interface for the AEMS program is quite complex and rigid in its column format. If an extra space is added or missing, the program executes without detecting an error and generates erroneous results. The organization of elements and variables is not very logical or sequential and it requires a considerable number of records to simply initialize variables or fill out unused elements of the control file. A separate control file with the full model specification is required for each trip purpose.

- The ModeChoice program uses the standard TRANSIMS control file structure that includes control keys and values that can be ordered and formatted in many ways. It also includes an integrated help system and data valuation checks. The control file is used to define the input and output file names, reports, and overall model structure. The model utility functions and conditional overrides are defined in a user script file that can be common for all trip purposes. This script uses a programming syntax similar to FORTRAN and C with a full range of variable names, lookup tables, and logical syntax statements (e.g., if-then-else-endif) that provide exceptional both flexibility and computational clarity at the same time.

- The AEMS and ModeChoice programs convert the model structure and computations defined in the control file and user script to internal processing commands. The syntax compiler included in the ModeChoice program generates more computationally efficient processing procedures than the AEMS counterpart. Side-by-side performance tests suggest the ModeChoice program reduces processing time by at least 40 percent.

- The ModeChoice program includes a number of report and output file options that are not available in AEMS.

- The ModeChoice program has a built-in calibration capability that iteratively adjusts the mode constants to match a set of mode share targets. This is similar to the CALIBMS program that is available for AEMS. The output file of the ModeChoice calibration application is in the exact format needed for production applications of the program. The output of the CALIBMS program needs to be integrated into the control file for AEMS.

- In addition to calibrating the mode constants, the ModeChoice program includes a secondary calibration adjustment to the demographic subdivision constants within a mode (e.g., income or auto occupancy categories). CALIBMS does not adjust these constants.
- The calibration target file used by ModeChoice includes the option to specify minimum and maximum values for each mode constant. This helps to keep the constant values within a reasonable range. One of the concerns in using CALIBMS is that it does not constrain the constant values and often generates values that are illogical and need to be adjusted manually for input to the AEMS program.
- The ModeChoice program includes an RMSE convergence parameter to stop the calibration progress at the required level of precision or when the maximum number of iterations is reached. CALIBMS uses a fixed number of iterations.

10.3 Prototype Conversion of the MWCOG HBW Model

In updating the WMATA modeling process to be compatible with the MWCOG Version 2.3 travel model, AECOM converted the AEMS mode choice models to the ModeChoice software and recalibrated the models to accommodate the new zone structure and change in person trip rates generated by the new regional model. Since MWCOG's implementation of the AEMS software in the Version 2.3 travel model is similar to the structure used in the original WMATA model, it was relatively straightforward to adjust the WMATA ModeChoice files to replicate the MWCOG Version 2.3 mode choice inputs.

The first input file to the ModeChoice program is a control file that specifies file names, defines the model structure and the parameters for standard attributes, and selects reports and output options. The control file for the HBW model is shown in Figure 10-1.

Figure 10-1: ModeChoice Control File

```
TITLE                        MWCOG Version 2.3 AM Peak HBW Mode Choice

PROJECT_DIRECTORY

TRIP_FILE                    HBW_INCOME.PTT
SKIM_FILE_2                  HWYAM.SKM
SKIM_FILE_3                  TRNAM_CR.SKM
SKIM_FILE_4                  TRNAM_AB.SKM
SKIM_FILE_5                  TRNAM_MR.SKM
SKIM_FILE_6                  TRNAM_BM.SKM
ZONE_FILE                    ZONEV2.A2F
NEW_TRIP_FILE                HBW_NL_MC.MTT

MODE_CONSTANT_FILE           ..\Controls\HBW_AM_Constant.txt
MODE_CHOICE_SCRIPT           ..\Scripts\Mode_Choice_Script.txt

SEGMENT_MAP_FILE             ..\Controls\Segment_Map.txt
ORIGIN_MAP_FIELD             AREA_TYPE
DESTINATION_MAP_FIELD        AREA_TYPE

TRIP_PURPOSE_LABEL           Peak Home-Based Work
```

```
TRIP_PURPOSE_NUMBER          1 //---- HBW=1, HBS=2, HBO=3, NHB=4 ----
TRIP_TIME_PERIOD             1 //---- PEAK=1, OFFPEAK=2 ----

PRIMARY_MODE_CHOICE          AUTO, TRANSIT
MODE_CHOICE_NEST_1           AUTO = SOV, HOV
MODE_CHOICE_NEST_2           HOV = SR2, SR3
MODE_CHOICE_NEST_3           TRANSIT = WALK, PNR, KNR
MODE_CHOICE_NEST_4           WALK = WK_CR, WK_BUS, WK_BUS_MR, WK_MR
MODE_CHOICE_NEST_5           PNR = PNR_CR, PNR_BUS, PNR_BUS_MR, PNR_MR
MODE_CHOICE_NEST_6           KNR = KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
NESTING_COEFFICIENT          0.5

VEHICLE_TIME_VALUE          -0.02128
WALK_TIME_VALUE             -0.04256
DRIVE_ACCESS_VALUE          -0.03192
WAIT_TIME_VALUE             -0.05320
TRANSFER_COUNT_VALUE        -0.00000
PENALTY_TIME_VALUE          -0.05320
TERMINAL_TIME_VALUE         -0.05320
COST_VALUE_TABLE_1          -0.00185
COST_VALUE_TABLE_2          -0.00093
COST_VALUE_TABLE_3          -0.00062
COST_VALUE_TABLE_4          -0.00046

MODE_ACCESS_MARKET_1        SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, WK_MR, PNR_CR, PNR_BUS,
                                 PNR_BUS_MR, PNR_MR, KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_2        SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, PNR_CR, PNR_BUS, PNR_BUS_MR,
                                 PNR_MR, KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_3        SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, PNR_CR, PNR_BUS, PNR_BUS_MR,
                                 KNR_CR, KNR_BUS, KNR_BUS_MR
MODE_ACCESS_MARKET_4        SOV, SR2, SR3, PNR_CR, PNR_BUS, PNR_BUS_MR, PNR_MR, KNR_CR, KNR_BUS,
                                 KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_5        SOV, SR2, SR3, PNR_CR, PNR_BUS, PNR_BUS_MR, KNR_CR, KNR_BUS,
                                 KNR_BUS_MR
MODE_ACCESS_MARKET_6        SOV, SR2, SR3

NEW_TABLE_MODES_1           SOV = Drive Alone
NEW_TABLE_MODES_2           SR2 = Shared Ride 2
NEW_TABLE_MODES_3           SR3 = Shared Ride 3
NEW_TABLE_MODES_4           WK_CR = Walk to Commuter Rail
NEW_TABLE_MODES_5           WK_BUS = Walk to Bus
NEW_TABLE_MODES_6           WK_BUS_MR = Walk to Bus/Metrorail
NEW_TABLE_MODES_7           WK_MR = Walk to Metrorail
NEW_TABLE_MODES_8           PNR_CR, KNR_CR = PNR/KNK to Commuter Rail
NEW_TABLE_MODES_9           PNR_BUS = PNR to Bus
NEW_TABLE_MODES_10          KNR_BUS = KNR to Bus
NEW_TABLE_MODES_11          PNR_BUS_MR = PNR to Bus/Metrorail
NEW_TABLE_MODES_12          KNR_BUS_MR = KNR to Bus/Metrorail
NEW_TABLE_MODES_13          PNR_MR = PNR to Metrorail
NEW_TABLE_MODES_14          KNR_MR = KNR to Metrorail

NEW_MODE_SUMMARY_FILE       Results/HBW_AM_Summary.txt
NEW_MARKET_SEGMENT_FILE     Results/HBW_AM_Segment.txt
NEW_MODE_SEGMENT_FILE       Results/HBW_AM_Mode_Seg.txt
NEW_PRODUCTION_FILE         Results/HBW_AM_Productions.txt
NEW_ATTRACTION_FILE         Results/HBW_AM_Attractions.txt

MODECHOICE_REPORT_1         MARKET_SEGMENT_REPORT
MODECHOICE_REPORT_2         MODE_SUMMARY_REPORT
MODECHOICE_REPORT_3         MODE_VALUE_SUMMARY
```

Since the MWCOG model includes 20 geographic market segments, a segment map file is required to convert the area group of the origin and destination zone into a market segment number. The segment map file in tab-delimited format is shown in Figure 10-2.

Figure 10-2: Market Segment Map File

ORG	DES	SEGMENT
1	1	1
1	2	2
1	3	3
1	4	3
1	5	3
1	6	4
1	7	4
2	1	9
2	2	10
2	3	11
2	4	11
2	5	11
2	6	12
2	7	12
3	1	1
3	2	2
3	3	3
3	4	3
3	5	3
3	6	4
3	7	4
4	1	5
4	2	6
4	3	7
4	4	7
4	5	7
4	6	8
4	7	8
5	1	9
5	2	10
5	3	11
5	4	11
5	5	11
5	6	12
5	7	12
6	1	13
6	2	14
6	3	15
6	4	15
6	5	15
6	6	16
6	7	16
7	1	17
7	2	18
7	3	19
7	4	19
7	5	19
7	6	20
7	7	20

Where the area groups for the origin and destination zone are defined using seven geographic markets:

1. DC Core
2. Virginia Core
3. DC Urban
4. Maryland Urban
5. Virginia Urban
6. Maryland Other
7. Virginia Other

A mode constant file is also needed to set the mode-specific constants for the HBW model. These constants are provided for each mode within each market segment and include income-based adjustment factors. The income-related constants are identified as HBWI1Psn through HBWI4Psn in Figure 10-3. These correspond to the table names in the input person trip table.

Figure 10-3: HWB Mode Constant File

SEGMENT	MODE	CONSTANT	HBWI1Psn	HBWI2Psn	HBWI3Psn	HBWI4Psn
1	AUTO	0	0.0	0.0	0.0	0.0
1	TRANSIT	3.72445	0.0	0.0	0.0	0.0
1	SOV	0	0.0	0.0	0.0	0.0
1	HOV	-1.29504	0.0	0.0	0.0	0.0
1	SR2	0	0.0	0.0	0.0	0.0
1	SR3	-1.55713	0.0	0.0	0.0	0.0
1	WALK	0	0.0	0.0	0.0	0.0
1	PNR	-3.76433	0.0	0.0	0.0	0.0
1	KNR	-7.33524	0.0	0.0	0.0	0.0
1	WK_CR	-0.80725	2.0	0.0	0.0	-2.0
1	WK_BUS	-1.44958	2.0	0.0	0.0	-2.0
1	WK_BUS_MR	-1.46039	2.0	0.0	0.0	-2.0
1	WK_MR	0	2.0	0.0	0.0	-2.0
1	PNR_CR	-0.39351	0.0	0.0	0.0	0.0
1	PNR_BUS	-2.45057	0.0	0.0	0.0	0.0
1	PNR_BUS_MR	0.85057	0.0	0.0	0.0	0.0
1	PNR_MR	0	0.0	0.0	0.0	0.0
1	KNR_CR	3.57299	0.0	0.0	0.0	0.0
1	KNR_BUS	1.26089	0.0	0.0	0.0	0.0
1	KNR_BUS_MR	5.74345	0.0	0.0	0.0	0.0
1	KNR_MR	0	0.0	0.0	0.0	0.0
2	AUTO	0	0.0	0.0	0.0	0.0
2	TRANSIT	4.41614	0.0	0.0	0.0	0.0
2	SOV	0	0.0	0.0	0.0	0.0
2	HOV	-1.77697	0.0	0.0	0.0	0.0
2	SR2	0	0.0	0.0	0.0	0.0
2	SR3	-0.97468	0.0	0.0	0.0	0.0
2	WALK	0	0.0	0.0	0.0	0.0
2	PNR	-6.15269	0.0	0.0	0.0	0.0
2	KNR	-9.76278	0.0	0.0	0.0	0.0
2	WK_CR	-2.65644	2.0	0.0	0.0	-2.0
2	WK_BUS	-14.71756	2.0	0.0	0.0	-2.0
2	WK_BUS_MR	-5.70638	2.0	0.0	0.0	-2.0
2	WK_MR	0	2.0	0.0	0.0	-2.0
2	PNR_CR	-0.73389	0.0	0.0	0.0	0.0
2	PNR_BUS	-0.73389	0.0	0.0	0.0	0.0
2	PNR_BUS_MR	0.05	0.0	0.0	0.0	0.0
2	PNR_MR	0	0.0	0.0	0.0	0.0
2	KNR_CR	0.38242	0.0	0.0	0.0	0.0
2	KNR_BUS	0.38242	0.0	0.0	0.0	0.0
2	KNR_BUS_MR	9.27713	0.0	0.0	0.0	0.0
2	KNR_MR	0	0.0	0.0	0.0	0.0
3	AUTO	0	0.0	0.0	0.0	0.0
3	TRANSIT	6.67769	0.0	0.0	0.0	0.0
3	SOV	0	0.0	0.0	0.0	0.0
3	HOV	-1.45163	0.0	0.0	0.0	0.0
3	SR2	0	0.0	0.0	0.0	0.0
3	SR3	-1.2373	0.0	0.0	0.0	0.0
3	WALK	0	0.0	0.0	0.0	0.0
3	PNR	-8.09017	0.0	0.0	0.0	0.0
3	KNR	-11.27367	0.0	0.0	0.0	0.0
3	WK_CR	-5.64991	2.0	0.0	0.0	-2.0
3	WK_BUS	-9.07725	2.0	0.0	0.0	-2.0
3	WK_BUS_MR	-8.59551	2.0	0.0	0.0	-2.0
3	WK_MR	0	2.0	0.0	0.0	-2.0
3	PNR_CR	-2.3531	0.0	0.0	0.0	0.0
3	PNR_BUS	-9.58041	0.0	0.0	0.0	0.0
3	PNR_BUS_MR	-7.89452	0.0	0.0	0.0	0.0
3	PNR_MR	0	0.0	0.0	0.0	0.0
3	KNR_CR	-0.1115	0.0	0.0	0.0	0.0

3	KNR_BUS	-3.90387	0.0	0.0	0.0	0.0
3	KNR_BUS_MR	0.84566	0.0	0.0	0.0	0.0
3	KNR_MR	0	0.0	0.0	0.0	0.0
4	AUTO	0	0.0	0.0	0.0	0.0
4	TRANSIT	6.39636	0.0	0.0	0.0	0.0
4	SOV	0	0.0	0.0	0.0	0.0
4	HOV	-1.85795	0.0	0.0	0.0	0.0
4	SR2	0	0.0	0.0	0.0	0.0
4	SR3	-1.25793	0.0	0.0	0.0	0.0
4	WALK	0	0.0	0.0	0.0	0.0
4	PNR	-10.41608	0.0	0.0	0.0	0.0
4	KNR	-12.058	0.0	0.0	0.0	0.0
4	WK_CR	-23.21476	2.0	0.0	0.0	-2.0
4	WK_BUS	-22.60831	2.0	0.0	0.0	-2.0
4	WK_BUS_MR	-22.95296	2.0	0.0	0.0	-2.0
4	WK_MR	0	2.0	0.0	0.0	-2.0
4	PNR_CR	-0.12203	0.0	0.0	0.0	0.0
4	PNR_BUS	-7.87212	0.0	0.0	0.0	0.0
4	PNR_BUS_MR	-6.3297	0.0	0.0	0.0	0.0
4	PNR_MR	0	0.0	0.0	0.0	0.0
4	KNR_CR	1.27847	0.0	0.0	0.0	0.0
4	KNR_BUS	-1.79718	0.0	0.0	0.0	0.0
4	KNR_BUS_MR	-3.84583	0.0	0.0	0.0	0.0
4	KNR_MR	0	0.0	0.0	0.0	0.0
5	AUTO	0	0.0	0.0	0.0	0.0
5	TRANSIT	3.38848	0.0	0.0	0.0	0.0
5	SOV	0	0.0	0.0	0.0	0.0
5	HOV	-1.53749	0.0	0.0	0.0	0.0
5	SR2	0	0.0	0.0	0.0	0.0
5	SR3	-1.78019	0.0	0.0	0.0	0.0
5	WALK	0	0.0	0.0	0.0	0.0
5	PNR	-6.69365	0.0	0.0	0.0	0.0
5	KNR	-8.68604	0.0	0.0	0.0	0.0
5	WK_CR	-3.88773	2.0	0.0	0.0	-2.0
5	WK_BUS	-10.33699	2.0	0.0	0.0	-2.0
5	WK_BUS_MR	-9.34656	2.0	0.0	0.0	-2.0
5	WK_MR	0	2.0	0.0	0.0	-2.0
5	PNR_CR	-0.67674	0.0	0.0	0.0	0.0
5	PNR_BUS	-5.49833	0.0	0.0	0.0	0.0
5	PNR_BUS_MR	0.80238	0.0	0.0	0.0	0.0
5	PNR_MR	0	0.0	0.0	0.0	0.0
5	KNR_CR	0.31162	0.0	0.0	0.0	0.0
5	KNR_BUS	0.9812	0.0	0.0	0.0	0.0
5	KNR_BUS_MR	7.14475	0.0	0.0	0.0	0.0
5	KNR_MR	0	0.0	0.0	0.0	0.0
6	AUTO	0	0.0	0.0	0.0	0.0
6	TRANSIT	2.26058	0.0	0.0	0.0	0.0
6	SOV	0	0.0	0.0	0.0	0.0
6	HOV	-1.47327	0.0	0.0	0.0	0.0
6	SR2	0	0.0	0.0	0.0	0.0
6	SR3	-2.5596	0.0	0.0	0.0	0.0
6	WALK	0	0.0	0.0	0.0	0.0
6	PNR	-4.23119	0.0	0.0	0.0	0.0
6	KNR	-5.48867	0.0	0.0	0.0	0.0
6	WK_CR	-2.68777	2.0	0.0	0.0	-2.0
6	WK_BUS	-11.29239	2.0	0.0	0.0	-2.0
6	WK_BUS_MR	-7.23534	2.0	0.0	0.0	-2.0
6	WK_MR	0	2.0	0.0	0.0	-2.0
6	PNR_CR	-0.87644	0.0	0.0	0.0	0.0
6	PNR_BUS	-0.87644	0.0	0.0	0.0	0.0
6	PNR_BUS_MR	-0.25151	0.0	0.0	0.0	0.0
6	PNR_MR	0	0.0	0.0	0.0	0.0
6	KNR_CR	-0.5444	0.0	0.0	0.0	0.0
6	KNR_BUS	-0.5444	0.0	0.0	0.0	0.0
6	KNR_BUS_MR	-0.5444	0.0	0.0	0.0	0.0
6	KNR_MR	0	0.0	0.0	0.0	0.0
7	AUTO	0	0.0	0.0	0.0	0.0
7	TRANSIT	2.1782	0.0	0.0	0.0	0.0
7	SOV	0	0.0	0.0	0.0	0.0
7	HOV	-1.70324	0.0	0.0	0.0	0.0
7	SR2	0	0.0	0.0	0.0	0.0

7	SR3	-1.72701	0.0	0.0	0.0	0.0
7	WALK	0	0.0	0.0	0.0	0.0
7	PNR	-6.4478	0.0	0.0	0.0	0.0
7	KNR	-7.67687	0.0	0.0	0.0	0.0
7	WK_CR	-3.64739	2.0	0.0	0.0	-2.0
7	WK_BUS	-5.05571	2.0	0.0	0.0	-2.0
7	WK_BUS_MR	-5.49456	2.0	0.0	0.0	-2.0
7	WK_MR	0	2.0	0.0	0.0	-2.0
7	PNR_CR	-1.30044	0.0	0.0	0.0	0.0
7	PNR_BUS	-4.34816	0.0	0.0	0.0	0.0
7	PNR_BUS_MR	-1.66072	0.0	0.0	0.0	0.0
7	PNR_MR	0	0.0	0.0	0.0	0.0
7	KNR_CR	-4.37215	0.0	0.0	0.0	0.0
7	KNR_BUS	-0.01143	0.0	0.0	0.0	0.0
7	KNR_BUS_MR	2.83679	0.0	0.0	0.0	0.0
7	KNR_MR	0	0.0	0.0	0.0	0.0
8	AUTO	0	0.0	0.0	0.0	0.0
8	TRANSIT	1.73906	0.0	0.0	0.0	0.0
8	SOV	0	0.0	0.0	0.0	0.0
8	HOV	-2.122	0.0	0.0	0.0	0.0
8	SR2	0	0.0	0.0	0.0	0.0
8	SR3	-1.07137	0.0	0.0	0.0	0.0
8	WALK	0	0.0	0.0	0.0	0.0
8	PNR	-5.88393	0.0	0.0	0.0	0.0
8	KNR	-8.39535	0.0	0.0	0.0	0.0
8	WK_CR	-7.98029	2.0	0.0	0.0	-2.0
8	WK_BUS	-6.9402	2.0	0.0	0.0	-2.0
8	WK_BUS_MR	-7.9319	2.0	0.0	0.0	-2.0
8	WK_MR	0	2.0	0.0	0.0	-2.0
8	PNR_CR	-2.00162	0.0	0.0	0.0	0.0
8	PNR_BUS	-1.14146	0.0	0.0	0.0	0.0
8	PNR_BUS_MR	-2.94853	0.0	0.0	0.0	0.0
8	PNR_MR	0	0.0	0.0	0.0	0.0
8	KNR_CR	0.50461	0.0	0.0	0.0	0.0
8	KNR_BUS	4.30963	0.0	0.0	0.0	0.0
8	KNR_BUS_MR	1.68178	0.0	0.0	0.0	0.0
8	KNR_MR	0	0.0	0.0	0.0	0.0
9	AUTO	0	0.0	0.0	0.0	0.0
9	TRANSIT	7.03008	0.0	0.0	0.0	0.0
9	SOV	0	0.0	0.0	0.0	0.0
9	HOV	-1.46918	0.0	0.0	0.0	0.0
9	SR2	0	0.0	0.0	0.0	0.0
9	SR3	-1.94766	0.0	0.0	0.0	0.0
9	WALK	0	0.0	0.0	0.0	0.0
9	PNR	-12.46855	0.0	0.0	0.0	0.0
9	KNR	-14.4278	0.0	0.0	0.0	0.0
9	WK_CR	-25.37241	2.0	0.0	0.0	-2.0
9	WK_BUS	-21.15433	2.0	0.0	0.0	-2.0
9	WK_BUS_MR	-17.20596	2.0	0.0	0.0	-2.0
9	WK_MR	0	2.0	0.0	0.0	-2.0
9	PNR_CR	0.38872	0.0	0.0	0.0	0.0
9	PNR_BUS	0.66486	0.0	0.0	0.0	0.0
9	PNR_BUS_MR	0.59496	0.0	0.0	0.0	0.0
9	PNR_MR	0	0.0	0.0	0.0	0.0
9	KNR_CR	0.26627	0.0	0.0	0.0	0.0
9	KNR_BUS	0.26627	0.0	0.0	0.0	0.0
9	KNR_BUS_MR	8.78342	0.0	0.0	0.0	0.0
9	KNR_MR	0	0.0	0.0	0.0	0.0
10	AUTO	0	0.0	0.0	0.0	0.0
10	TRANSIT	1.73132	0.0	0.0	0.0	0.0
10	SOV	0	0.0	0.0	0.0	0.0
10	HOV	-1.79093	0.0	0.0	0.0	0.0
10	SR2	0	0.0	0.0	0.0	0.0
10	SR3	-1.37094	0.0	0.0	0.0	0.0
10	WALK	0	0.0	0.0	0.0	0.0
10	PNR	-5.88064	0.0	0.0	0.0	0.0
10	KNR	-8.47752	0.0	0.0	0.0	0.0
10	WK_CR	-3.13572	2.0	0.0	0.0	-2.0
10	WK_BUS	-5.72946	2.0	0.0	0.0	-2.0
10	WK_BUS_MR	-7.52165	2.0	0.0	0.0	-2.0
10	WK_MR	0	2.0	0.0	0.0	-2.0

10	PNR_CR	-1.99023	0.0	0.0	0.0	0.0
10	PNR_BUS	-0.69594	0.0	0.0	0.0	0.0
10	PNR_BUS_MR	-1.99023	0.0	0.0	0.0	0.0
10	PNR_MR	0	0.0	0.0	0.0	0.0
10	KNR_CR	-0.28971	0.0	0.0	0.0	0.0
10	KNR_BUS	-0.28971	0.0	0.0	0.0	0.0
10	KNR_BUS_MR	-0.28971	0.0	0.0	0.0	0.0
10	KNR_MR	0	0.0	0.0	0.0	0.0
11	AUTO	0	0.0	0.0	0.0	0.0
11	TRANSIT	5.35269	0.0	0.0	0.0	0.0
11	SOV	0	0.0	0.0	0.0	0.0
11	HOV	-1.87907	0.0	0.0	0.0	0.0
11	SR2	0	0.0	0.0	0.0	0.0
11	SR3	-1.523	0.0	0.0	0.0	0.0
11	WALK	0	0.0	0.0	0.0	0.0
11	PNR	-12.58348	0.0	0.0	0.0	0.0
11	KNR	-13.89833	0.0	0.0	0.0	0.0
11	WK_CR	-12.85594	2.0	0.0	0.0	-2.0
11	WK_BUS	-17.43408	2.0	0.0	0.0	-2.0
11	WK_BUS_MR	-16.91948	2.0	0.0	0.0	-2.0
11	WK_MR	0	2.0	0.0	0.0	-2.0
11	PNR_CR	-0.22059	0.0	0.0	0.0	0.0
11	PNR_BUS	-1.40483	0.0	0.0	0.0	0.0
11	PNR_BUS_MR	0.25582	0.0	0.0	0.0	0.0
11	PNR_MR	0	0.0	0.0	0.0	0.0
11	KNR_CR	-0.55664	0.0	0.0	0.0	0.0
11	KNR_BUS	-0.55664	0.0	0.0	0.0	0.0
11	KNR_BUS_MR	-0.48224	0.0	0.0	0.0	0.0
11	KNR_MR	0	0.0	0.0	0.0	0.0
12	AUTO	0	0.0	0.0	0.0	0.0
12	TRANSIT	4.23525	0.0	0.0	0.0	0.0
12	SOV	0	0.0	0.0	0.0	0.0
12	HOV	-2.19769	0.0	0.0	0.0	0.0
12	SR2	0	0.0	0.0	0.0	0.0
12	SR3	-1.01759	0.0	0.0	0.0	0.0
12	WALK	0	0.0	0.0	0.0	0.0
12	PNR	-9.35569	0.0	0.0	0.0	0.0
12	KNR	-11.70605	0.0	0.0	0.0	0.0
12	WK_CR	-16.14143	2.0	0.0	0.0	-2.0
12	WK_BUS	-20.83291	2.0	0.0	0.0	-2.0
12	WK_BUS_MR	-19.81743	2.0	0.0	0.0	-2.0
12	WK_MR	0	2.0	0.0	0.0	-2.0
12	PNR_CR	-9.10845	0.0	0.0	0.0	0.0
12	PNR_BUS	-6.88424	0.0	0.0	0.0	0.0
12	PNR_BUS_MR	-9.10845	0.0	0.0	0.0	0.0
12	PNR_MR	0	0.0	0.0	0.0	0.0
12	KNR_CR	-2.15853	0.0	0.0	0.0	0.0
12	KNR_BUS	-0.17748	0.0	0.0	0.0	0.0
12	KNR_BUS_MR	-4.78017	0.0	0.0	0.0	0.0
12	KNR_MR	0	0.0	0.0	0.0	0.0
13	AUTO	0	0.0	0.0	0.0	0.0
13	TRANSIT	2.53517	0.0	0.0	0.0	0.0
13	SOV	0	0.0	0.0	0.0	0.0
13	HOV	-1.6018	0.0	0.0	0.0	0.0
13	SR2	0	0.0	0.0	0.0	0.0
13	SR3	-1.32632	0.0	0.0	0.0	0.0
13	WALK	0	0.0	0.0	0.0	0.0
13	PNR	-4.78568	0.0	0.0	0.0	0.0
13	KNR	-6.42225	0.0	0.0	0.0	0.0
13	WK_CR	-7.49375	2.0	0.0	0.0	-2.0
13	WK_BUS	-8.22635	2.0	0.0	0.0	-2.0
13	WK_BUS_MR	-8.77999	2.0	0.0	0.0	-2.0
13	WK_MR	0	2.0	0.0	0.0	-2.0
13	PNR_CR	-1.37189	0.0	0.0	0.0	0.0
13	PNR_BUS	-6.56855	0.0	0.0	0.0	0.0
13	PNR_BUS_MR	-0.31971	0.0	0.0	0.0	0.0
13	PNR_MR	0	0.0	0.0	0.0	0.0
13	KNR_CR	-4.43232	0.0	0.0	0.0	0.0
13	KNR_BUS	-6.67781	0.0	0.0	0.0	0.0
13	KNR_BUS_MR	-1.36864	0.0	0.0	0.0	0.0
13	KNR_MR	0	0.0	0.0	0.0	0.0

14	AUTO	0	0.0	0.0	0.0	0.0
14	TRANSIT	1.17306	0.0	0.0	0.0	0.0
14	SOV	0	0.0	0.0	0.0	0.0
14	HOV	-1.83504	0.0	0.0	0.0	0.0
14	SR2	0	0.0	0.0	0.0	0.0
14	SR3	-1.32021	0.0	0.0	0.0	0.0
14	WALK	0	0.0	0.0	0.0	0.0
14	PNR	-1.31363	0.0	0.0	0.0	0.0
14	KNR	-3.50697	0.0	0.0	0.0	0.0
14	WK_CR	-8.30086	2.0	0.0	0.0	-2.0
14	WK_BUS	-4.27224	2.0	0.0	0.0	-2.0
14	WK_BUS_MR	-5.32487	2.0	0.0	0.0	-2.0
14	WK_MR	0	2.0	0.0	0.0	-2.0
14	PNR_CR	-5.72124	0.0	0.0	0.0	0.0
14	PNR_BUS	-1.17606	0.0	0.0	0.0	0.0
14	PNR_BUS_MR	-1.2301	0.0	0.0	0.0	0.0
14	PNR_MR	0	0.0	0.0	0.0	0.0
14	KNR_CR	-9.2145	0.0	0.0	0.0	0.0
14	KNR_BUS	-1.1464	0.0	0.0	0.0	0.0
14	KNR_BUS_MR	-1.11396	0.0	0.0	0.0	0.0
14	KNR_MR	0	0.0	0.0	0.0	0.0
15	AUTO	0	0.0	0.0	0.0	0.0
15	TRANSIT	2.06591	0.0	0.0	0.0	0.0
15	SOV	0	0.0	0.0	0.0	0.0
15	HOV	-2.00158	0.0	0.0	0.0	0.0
15	SR2	0	0.0	0.0	0.0	0.0
15	SR3	-1.65818	0.0	0.0	0.0	0.0
15	WALK	0	0.0	0.0	0.0	0.0
15	PNR	-4.78366	0.0	0.0	0.0	0.0
15	KNR	-5.89947	0.0	0.0	0.0	0.0
15	WK_CR	-9.7561	2.0	0.0	0.0	-2.0
15	WK_BUS	-6.22465	2.0	0.0	0.0	-2.0
15	WK_BUS_MR	-7.57288	2.0	0.0	0.0	-2.0
15	WK_MR	0	2.0	0.0	0.0	-2.0
15	PNR_CR	-3.84892	0.0	0.0	0.0	0.0
15	PNR_BUS	-1.70538	0.0	0.0	0.0	0.0
15	PNR_BUS_MR	-1.4554	0.0	0.0	0.0	0.0
15	PNR_MR	0	0.0	0.0	0.0	0.0
15	KNR_CR	-6.67092	0.0	0.0	0.0	0.0
15	KNR_BUS	-2.35052	0.0	0.0	0.0	0.0
15	KNR_BUS_MR	-2.29305	0.0	0.0	0.0	0.0
15	KNR_MR	0	0.0	0.0	0.0	0.0
16	AUTO	0	0.0	0.0	0.0	0.0
16	TRANSIT	0.00011	0.0	0.0	0.0	0.0
16	SOV	0	0.0	0.0	0.0	0.0
16	HOV	-2.24901	0.0	0.0	0.0	0.0
16	SR2	0	0.0	0.0	0.0	0.0
16	SR3	-1.45489	0.0	0.0	0.0	0.0
16	WALK	0	0.0	0.0	0.0	0.0
16	PNR	-3.649	0.0	0.0	0.0	0.0
16	KNR	-3.9994	0.0	0.0	0.0	0.0
16	WK_CR	-5.28939	2.0	0.0	0.0	-2.0
16	WK_BUS	-1.50798	2.0	0.0	0.0	-2.0
16	WK_BUS_MR	-2.94853	2.0	0.0	0.0	-2.0
16	WK_MR	0	2.0	0.0	0.0	-2.0
16	PNR_CR	-1.79539	0.0	0.0	0.0	0.0
16	PNR_BUS	-0.86965	0.0	0.0	0.0	0.0
16	PNR_BUS_MR	-0.59093	0.0	0.0	0.0	0.0
16	PNR_MR	0	0.0	0.0	0.0	0.0
16	KNR_CR	-4.26674	0.0	0.0	0.0	0.0
16	KNR_BUS	-1.39508	0.0	0.0	0.0	0.0
16	KNR_BUS_MR	-1.66796	0.0	0.0	0.0	0.0
16	KNR_MR	0	0.0	0.0	0.0	0.0
17	AUTO	0	0.0	0.0	0.0	0.0
17	TRANSIT	3.51488	0.0	0.0	0.0	0.0
17	SOV	0	0.0	0.0	0.0	0.0
17	HOV	-2.01828	0.0	0.0	0.0	0.0
17	SR2	0	0.0	0.0	0.0	0.0
17	SR3	-1.57923	0.0	0.0	0.0	0.0
17	WALK	0	0.0	0.0	0.0	0.0
17	PNR	-7.86894	0.0	0.0	0.0	0.0

17	KNR	-8.86193	0.0	0.0	0.0	0.0
17	WK_CR	-17.57389	2.0	0.0	0.0	-2.0
17	WK_BUS	-13.92998	2.0	0.0	0.0	-2.0
17	WK_BUS_MR	-12.83641	2.0	0.0	0.0	-2.0
17	WK_MR	0	2.0	0.0	0.0	-2.0
17	PNR_CR	-3.30493	0.0	0.0	0.0	0.0
17	PNR_BUS	-0.70056	0.0	0.0	0.0	0.0
17	PNR_BUS_MR	0.23622	0.0	0.0	0.0	0.0
17	PNR_MR	0	0.0	0.0	0.0	0.0
17	KNR_CR	-6.57274	0.0	0.0	0.0	0.0
17	KNR_BUS	-2.98946	0.0	0.0	0.0	0.0
17	KNR_BUS_MR	-1.94384	0.0	0.0	0.0	0.0
17	KNR_MR	0	0.0	0.0	0.0	0.0
18	AUTO	0	0.0	0.0	0.0	0.0
18	TRANSIT	2.36783	0.0	0.0	0.0	0.0
18	SOV	0	0.0	0.0	0.0	0.0
18	HOV	-2.10544	0.0	0.0	0.0	0.0
18	SR2	0	0.0	0.0	0.0	0.0
18	SR3	-1.74176	0.0	0.0	0.0	0.0
18	WALK	0	0.0	0.0	0.0	0.0
18	PNR	-4.98412	0.0	0.0	0.0	0.0
18	KNR	-5.9193	0.0	0.0	0.0	0.0
18	WK_CR	-11.40125	2.0	0.0	0.0	-2.0
18	WK_BUS	-6.94533	2.0	0.0	0.0	-2.0
18	WK_BUS_MR	-7.86833	2.0	0.0	0.0	-2.0
18	WK_MR	0	2.0	0.0	0.0	-2.0
18	PNR_CR	-1.01417	0.0	0.0	0.0	0.0
18	PNR_BUS	1.09391	0.0	0.0	0.0	0.0
18	PNR_BUS_MR	-0.1955	0.0	0.0	0.0	0.0
18	PNR_MR	0	0.0	0.0	0.0	0.0
18	KNR_CR	-4.77777	0.0	0.0	0.0	0.0
18	KNR_BUS	-2.21824	0.0	0.0	0.0	0.0
18	KNR_BUS_MR	-2.43815	0.0	0.0	0.0	0.0
18	KNR_MR	0	0.0	0.0	0.0	0.0
19	AUTO	0	0.0	0.0	0.0	0.0
19	TRANSIT	2.76083	0.0	0.0	0.0	0.0
19	SOV	0	0.0	0.0	0.0	0.0
19	HOV	-2.49049	0.0	0.0	0.0	0.0
19	SR2	0	0.0	0.0	0.0	0.0
19	SR3	-2.56594	0.0	0.0	0.0	0.0
19	WALK	0	0.0	0.0	0.0	0.0
19	PNR	-6.71782	0.0	0.0	0.0	0.0
19	KNR	-7.34757	0.0	0.0	0.0	0.0
19	WK_CR	-15.09131	2.0	0.0	0.0	-2.0
19	WK_BUS	-11.42943	2.0	0.0	0.0	-2.0
19	WK_BUS_MR	-10.74147	2.0	0.0	0.0	-2.0
19	WK_MR	0	2.0	0.0	0.0	-2.0
19	PNR_CR	-1.95993	0.0	0.0	0.0	0.0
19	PNR_BUS	-0.65682	0.0	0.0	0.0	0.0
19	PNR_BUS_MR	-0.49789	0.0	0.0	0.0	0.0
19	PNR_MR	0	0.0	0.0	0.0	0.0
19	KNR_CR	-5.90216	0.0	0.0	0.0	0.0
19	KNR_BUS	-3.43406	0.0	0.0	0.0	0.0
19	KNR_BUS_MR	-3.07321	0.0	0.0	0.0	0.0
19	KNR_MR	0	0.0	0.0	0.0	0.0
20	AUTO	0	0.0	0.0	0.0	0.0
20	TRANSIT	1.65769	0.0	0.0	0.0	0.0
20	SOV	0	0.0	0.0	0.0	0.0
20	HOV	-2.39718	0.0	0.0	0.0	0.0
20	SR2	0	0.0	0.0	0.0	0.0
20	SR3	-2.07527	0.0	0.0	0.0	0.0
20	WALK	0	0.0	0.0	0.0	0.0
20	PNR	-7.72107	0.0	0.0	0.0	0.0
20	KNR	-6.76308	0.0	0.0	0.0	0.0
20	WK_CR	-16.37276	2.0	0.0	0.0	-2.0
20	WK_BUS	-9.94345	2.0	0.0	0.0	-2.0
20	WK_BUS_MR	-11.56452	2.0	0.0	0.0	-2.0
20	WK_MR	0	2.0	0.0	0.0	-2.0
20	PNR_CR	-2.92185	0.0	0.0	0.0	0.0
20	PNR_BUS	-6.2218	0.0	0.0	0.0	0.0
20	PNR_BUS_MR	-3.80359	0.0	0.0	0.0	0.0

```
      20   PNR_MR                  0         0.0         0.0         0.0         0.0
      20   KNR_CR           -9.14702         0.0         0.0         0.0         0.0
      20   KNR_BUS          -6.00517         0.0         0.0         0.0         0.0
      20   KNR_BUS_MR        -7.6089         0.0         0.0         0.0         0.0
      20   KNR_MR                  0         0.0         0.0         0.0         0.0
```

The heart of the mode choice model is in the ModeChoice script file. This file contains the syntax for a user-defined program that uses various input data to calculate utility functions for each mode and transit-access market segments. This script could be unique to each trip purpose or common to all trip purposes. In this case (Figure 10-4), the script is configured to satisfy the requirements of all trip purposes. It is compiled by the ModeChoice software and executed for each origin-destination cell.

Figure 10-4: Mode Choice Script File

```
//--- ASSUMED MATRIX ORGANIZATION
//--- TRIP_FILE = TRIP TABLE (SEPARATE FOR EACH PURPOSE)
//--- HBWI1Psn INCOME 1 (HOME-BASED)/ALL NHB TRIPS
//--- HBWI2Psn INCOME 2 (HOME-BASED)
//--- HBWI3Psn INCOME 3 (HOME-BASED)
//--- HBWI4Psn INCOME 4 (HOME-BASED)

//--- SKIM_FILE_2 = HIGHWAY SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
//--- SovTime  1 SOV   TIME (MIN)
//--- SovDst10 2 SOV   DIST (0.1 MILES)
//--- SovTollt 3 SOV   TOLL (2007 CENTS)
//--- Hv2Time  4 HOV2  TIME (MIN)
//--- Hv2Dst10 5 HOV2  DIST (0.1 MILES)
//--- Hv2Toll  6 HOV2  TOLL (2007 CENTS)
//--- Hv3Time  7 HOV3+ TIME (MIN)
//--- Hv3Dst10 8 HOV3+ DIST (0.1 MILES)
//--- Hv3Toll  9 HOV3+ TOLL (2007 CENTS)

//--- SKIM_FILE_3 = COM. RAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
//--- SKIM_FILE_4 = BUS SKIMS       (SEPARATE FOR PEAK AND OFFPEAK)
//--- SKIM_FILE_5 = METRORAIL SKIMS (SEPARATE FOR PEAK AND OFFPEAK)
//--- SKIM_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)

//--- ZONE_FILE = ZDATA
//--- HBW_PARK_COST      1 HBW PARK COST (2007 CENTS)
//--- HBS_PARK_COST      2 HBS PARK COST (2007 CENTS)
//--- HBO_PARK_COST      3 HBO PARK COST (2007 CENTS)
//--- NHB_PARK_COST      4 NHB PARK COST (2007 CENTS)
//--- TERM_TIME_HOME     5 TERMINAL TIME (HOME BASED) (MINUTES)
//--- TERM_TIME_OTHER    6 TERMINAL TIME (NON HOME BASED) (MINUTES)
//--- SHORT_WK_METRO     7 ARC VIEW SHORT WALK PERCENT TO METRO
//--- LONG_WK_METRO      8 ARC VIEW LONG WALK PERCENT TO METRO
//--- SHORT_WK_ALL_PK    9 ARC VIEW SHORT WALK PERCENT TO ALL AM PK TRANSIT
//--- LONG_WK_ALL_PK    10 ARC VIEW LONG WALK PERCENT TO ALL AM PK TRANSIT
```

```
//--- SHORT_WK_ALL_OP   11 ARC VIEW SHORT WALK PERCENT TO ALL OP TRANSIT
//--- LONG_WK_ALL_OP    12 ARC VIEW LONG WALK PERCENT TO ALL OP TRANSIT
//--- AREA_TYPE         13 AREA TYPE
//---                       1=DC CORE
//---                       2=VA CORE
//---                       3=DC URBAN
//---                       4=MD URBAN
//---                       5=VA URBAN
//---                       6=MD OTHER
//---                       7=VA OTHER

//---- MWCOG Mode Choice Script -- 4/20/13 ----

INTEGER HBW, HBS, HBO, NHB, PEAK, ENDDEF
REAL TERM_TIME, PARK_COST, AUTO_OP, OCC3, TIME1, TIME2, TIME3,
  SHORT_WK_P, SHORT_WK_A, LONG_WK_P, LONG_WK_A, WK_P, DR_P, WK_A, DR_A,
  WK_MR_P, WK_MR_A, WK_ALL_P, WK_ALL_A, WK_ACCESS, WK_METRO,
ENDDEF

//---- purpose/period codes ----

HBW = 1
HBS = 2
HBO = 3
NHB = 4
PEAK = 1

//---- auto operating cost ---

AUTO_OP = 10

//---- auto occupancy for 3+ ----

OCC3 = 3.5

//---- terminal time ----

IF (TRIP.PURPOSE != NHB) THEN
  TERM_TIME = ORG.TERM_TIME_HOME + DES.TERM_TIME_HOME
ELSE
  TERM_TIME = ORG.TERM_TIME_OTHER + DES.TERM_TIME_OTHER
ENDIF

//---- parking cost ----

IF (TRIP.PURPOSE == HBW) THEN
  PARK_COST = DES.HBW_PARK_COST / 2.0
ELSE IF (TRIP.PURPOSE == HBS) THEN
  PARK_COST = DES.HBS_PARK_COST / 2.0
ELSE IF (TRIP.PURPOSE == HBO) THEN
  PARK_COST = DES.HBO_PARK_COST / 2.0
ELSE
  PARK_COST = DES.NHB_PARK_COST
ENDIF ENDIF ENDIF

//---- intrazonal/auto times ----

IF (ORG.ZONE == DES.ZONE) THEN
  TIME1 = 1.0
  TIME2 = 1.0
  TIME3 = 1.0
ELSE
  TIME1 = SKIM2.SovTime
  TIME2 = SKIM2.Hv2Time
  TIME3 = SKIM2.Hv3Time
ENDIF

//---- drive alone ----

IF (TIME1 > 0) THEN
  SOV.TIME = TIME1
```

```
  SOV.COST = AUTO_OP * SKIM2.SovDst10 / 10 + SKIM2.SovToll + PARK_COST
  SOV.TERM = TERM_TIME
  IF (DES.AREA_TYPE == 1) THEN
    SOV.CBD = -1.0
  ELSE IF (DES.AREA_TYPE == 2) THEN
    SOV.CBD = -0.25
  ENDIF ENDIF
ENDIF

//---- shared ride 2 ----

IF (TIME2 > 0) THEN
  SR2.TIME = TIME2
  SR2.COST = (AUTO_OP * SKIM2.Hv2Dst10 / 10 + SKIM2.Hv2Toll + PARK_COST) / 2.0
  SR2.TERM = TERM_TIME
ENDIF

//---- shared ride 3+ ----

IF (TIME3 > 0) THEN
  SR3.TIME = TIME3
  SR3.COST = (AUTO_OP * SKIM2.Hv3Dst10 / 10 + SKIM2.Hv3Toll + PARK_COST) / OCC3
  SR3.TERM = TERM_TIME
ENDIF

//---- intrazonal return ----

IF (ORG.ZONE == DES.ZONE) THEN
  TRIP.ACCESS1 = 0
  TRIP.ACCESS2 = 0
  TRIP.ACCESS3 = 0
  TRIP.ACCESS4 = 0
  TRIP.ACCESS5 = 0
  TRIP.ACCESS6 = 1.0
  RETURN (1)
ENDIF

//---- transit access markets ----

SHORT_WK_P = ORG.SHORT_WK_METRO
SHORT_WK_A = DES.SHORT_WK_METRO
LONG_WK_P = ORG.LONG_WK_METRO
LONG_WK_A = DES.LONG_WK_METRO

WK_MR_P = MIN (MAX (((SHORT_WK_P + 0.25 * (LONG_WK_P - SHORT_WK_P)) / 100.0), 0), 1)
WK_MR_A = MIN (MAX (((SHORT_WK_A + 0.25 * (LONG_WK_A - SHORT_WK_A)) / 100.0), 0), 1)
WK_METRO = WK_MR_P * WK_MR_A

IF (TRIP.PERIOD == PEAK) THEN
  SHORT_WK_P = ORG.SHORT_WK_ALL_PK
  SHORT_WK_A = DES.SHORT_WK_ALL_PK
  LONG_WK_P = ORG.LONG_WK_ALL_PK
  LONG_WK_A = DES.LONG_WK_ALL_PK
ELSE
  SHORT_WK_P = ORG.SHORT_WK_ALL_OP
  SHORT_WK_A = DES.SHORT_WK_ALL_OP
  LONG_WK_P = ORG.LONG_WK_ALL_OP
  LONG_WK_A = DES.LONG_WK_ALL_OP
ENDIF

WK_ALL_P = MIN (MAX (((SHORT_WK_P + 0.25 * (LONG_WK_P - SHORT_WK_P)) / 100.0), WK_MR_P), 1)
WK_ALL_A = MIN (MAX (((SHORT_WK_A + 0.25 * (LONG_WK_A - SHORT_WK_A)) / 100.0), WK_MR_A), 1)
WK_ACCESS = WK_ALL_P * WK_ALL_A

TRIP.ACCESS1 = WK_METRO
TRIP.ACCESS2 = (WK_ALL_P - WK_MR_P) * WK_MR_A
TRIP.ACCESS3 = WK_ALL_P * (WK_ALL_A - WK_MR_A)
TRIP.ACCESS4 = (1 - WK_ALL_P) * WK_MR_A
TRIP.ACCESS5 = (1 - WK_ALL_P) * (WK_ALL_A - WK_MR_A)
TRIP.ACCESS6 = 1 - TRIP.ACCESS1 - TRIP.ACCESS2 - TRIP.ACCESS3 - TRIP.ACCESS4 - TRIP.ACCESS5
```

```
//---- walk access modes ----

IF (WK_ACCESS > 0) THEN

  //---- walk to commuter rail ----

  IF (SKIM3.5 > 0) THEN
    WK_CR.TIME = SKIM3.5 / 100.0
    WK_CR.WALK = (SKIM3.1 + SKIM3.2) / 100.0
    WK_CR.WAIT = (SKIM3.3 + SKIM3.4) / 100.0
    WK_CR.COST = SKIM3.13
    WK_CR.XFER = SKIM3.12
    WK_CR.TPEN = SKIM3.14 / 100.0
  ENDIF

  //---- walk to bus ----

  IF (SKIM4.5 > 0) THEN
    WK_BUS.TIME = SKIM4.5 / 100.0
    WK_BUS.WALK = (SKIM4.1 + SKIM4.2) / 100.0
    WK_BUS.WAIT = (SKIM4.3 + SKIM4.4) / 100.0
    WK_BUS.COST = SKIM4.13
    WK_BUS.XFER = SKIM4.12
    WK_BUS.TPEN = SKIM4.14 / 100.0
  ENDIF

  //---- walk to bus/Metrorail ----

  IF (SKIM6.5 > 0) THEN
    WK_BUS_MR.TIME = SKIM6.5 / 100.0
    WK_BUS_MR.WALK = (SKIM6.1 + SKIM6.2) / 100.0
    WK_BUS_MR.WAIT = (SKIM6.3 + SKIM6.4) / 100.0
    WK_BUS_MR.COST = SKIM6.13
    WK_BUS_MR.XFER = SKIM6.12
    WK_BUS_MR.TPEN = SKIM6.14 / 100.0
  ENDIF

  //---- walk to Metrorail ----

  IF (SKIM5.5 > 0 && WK_METRO > 0) THEN
    WK_MR.TIME = SKIM5.5 / 100.0
    WK_MR.WALK = (SKIM5.1 + SKIM5.2) / 100.0
    WK_MR.WAIT = (SKIM5.3 + SKIM5.4) / 100.0
    WK_MR.COST = SKIM5.13
    WK_MR.XFER = SKIM5.12
    WK_MR.TPEN = SKIM5.14 / 100.0
  ENDIF
ENDIF

//---- drive access ----

IF (WK_ALL_A > 0) THEN

  //---- park-ride to commuter rail ----

  IF (SKIM3.19 > 0) THEN
    PNR_CR.TIME = SKIM3.19 / 100.0
    PNR_CR.AUTO = SKIM3.29 / 100.0
    PNR_CR.TERM = SKIM3.32 / 100.0
    PNR_CR.WALK = (SKIM3.15 + SKIM3.16) / 100.0
    PNR_CR.WAIT = (SKIM3.17 + SKIM3.18) / 100.0
    PNR_CR.COST = AUTO_OP * SKIM3.30 / 100.0 + SKIM3.27 + SKIM3.31
    PNR_CR.XFER = SKIM3.26
    PNR_CR.TPEN = SKIM3.28 / 100.0
  ENDIF

  //---- park-ride to bus ----

  IF (SKIM4.19 > 0) THEN
    PNR_BUS.TIME = SKIM4.19 / 100.0
    PNR_BUS.AUTO = SKIM4.29 / 100.0
```

```
    PNR_BUS.TERM = SKIM4.32 / 100.0
    PNR_BUS.WALK = (SKIM4.15 + SKIM4.16) / 100.0
    PNR_BUS.WAIT = (SKIM4.17 + SKIM4.18) / 100.0
    PNR_BUS.COST = AUTO_OP * SKIM4.30 / 100.0 + SKIM4.27 + SKIM4.31
    PNR_BUS.XFER = SKIM4.26
    PNR_BUS.TPEN = SKIM4.28 / 100.0
  ENDIF

  //---- park-ride to bus/Metrorail ----

  IF (SKIM6.19 > 0) THEN
    PNR_BUS_MR.TIME = SKIM6.19 / 100.0
    PNR_BUS_MR.AUTO = SKIM6.29 / 100.0
    PNR_BUS_MR.TERM = SKIM6.32 / 100.0
    PNR_BUS_MR.WALK = (SKIM6.15 + SKIM6.16) / 100.0
    PNR_BUS_MR.WAIT = (SKIM6.17 + SKIM6.18) / 100.0
    PNR_BUS_MR.COST = AUTO_OP * SKIM6.30 / 100.0 + SKIM6.27 + SKIM6.31
    PNR_BUS_MR.XFER = SKIM6.26
    PNR_BUS_MR.TPEN = SKIM6.28 / 100.0
  ENDIF

  //---- park-ride to Metrorail ----

  IF (SKIM5.19 > 0 && WK_MR_A > 0) THEN
    PNR_MR.TIME = SKIM5.19 / 100.0
    PNR_MR.AUTO = SKIM5.29 / 100.0
    PNR_MR.TERM = SKIM5.32 / 100.0
    PNR_MR.WALK = (SKIM5.15 + SKIM5.16) / 100.0
    PNR_MR.WAIT = (SKIM5.17 + SKIM5.18) / 100.0
    PNR_MR.COST = AUTO_OP * SKIM5.30 / 100.0 + SKIM5.27 + SKIM5.31
    PNR_MR.XFER = SKIM5.26
    PNR_MR.TPEN = SKIM5.28 / 100.0
  ENDIF

  //---- kiss-ride to commuter rail ----

  IF (SKIM3.19 > 0) THEN
    KNR_CR.TIME = SKIM3.19 / 100.0
    KNR_CR.AUTO = SKIM3.29 / 100.0
    KNR_CR.WALK = (SKIM3.15 + SKIM3.16) / 100.0
    KNR_CR.WAIT = (SKIM3.17 + SKIM3.18) / 100.0
    KNR_CR.COST = AUTO_OP * SKIM3.30 / 100.0 + SKIM3.27
    KNR_CR.XFER = SKIM3.26
    KNR_CR.TPEN = SKIM3.28 / 100.0
 ENDIF

 //---- kiss-ride to bus ----

  IF (SKIM4.37 > 0) THEN
    KNR_BUS.TIME = SKIM4.37 / 100.0
    KNR_BUS.AUTO = SKIM4.47 / 100.0
    KNR_BUS.WALK = (SKIM4.33 + SKIM4.34) / 100.0
    KNR_BUS.WAIT = (SKIM4.35 + SKIM4.36) / 100.0
    KNR_BUS.COST = AUTO_OP * SKIM4.48 / 100.0 + SKIM4.45
    KNR_BUS.XFER = SKIM4.44
    KNR_BUS.TPEN = SKIM4.46 / 100.0
  ENDIF

  //---- kiss-ride to bus/Metrorail ----

  IF (SKIM6.37 > 0) THEN
    KNR_BUS_MR.TIME = SKIM6.37 / 100.0
    KNR_BUS_MR.AUTO = SKIM6.47 / 100.0
    KNR_BUS_MR.WALK = (SKIM6.33 + SKIM6.34) / 100.0
    KNR_BUS_MR.WAIT = (SKIM6.35 + SKIM6.36) / 100.0
    KNR_BUS_MR.COST = AUTO_OP * SKIM6.48 / 100.0 + SKIM6.45
    KNR_BUS_MR.XFER = SKIM6.44
    KNR_BUS_MR.TPEN = SKIM6.46 / 100.0
  ENDIF

  //---- kiss-ride to Metrorail ----
```

```
  IF (SKIM5.37 > 0 && WK_MR_A > 0) THEN
    KNR_MR.TIME = SKIM5.37 / 100.0
    KNR_MR.AUTO = SKIM5.47 / 100.0
    KNR_MR.WALK = (SKIM5.33 + SKIM5.34) / 100.0
    KNR_MR.WAIT = (SKIM5.35 + SKIM5.36) / 100.0
    KNR_MR.COST = AUTO_OP * SKIM5.48 / 100.0 + SKIM5.45
    KNR_MR.XFER = SKIM5.44
    KNR_MR.TPEN = SKIM5.46 / 100.0
  ENDIF
ENDIF

RETURN (1)
```

10.4 ModeChoice Application Test

The prototype ModeChoice application was applied to the input files from the MWCOG Version 2.3 travel model for 2010. The results of the AEMS mode choice model were compared to the ModeChoice output to confirm the conversion process and assess the overall impact of the software change on the model results. Table 10-1, Table 10-2, and Table 10-3 highlight the differences in the model results. As you can see in Table 10-3, the largest difference is 0.8 trips out of 801,825 for high income single occupancy auto trips. These small differences are caused by the differences in the log calculations and the number of decimal points printed in the output report. In addition, the ModeChoice application completed in less than 7.5 minutes versus 16.5 minutes for the AEMS application.

Table 10-1: HBW Mode Choice Results Generated by the ModeChoice Software

Trips by Mode	HBWI1Psn	HBWI2Psn	HBWI3Psn	HBWI4Psn	Total	Percent
AUTO	438,626	902,076	740,144	942,924	3,023,770	79.28%
TRANSIT	202,410	221,190	178,903	187,738	790,241	20.72%
SOV	365,712	765,623	632,549	801,826	2,565,710	67.27%
HOV	72,915	136,453	107,595	141,098	458,060	12.01%
SR2	52,935	100,735	79,710	104,376	337,756	8.86%
SR3	19,980	35,718	27,884	36,722	120,304	3.15%
WALK	194,044	159,757	115,854	51,381	521,035	13.66%
PNR	5,959	46,466	49,606	105,496	207,527	5.44%
KNR	2,408	14,967	13,444	30,861	61,679	1.62%
WK_CR	899	605	540	19	2,064	0.05%
WK_BUS	95,674	56,635	37,334	11,577	201,220	5.28%
WK_BUS_MR	53,972	41,748	34,778	7,549	138,047	3.62%
WK_MR	43,498	60,768	43,202	32,236	179,705	4.71%
PNR_CR	532	3,740	4,160	7,054	15,487	0.41%
PNR_BUS	914	4,118	3,696	7,902	16,630	0.44%
PNR_BUS_MR	729	6,226	6,879	16,477	30,310	0.79%
PNR_MR	3,784	32,381	34,871	74,063	145,100	3.80%
KNR_CR	60	353	384	686	1,483	0.04%
KNR_BUS	309	1,212	992	2,193	4,705	0.12%
KNR_BUS_MR	291	2,018	2,009	4,938	9,256	0.24%
KNR_MR	1,748	11,385	10,058	23,044	46,235	1.21%
Total	641,036	1,123,266	919,047	1,130,662	3,814,011	100.00%
Percent	16.81%	29.45%	24.10%	29.64%	100.00%	

Table 10-2: HBW Mode Choice Result Generated by the AEMS Software

Trips by Mode	HBWI1Psn	HBWI2Psn	HBWI3Psn	HBWI4Psn	Total	Percent
AUTO	438,626	902,076	740,143	942,924	3,023,769	79.28%
TRANSIT	202,410	221,190	178,904	187,739	790,242	20.72%
SOV	365,711	765,622	632,548	801,825	2,565,707	67.27%
HOV	72,915	136,454	107,595	141,098	458,062	12.01%
SR2	52,935	100,735	79,711	104,376	337,756	8.86%
SR3	19,980	35,719	27,884	36,723	120,306	3.15%
WALK	194,044	159,757	115,854	51,381	521,036	13.66%
PNR	5,958	46,466	49,606	105,496	207,527	5.44%
KNR	2,408	14,967	13,444	30,861	61,679	1.62%
WK_CR	899	605	540	19	2,063	0.05%
WK_BUS	95,674	56,635	37,334	11,577	201,220	5.28%
WK_BUS_MR	53,972	41,748	34,778	7,549	138,047	3.62%
WK_MR	43,498	60,768	43,202	32,236	179,705	4.71%
PNR_CR	532	3,740	4,160	7,054	15,487	0.41%
PNR_BUS	914	4,118	3,696	7,902	16,630	0.44%
PNR_BUS_MR	729	6,226	6,879	16,477	30,310	0.79%
PNR_MR	3,784	32,381	34,871	74,063	145,100	3.80%
KNR_CR	60	353	384	686	1,483	0.04%
KNR_BUS	309	1,211	992	2,193	4,705	0.12%
KNR_BUS_MR	291	2,018	2,009	4,938	9,256	0.24%
KNR_MR	1,748	11,385	10,058	23,044	46,235	1.21%
Total	641,036	1,123,266	919,047	1,130,662	3,814,010	100.00%
Percent	16.81%	29.45%	24.10%	29.64%	100.00%	

Table 10-3: Difference in Trips between ModeChoice and AEMS

Trips by Mode	HBWI1Psn	HBWI2Psn	HBWI3Psn	HBWI4Psn	Total	Percent
AUTO	0.13	0.19	0.16	0.28	0.60	0.00%
TRANSIT	-0.17	-0.20	-0.16	-0.26	-0.79	0.00%
SOV	0.41	0.69	0.56	0.85	2.60	0.00%
HOV	-0.27	-0.53	-0.42	-0.56	-1.71	0.00%
SR2	-0.09	-0.18	-0.17	-0.24	-0.58	0.00%
SR3	-0.19	-0.35	-0.24	-0.31	-1.10	0.00%
WALK	-0.19	-0.12	-0.08	-0.10	-0.40	0.00%
PNR	0.02	-0.06	-0.10	-0.12	-0.25	0.00%
KNR	0.00	-0.02	0.02	-0.03	-0.04	0.00%
WK_CR	-0.05	-0.02	0.04	0.05	0.01	0.00%
WK_BUS	-0.10	-0.10	0.01	-0.05	-0.23	0.00%
WK_BUS_MR	-0.06	-0.05	-0.08	0.02	-0.17	0.00%
WK_MR	0.02	0.06	0.06	-0.01	0.01	0.00%
PNR_CR	-0.02	0.01	0.02	-0.03	-0.03	0.00%
PNR_BUS	-0.04	-0.05	-0.01	-0.03	-0.02	0.00%
PNR_BUS_MR	0.00	0.00	-0.03	-0.04	-0.17	0.00%
PNR_MR	-0.02	-0.02	0.02	-0.02	-0.05	0.00%
KNR_CR	-0.03	0.05	-0.02	-0.03	-0.03	0.00%
KNR_BUS	0.00	0.01	-0.02	-0.04	-0.05	0.00%
KNR_BUS_MR	-0.03	-0.03	-0.02	0.03	-0.04	0.00%
KNR_MR	-0.03	0.04	-0.03	0.00	-0.01	0.00%
Total	0.11	0.10	0.00	0.08	0.35	0.00%
Percent	0.00%	0.00%	0.00%	0.00%	0.00%	

11 Summary of Recommendations

Throughout this document, AECOM identified places where MWCOG may wish to consider focusing additional attention to improve the regional modeling process. This chapter consolidates many of these suggestions for easier reference.

1. AECOM demonstrated that an HOV choice model can be calibrated to achieve desired HOV volumes on the HOV facilities. One of the benefits of such a change is the ability to eliminate the "two-step assignment," where HOV3+ trips are assigned separately from other user classes during the AM and PM peak period, which should help reduce model run times. However, a careful review of the HOV count data and additional calibration work must be conducted before integrating an HOV choice model into the MWCOG modeling process.
2. AECOM recommends integrating an HOV choice model and multi-class assignment procedure into the current mode choice and assignment setups to reduce processing time and improve behavioral sensitivity of the model.
3. The primary objective of the HOT lane modeling process (preventing degradation of HOV speeds in a full multi-class assignment) was achieved along with other objectives such as incorporation of toll-setting and toll-choice in the standard highway assignment process, while moderately reducing the overall model runtime. AECOM recommends pursuing this concept further.
4. The availability of better-quality observed speed data, such as provided by INRIX, has served only to further highlight the fact that static models do not generate realistic speeds. Since the primary purpose of estimating speeds in a static assignment model is to produce reasonable traffic volumes, it is inadvisable to be overly ambitious in calibrating volume-delay functions that reproduce observed speeds at the expense of reproducing observed traffic counts.
5. The fact that the MWCOG model generates consistently lower peak-period speeds on freeways, compared to the INRIX data, suggests it may be desirable to adjust the volume-delay function used for freeways to generate more realistic speeds and travel times. Additional detailed traffic counts on freeway facilities where INRIX speed data are available would need to be collected to properly calibrate the volume-delay function.
6. The reconfiguration of transit access links around Metrorail and commuter rails stations provides the necessary connections to enable the PT Generate statement and path building procedures to construct transit paths with various access mode restrictions and line-haul mode options. Initial attempts to implement park-n-ride access to bus routes suggests that a similar process may be needed for bus park-n-ride lots if the PT Generate statement cannot be forced to use connection links between park-n-ride nodes and bus stops. A variety of potential solutions to this concern should be investigated before the PT access procedures are finalized.
7. The PT fare calculation options cannot replicate the current fare calculation methods within the MWCOG model, but do offer a number of features that could be useful in designing a new fare estimation process. These options require further analysis and implementation testing especially if fares are included in selecting the path.

8. Migrating the MWCOG mode choice model from the AEMS software to the ModeChoice program will reduce processing time, increase flexibility, simplify calibration efforts, and improve software maintenance.

12 Appendix

The appendix includes two types of software documentation for the TRANSIMS-related tools proposed for use within the MWCOG modeling process. The first set of documentation is Quick References that provide a summary of the program's capabilities, a one-line summary of each control key's attributes, and the names of the report options. The second set of documentation is User's Guides that discussion each control key in more detail, describe processing methods and options, and provides examples of the output files and reports.

Execution Service Quick Reference

Version 5.x

Syntax:

Program [-flag] [control_file]

Purpose:

1. Manages the command line interface;
2. Processes the command line flags;
3. Creates the program report file;
4. Reads and processes the configuration file;
5. Reads and processes the control file and control keys;
6. Manages the report and print interface;
7. Manages the message, problem and error services;
8. Generates help messages and the XML interface;
9. Manages partition processing data;
10. Controls multi-threading and MPI options;
11. Processes a global set of control keys; and
12. Executes the program.

Command Line Flags:

-H[elp]	show program syntax and control keys
-C[ontrol]	create/update a default control file
-K[eyCheck]	list unrecognized control file keys
-P[ause]	pause before exiting
-N[oPause]	never pause before exiting
-Q[uiet]	execute without screen messages
-D[etail]	execute with detailed status messages
-X[ML]	write an XML file with control keys

-R[eport]	write control keys and report names

Execution Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
TITLE	Optional	Text		
REPORT_DIRECTORY	Optional	Directory		
REPORT_FILE	Optional	Output File		[report_directory]filename[_partition][.prn]
REPORT_FLAG	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
PAGE_LENGTH	Optional	Integer	65	>= 0 (0 = no page breaks)
PROJECT_DIRECTORY	Optional	Directory		
DEFAULT_FILE_FORMAT	Optional	Text	TAB_DELIMITED	Note #1
TIME_OF_DAY_FORMAT	Optional	Text	DAY_TIME	Note #2
MODEL_START_TIME	Optional	Time	0:00	>= 0 [seconds], 0.0 [hours], 0:00
MODEL_END_TIME	Optional	Time	24:00	> [model_start_time]",
MODEL_TIME_INCREMENT	Optional	Time	15 minutes	0, 2..240 minutes
UNITS_OF_MEASURE	Optional	Text	METRIC	METRIC, ENGLISH
RANDOM_NUMBER_SEED	Optional	Integer	0	
MAX_WARNING_MESSAGES	Optional	Integer	100,000	>= 0
MAX_WARNING_EXIT_FLAG	Optional	Boolean	TRUE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
MAX_PROBLEM_COUNT	Optional	Integer	0	>= 0
NUMBER_OF_THREADS	Optional	Integer	1	1..64

Notes:

1	TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3
2	SECONDS, MINUTES, HOURS, HOUR_CLOCK, DAY_TIME, TIME_CODE

Selection Service Quick Reference

Version 5.x

Purpose:

13. Select a range of values for a specified set of file fields;
14. Path-based selection options for links, nodes, stops, and routes with compound or serial selection criteria;
15. Point-in-polygon selection using an ArcGIS shape file;
16. Selection percentages and maximum percent selected for random selection methods;
17. Minimum, maximum, and percent difference values to define significant differences for comparison selections; and
18. Record deletion keys and a deletion file to select everything except a specified list of values.

Selection Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
SELECT_HOUSEHOLDS	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_MODES	Optional	List	ALL	e.g., ALL or 1, 12..14 or WALK, HOV2..HOV4
SELECT_PURPOSES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_TIME_OF_DAY	Optional	List	0:00	0:00..24:00
SELECT_START_TIMES	Optional	List	ALL	e.g., ALL, 0..97200 seconds, 0.0..27.0 hours, 0:00..27:00
SELECT_END_TIMES	Optional	List	ALL	e.g., ALL, 0..97200 seconds, 0.0..27.0 hours, 0:00..27:00
SELECT_ORIGINS	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_DESTINATIONS	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_TRAVELER_TYPES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_FACILITY_TYPES	Optional	List	ALL	e.g., FREEWAY..EXTERNAL
SELECT_VEHICLE_TYPES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_PROBLEM_TYPES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_LINKS	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_NODES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_STOPS	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_ROUTES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300

SELECT_SUBAREA_POLYGON	Optional	Input File		[project_directory]filename
SELECT_ORIGIN_ZONES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_DESTINATION_ZONES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
PERCENT_TIME_DIFFERENCE	Optional	Decimal	0.0 percent	0.0..100.0 percent
MINIMUM_TIME_DIFFERENCE	Optional	Time	1 minutes	0..120 minutes
MAXIMUM_TIME_DIFFERENCE	Optional	Time	60 minutes	0..1440 minutes
PERCENT_COST_DIFFERENCE	Optional	Decimal	0.0 percent	0.0..100.0 percent
MINIMUM_COST_DIFFERENCE	Optional	Decimal	10 impedance	0..500 impedance
MAXIMUM_COST_DIFFERENCE	Optional	Decimal	1000 impedance	0..10000 impedance
SELECTION_PERCENTAGE	Optional	Decimal	100.0 percent	0.01..100.0 percent
MAXIMUM_PERCENT_SELECTED	Optional	Decimal	100.0 percent	0.1..100.0 percent
DELETION_FILE	Optional	Input File		[project_directory]filename
DELETION_FORMAT	Optional	Text	TAB_DELIMITED	Note #1
DELETE_HOUSEHOLDS	Optional	List	NONE	e.g., 1, 2, 4..10, 100..200, 300
DELETE_MODES	Optional	List	NONE	e.g., ALL or 1, 12..14 or WALK, HOV2..HOV4
DELETE_TRAVELER_TYPES	Optional	List	NONE	e.g., 1, 2, 4..10, 100..200, 300

Notes:

1	TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3

LineSum Quick Reference

Version 5.x.17

Syntax:

LineSum [-flag] [control_file]

Purpose:

1. Merge TPPlus transit ridership files by time period and daily total.
2. Generate line ridership reports showing boardings, alightings, and riders by time period and stop given multiple sets of lines.
3. Generate link ridership reports showing riders by mode, line, direction, and time period for selected modes, lines, and links.
4. Generate stop reports showing riders boarding and alighting by time period, mode and line for selected stops and modes.
5. Generate node access reports showing riders arriving and departing by time period, mode and node for selected stops and modes.
6. Generate stop reports showing the riders that arrive, board, alight, and depart by mode and line for selected stops, modes, and lines.
7. Generate total ridership reports showing total boarding, maximum load points, and passenger miles and hours of travel by line.
8. Create output data files for the ridership summarizes generated by stop, access, and total ridership reports.
9. Create link riders files in text or ArcGIS format that summarized the combined ridership on links from selected modes and lines.
10. Link rider files include peak and offpeak hours of service, peak hour factors, and mode capacities to calculate performance statistics such as passenger miles and hours of travel, vehicle miles and hours of travel, peak hour ridership, and load factors.
11. Use a link shape file to include shape coordinates in the ArcGIS version of the link rider file.
12. Report the differences between two route files with respect to peak and offpeak headways and run times and the number of stops on the route.

Command Line Flags:

-H[elp]	show program syntax and control keys
-C[ontrol]	create/update a default control file
-K[eyCheck]	list unrecognized control file keys
-P[ause]	pause before exiting
-N[oPause]	never pause before exiting
-Q[uiet]	execute without screen messages
-D[etail]	execute with detailed status messages
-X[ML]	write an XML file with control keys
-R[eport]	write control keys and report names

Execution Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
TITLE	Optional	Text		
REPORT_DIRECTORY	Optional	Directory		
REPORT_FILE	Optional	Output File		[report_directory]filename[_partition][.prn]
REPORT_FLAG	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
PAGE_LENGTH	Optional	Integer	65	>= 0 (0 = no page breaks)
PROJECT_DIRECTORY	Optional	Directory		
DEFAULT_FILE_FORMAT	Optional	Text	TAB_DELIMITED	Note #1

Projection Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
INPUT_COORDINATE_SYSTEM	Optional	Parameters		System, Code, Units (Note #2)
INPUT_COORDINATE_ADJUSTMENT	Optional	Parameters	0.0, 0.0, 1.0, 1.0	X_offset, Y_offset, X_factor, Y_factor
OUTPUT_COORDINATE_SYSTEM	Optional	Parameters		System, Code, Units (Note #2)
OUTPUT_COORDINATE_ADJUSTMENT	Optional	Parameters	0.0, 0.0, 1.0, 1.0	X_offset, Y_offset, X_factor, Y_factor
OUTPUT_XYZ_SHAPES	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
OUTPUT_XYM_SHAPES	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N

LineSum Control Keys:

Control Key	Usage	Value Type	Default Value	Value Range
PEAK_RIDERSHIP_FILE_#	Optional	Input File		[project_directory]filename
PEAK_RIDERSHIP_FORMAT_#	Optional	Text	DBASE	Note #1
OFFPEAK_RIDERSHIP_FILE_#	Optional	Input File		[project_directory]filename
OFFPEAK_RIDERSHIP_FORMAT_#	Optional	Text	DBASE	Note #1
NEW_PEAK_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
NEW_PEAK_RIDERSHIP_FORMAT	Optional	Text	DBASE	Note #1
NEW_OFFPEAK_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
NEW_OFFPEAK_RIDERSHIP_FORMAT	Optional	Text	DBASE	Note #1

NEW_TOTAL_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
NEW_TOTAL_RIDERSHIP_FORMAT	Optional	Text	DBASE	Note #1
STOP_NAME_FILE	Optional	Input File		[project_directory]filename
STOP_NAME_FORMAT	Optional	Text	TAB_DELIMITED	Note #1
LINE_REPORT_TITLE_#	Optional	Text	Line Report	Report Title
LINE_REPORT_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINE_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
LINE_REPORT_ALL_NODES_#	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
LINK_REPORT_TITLE_#	Optional	Text	Link Report	Report Title
LINK_REPORT_LINKS_#	Optional	List		e.g., 100-200, 300-400-500
LINK_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
LINK_REPORT_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINK_REPORT_ONEWAY_#	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
ON_OFF_REPORT_TITLE_#	Optional	Text	On-Off Report	Report Title
ON_OFF_REPORT_STOPS_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
ON_OFF_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
ON_OFF_REPORT_DETAILS_#	Optional	Boolean	FALSE	Note #3
NEW_ON_OFF_REPORT_FILE_#	Optional	Output File		[project_directory]filename
NEW_ON_OFF_REPORT_FORMAT_#	Optional	Text	TAB_DELIMITED	Note #1
ACCESS_REPORT_TITLE_#	Optional	Text	Access Report	Report Title
ACCESS_REPORT_STOPS_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
ACCESS_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
ACCESS_REPORT_DETAILS_#	Optional	Boolean	FALSE	Note #2
NEW_ACCESS_REPORT_FILE_#	Optional	Output File		[project_directory]filename
NEW_ACCESS_REPORT_FORMAT_#	Optional	Text	TAB_DELIMITED	Note #1
STOP_REPORT_TITLE_#	Optional	Text	Stop Report	Report Title
STOP_REPORT_STOPS_#	Optional	List		e.g., 1, 2, 4..10, 100..200, 300
STOP_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
STOP_REPORT_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|

STOP_REPORT_TRANSFERS_#	Optional	List	NONE	e.q., 1, 2, 4..10, 100..200, 300
TOTAL_REPORT_TITLE_#	Optional	Text	Total Report	Report Title
TOTAL_REPORT_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10
TOTAL_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
NEW_TOTAL_REPORT_FILE_#	Optional	Output File		[project_directory]filename
NEW_TOTAL_REPORT_FORMAT_#	Optional	Text	TAB_DELIMITED	Note #1
NEW_LINK_RIDER_FILE_#	Optional	Output File		[project_directory]filename
NEW_LINK_RIDER_FORMAT_#	Optional	Text	TAB_DELIMITED	Note #1
LINK_RIDER_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
LINK_RIDER_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINK_RIDER_PEAK_HOURS_#	Optional	Decimal	6	1.0..10.0
LINK_RIDER_PEAK_FACTOR_#	Optional	Decimal	1	1.0..10.0
LINK_RIDER_PEAK_CAPACITY_#	Optional	Decimal	1	1.0..1000.0
LINK_RIDER_OFFPEAK_HOURS_#	Optional	Decimal	10	1.0..20.0
LINK_RIDER_XY_FILE_#	Optional	Input File		[project_directory]filename
LINK_RIDER_XY_FORMAT_#	Optional	Text	TAB_DELIMITED	Note #1
LINK_RIDER_SIDE_OFFSET_#	Optional	Decimal	0.0 feet	0.0..1000 feet
LINK_SHAPE_FILE	Optional	Input File		[project_directory]filename
LINK_SHAPE_ANODE	Optional	Text	ANODE	
LINK_SHAPE_BNODE	Optional	Text	BNODE	
SERVICE_FILE	Optional	Input File		[project_directory]filename
SERVICE_FORMAT	Optional	Text	TAB_DELIMITED	Note #1
SERVICE_LINE_FIELD	Optional	Text	LINE	
SERVICE_PEAK_FIELD	Optional	Text	PEAK	
SERVICE_OFFPEAK_FIELD	Optional	Text	OFFPEAK	
BASE_ROUTE_FILE_#	Optional	Input File		[project_directory]filename
BASE_ROUTE_FORMAT_#	Optional	Text	DBASE	Note #1
ALTERNATIVE_ROUTE_FILE_#	Optional	Input File		[project_directory]filename
ALTERNATIVE_ROUTE_FORMAT_#	Optional	Text	DBASE	Note #1

LINESUM_REPORT_#	Optional	Text		program report name

Report Options:

LINE_REPORT
LINK_REPORT
ON_OFF_REPORT
ACCESS_REPORT
STOP_REPORT
TOTAL_REPORT
DIFFERENCE_REPORT

Notes:

1	TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3
2	UTM, STATEPLAN, and LATLONG – Example: UTM, 10N, METERS or STATEPLANE, 3601, FEET or LATLONG, MILLION_DEGREES
3	TRUE/FALSE/MODE, YES/NO/MODE, 1/0/2, T/F/M, Y/N/M

ModeChoice Quick Reference

Version 5.x.19

Syntax:

ModeChoice [-flag] [control_file]

Purpose:

13. Apply a nested-logit or multi-nomial mode choice model to zone-based trip tables.
    a. Read trip tables and one or more mode-specific skim files in TPPlus or TransCAD format.
    b. Trip tables can be subdivided by income or some other traveler attribute.
    c. Mode choice constants can vary by geographic market segments and traveler attribute.
    d. Short and long walk accessibility by origin and destination zone is used to determine mode shares by combined access markets.
    e. A user script interface is available for manipulating the skim and zonal attributes used for each origin-destination choice.
14. Apply an iterative model calibration process to match mode choice results to calibration targets.
    a. Calibration targets can be defined by mode, market segment, and traveler attribute.
    b. Exit criteria are control by a maximum number of iterations and a RMSE convergence value.
    c. Input constants with minimum and maximum constraints are used as seed values for the calibration process.
    d. The output calibration file can be used for mode choice applications or as input to additional calibration iterations.
15. Generate reports summarizing the mode shares, market segments, and calibration results.
16. Generate text files summarizing the mode shares, market segments, and calibration results.
17. Generate model results in the input format required by FTA's SUMMIT program.
18. Generate production and attraction zone mode split summary files.

Command Line Flags:

-H[elp]	show program syntax and control keys
-C[ontrol]	create/update a default control file
-K[eyCheck]	list unrecognized control file keys
-P[ause]	pause before exiting
-N[oPause]	never pause before exiting
-Q[uiet]	execute without screen messages
-D[etail]	execute with detailed status messages
-X[ML]	write an XML file with control keys

-R[eport]	write control keys and report names

Execution Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
TITLE	Optional	Text		
REPORT_DIRECTORY	Optional	Directory		
REPORT_FILE	Optional	Output File		[report_directory]filename[_partition][.prn]
REPORT_FLAG	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
PAGE_LENGTH	Optional	Integer	65	>= 0 (0 = no page breaks)
PROJECT_DIRECTORY	Optional	Directory		
DEFAULT_FILE_FORMAT	Optional	Text	TAB_DELIMITED	Note #1

Selection Service Keys:

Control Key	Usage	Value Type	Default Value	Value Range
SELECT_ORIGIN_ZONES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
SELECT_DESTINATION_ZONES	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300

ModeChoice Control Keys:

Control Key	Usage	Value Type	Default Value	Value Range
TRIP_FILE	Required	Input File		[project_directory]filename
TRIP_FORMAT	Optional	Text	TPPLUS	Note #2
NEW_TRIP_FILE	Required	Output File		[project_directory]filename
NEW_TRIP_FORMAT	Optional	Text	TPPLUS	Note #2
SELECT_TRIP_TABLES	Optional	Text	ALL	Comma separated list of trip table names
SKIM_FILE_#	Required	Input File		[project_directory]filename
SKIM_FORMAT	Optional	Text	TPPLUS	Note #2
ZONE_FILE	Required	Input File		[project_directory]filename
ZONE_FORMAT	Optional	Text	TAB_DELIMITED	Note #1
MODE_CONSTANT_FILE	Optional	Input File		[project_directory]filename

MODE_CHOICE_SCRIPT	Required	Input File		[project_directory]filename
SEGMENT_MAP_FILE	Optional	Input File		[project_directory]filename
ORIGIN_MAP_FIELD	Optional	Text	SEGMENT	
DESTINATION_MAP_FIELD	Optional	Text	SEGMENT	
TRIP_PURPOSE_LABEL	Optional	Text	Peak HBW	
TRIP_PURPOSE_NUMBER	Optional	Integer	1	1..100
TRIP_TIME_PERIOD	Optional	Integer	1	1..100
PRIMARY_MODE_CHOICE	Required	Text		
MODE_CHOICE_NEST_#	Optional	Text		
NESTING_COEFFICIENT_#	Optional	Decimal	0.5	0.0..1.0
VEHICLE_TIME_VALUE	Optional	Decimal	-0.02	0, -0.03..-0.02
WALK_TIME_VALUE	Optional	Decimal	0	0, -1.0..-0.02
DRIVE_ACCESS_VALUE	Optional	Decimal	0	0, -1.0..-0.02
WAIT_TIME_VALUE	Optional	Decimal	0	0, -1.0..-0.02
LONG_WAIT_VALUE	Optional	Decimal	0	0, -1.0..-0.02
TRANSFER_TIME_VALUE	Optional	Decimal	0	0, -1.0..-0.02
TRANSFER_COUNT_VALUE	Optional	Decimal	0	0, -1.0..-0.02
PENALTY_TIME_VALUE	Optional	Decimal	0	0, -1.0..-0.02
TERMINAL_TIME_VALUE	Optional	Decimal	0	0, -1.0..-0.02
COST_VALUE_TABLE_#	Optional	Decimal	0	0, -5.0..0.0
MODE_ACCESS_MARKET_#	Optional	Text		SOV, SR2, SR3...
NEW_TABLE_MODES_#	Optional	Text		
OUTPUT_TRIP_FACTOR	Optional	Decimal	1.0	1.0..1000.0
NEW_MODE_SUMMARY_FILE	Optional	Output File		[project_directory]filename
NEW_MARKET_SEGMENT_FILE	Optional	Output File		[project_directory]filename
NEW_MODE_SEGMENT_FILE	Optional	Output File		[project_directory]filename
NEW_FTA_SUMMIT_FILE	Optional	Output File		[project_directory]filename
NEW_PRODUCTION_FILE	Optional	Output File		[project_directory]filename
NEW_PRODUCTION_FORMAT	Optional	Text	TAB_DELIMITED	Note #2

NEW_ATTRACTION_FILE	Optional	Output File		[project_directory]filename
NEW_ATTRACTION_FORMAT	Optional	Text	TAB_DELIMITED	Note #2
CALIBRATION_TARGET_FILE	Optional	Input File		[project_directory]filename
CALIBRATION_SCALING_FACTOR	Optional	Decimal	1	1.0..5.0
MAX_CALIBRATION_ITERATIONS	Optional	Integer	20	1..1000
CALIBRATION_EXIT_RMSE	Optional	Decimal	5	1.0..50.0
NEW_MODE_CONSTANT_FILE	Optional	Output File		[project_directory]filename
NEW_CALIBRATION_DATA_FILE	Optional	Output File		[project_directory]filename
MODECHOICE_REPORT_#	Optional	Text		program report name

Report Options:

MODE_CHOICE_SCRIPT
MODE_CHOICE_STACK
MODE_SUMMARY_REPORT
MARKET_SEGMENT_REPORT
CALIBRATION_REPORT
TARGET_DATA_REPORT
MODE_VALUE_SUMMARY
SEGMENT_VALUE_SUMMARY

Notes:

1	TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3
2	TRANSCAD, CUBE, TPPLUS, TRANPLAN, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3

Execution Service (version 5.x)

All TRANSIMS programs are derived from the Execution Service class. This class:

1. Manages the command line interface;
2. Processes the command line flags;
3. Creates the program report file;
4. Reads and processes the configuration file;
5. Reads and processes the control file and control keys;
6. Manages the report and print interface;
7. Manages the message, problem and error services;
8. Generates help messages and the XML interface;
9. Manages partition processing data;
10. Controls multi-threading and MPI options;
11. Processes a global set of control keys; and
12. Executes the program.

Command Line Syntax

TRANSIMS programs are console-based programs that run in a command window on either 32 or 64 bit Windows or Linux operating systems. The programs are executed from the command prompt or through a batch file using one of the following syntax statements:

Program [-flag] [control_file]

Program [-flag] [control_file] [partition]

Program [-flag] [control_file] [parameter]

The flag parameters are optional. Any combination of the following flag parameters can be included on the command line:

-H[elp]	-C[ontrol]	-K[eyCheck]
-P[ause]	-N[oPause]	-Q[uiet]
-D[etail]	-X[ML]	-R[eport]

–Help

The help flag writes the program syntax, control keys, and report options to the screen to provide a quick reminder of the control options and the text used to reference various keys and reports. The help flag can be specified with or without a control file. If a control file is not provided, the program writes the help messages to the screen and waits for the user to press the enter key to exit.

A typical help file is shown below. The control key section of a help message shows the key name and identifies if the key is required or optional, the data type of the key value, and the default value for the key.

```
********************************************
|                                          |
|         LineSum - Version 5.0.0          |
|  Copyright 2012 by TRANSIMS Open-Source  |
|         Wed May 16 19:49:50 2012         |
|                                          |
********************************************

Syntax is LineSum [-flag] [control_file]

Optional Flags:
  -H[elp]      = show program syntax and control keys
  -C[ontrol]   = create/update a default control file
  -K[eyCheck]  = list unrecognized control file keys
  -P[ause]     = pause before exiting
  -N[oPause]   = never pause before exiting
  -Q[uiet]     = execute without screen messages
  -D[etail]    = execute with detailed status messages
  -X[ML]       = write an XML file with control keys
  -R[eport]    = write control keys and report names

Control File Keys:
  TITLE                              Opt.Text
  REPORT_FILE                        Opt.New
  REPORT_FLAG                        Opt.Bool = FALSE
  PROJECT_DIRECTORY                  Opt.Path
  DEFAULT_FILE_FORMAT                Opt.Text = TAB_DELIMITED
  PEAK_RIDERSHIP_FILE_#              Opt.File
  PEAK_RIDERSHIP_FORMAT_#            Opt.Text = DBASE
  OFFPEAK_RIDERSHIP_FILE_#           Opt.File
  OFFPEAK_RIDERSHIP_FORMAT_#         Opt.Text = DBASE
  NEW_PEAK_RIDERSHIP_FILE            Opt.New
  NEW_PEAK_RIDERSHIP_FORMAT          Opt.Text = DBASE
  NEW_OFFPEAK_RIDERSHIP_FILE         Opt.New
  NEW_OFFPEAK_RIDERSHIP_FORMAT       Opt.Text = DBASE
  STOP_NAME_FILE                     Opt.File
  STOP_NAME_FORMAT                   Opt.Text = TAB_DELIMITED
  LINE_REPORT_TITLE_#                Opt.Text = Line Report
  LINE_REPORT_LINES_#                Opt.List = ALL
  LINE_REPORT_MODES_#                Opt.List = ALL
  LINE_REPORT_ALL_NODES_#            Opt.Bool = False
  LINK_REPORT_TITLE_#                Opt.Text = Link Report
  LINK_REPORT_LINKS_#                Opt.List
  LINK_REPORT_MODES_#                Opt.List = ALL
  LINK_REPORT_LINES_#                Opt.List = ALL
  LINK_REPORT_ONEWAY_#               Opt.Bool = False
  NEW_LINK_REPORT_FILE_#             Opt.New
  ACCESS_REPORT_TITLE_#              Opt.Text = Access Report
  ACCESS_REPORT_STOPS_#              Opt.List = ALL
  ACCESS_REPORT_MODES_#              Opt.List = ALL
  ACCESS_REPORT_DETAILS_#            Opt.Bool = False
  NEW_ACCESS_REPORT_FILE_#           Opt.New
  STOP_REPORT_TITLE_#                Opt.Text = Stop Report
  STOP_REPORT_STOPS_#                Opt.List
  STOP_REPORT_MODES_#                Opt.List = ALL
  STOP_REPORT_LINES_#                Opt.List = ALL
  NEW_LINK_RIDER_FILE_#              Opt.New
  NEW_LINK_RIDER_FORMAT_#            Opt.Text = TAB_DELIMITED
  LINK_RIDER_MODES_#                 Opt.List = ALL
  LINK_RIDER_LINES_#                 Opt.List = ALL
  LINK_RIDER_PEAK_HOURS_#            Opt.Dec. = 6.0
  LINK_RIDER_PEAK_FACTOR_#           Opt.Dec. = 1.0
  LINK_RIDER_PEAK_CAPACITY_#         Opt.Dec. = 1.0
  LINK_RIDER_OFFPEAK_HOURS_#         Opt.Dec. = 10.0
  LINK_RIDER_XY_FILE_#               Opt.File
  LINK_RIDER_XY_FORMAT_#             Opt.Text = TAB_DELIMITED
  BASE_ROUTE_FILE_#                  Opt.File
  BASE_ROUTE_FORMAT_#                Opt.Text = DBASE
  ALTERNATIVE_ROUTE_FILE_#           Opt.File
  ALTERNATIVE_ROUTE_FORMAT_#         Opt.Text = DBASE
  LINESUM_REPORT_#                   Opt.Text

Report Options:
  LINE_REPORT
  LINK_REPORT
  ACCESS_REPORT
  STOP_REPORT
  TOTAL_REPORT
  DIFFERENCE_REPORT

Press Enter to Continue
```

–Control

The control flag creates a control file template using the default values or updates an existing control file by adding keys that were not previously included. If a control file name is not provided, the program creates a file using the program name with the ".ctl" extension. If a control file name is include, the file is read, the default values are replaced with the key values from the control file, the program saves the updated control file and proceeds to execute the program. If the user wants to update the control file without executing the program, the flag –CX can be used to exit the program after the control file is updated.

A typical control template is shown below. It lists the keys in service groups, includes the default value if one exists, and adds a comment message that lists the acceptable value range, value options, or syntax examples.

```
TITLE                              LineSum Default Control Keys
REPORT_FILE                                         //---- [report_directory]filename[_partition][.prn]
REPORT_FLAG                        FALSE            //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
PROJECT_DIRECTORY
DEFAULT_FILE_FORMAT                TAB_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…

#---- LineSum Control Keys ----

PEAK_RIDERSHIP_FILE_1                               //---- [project_directory]filename
PEAK_RIDERSHIP_FORMAT_1            DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
OFFPEAK_RIDERSHIP_FILE_1                            //---- [project_directory]filename
OFFPEAK_RIDERSHIP_FORMAT_1         DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
NEW_PEAK_RIDERSHIP_FILE                             //---- [project_directory]filename
NEW_PEAK_RIDERSHIP_FORMAT          DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
NEW_OFFPEAK_RIDERSHIP_FILE                          //---- [project_directory]filename
NEW_OFFPEAK_RIDERSHIP_FORMAT       DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
NEW_TOTAL_RIDERSHIP_FILE                            //---- [project_directory]filename
NEW_TOTAL_RIDERSHIP_FORMAT         DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
STOP_NAME_FILE                                      //---- [project_directory]filename
STOP_NAME_FORMAT                   TAB_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
LINE_REPORT_TITLE_1                Line Report      //---- Report Title
LINE_REPORT_LINES_1                ALL              //---- e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINE_REPORT_MODES_1                ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
LINE_REPORT_ALL_NODES_1            False            //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
LINK_REPORT_TITLE_1                Link Report      //---- Report Title
LINK_REPORT_LINKS_1                                 //---- e.g., 100-200, 300-400-500
LINK_REPORT_MODES_1                ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
LINK_REPORT_LINES_1                ALL              //---- e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINK_REPORT_ONEWAY_1               False            //---- TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
ACCESS_REPORT_TITLE_1              Access Report    //---- Report Title
ACCESS_REPORT_STOPS_1              ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
ACCESS_REPORT_MODES_1              ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
ACCESS_REPORT_DETAILS_1            False            //---- TRUE/FALSE/MODE, YES/NO/MODE, 1/0/2, T/F/M,…
NEW_ACCESS_REPORT_FILE_1                            //---- [project_directory]filename
NEW_ACCESS_REPORT_FORMAT_1         TAB_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
STOP_REPORT_TITLE_1                Stop Report      //---- Report Title
STOP_REPORT_STOPS_1                                 //---- e.g., 1, 2, 4..10, 100..200, 300
STOP_REPORT_MODES_1                ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
STOP_REPORT_LINES_1                ALL              //---- e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
TOTAL_REPORT_TITLE_1               Total Report     //---- Report Title
TOTAL_REPORT_LINES_1               ALL              //---- e.g., LINE1, LINE2, LINE1..LINE10
NEW_TOTAL_REPORT_FILE_1                             //---- [project_directory]filename
NEW_LINK_RIDER_FILE_1                               //---- [project_directory]filename
NEW_LINK_RIDER_FORMAT_1            TAB_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
LINK_RIDER_MODES_1                 ALL              //---- e.g., 1, 2, 4..10, 100..200, 300
LINK_RIDER_LINES_1                 ALL              //---- e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
LINK_RIDER_PEAK_HOURS_1            6.0              //---- 1.0..10.0
LINK_RIDER_PEAK_FACTOR_1           1.0              //---- 1.0..10.0
LINK_RIDER_PEAK_CAPACITY_1         1.0              //---- 1.0..1000.0
LINK_RIDER_OFFPEAK_HOURS_1         10.0             //---- 1.0..20.0
LINK_RIDER_XY_FILE_1                                //---- [project_directory]filename
LINK_RIDER_XY_FORMAT_1             TAB_DELIMITED    //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
BASE_ROUTE_FILE_1                                   //---- [project_directory]filename
BASE_ROUTE_FORMAT_1                DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
ALTERNATIVE_ROUTE_FILE_1                            //---- [project_directory]filename
ALTERNATIVE_ROUTE_FORMAT_1         DBASE            //---- TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED,…
LINESUM_REPORT_1                                    //---- program report name
```

–KeyCheck

The key-check flag generates a list of warning messages for any key included in the control file that is not recognized by the program. By default, a control file can include keys that are used by multiple programs. Some of the keys may be unique to one program and ignored by other programs. Based on this behavior, a key that is intended for a given program, but is defined improperly or misspelled will be ignored. The key check option enables the user to check if their control keys are specified correctly or remove keys that are not used by a particular program.

–Pause

If the pause flag is included on the command line, the program with require the user to press the enter key before the program exits and returns control to the operating system or the batch file. This can be a useful way of reviewing the on-screen messages before the operating system deletes the command window.

–NoPause

If the no-pause flag is included, the program will always exit without waiting for user intervention. Normally, when an error is detected, the program writes an error message and waits for the user to press the enter key before exiting the program. The no-pause option permits the program to automatically exit even when an error is detected. In this case, a batch procedure should check for an error return code from the program and take appropriate action.

–Quiet

The quiet flag suppresses output to the computer screen unless or until an error message is encountered. It is intended for applications that run on multiple remote processors or for batch applications that capture screen output to a log file. In this case the log file will include the batch commands but not the long list of status messages that are typically written to the screen. The log file will include the banner page and error message when the program terminates with an error.

–Detail

The detail flag is intended for program applications that wish to save screen output to a log file. When a program is processing a given file or making calculations, it writes a progress message to the screen once every second to provide the user with general information about the processing rate. Normally the progress counter overwrites the previous value each second. When screen output is written to a file, the backspace commands used to overwrite the counter generates undesirable output. In this case, the detail flag can be used to write each progress counter on a separate line.

–XML

The XML flag writes the same type of information as the control flag in an XML file format. If a control file is provided, the XML file will include the key values specified in the control file plus the default values for other keys. The output filename will be the name of the program with the “.xml” extension or the name of the input control file with the file extension replaced with “.xml”.

A small section of a sample XML file is shown below. Notice how the LEVEL_KEYS section is coded. In this case the KEY_CODE 806 defines the syntax for a generic key group. The input control file, however, defined two instances of this key. These instances are listed in the LEVEL_KEYS section along with the values provided by the control file.

```xml
<?xml version="1.0" encoding="UTF-8" ?>
<TRANSIMS>
<PROGRAM NAME="Router" VERSION="5.0.46" COPYRIGHT="2012 by TRANSIMS Open-Source" PARTITIONS="TRUE" />
<CONTROL_KEYS>
<KEY CODE="200" NAME="TITLE" REQUIRED="false" TYPE="Text" VALUE="Router Test" />
<KEY CODE="204" NAME="PROJECT_DIRECTORY" REQUIRED="false" TYPE="Path" VALUE="../" />
<KEY CODE="213" NAME="MAX_WARNING_EXIT_FLAG" REQUIRED="false" TYPE="Bool" DEFAULT="TRUE" RANGE="TRUE/FALSE, YES/NO, 1/0, T/F, Y/N" />
<KEY CODE="214" NAME="MAX_PROBLEM_COUNT" REQUIRED="false" TYPE="Integer" DEFAULT="0" RANGE=">= 0" />
<KEY CODE="215" NAME="NUMBER_OF_THREADS" REQUIRED="false" TYPE="Integer" DEFAULT="1" RANGE="1..64" VALUE="30" />
<KEY CODE="300" NAME="NODE_FILE" REQUIRED="true" TYPE="Net" RANGE="[project_directory]filename" VALUE="network/node.txt" />
<KEY CODE="400" NAME="NODE_FORMAT" REQUIRED="false" TYPE="Text" DEFAULT="TAB_DELIMITED" RANGE="TEXT, BINARY, FIXED_COLUMN,
COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3" HELP="2" />
<KEY CODE="303" NAME="LINK_FILE" REQUIRED="true" TYPE="Net" RANGE="[project_directory]filename" VALUE="network/link.txt" />
<KEY CODE="403" NAME="LINK_FORMAT" REQUIRED="false" TYPE="Text" DEFAULT="TAB_DELIMITED" RANGE="TEXT, BINARY, FIXED_COLUMN,
COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, VERSION3" HELP="2" />
<KEY CODE="805" NAME="LINK_DELAY_FLOW_FACTOR" REQUIRED="false" TYPE="Decimal" DEFAULT="1.0" RANGE="1..100000" VALUE="1.0" />
<KEY CODE="806" NAME="EQUATION_PARAMETERS_#" REQUIRED="false" TYPE="List" DEFAULT="BPR, 0.15, 4.0, 0.75" RANGE="BPR, 0.15, 4.0, 0.75" >
<LEVEL_KEYS>
<LEVEL NAME="EQUATION_PARAMETERS_1" VALUE="BPR, 0.2, 8.5, 1.0" />
<LEVEL NAME="EQUATION_PARAMETERS_2" VALUE="BPR, 0.25, 9.0, 0.75" />
</LEVEL_KEYS>
</KEY>
<KEY CODE="1" NAME="UPDATE_PLAN_RECORDS" REQUIRED="false" TYPE="Bool" DEFAULT="FALSE" RANGE="TRUE/FALSE, YES/NO, 1/0, T/F, Y/N" />
<KEY CODE="2" NAME="REROUTE_FROM_TIME_POINT" REQUIRED="false" TYPE="Time" DEFAULT="0:00" />
<KEY CODE="11" NAME="NEW_TRIP_CONVERGENCE_FILE" REQUIRED="false" TYPE="NewFile" RANGE="[project_directory]filename"
VALUE="demand/trip_gap.txt" />
<KEY CODE="216" NAME="ROUTER_REPORT_#" REQUIRED="false" TYPE="Text" RANGE="program report name" HELP="1" >
<LEVEL_KEYS>
<LEVEL NAME="ROUTER_REPORT_1" VALUE="LINK_GAP_REPORT" />
<LEVEL NAME="ROUTER_REPORT_2" VALUE="TRIP_GAP_REPORT" />
<LEVEL NAME="ROUTER_REPORT_3" VALUE="ITERATION_PROBLEMS" />
</LEVEL_KEYS>
</KEY>
```


If the XML flag is specified as –XH, the output file will include additional help messages defined in the TRANSIMS help file. The path to the help file is set using the operating system environment variable TRANSIMS_HELP_FILE. An example of the help message section of an XML file is shown below.


```xml
<HELP_CODES>
<HELP CODE="1" >
<LINE NUM="1" TEXT=" Reports are requested through a nested key with syntax: " />
<LINE NUM="2" TEXT=" PROGRAM_REPORT_# = REPORT_NAME" />
<LINE NUM="3" TEXT="" />
<LINE NUM="4" TEXT=" For Example: " />
<LINE NUM="5" TEXT=" LINKSUM_REPORT_1 = TOP_100_LINK_FLOWS" />
<LINE NUM="6" TEXT=" LINKSUM_REPORT_2 = TOP_100_LANE_FLOWS" />
<LINE NUM="7" TEXT=" LINKSUM_REPORT_3 = LINK_VOLUME_GREATER_THAN_1.3" />
<LINE NUM="8" TEXT=" LINKSUM_REPORT_4 = LINK_VOLUME_GREATER_THAN_2.5" />
<LINE NUM="9" TEXT="" />
<LINE NUM="10" TEXT="Note that the last two reports request the same report with different filtering crite…
<LINE NUM="11" TEXT="This report is defined with a wildcard code (LINK_VOLUME_GREATER_THEN_#) that enables…
<LINE NUM="12" TEXT="to specify a filter parameter. Multiple reports of this type can be generated." />
<LINE NUM="13" TEXT="" />
<LINE NUM="14" TEXT="In most cases, the reports are printed in the report file in the report key order." />
</HELP>
<HELP CODE="2" >
<LINE NUM="1" TEXT=" Format keys are used to define how data files are read or created." />
<LINE NUM="2" TEXT=" If a format key is not provided, the value of the DEFAULT_FILE_FORMAT key is used." />
<LINE NUM="3" TEXT=" The default value of the DEFAULT_FILE_FORMAT key is TAB_DELIMITED. " />
<LINE NUM="4" TEXT="" />
<LINE NUM="5" TEXT=" In most cases, TRANSIMS constructs a *.def file for each file it creates. " />
<LINE NUM="6" TEXT="" />
<LINE NUM="7" TEXT=" The *.def file enables the software to identify the file format and field names for …
<LINE NUM="8" TEXT=" If a *.def file is available, the format key is ignored." />
<LINE NUM="9" TEXT=" If a *.def file is not available, the format key tells the program how to build a *….
<LINE NUM="10" TEXT="For Delimited files, the software reads the header line     and the first 100 records…
<LINE NUM="11" TEXT="The software cannot build *.def files for Binary and Fixed Columm files." />
<LINE NUM="12" TEXT="" />
<LINE NUM="13" TEXT="The VERSION3 format option is provided for backwards compatibility.  In many cases,…
<LINE NUM="14" TEXT="read a Version 3 or Version 4 file without modification.  The Version 5 software …
</HELP_CODES>
</TRANSIMS>
```

–Report

The report flag writes the program syntax, control keys and report information to a tab delimited file to assist with program documentation. The output file includes the program name with the ".doc" extension. If a control file is provided, the document file does not include any information from the control file.

Selected rows of a sample document file read into an Excel spreadsheet are shown below.

LineSum (5.0.1)					
Syntax:					
	LineSum [-flag] [control_file]				
Optional Flags:					
	-H[elp]	show program syntax and control keys			
	-C[ontrol]	create/update a default control file			
	-K[eyCheck]	list unrecognized control file keys			
	-P[ause]	pause before exiting			
	-N[oPause]	never pause before exiting			
	-Q[uiet]	execute without screen messages			
	-D[etail]	execute with detailed status messages			
	-X[ML]	write an XML file with control keys			
	-R[eport]	write control keys and report names			
Execution Service Keys:					
	TITLE	Optional	Text		
	REPORT_DIRECTORY	Optional	Directory		
	REPORT_FILE	Optional	Output File		[report_directory]filename[_partition][.prn]
	REPORT_FLAG	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
	PROJECT_DIRECTORY	Optional	Directory		
	DEFAULT_FILE_FORMAT	Optional	Text	TAB_DELIMITED	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
LineSum Control Keys:					
	PEAK_RIDERSHIP_FILE_#	Optional	Input File		[project_directory]filename
	PEAK_RIDERSHIP_FORMAT_#	Optional	Text	DBASE	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	OFFPEAK_RIDERSHIP_FILE_#	Optional	Input File		[project_directory]filename
	OFFPEAK_RIDERSHIP_FORMAT_#	Optional	Text	DBASE	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	NEW_PEAK_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
	NEW_PEAK_RIDERSHIP_FORMAT	Optional	Text	DBASE	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	NEW_OFFPEAK_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
	NEW_OFFPEAK_RIDERSHIP_FORMAT	Optional	Text	DBASE	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	NEW_TOTAL_RIDERSHIP_FILE	Optional	Output File		[project_directory]filename
	NEW_TOTAL_RIDERSHIP_FORMAT	Optional	Text	DBASE	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	STOP_NAME_FILE	Optional	Input File		[project_directory]filename
	STOP_NAME_FORMAT	Optional	Text	TAB_DELIMITED	TEXT, BINARY, FIXED_COLUMN, COMMA_DELI
	LINE_REPORT_TITLE_#	Optional	Text	Line Report	Report Title
	LINE_REPORT_LINES_#	Optional	List	ALL	e.g., LINE1, LINE2, LINE1..LINE10, AB..AB|
	LINE_REPORT_MODES_#	Optional	List	ALL	e.g., 1, 2, 4..10, 100..200, 300
	LINE_REPORT_ALL_NODES_#	Optional	Boolean	FALSE	TRUE/FALSE, YES/NO, 1/0, T/F, Y/N
	LINESUM_REPORT_#	Optional	Text		program report name
Report Options:					
	LINE_REPORT				
	LINK_REPORT				
	ACCESS_REPORT				
	STOP_REPORT				
	TOTAL_REPORT				
	DIFFERENCE_REPORT				

Control File

The control_file field on the command line is the directory path and file name of a text file that contains the control strings expected by the program. If a file name is not provided, the program will prompt the user to enter a file name. The program automatically creates a printout file based on the control file name. If the file name includes an extension (e.g., ".ctl"), the extension is removed and ".prn" is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

If the program command syntax includes the partition option, the program can be instructed to process a subset of file partitions by specifying a partition number or partition range after the control file name. For example, the Router can execute a subset of partitions using a command line like:

    Router.exe Router.ctl 10

    Router.exe Router.ctl 0..4

The first command generates plans for the households assigned to partition 10. The second command generates plans for households assigned to partitions 0 through 4. In these cases, the printout file generated by the program includes the partition number or range in the file name:

    Router_10.prn

    Router_0-4.prn

If the program command syntax includes the parameter option, the printout file will include the parameter information. For example, the command:

    RunSetup.exe TripModel.ctl 2010

will create the printout file:

    TripModel_2010.prn

Program Controls

Program control parameters are defined using a control key followed by a string or number. The control parameters can be specified in any order and are not case sensitive. If a given key is defined more than once, the last instance of the key is used. A given program can define a given key as required or optional. If the key is required and not included in the control file, an error message is written and the program is terminated. The program may also assign a default value and value range to each key. If the default value is appropriate, the key does not need to be included in the file. If the user-provided value is out of range, an error message is written and the program is terminated.

Each key includes a value type that defines how the key is processed. Value types include integer or decimal numbers, text strings, Boolean (true/false) flags, time strings, input and output files, directory paths, and lists. A list key includes one or more values or value ranges. TRANSIMS defines a value range using two periods (e.g., 100..200).

Some keys may also permit multiple instances or multiple nesting levels. In these cases, the key name is followed by a number that defines a particular instance. Several examples of multi-level keys are shown in the –Report section above. The PEAK_RIDERSHIP_FILE_# key implies that multiple peak ridership files can be processed by the program. The control file identifies multiple files by replacing the "#" with a number. For example:

PEAK_RIDERSHIP_FILE_1	Myfile1.dbf
PEAK_RIDERSHIP_FILE_2	Myfile2.dbf

If the application does not require multiple files, the _# extension is not required. In other words, the control file could include the following key:

PEAK_RIDERSHIP_FILE	Myfile.dbf

In addition to defining multiple instances of a given key, the level code is used to define groups of keys. The –Report section above also includes an example of a key group. The four keys:

LINE_REPORT_TITLE_#
LINE_REPORT_LINES_#
LINE_REPORT_MODES_#
LINE_REPORT_ALL_NODES_#

function as a group. The instances of each key that share the same level code are processed together. In this case a line report has four control parameters and multiple line reports with difference selection criteria can be generated by the program.

Note that comment lines or extraneous keys can be included in a control file.   They will be ignored by the program. Comment lines or messages are identified by text strings that start with one of the following character sequences:

#- #* // /- /* ;; ;- ;*

All test in the record after the comment characters is ignored. For example:

```
LINE_REPORT_MODES_2         1..3    //---- local bus, express bus, and Metrorail ----
##LINE_REPORT_ALL_NODES_2           TRUE
```

The first line shows a comment message after the key value. The second line shows a convenient method of disabling a given key.

Execution Service Keys

The execution service manages a number of control keys that are common to all programs. These keys are described below:

TITLE (optional, text)

Any text string can be used on this line. This text is printed on the top of each output page.

REPORT_DIRECTORY (optional, path)

If the report directory key is specified, it is added to the report file name specified by the Report File key or the default report file name derived from the control file name. By default, the report file is created in the same directory as the control file. If the control file name includes path information, the path string is removed and replaced by the report directory string.

REPORT_FILE (optional, output file)

If a report file name is not provided, the program automatically creates a report file name based on the input control file name plus the partition number. The report file will overwrite an existing file with the same name if the Report Flag key is False or not specified.

REPORT_FLAG (optional, flag, FALSE)

If the report flag key is YES or TRUE, the report file or default printout file will be opened in "Append" mode rather than "Create" mode. This permits the user to consolidate the output of several programs into a single report file.

PAGE_LENGTH (optional, integer, 65, >= 0)

This key is used to change the default page length in the print files. By default, TRANSIMS creates a page break and page header with a title, date-time stamp, program name and page number on the top of each page or report. The default is 65 lines for print files in portrait format. If this value is set to zero or a very high number, a page break and title will not be generated by the program. This creates a continuous stream of output text that may be useful for data extraction or report re-formatting purposes.

PROJECT_DIRECTORY (optional, text)

If the project directory key is specified, it is added to all file names referenced by the program. If it is not specified, all file names should fully specify the file path relative to the current directory.

DEFAULT_FILE_FORMAT (optional, text, TAB_DELIMITED)

This key can be used to change the default file format. By default, TRANSIMS creates new files in TAB_DELIMITED format. Other options include BINARY, DBASE, COMMA_DELIMITED, SPACE_DELIMITED, FIXED_COLUMN and SQLITE3.

TIME_OF_DAY_FORMAT (optional, text, DAY_TIME)

The time of day format defines how the time data are written to the output files and reports. The default format will display values in DAY_TIME format (e.g., 0:00:00 to 1@3:00:00 refers to midnight to 3:00 AM the next day). The format options include SECONDS, MINUTES, HOURS, HOUR_CLOCK (e.g., 0:00 to 27:00), and TIME_CODE. Time codes combine a day code with an hour clock (e.g. TUE08:00). Day code options include SUN, MON, TUE, WED, THU, FRI, SAT, WKE, WKD, and ALL.

MODEL_START_TIME (optional, time, 0:00)

The model start time defines the time-of-day at the beginning of the modeling process. The default value is 0:00 or midnight. Many activity-based models consider the start of the day to be 3:00 AM when most people are at home in bed.

MODEL_END_TIME (optional, time, 24:00)

The model end time defines the time-of-day at the end of the modeling process. The default value is 24:00. Since there tends to be a significant number of trips that start near midnight and may take some time to reach their destination, the model end time is often increased to a value such as 27:00 to ensure that all trips are completed. Other applications may wish to model travel over multiple days (e.g., hurricane evacuation studies). In this case, this control key can be set to 48:00 or 72:00.

MODEL_TIME_INCREMENT (optional, time, 15 minutes, 2..240 minutes)

The model time increment defines the standard time period resolution used for dynamic assignments. The default value is 15 minutes. The combination of time increments and model start and end times established the number of time periods used for defining link travel times and speeds. For example, the default parameters create 96 different travel time values for each link.

UNITS_OF_MEASURE (optional, text, METRIC, ENGLISH/METRIC)

The default distance and speed units included in data files or control keys are assumed to be in METRIC units. This key can be used to specify the units of measure as ENGLISH or METRIC. If a particular key value includes data units, the program will automatically convert the value to the specified units of measure. The standard data files created by the TRANSIMS Version 5 software identify the units associated with each data field in the definition file (*.def).

RANDOM_NUMBER_SEED (optional, integer, 0, >= 0)

The random number seed key specifies the starting point for a list of random numbers. Any positive integer can be specified. If the value is zero or if no key is provided, the program uses the computer clock to set the random number seed. The selected seed value is written to the printout report to enable the user to re-run the model using the same random number sequence.

MAX_WARNING_MESSAGES (optional, integer, 100000, >= 0)

When the program generates a warning message, a counter is incremented and the total number of warning messages is reported and a warning return coded (2) is set at the end of the execution. By default the program prints up to 100,000 warning messages to the printout file. If more than 100,000 warning messages are sent, the program stops printing additional messages to the file or terminates the program with an error message based on the MAX_WARNING_EXIT_FLAG. This parameter enables the user to modify the default warning limit.

MAX_WARNING_EXIT_FLAG (optional, flag, true)

If the maximum number of warning messages is exceeded, this flag directs the program in what to do. If the flag is TRUE (the default), the program is terminated with an error message about the warning messages. If the flag is FALSE, the program continues execution, but no additional warning messages are sent to the screen or written to the printout file. The warning message counter continues to count the messages and reports the total at the end of the execution.

MAX_PROBLEM_COUNT (optional, integer, 0, >= 0)

The maximum problem count defines the number of modeling problems that are permitted before the problem terminates execution. The default value of zero disables this feature.

NUMBER_OF_THREADS (optional, integer, 1, 1..64)

This parameter is only used for programs where multi-thread processing is enables. TRANSIMS uses the Boost library to implemented processing threads. The software can be compiled with or without this library. If the library is included and the program is thread enabled, the number of threads key instructs the program on the number of CPUs that will be used for parallel data processing. The key value can range from 1 to 64. The user can disable the multi-thread processing by setting this key to 1. If the key value is greater than one and the particular program or compiled executable does not support multi-threading, a warning message is written to the screen.

Configuration File

In most TRANSIMS applications there are a significant number of keys that are common to all programs. Many of the Execution Service keys fall into this category. They tend to be global keys that define the default behavior of the model. If the modeler wishes to set these keys once and use them in all model applications, a TRANSIMS configuration file can be created. A configuration file is exactly like any other control file and can include any number of control keys and key values. Each TRANSIMS program looks for a configuration file using the operating system environment variable TRANSIMS_CONFIG_FILE. The variable points to a file name that stores the configuration keys. The program reads the configuration keys into memory before it reads the control file keys. If a control key is defined in both files, the value from the control file will override the value in the configuration file.

The path to a configuration file can be set dynamically for a particular application using the SET command within a batch file or at the command prompt. For example:

```
SET  TRANSIMS_CONFIG_FILE=c:\myproject\config.txt
```

Status Codes

TRANSIMS programs return a status code to the operating system based on the results of the application. A return code of zero indicates a successful completion with no warning messages. A return code of 1 indicates that the application was terminated with an error message. A return code of 2 indicates that the program ran to completion, but at least one warning message was generated. The user can detect these return codes within a batch file using the following command:

```
Router.exe  Router.ctl
if %ERRORLEVEL% == 1 exit 1
```

Definition Files

TRANSIMS uses definition files to interpret and define data fields within most input and output files generated by the modeling process. A definition file is automatically created when the file is created. It has the same path and file name as the data file with a ".def" extension added at the end. For example, the program control keys:

```
NEW_LINK_FILE          network\link.txt
NEW_LINK_FORMAT        TAB_DELIMITED
```

create a new link file in the network directory called "link.txt". The format key indicates that the link file will be created in tab delimited format. A definition file called "link.txt.def" will also be created in the network directory. The definition file is a standard text file containing the following information:

```
TRANSIMS50, TAB_DELIMITED, 1
LINK, INTEGER, 1, 10
NAME, STRING, 2, 40
NODE_A, INTEGER, 3, 10
NODE_B, INTEGER, 4, 10
```

LENGTH, DOUBLE, 5, 8.1, FEET
TYPE, STRING, 10, 12, FACILITY_TYPE
AREA_TYPE, UNSIGNED, 12, 3
LANES_AB, UNSIGNED, 14, 2
SPEED_AB, DOUBLE, 15, 5.1, MPH
FSPD_AB, DOUBLE, 16, 5.1, MPH
CAP_AB, UNSIGNED, 17, 8, VPH
USE, STRING, 22, 128, USE_TYPE

The first record in the *.def file specifies the software version that created the file (TRANSIMS 5.0), the data file format (tab delimited), and the number of header records in the data file (1). The header record is followed by one record for each data field. These records include the field name, the data type, the field offset within the data record, the maximum field length and number of decimal places, and, if appropriate, the units or enumeration type of the field. The units field facilitates conversions between English and metric systems. It also automates the process of converting text strings to internal type codes (i.e., enumerations) and back again. Binary files, for example, store the type codes as numbers rather than strings to reduce file size and improve performance.

When an existing file is read by a program, the program looks for the definition file to automatically determine how to read the file and process the data fields. If a definition file is not found, the program will look for a *.FORMAT control key where the user identifies the file format. In many cases, the program can used the file format information to read header records from the data file and construct a definition file. If the file is delimited, the program will read the first 100 records of the file to estimate the data types and field widths. This information is written to a new definition file constructed for the data file. If the estimation process is inaccurate, the user can edit the definition file to correct any inaccuracies.

Binary and fixed column file format definition files cannot be constructed automatically. These file formats do not store field header information in the data file. All information about how to read and interpret the file must be provided in the definition file. The user must manually create a definition file for these file types if they are to be read into a TRANSIMS program. This is also true for delimited files that do not include field names as the first record in the file.

TRANSIMS also supports nested files that include two record types. The first record is the master record that includes a field that identifies the number of nested records that follow. A link delay file is a typical example of a nested data file. The master records define the link, time period, flow and travel time on the link while the nested records define the turning movement links, flows, and travel times.

LINK	DIR	TYPE	START	END	FLOW	TIME	NCONNECT
OUT_LINK		OUT_FLOW		OUT_TIME			
37	0	0	2:00	2:15	2.0	19.4	2
44	1.0	19.4					
41	1.0	19.4					
37	1	0	2:00	2:15	0.5	19.4	0
39	0	0	2:00	2:15	8.0	63.8	3
42	4.0	63.8					
46	11.0	63.8					
43	1.0	63.8					
40	1	0	2:00	2:15	2.0	63.8	1
10	2.0	63.8					

41	0	0	2:00	2:15	1.0	63.8	1
45	1.0	63.8					
41	1	0	2:00	2:15	3.2	63.8	3
37	1.0	63.8					
40	2.0	63.8					
44	1.0	63.8					
42	1	0	2:00	2:15	3.8	63.8	1
41	4.0	63.8					
43	0	0	2:00	2:15	0.8	63.8	0

The definition file for the link delay file shown above looks like this:

```
TRANSIMS50, TAB_DELIMITED, 2, NESTED
LINK, INTEGER, 1, 10
DIR, INTEGER, 2, 1
TYPE, INTEGER, 3, 1
START, TIME, 4, 16, HOUR_CLOCK
END, TIME, 5, 16, HOUR_CLOCK
FLOW, DOUBLE, 6, 8.1, VEHICLES
TIME, TIME, 7, 8.1, SECONDS
NCONNECT, INTEGER, 8, 2, NEST_COUNT
OUT_LINK, INTEGER, 1, 10, NO, NESTED
OUT_FLOW, DOUBLE, 2, 8.1, VEHICLES, NESTED
OUT_TIME, TIME, 3, 8.1, SECONDS, NESTED
```

The first record indicates that the data file has two header records and includes the NESTED key word. The field specifications for the master record are exactly like any other definition file. The nested fields add the NESTED key word after the units field. Note that the record offsets restarts from 1 as well. The field with the NEST_COUNT identifier is used to determine how many nested records follow each master record.

Software Implementation

A TRANSIMS C++ program incorporates the execution services by inheriting from Execution_Service in the following way:

```
#include "Execution_Service.hpp"

class Program : public Execution_Service
```

By default, all execution service keys are available to the program. If the programmer wishes to exclude some keys from processing, the following logic can be added:

```
int ignore_keys [] = {
        TIME_OF_DAY_FORMAT, MODEL_START_TIME, MODEL_END_TIME, MODEL_TIME_INCREMENT,
        MAX_WARNING_MESSAGES, MAX_WARNING_EXIT_FLAG, NUMBER_OF_THREADS, 0
};
Ignore_Keys (ignore_keys);
```

The keys are then processed in the program control method using the following call:

```
Execution_Service::Program_Control ();
```

Selection Service (version 5.x)

Many TRANSIMS programs provide the user with the option of selecting a subset of data records for processing or reporting. Selection Service keys are often used to control the selection process. This class:

1. Defines the selection keys and reads and reports the key values;
2. Provides the inherited program with flags and range variables to implement the selection checks;
3. Includes path-based selection options for links, nodes, stops, and routes with compound or serial selection criteria;
4. Includes point-in-polygon selection using an ArcGIS shape file;
5. Includes selection percentages and maximum percent selected for random selection methods;
6. Includes minimum, maximum, and percent difference values to define significant differences for comparison selections; and
7. Include record deletion keys and a deletion file to select everything except a specified list of values.

A typical set of Selection Service control keys is shown below. These keys….

```
SELECT_ORIGIN_ZONES                20..30, 45, 55
SELECT_FACILITY_TYPES              FREEWAY..EXPRESSWAY
SELECT_SUBAREA_POLYGON             subarea.shp
SELECT_MODES                       WALK, HOV2..HOV4
SELECT_START_TIMES                 6:00..9:00
PERCENT_TIME_DIFFERENCE            10 percent
MINIMUM_TIME_DIFFERENCE            0.1 minutes
MAXIMUM_TIME_DIFFERENCE            60 minutes
PERCENT_COST_DIFFERENCE            1 percent
MINIMUM_COST_DIFFERENCE            10 impedance
MAXIMUM_COST_DIFFERENCE            10000 impedance
SELECTION_PERCENTAGE               20.0 percent
MAXIMUM_PERCENT_SELECTED           20.0 percent
```

Selection Service Keys

The selection service manages a number of control keys that are available in various programs. These keys are described below:

SELECT_HOUSEHOLDS (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of household IDs for processing. If it is not provided, all households will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of household IDs are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1000..2000).

SELECT_MODES (optional, list, ALL, e.g., ALL or 1, 12..14 or WALK, HOV2..HOV4)

This key enables the user to specify the transit modes that are considered for processing. If it is not provided, all transit modes will be considered by the selection process. The key is interpreted as a comma-delimited list of mode names (e.g., BUS, LOCAL_BUS, EXPRESS, EXPRESS_BUS, TROLLEY, STREETCAR, LIGHTRAIL, RAPIDRAIL, REGIONRAIL) or a list of number codes equal to the mode names.

SELECT_PURPOSES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of purpose IDs for processing. If it is not provided, all purposes will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of purpose IDs are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1..2).

SELECT_TIME_OF_DAY (optional, time, 0:00, 0:00..24:00)

This key enables the user to provide a time value for processing records that change by time of day. If it is not provided, all times will be considered by the selection process. The key is used to select data records that are applicable at a given time of day. For example, 8:00 can be used to select the network attributes, available transit routes, and service levels in the AM peak period or exclude the network records that are not active at 8:00 AM.

SELECT_START_TIMES (optional, list, ALL, e.g., ALL, 0..97200 seconds, 0.0..27.0 hours, 0:00..27:00)

This key enables the user to provide a range of start times for processing. If it is not provided, all start times will be considered by the selection process. The key is interpreted as a comma-delimited list of time ranges. A time range is specified by providing the beginning and ending time separated by two periods (e.g., 8:00..9:30).

SELECT_END_TIMES (optional, list, ALL, e.g., ALL, 0..97200 seconds, 0.0..27.0 hours, 0:00..27:00)

This key enables the user to provide a range of end times for processing. If it is not provided, all end times will be considered by the selection process. The key is interpreted as a comma-delimited list of time ranges. A time range is specified by providing the beginning and ending time separated by two periods (e.g., 8:00..9:30).

SELECT_ORIGINS (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of origin locations for processing. If it is not provided, all locations will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of locations are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1000..2000).

SELECT_DESTINATIONS (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of destination locations for processing. If it is not provided, all locations will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of locations are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1000..2000).

SELECT_TRAVELER_TYPES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of traveler type codes for processing. If it is not provided, all types will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of types are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1..2).

SELECT_FACILITY_TYPES (optional, list, ALL, e.g., FREEWAY..EXTERNAL)

This key enables the user to provide a list of facility types for processing. If it is not provided, all types will be considered by the selection process. The key is interpreted as a comma-delimited list of facility type names or name ranges. A sequential range of types are specified by providing the first name in the range and the last name in the range separated by two periods (e.g., FREEWAY..EXPRESSWAY).

SELECT_VEHICLE_TYPES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of vehicle type codes for processing. If it is not provided, all types will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of types are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1..2).

SELECT_PROBLEM_TYPES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of problem type codes for processing. If it is not provided, all types will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of types are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1..2).

SELECT_LINKS_# (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

The select links keys enable the user to provide a list of link numbers that a path must include before it is processed.   If it is not provided, the selection process will consider all paths. The "#" at the end of the keyword represents a selection set number (e.g., SELECT_LINKS_1). Any number of selection sets can be specified. If a path satisfies any one of the selection sets, the plan is included.

Each link key is interpreted as a comma-delimited list of link numbers or link number ranges. A sequential range of links are specified by providing the first node number in the range and the last link number in the range separated by two periods (e.g., 1000..1010). The path must include all of the links in the list in sequential order in order to be selected. The path may include other links between links included in the list, but it must travel through all of the links in the order specified

SELECT_NODES _# (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

The select nodes keys enable the user to provide a list of node numbers that a path must include before it is processed.   If it is not provided, the selection process will consider all paths. The "#" at the end of the keyword represents a selection set number (e.g., SELECT_NODES_1). Any number of selection sets can be specified. If a path satisfies any one of the selection sets, the plan is included.

Each node key is interpreted as a comma-delimited list of node numbers or node number ranges. A sequential range of nodes are specified by providing the first node number in the range and the last node number in the range separated by two periods (e.g., 1000..1010). The path must include all of the nodes in the list in sequential order in order to be selected. The path may include other nodes between nodes included in the list, but it must travel through all of the nodes in the order specified.

SELECT_STOPS (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to specify the transit stops that are considered for processing. If it is not provided, all transit stops will be considered by the selection process. The key is interpreted as a comma-delimited list of stop numbers or stop number ranges. A sequential range of stops are specified by providing the first stop number in the range and the last stop number in the range separated by two periods (e.g., 47..78). The path is selected if the transit path boards or alights at one of the stops in the list.

SELECT_ROUTES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to specify the transit routes that are considered for processing. If it is not provided, all transit routes will be considered by the selection process. The key is interpreted as a comma-delimited list of route numbers or route number ranges. A sequential range of routes are specified by providing the first route number in the range and the last route number in the range separated by two periods (e.g., 47..78). The path is selected if one of the transit legs is in the list.

SELECT_SUBAREA_POLYGON (optional, input file)

This key specifies the name and location of an ArcView shapefile used to select records with coordinates that fall within a polygon boundary. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. Only the first polygon in the file is used by the selection process. If the selection key is applied to a path, the path is selected if any node along the path falls within the polygon.

SELECT_ORIGIN_ZONES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of origin zones for processing. If it is not provided, all zones will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of zones are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 100..200).

SELECT_DESTINATION_ZONES (optional, list, ALL, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of destination zones for processing. If it is not provided, all zones will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of zones are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 100..200).

PERCENT_TIME_DIFFERENCE (optional, decimal, 0.0 percent, 0.0..100.0 percent)

This key controls a group of keys used to compare two travel plans and select those plans that are significantly different. The percent time different is the absolute travel time difference between the current plan and the comparison plan. This key must be included in the control file and be greater than zero for time difference processing to be implemented. The percent difference must be greater than or equal to this value for the current plan to be selected.

MINIMUM_TIME_DIFFERENCE (optional, time, 1 minutes, 0..120 minutes)

This key is only read if the percent time difference key is included in the control file. It defines the minimum absolute time difference between the current plan and the comparison plan to be considered for selection. To be selected, the time difference must be greater than the percent time difference and the minimum time difference.

MAXIMUM_TIME_DIFFERENCE (optional, time, 60 minutes, 0..1440 minutes)

This key is only read if the percent time difference key is included in the control file. If the absolute time difference is greater than the value of this key, the plan will always be selected even if it does not have a percent difference greater than the percent time difference key.

PERCENT_COST_DIFFERENCE (optional, decimal, 0.0 percent, 0.0..100.0 percent)

This key controls a group of keys used to compare two travel plans and select those plans that are significantly different. The percent cost different is the absolute generalized cost (impedance) difference between the current plan and the comparison plan. This key must be included in the control file and be greater than zero for cost difference processing to be implemented. The percent difference must be greater than or equal to this value for the current plan to be selected.

MINIMUM_COST_DIFFERENCE (optional, decimal, 10 impedance, 0..500 impedance)

This key is only read if the percent cost difference key is included in the control file. It defines the minimum absolute generalized cost (impedance) difference between the current plan and the comparison plan to be considered for selection. To be selected, the cost difference must be greater than the percent cost difference and the minimum cost difference.

MAXIMUM_COST_DIFFERENCE (optional, decimal, 1000 impedance, 0..10000 impedance)

This key is only read if the percent cost difference key is included in the control file. If the absolute generalized cost (impedance) difference is greater than the value of this key, the plan will always be selected even if it does not have a percent difference greater than the percent cost difference key.

SELECTION_PERCENTAGE (optional, decimal, 100.0 percent, 0.01..100.0 percent)

This key defines the percentage of the selected records that are ultimately written to the output file. The other selection keys define which records are eligible for selection. This key determines how many of those records are actually selected. If the selection percentage is less than 100 percent, a random number is used to determine if a given record is selected. The RANDOM_NUMBER_SEED key within Execution Service sets the initial seed value for the random number generator.

MAXIMUM_PERCENT_SELECTED (optional, decimal, 100.0 percent, 0.1..100.0 percent)

This key defines an upper limit to the selection process. Records are qualified for selection using various selection keys and the selection percentage selects a subset of these for output. If the total number of records selected by the selection process is a percentage of the total records in the database that exceed the value of this key, the number of records selected is reduced until the total is less than the value of this key. In other words, if this key is 10 percent, no more than 10 percent of the total number of records is selected.

DELETION_FILE (optional, input file)

This key specifies the name and location of a selection file used to select records to be excluded from the selection set. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. Since this file is interpreted in the same way as a selection file, it can include various combinations of Household, Person, Tour and Trip fields.

DELETION_FORMAT (optional, text, TAB_DELIMITED)

If the Deletion file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

DELETE_HOUSEHOLDS (optional, list, NONE, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of household IDs to exclude from processing. If it is not provided, all households will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of household IDs are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1000..2000).

DELETE_MODES (optional, list, NONE, e.g., ALL or 1, 12..14 or WALK, HOV2..HOV4)

This key enables the user to specify the transit modes to exclude from processing. If it is not provided, all transit modes will be considered by the selection process. The key is interpreted as a comma-delimited list of mode names (e.g., BUS, LOCAL_BUS, EXPRESS, EXPRESS_BUS, TROLLEY, STREETCAR, LIGHTRAIL, RAPIDRAIL, REGIONRAIL) or a list of number codes equal to the mode names.

DELETE_TRAVELER_TYPES (optional, list, NONE, e.g., 1, 2, 4..10, 100..200, 300)

This key enables the user to provide a list of traveler type codes to exclude from processing. If it is not provided, all types will be considered by the selection process. The key is interpreted as a comma-delimited list of numbers or number ranges. A sequential range of types are specified by providing the first ID in the range and the last ID in the range separated by two periods (e.g., 1..2).

Selection Algorithms

Selection keys typically function using "AND" logic. For example,

```
SELECT_ORIGIN_ZONES       20..30, 45, 55
SELECT_MODES              HOV2..HOV3
```

means the records must have a mode code of HOV 2 *or* HOV 3 *and* an origin zone number between 20 and 30, *or* 45 *or* 55.

The delete keys function in a similar way. For example,

```
SELECT_HOUSEHOLDS         1..100000
DELETE_HOUSEHOLDS         50000..60000, 70000..71000
```

is equivalent to

```
SELECT_HOUSEHOLDS         1..49999, 60001..69000, 71001..100000
```

The select links and select nodes keys differ from this basic behavior in two important ways. For example, the following select link keys

```
SELECT_LINKS_1            100, 110, 120, 200..202
SELECT_LINKS_2            300, 400, 500
```

create two selection sets. The plan is selected if *either* selection set is true. A given selection is true, if *all* of the links are included in the path *in the order* they are specified. In other words, the plan is selected if the path includes links 100, 110, 120, 200, 201, *and* 202 *or* the path includes 300, 400, *and* 500.

Comparison Selection Criteria

A typical set of keys used to select plans that are significantly different from a previous set of plans might look like the following:

```
PERCENT_TIME_DIFFERENCE        15
MINIMUM_TIME_DIFFERENCE        2 minutes
MAXIMUM_TIME_DIFFERENCE        20 minutes
SELECTION_PERCENTAGE           50
MAXIMUM_PERCENT_SELECTED       10
RANDOM_NUMBER_SEED             12345
```

In this case, the overall objective is to select plans that have travel time differences of 15 percent or more. The travel time might be 15 percent longer or 15 percent less than the previous plan (i.e., absolute difference). The criteria, however, constrains the percent difference to reasonable minimum and maximum absolute time differences to avoid selecting plans that are extremely short or extremely long. In this case, the time difference must be at least two minutes which typically ignores plans with large percent differences due to very short overall travel times. On the other hand, if the absolute time difference is greater than 20 minutes, the plan is also selected as a significant difference even if the percent difference is less than 15 percent. This can be expressed in mathematical terms as:

((abs_time_diff > 2) and (abs_time_diff / travel_time > 0.15)) or (abs_time_diff > 20)

The first group of keys selects a plan based on its difference to a previous plan. This is typically called the set of plans that are qualified for selection. The second group of keys determines which of the qualified plans are actually selected. In this case, no more than 50 percent of the qualified plans are selected. If, however, the total number of selected plans is greater than 10 percent of the total number of plans in the model, the number of plans are further reduced to keep the total selected less than or equal to 10 percent of the total plans in the model. A random probability is used to select the plans from the qualified set for the final selection. This can be expressed in computational terms as:

```
if ((num_qualified * select_percent / num_trips) > max_percent_select) {
        select_percent = max_percent_select * num_trips / num_qualified;
}
for (qualified plans) {
        if (random.Probability () <= select_percent) output ();
}
```

Software Implementation

A TRANSIMS C++ program can incorporate the selection logic in the following ways. First the program should inherit Data_Service and Select_Service:

```
#include "Data_Service.hpp"
#include "Select_Service.hpp"

class Program : public Data_Service, public Select_Service
```

and then select the selection keys that are appropriate to the given application:

```
int select_service_keys [] = {
       SELECT_HOUSEHOLDS, SELECT_MODES, SELECT_PURPOSES, SELECT_START_TIMES,
       SELECT_END_TIMES, SELECT_ORIGINS, SELECT_DESTINATIONS, SELECT_TRAVELER_TYPES,
       SELECT_FACILITY_TYPES, SELECT_LINKS, SELECT_NODES, SELECT_ORIGIN_ZONES,
       SELECT_DESTINATION_ZONES, PERCENT_TIME_DIFFERENCE, MINIMUM_TIME_DIFFERENCE,
       MAXIMUM_TIME_DIFFERENCE, SELECTION_PERCENTAGE, MAXIMUM_PERCENT_SELECTED, 0
};
Select_Service_Keys (select_service_keys);
```

The keys are then read and processed in the program control method after data service keys are processed using the following call:

```
Data_Service::Program_Control ();

Read_Select_Keys ();
```

This populates two variables for each key. One is a Boolean flag indicating if the key is set and the second is a data range containing the selection values. For example:

```
bool select_households, select_purposes, select_travelers;
Data_Range hhold_range, purpose_range, traveler_range;
```

The selection checks are made within the code using logic like the following:

```
if (select_households && !hhold_range.In_Range (plan.Household ())) continue;
if (select_purposes && !purpose_range.In_Range (plan.Purpose ())) continue;
if (select_travelers && !traveler_range.In_Range (plan.Type ())) continue;
```

The interface also defines special methods for processing the path-based selections.

```
bool Select_Plan_Links (Plan_Data &plan);
bool Select_Plan_Nodes (Plan_Data &plan);
bool Select_Plan_Stops (Plan_Data &plan);
bool Select_Plan_Routes (Plan_Data &plan);
bool Select_Plan_Subarea (Plan_Data &plan);
```

Projection Service (version 5.x)

All TRANSIMS programs that read or write ArcView/ArcGIS shape files include the Projection Service keys to convert the coordinate data to the appropriate internal and external units. This class:

1. Defines the projection system and units of the input coordinate data;
2. Converts the coordinate system to the output projection units; and
3. Writes a projection file (*.prj) for each newly created ArcGIS shape file.

A typical set of Projection Service control keys is shown below. These keys can be defined in a variety of different ways to define different projection systems.

```
INPUT_COORDINATE_SYSTEM          UTM, 10N, METERS
INPUT_COORDINATE_ADJUSTMENT      0.0, 0.0, 1.0, 1.0
OUTPUT_COORDINATE_SYSTEM         STATEPLANE, 3601, FEET
OUTPUT_COORDINATE_ADJUSTMENT     0.0, 0.0, 1.0, 1.0
OUTPUT_XYZ_SHAPES                FALSE
OUTPUT_XYM_SHAPES                FALSE
```

Projection Service Keys

The projection service manages a number of control keys that are common to all programs that work with ArcGIS shape files. These keys are described below:

INPUT_COORDINATE_SYSTEM (optional, parameter list)

The input coordinate system determines how the X and Y coordinate data fields in the Node, Zone and Shape files are translated into generic Latitude and Longitude values. It is only needed if coordinate conversions are desired and then only if the input coordinates are not in degrees of Latitude and Longitude. By default, TRANSIMS data files store coordinate data in UTM coordinates in meters.

The input coordinate command includes three parts separated by a comma. The first part is the coordinate system description. The options include UTM, STATEPLAN, and LATLONG. The second part identified the code number within the coordinate system that relates to the local conversion parameters. For UTM coordinates these codes range from 1N to 23N. Stateplane coordinates are defined using four digit FIPS codes (e.g., Oregon North = 3601). A code is not needed for the Latitude/Longitude system. The third parameter defines the coordinate units. By default, UTM is in meters, Stateplane is in feet, and Latitude/Longitude is in degrees. The user can override these assumptions using the following keywords: FEET, METERS, MILES, KILOMETERS, DEGREES, and MILLION_DEGREES.

INPUT_COORDINATE_ADJUSTMENT (optional, parameter list)

The input coordinate adjustment enables the user to manipulate the coordinates before they are sent to the input coordinate conversion calculation. It is only needed if the coordinates are not in the units expected by the conversion algorithm. By default, TRANSIMS data files store coordinate data in meters that don't require any adjustments.

The adjustment command includes four floating-point numbers separated by commas. The first two numbers are the X and Y offsets. The last two numbers are X and Y adjustment factors. The process adds the offset value to the coordinate and then applies the adjustment factor. In other words:

X = (X + X_offset) * X_factor

Y = (Y + Y_offset) * Y_factor

OUTPUT_COORDINATE_SYSTEM (optional, parameter list)

The output coordinate system determines how the internal Latitude and Longitude values are converted into X-Y coordinates in the output ArcView shape file. It is only needed if coordinate conversions are desired and then only if the output coordinates are not in degrees of Latitude and Longitude. If both the input coordinate system and the output coordinate system keys are NULL, no coordinate conversion takes place. The output coordinates will be the same as the input coordinates. In TRANSIMS, this means that the output ArcView shape file will be in UTM coordinates and meters.

The output coordinate command includes three parts separated by a comma. The first part is the coordinate system description. The options include UTM, STATEPLAN, and LATLONG. The second part identified the code number within the coordinate system that relates to the local conversion parameters. For UTM coordinates these codes range from 1N to 23N. Stateplane coordinates are defined using four digit FIPS codes (e.g., Oregon North = 3601). A code is not needed for the Latitude/Longitude system. The third parameter defines the coordinate units. By default, UTM is in meters, Stateplane is in feet, and Latitude/Longitude is in degrees. The user can override these assumptions using the following keywords: FEET, METERS, MILES, KILOMETERS, DEGREES, and MILLION_DEGREES.

When this key is provided, a projection file (*.prj) is created for each new ArcView shape file

OUTPUT_COORDINATE_ADJUSTMENT (optional, parameter list)

The output coordinate adjustment enables the user to manipulate the coordinates after they are returned from the output coordinate conversion calculation. It is only needed if the output coordinates should be in units that are different from the conversion algorithm.

The adjustment command includes four floating-point numbers separated by commas. The first two numbers are the X and Y offsets. The last two numbers are X and Y adjustment factors. The process adds the offset value to the coordinate and then applies the adjustment factor. In other words:

X = (X + X_offset) * X_factor

Y = (Y + Y_offset) * Y_factor

OUTPUT_XYZ_SHAPES (optional, flag, FALSE)

By default, the ArcView shape files are generated with X and Y coordinates. If this key is specified as TRUE, the output shape file will be constructed with X, Y, and Z coordinates. (The ArcView shape file will also include M (measure) values and each M value will be equal to the corresponding Z value). If the TRANSIMS network does not include Z coordinates, the output Z coordinates will be zero.

OUTPUT_XYM_SHAPES (optional, flag, FALSE)

By default, the ArcView shape files are generated with X and Y coordinates. If this key is specified as TRUE, the output shape file will be constructed with X, Y, and M coordinates where the M (measure) value will be equal to the Z coordinate in the TRANSIMS network. If this key is TRUE, it overrides the OUTPUT_XYZ_SHAPES key. This file structure generates a smaller shapefile than the XYZ structure.

User Programs (version 5.x)

The data processing available through a user program is based on a robust programming syntax that is a cross between the ‘C’ and FORTRAN languages. The user script is first compiled into an efficient, stack-based command structure that is then executed each time the main program calls the routine. Most user programs are designed to read from and write to specific data fields within a pre-defined set of data files. The main program typically manages the file input and output and the user program performs calculates based on data records from the files.

User programs can be implemented in both Windows and Linux executables. In a Windows program all of the syntax commands and variables are not case sensitivity (e.g., “a1” is the same as “A1”). A Linux executable is case sensitivity. The syntax keywords and functions must be defined with all-caps in a Linux script. The case for variable and field names is set when they are first defined.

Variables

All variables included in the user-provided data files are automatically defined and available for use. The user program operates on fields in the current data records using the syntax:

```
File_Name.Field_Name
```

For example, the main program defines a user file named “NODE” and the first line of this file or the associated Definition file defines the following field names:

```
X, Y, TAZ, AREA_TYPE
```

The user can reference the fields in the current data record using the following syntax:

```
NODE.X
NODE.Y
NODE.TAZ
NODE.AREA_TYPE
```

In addition to data records, the user can define any number of local variables. Local variables are defined using declaration statements. Three types of variables can be declared. These include integers, decimal numbers, and strings. The declaration syntax is:

```
INTEGER  xxx, x1..x20, Y_100, z  ENDDEF
REAL  R10..R30  ENDDEF
STRING  svalue  ENDDEF
```

Where “xxx, x1..x20, Y_100, z”, “R10..R30”, and “svalue” represent the variable names defined by the user. Variable names must start with a letter and may include numbers and underscores (i.e., “_”). The list of variable names can continue onto multiple lines of the program. The declaration process continues until the “ENDDEF” keyword is reached.

As above examples demonstrate, variable ranges can be used. “x1..x20” will declare all variables between “x1” and “x20” as integers. The text parts of a range command must be identical. The number parts must increase from the lower value to the upper value.

Local variables are initialized to zero (or blank) before the first program execution, but maintain their current value as each record is processed. This enables the user to accumulate data for a given data field, or pass information from a previous record to a subsequent record. The software also maintains a pre-defined record counter ("RECORD") and a file group code ("GROUP") that the user program can reference. "RECORD" is set to one for the first execution and is automatically incremented for each subsequent execution. "GROUP" defaults to zero, but may be set to the group number by programs that use the same script to processing multiple file groups.

Arithmetic Operators

The purpose of a user program is to manipulate data fields, perform calculations, or implement conditional procedures. To facilitate these tasks, the user program provides a full set of arithmetic operators. These operators include:

Operator	Syntax	Result
+	value1 + value2	Sum of 'value1' and 'value2'
-	value1 - value2	Difference between 'value1' and 'value2'
*	value1 * value2	Product of 'value1' and 'value2'
/	value1 / value2	'Value1' divided by 'value2'
**	value1 ** value2	'Value1' raised to the power of 'value2'
%	value1 % value2	The remainder of 'value1' divided by 'value2'
-	-value	Changes the sign of 'value'
+	string1 + string2	Concatenates 'string2' to the end of 'string1'

The operators can be combined in any order into compound statements. Compound statements will use standard order of operation logic. This means that power, modula, and negative operators will be executed first, multiplication and division is second, and addition and subtraction last. The user can include parenthesis to modify the order of operation. For example, (t1 + t2) * (t3 + t4).

Sample equations include:

```
J = J + 10
Y = A * X ** 2 + B * X + C
S2 = S1 + " MODE"
VMT = VMT + LINK.VOLUME * LINK.LENGTH
```

Logical Statements

In addition to arithmetic operators, a full set of logical operators is available. The operators can be defined using the syntax typically associated with 'C' and FORTRAN programming languages. The syntax can be mixed and matched in any way the user prefers. Parentheses are available for defining logical groups. The following describes the logical syntax:

Operator	Syntax	Result
&& AND	Logic1 && logic2 logic1 AND logic2	True if 'logic1' and 'logic2' are both True (or not zero)
|| OR	Logic1 || logic2 logic1 OR logic2	True if 'logic1' or 'logic2' are True (or not zero)
! NOT	! logic NOT logic	True if 'logic' is False (or zero)

== EQ	Value1 == value2 value1 EQ value2	True if 'value1' equals 'value2'
!= NE	Value1 != value2 value1 NE value2	True if 'value1' is not equal to 'value2'
<= LE	Value1 <= value2 value1 LE value2	True if 'value1' is less than or equal to 'value2'
>= GE	Value1 >= value2 value1 GE value2	True if 'value1' is greater than or equal to 'value2'
< LT	value1 < value2 value1 LT value2	True if 'value1' is less than 'value2'
> GT	value1 > value2 value1 GT value2	True if 'value1' is greater than 'value2'

When variables of different data types are used in arithmetic equations or logical statements, industry standard type conversion rules are used. If the statement includes decimal numbers, the arithmetic operators will use floating point methods. If a floating point statement is assigned to an integer variable, the decimal number is truncated to the next lower integer before the assignment is made.

In general, integer and decimal numbers cannot be combined with string variables in arithmetic or logical statements. The result of a logical statement can, however, be assigned to an integer variable. If the result is True, the integer will be set to 1. If the result if False, the integer will be set to zero. Conversely, a logical test can be based on a number or string. If the number is not zero, the condition is True. If the number is zero, the condition is False.

Sample logical statements include:

```
IF (J < 10) …
C1 = (HHOLD.INCOME < 10000)
IF (J GE 100 AND J LE 1000) …
WHILE (STAT) …
IF (!(D1.TAZ > 10 && D1.TAZ < 25) || J == 1) …
```

Conditional Processing

Logical statements are typically used to implement conditional processing. Two types of conditional processing are available within user programs. Branching procedures are implemented with IF-THEN-ELSE statements. Repetitive procedures are implemented with WHILE-LOOP statements. The software is designed to accommodate up to 100 nested IF statements and 10 nested WHILE statements. The logical keywords include:

Operator	Syntax	Result
IF	IF (logic)	If 'logic' is False (or zero), go to ELSE or ENDIF command
THEN	THEN …	End of logic statement and beginning of True statements
ELSE	… ELSE …	End of True statements and beginning of False statements
ENDIF	… ENDIF	End of True or False statements
WHILE	WHILE (logic)	If 'logic' is False (or zero), go to statement after ENDLOOP
LOOP or {	LOOP … or { …	End of logic statement and beginning of Loop statements
BREAK	BREAK	Go to statement after ENDLOOP command

ENDLOOP or }	… ENDLOOP or … }	End of Loop statements and go to WHILE command

The basic branching syntax is:

```
IF (logical_test) THEN [do-when-true] ELSE [do-when-false] ENDIF
```

The syntax can be provided on one line or multiple lines. The ELSE clause is optional, but an ENDIF is required for each IF statement. Up to 100 branching statements can be nested within each other to form complex conditional tests. Nested conditional processing would look like the following:

```
IF (t1 > 10) THEN
      IF (t1 > 15) THEN
            t2 = 15
      ELSE
            t2 = 10
      ENDIF
ELSE
      IF (t1 > 5) THEN
            t2 = 5
      ELSE
            t2 = 1
      ENDIF
ENDIF
```

The basic repetitive syntax is:

```
WHILE (logical test) LOOP [do-while-true] ENDLOOP
```

The syntax can be provided on one line or multiple lines. Note that the user is responsible for implementing the changes that will cause the logical test to eventually be False. In other words, the loop statements should set a status code or increment a counter that results in a False logic statement. The loop will continue as long as the logical test is True.

Up to 10 sets of repetitive statements can be nested within each other to form multi-dimensional loops. Each loop can include up to 10 BREAK statements to exit the loop without returning to the WHILE logic test. A BREAK statement, however, only has meaning if it is included within a branching statement. The following example demonstrates a nested loop with BREAK statements:

```
rec = 0

WHILE (1) LOOP
      rec = rec + 1
      stat = INPUT (A1, rec)

      IF (stat != 0) THEN BREAK ENDIF

      total = 0
      t1 = 1

      WHILE (t1 < 80) {
            data = READ (A1, t1, "%10", data)
```


AECOM

```
            IF (data == 0) THEN BREAK ENDIF

            t1 = t1 + 10
            total = total + data
        }
        PRINT ("\n\tRecord %", rec)
        PRINT (" Total = %", total)
ENDLOOP
```

Arithmetic Functions

User programs also provide a number of arithmetic functions. These functions return a value that can be used in standard calculations or logic statements. In most cases the return value is a decimal number. These functions include:

Functions	Syntax	Result
MIN	MIN (value1, value2)	Minimum of 'value1' and 'value2'
MAX	MAX (value1, value2)	Maximum of 'value1' and 'value2'
ABS	ABS (value)	Absolute value of 'value'
SQRT	SQRT (value)	Square root of 'value'
EXP	EXP (value)	The exponential of 'value'
LOG	LOG (value)	The natural logarithm of 'value'
POWER	POWER (value1, value2)	'Value1' raised to the 'value2' power
LOG10	LOG10 (value)	Base 10 logarithm of 'value'
RANDOM	RANDOM ()	Random number between 0 and 1

A few example function applications are shown below:

```
C1 = MIN (C2, C3 - 30)
U1 = EXP (-0.0225 * OD1.TTIME + 1.2008)
IF (ABS (D1) > 100) …
PROB = RANDOM ()
R1 = LOG (OD1.TDIST)
```

Conversion Functions

In addition to arithmetic functions, the software provides a full set of type conversion functions. These functions typically convert data between string fields and integer or decimal numbers. They include:

Functions	Syntax	Result
ATOI	ATOI (value)	The string 'value' converted to an integer number
ATOF	ATOF (value)	The string 'value' converted to a decimal number
ITOA	ITOA (value)	The integer 'value' is converted to a string
FTOA	FTOA (value)	The decimal number 'value' is converted to a string
INT	INT (value)	'Value' truncated to an integer
FLOAT	FLOAT (value)	'Value' changed to a decimal number
ROUND	ROUND (value)	'Value' rounded to the nearest integer
SUBSTR	SUBSTR (string, i1, i2)	The substring of 'string' between characters 'i1' and 'i2'

TRIM	TRIM (string)	Removes leading and trailing blanks from a string
LOW	LOW (string)	Low string in a string range
HIGH	HIGH (string)	High string in a string range
TTOI	TTOI (value)	The Dtime 'value' converted to an integer
TTOF	TTOF (value)	The Dtime 'value' converted to a decimal number
ITOT	ITOT (value)	The integer 'value' is converted to a Dtime
FTOT	FTOT (value)	The decimal number 'value' is converted to a Dtime
ATOT	ATOT (value)	The string 'value" converted to a Dtime

A few conversion examples are shown below:

```
I1 = ATOI (string)
S1 = "NUMBER = " + FTOA (number)
J = INT (number + 0.5)
S = SUBSTR (string, 5, 10)
IF (ROUND (number) > 10) …
R1 = 10.5 + i1 / FLOAT (total)
```

Date-Time Functions

Date and time fields often require special processing for logic statements or conversion programs. User programs work with date and time data in "time stamp" format. A "time stamp" is the standard computer representation of a date and time. The data are stored as the integer number of seconds from January 1, 1970. Functions are provided to give the user access to the year, month, day, hour, minute, and second components of this data type. The following table describes the behavior of these functions:

Functions	Syntax	Result
DOW	DOW (time_stamp)	The day-of-the-week (0-6) from the 'time_stamp'
HOUR	HOUR (time_stamp)	Decimal hour (0.00-23.99) from the 'time_stamp'
MONTH	MONTH (time_stamp)	Integer month (1-12) from the 'time_stamp'
YEAR	YEAR (time_stamp)	Integer year (1970+) from the 'time_stamp'
DATE	DATE (year, month, day)	The time stamp equal to the date components
DATE_TIME	DATE_TIME (year, month, day, hour, minute, second)	The time stamp equal to the date and time components
DATE_STR	DATE_STR (time_stamp)	The 'mm/dd/yyyy' string equal to 'time_stamp'
TIME_STR	TIME_STR (time_stamp)	The 'hh:mm:ss' string equal to 'time_stamp'

A few date-time examples are shown below:

```
IF (YEAR (COUNT.DATE) == 1999) …
S1 = "DATE = " + DATE_STR (d1)
D1 = DATE (2003, 11, 25)
IF (DOW (d1) > 0 && DOW (d1) < 6) …
T1 = HOUR (d1) * 3600.0
```

Input-Output Functions

User programs provide a few input and output functions that enable the user to manipulate data files directly, generate reports, or display debug information. These functions are listed below:

Functions	Syntax	Result
PRINT	PRINT (format, value)	Send the formatted 'value' to the report file
LIST	LIST (format, value)	Send the formatted 'value' to the screen
FORMAT	FORMAT (format, value)	A string with the formatted 'value'
READ	READ (file, column, format, value)	The 'value' read from the ASCII 'file' starting at 'column' using 'format' and 'value' data type
WRITE	WRITE (file, column, format, value)	The column after 'value' is written to the ASCII 'file' starting at 'column' using 'format'
INPUT	INPUT (file, record)	The status code from reading 'record' from 'file'
OUTPUT	OUTPUT (file, record)	The status code from writing 'record' to 'file'

The "format" parameter in the input-output functions enables the user to control how multiple statements interact with one another. Each message adds to the end of the previous message until a new line command is included in the format string. The syntax of the format string generally follows the 'C' programming language for a "printf" statement. The primary difference is that the functions operate on only one variable at a time and the data type of that variable is supplied by the program or the user.

The placement of the data variable within the format string is defined with a percent sign (%). The data item may be preceded and followed by a text message. If the user wishes to control the size or format of the data item, the field width and the number of decimal places will follow the percent sign. The following shows a few sample output messages:

```
PRINT  ("%", t1)
PRINT  ("Record Number = %5\n", t0)
PRINT  ("\n\t%10.5", r1)
PRINT  ("\n\nStreet Name: %40  \nUser ID:", ArcView.StreetName)
```

The first message prints a single integer on the report file. The second message prints a message followed by a five-digit integer number and a new line command ("\n"). The new line command can be included at the end of the message, at the beginning of the message, or in the middle of the message. The third message shows the new line command at the beginning of the message followed by a tab command ("\t"). The decimal variable will be displayed using a maximum of ten digits and five decimal places. The fourth message uses double spaces before printing the label and 40 characters of a string variable. It then starts a new line and prints a label message in anticipation of the next PRINT command. This message prints the variable called "StreetName" from a user-provided data file named "ArcView".

As implemented with the FORMAT command, string formatting can also be used to convert numbers to strings or truncate strings. For example:

```
S1 = FORMAT ("%10.1", R1)
S3 = FORMAT ("%10", S2)
```

The first example converts a decimal number to a string with one decimal point. The second example saves the first 10 characters from "S2" in "S3".

READ and WRITE are designed to process ASCII files that don't contain fields or a field header. These functions enable the user to read or write information from the data record using column positions and field widths. For example:

```
R2 = READ (A1, 20, "%10", R1) + 100
C1 = WRITE (A2, C1, "%,", R3)
```

The first example reads 10 characters starting at column 20 from a file named "A1", converts this data to a variable of type "R1", adds 100 to the result, and assigns the sum to the variable "R2". The second example writes the variable "R3" followed by a comma to a file named "A2" starting at column "C1". The number of characters written depends on the data type of "R3" and the value of "R3". The WRITE function returns the column number after the data are written to the record. This column number can be used in subsequent WRITE statements to append additional data to the end of the record.

INPUT and OUTPUT are designed to read and write records to a file. In most user program applications the main program controls the file input and output. The main program reads the records in the file one at a time and executes the user program to manipulate the data fields in the current record. When the execution is finished, the main program writes the results to the output file and reads the next record from the input file.

In certain situations, like unformatted ASCII files or multi-line custom reports, the user may want direct control over data input and output. The INPUT and OUTPUT functions enable the user to read and write records to the file. For example:

```
stat = 0
record = 0

WHILE (stat EQ 0) LOOP
      record = record + 1
      IF (INPUT (A1, record)) THEN BREAK ENDIF
      A2 = A1
      stat = OUTPUT (A2, record)
ENDLOOP
```

This example reads records from file "A1" until an end-of-file (non-zero return code) is found or the OUTPUT command returns an error (non-zero return code). Each record from file "A1" is read sequentially, copied to the data record for file "A2", and file "A2" is written to the disk.

Lookup Tables

The software also supports integer, decimal, and string lookup tables using the following syntax:

Functions	Syntax	Result
ITABn	ITABn (index)	the integer at 'index' in integer lookup table 'n'
RTABn	RTABn (index)	the number at 'index' in decimal lookup table 'n'
STABn	STABn (index)	the string at 'index' in string lookup table 'n'
ENDTAB	tTABn (max) = … ENDTAB	end lookup table initialization

Up to 10 sets of lookup tables can be defined. The tables are numbered 0 to 9 using keywords ITAB, RTAB, and STAB (e.g., ITAB2, RTAB5, STAB0). If the table keyword is specified without a number (e.g., ITAB), it is interpreted as table zero.

Lookup tables must be defined before they are used. The table is defined the first time the variable name is encountered. The user program interprets everything following the table name until an "ENDTAB" is reached as table initialization data. The initialization data may span any number of command records. The initialization syntax is:

```
tTABn (max) = x, y, z, … ENDTAB
```

where: `t` = table type (I, R, or S)

`n` = table number (blank, 0..9)

`max` = the maximum number of elements in the table

`x, y, z` = comma delimited table values of type 't' started at element 1.

The following example demonstrates how a table is defined and used:

```
ITAB1 (10) = 1, 2, 3, 4, 5,
             6, 7, 8, 9, 10  ENDTAB

TOTAL = 0
T1 = 0

WHILE (T1 < 10) LOOP
      T1 = T1 + 1
      TOTAL = TOTAL + ITAB1 (T1) * 10
ENDLOOP
```

Data Maps

The software supports data structures that map an index value to a return value. These include tables that map an integer to a decimal number, an integer to a string, and a string to an integer.

Functions	Syntax	Result
IR_MAPn	IR_MAPn (index)	the decimal number at 'index' in data map 'n'
IS_MAPn	IS_MAPn (index)	the string at 'index' in data map 'n'
SI_MAPn	SI_MAPn (string)	the integer at 'string' in data map 'n'
ENDMAP	ix_MAPn = i, x, … ENDMAP	end data map initialization

Up to 10 sets of data maps can be defined. The maps are numbered 0 to 9 using keywords IR_MAP, IS_MAP, and SI_MAP (e.g., IR_MAP2, IS_MAP5, SI_MAP0). If the map keyword is specified without a number (e.g., IR_MAP), it is interpreted as table zero.

Data maps must be defined before they are used. The map is defined the first time the variable name is encountered. The user program interprets everything following the map name until an "ENDMAP" is reached as map initialization data. The initialization data may span any number of command records. The initialization syntax is:

```
ix_MAPn = i, x, … ENDMAP
```

where: ix = map type (IR, IS, or SI)

n = map number (blank, 0..9)

i = the index value for the map (integer or string)

x = the value returned by the index (decimal, string, or integer).

The following example demonstrates how a table is defined and used:

```
SI_MAP1 = "FREEWAY",   1,
          "MAJOR",     2,
          "MINOR",     3,
          "COLLECTOR", 4,
          "LOCAL",     5
ENDMAP

NEWLINK.FUNC = SI_MAP1 (LINK.FTYPE)
```

Return Values

A user program returns to the calling program when a RETURN or END statement is encounters. The END statement terminates the process with a return value of zero. If the user needs to return information to the calling program, the RETURN function should be used. The result of a RETURN command is always an integer value. The syntax is:

RETURN (…)

Any type of calculation can be included within the parentheses, but the result will be interpreted as an integer.

Reports

The user program interface provides two types of printout reports. These are typically identified in the program report list as a "SCRIPT" report and a "STACK" report. A script report prints a copy of the commands sent to the compiler to the printout for the main program. These reports typically look like the following:

```
Household Type Script

      IF (Household.Persons > 2) THEN
           IF (Household.Persons > 3) THEN
                RETURN (13)
           ELSE
                IF (Household.AgeLT5 == 1) THEN
                        RETURN (12)
                ELSE
                      IF (household.Age5to17 == 1) THEN
                           RETURN (11)
                      ELSE
                           RETURN (10)
                      ENDIF
                ENDIF
```

```
          ENDIF
     ELSE
          IF (Household.Persons == 2) THEN
               IF (Household.AgeLT5 == 1) THEN
                    RETURN (6)
               ELSE
                    RETURN (5)
               ENDIF
          ELSE
               RETURN (3)
          ENDIF
     ENDIF
```

The stack report lists the compiled commands as they are executed by the user program. This represents sequence of steps the program uses to process the information. It looks similar to the order of operation used by "reverse polish" calculators. The stack for the script show above is as follows:

```
Household Type Stack

  1) Integer      Household.PERSONS
  2) Integer      2
  3) Relation     GT
  4) Logical      If False, Jump to 29
  5) Integer      Household.PERSONS
  6) Integer      3
  7) Relation     GT
  8) Logical      If False, Jump to 12
  9) Integer      13
 10) Return       Integer
 11) Logical      Jump to 28
 12) Integer      Household.AGELT5
 13) Integer      1
 14) Relation     EQ
 15) Logical      If False, Jump to 19
 16) Integer      12
 17) Return       Integer
 18) Logical      Jump to 28
 19) Integer      Household.AGE5TO17
 20) Integer      1
 21) Relation     EQ
 22) Logical      If False, Jump to 26
 23) Integer      11
 24) Return       Integer
 25) Logical      Jump to 28
 26) Integer      10
 27) Return       Integer
 28) Logical      Jump to 45
 29) Integer      Household.PERSONS
 30) Integer      2
 31) Relation     EQ
 32) Logical      If False, Jump to 43
```

```
33) Integer      Household.AGELT5
34) Integer      1
35) Relation     EQ
36) Logical      If False, Jump to 40
37) Integer      6
38) Return       Integer
39) Logical      Jump to 42
40) Integer      5
41) Return       Integer
42) Logical      Jump to 45
43) Integer      3
44) Return       Integer
45) End
```

LineSum (version 5.x.17)

The LineSum program is used to:

1. Merge TPPlus transit ridership files by time period and daily total.
2. Generate line ridership reports showing boardings, alightings, and riders by time period and stop given multiple sets of lines.
3. Generate link ridership reports showing riders by mode, line, direction, and time period for selected modes, lines, and links.
4. Generate stop reports showing riders boarding and alighting by time period, mode and line for selected stops and modes.
5. Generate node access reports showing riders arriving and departing by time period, mode and node for selected stops and modes.
6. Generate stop reports showing the riders that arrive, board, alight, and depart by mode and line for selected stops, modes, and lines.
7. Generate total ridership reports showing total boarding, maximum load points, and passenger miles and hours of travel by line.
8. Create output data files for the ridership summarizes generated by stop, access, and total ridership reports.
9. Create link riders files in text or ArcGIS format that summarized the combined ridership on links from selected modes and lines.
10. Link rider files include peak and offpeak hours of service, peak hour factors, and mode capacities to calculate performance statistics such as passenger miles and hours of travel, vehicle miles and hours of travel, peak hour ridership, and load factors.
11. Use a link shape file to include shape coordinates in the ArcGIS version of the link rider file.
12. Report the differences between two route files with respect to peak and offpeak headways and run times and the number of stops on the route.

LineSum is a console-based program that runs in a command window on either Windows or Linux. The command syntax is:

LineSum [-flag] [control_file]

The control_file is the directory path and file name of a text file that contains the control strings expected by the program. If a file name is not provided, the program will prompt the user to enter a file name. The program automatically creates a printout file based on the control_file name. If the file name includes an extension (e.g., ".ctl"), the extension is removed and ".prn" is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

Additional information about program syntax, command lines flags, control keys and comments can be found in the Execution Service document.

A typical LineSum control file is shown below. These keys can be defined in a variety of different ways to perform the tasks listed above.

```
TITLE                         WMATA Ridership Summaries
PROJECT_DIRECTORY             ..\
DEFAULT_FILE_FORMAT           DBASE

#---- LineSum Control Keys ----

PEAK_RIDERSHIP_FILE           wmata\pk_vol.dbf
OFFPEAK_RIDERSHIP_FILE        wmata\op_vol.dbf

STOP_NAME_FILE                controls\Station_Names.txt
STOP_NAME_FORMAT              TAB_DELIMITED

LINE_REPORT_TITLE_1           BLUE ORANGE and RED Lines
LINE_REPORT_LINES_1           WMBLUA, WMORGA, WMREDA, WMREDB

LINE_REPORT_TITLE_2           GREEN Line: GREENBELT to BRANCH AVE
LINE_REPORT_LINES_2           WMGRNA

ACCESS_REPORT_TITLE           Metrorail Access Report
ACCESS_REPORT_STOPS           8014..8020
ACCESS_REPORT_MODES           ALL
ACCESS_REPORT_DETAILS         TRUE
NEW_ACCESS_REPORT_FILE        wmata\access.txt
NEW_ACCESS_REPORT_FORMAT      TAB_DELIMITED

STOP_REPORT_TITLE_1           Metro Center Station
STOP_REPORT_STOPS_1           8015
STOP_REPORT_MODES_1           ALL
STOP_REPORT_LINES_1           ALL

STOP_REPORT_TITLE_2           Gallery Place Station
STOP_REPORT_STOPS_2           8016
STOP_REPORT_MODES_2           ALL
STOP_REPORT_LINES_2           ALL

TOTAL_REPORT_TITLE            Total Ridership Report
TOTAL_REPORT_LINES            ALL
NEW_TOTAL_REPORT_FILE         wmata\total_lines.txt
NEW_TOTAL_REPORT_FORMAT       TAB_DELIMITED

NEW_LINK_RIDER_FILE           wmata\Metrorail_Ridership.txt
NEW_LINK_RIDER_FORMAT         TAB_DELIMITED
LINK_RIDER_MODES              3
LINK_RIDER_LINES              WM..WM|
LINK_RIDER_PEAK_FACTOR        5.0
LINK_RIDER_PEAK_CAPACITY      8.0
```

Execution Service Keys

The LineSum program inherits the following control keys from Execution Service.

TITLE (optional, text)

REPORT_DIRECTORY (optional, path)

REPORT_FILE (optional, output file)

REPORT_FLAG (optional, flag, FALSE)

PAGE_LENGTH (optional, integer, 65, >= 0)

PROJECT_DIRECTORY (optional, text)

DEFAULT_FILE_FORMAT (optional, text, TAB_DELIMITED)

Projection Service Keys

The LineSum program supports the following coordinate project control keys.

INPUT_COORDINATE_SYSTEM (optional, text)

INPUT_COORDINATE_ADJUSTMENT (optional, path)

OUTPUT_COORDINATE_SYSTEM (optional, output file)

OUTPUT_COORDINATE_ADJUSTMENT (optional, flag, FALSE)

OUTPUT_XYZ_SHAPES (optional, text)

OUTPUT_XYM_SHAPES (optional, text)

Input-Output Data Keys:

The following keys identify the input data files used to generate the reports and enable the user to create output data files that contain the aggregate data from multiple input files.

PEAK_RIDERSHIP_FILE_# (optional, input file)

This key specifies the name and location of one or more input data files that will be read by the program to generate the peak period ridership. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file contains the boarding, alighting, and rider data for each link on a given set of transit lines.

PEAK_RIDERSHIP_FORMAT_# (optional, text, DBASE)

If the Peak Ridership file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

OFFPEAK_RIDERSHIP_FILE_# (optional, input file)

This key specifies the name and location of one or more input data files that will be read by the program to generate the offpeak period ridership. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file contains the boarding, alighting, and rider data for each link on a given set of transit lines.

OFFPEAK_RIDERSHIP_FORMAT_# (optional, text, DBASE)

If the Offpeak Ridership file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

NEW_PEAK_RIDERSHIP_FILE (optional, output file)

This key specifies the name and location of a new data file that will be created by the program to store the combined peak period ridership. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file will contain the boarding, alighting, and rider data for each link and transit line found in the input peak ridership files.

NEW_PEAK_RIDERSHIP_FORMAT (optional, text, DBASE)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

NEW_OFFPEAK_RIDERSHIP_FILE (optional, output file)

This key specifies the name and location of a new data file that will be created by the program to store the combined offpeak period ridership. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file will contain the boarding, alighting, and rider data for each link and transit line found in the input offpeak ridership files.

NEW_OFFPEAK_RIDERSHIP_FORMAT (optional, text, DBASE)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

NEW_TOTAL_RIDERSHIP_FILE (optional, output file)

This key specifies the name and location of a new data file that will be created by the program to store the combined peak and offpeak ridership. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file will contain the boarding, alighting, and rider data for each link and transit line found in the input peak and offpeak ridership files.

NEW_TOTAL_RIDERSHIP_FORMAT (optional, text, DBASE)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

STOP_NAME_FILE (optional, input file)

This key specifies the name and location of an input data file that will be read by the program to associate station names with stop node numbers. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file contains the stop number followed by a name string.

STOP_NAME_FORMAT (optional, text, TAB_DELIMITED)

If the Stop Name file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

Line Report Keys

The following keys control the generation of one or more line ridership reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

LINE_REPORT_TITLE_# (optional, text, Line Report)

Any text string can be used on this line. This text is printed on the top of the report header.

LINE_REPORT_LINES_# (optional, list, ALL)

This key contains a list of transit route names that will be aggregated into a combined ridership profile. The default value "ALL" means that every line is included. The names may be comma, space or tab delimited and the list may include name ranges. A name range starts the characters that define the beginning of the range followed by two periods and the characters that define the end of the range. For example, ABC..DEF and MW..MW|. The first range includes all lines that start with ABC and are less than or equal to DEF. The second string demonstrates a convenient method of referring to all lines that start with MW. Since the | character is greater than any other number, character or symbol, it captures all possible names in the range.

LINE_REPORT_MODES_# (optional, list, ALL)

This key contains a list of transit modes that when combined with the line list define the selection criteria for the report. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

LINE_REPORT_ALL_NODES_# (optional, flag, FALSE)

The line ridership profile lists the boardings, alightings, and riders at each stop along the line or line branch. By default, the report only shows nodes where the route stops in the peak or offpeak period. If this key is set to TRUE, the report will include non-stop nodes in the node sequence.

Link Report Keys

The following keys control the generation of one or more link ridership reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

LINK_REPORT_TITLE_# (optional, text, Link Report)

Any text string can be used on this line. This text is printed on the top of the report header.

LINK_REPORT_LINKS_# (optional, list)

This key must contain a list of node pairs for the link report to be generated. A node pair is coded as two node numbers separated by a dash (e.g., 100-200). These nodes represent the A-node and B-node or B-node and A-node of a link in the transit network. A list of sequential links can also be specified by appending addition nodes to the node pair (e.g., 300-400-500).

LINK_REPORT_MODES_# (optional, list, ALL)

This key contains a list of transit modes that when combined with the link and line lists define the selection criteria for the report. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

LINK_REPORT_LINES_# (optional, list, ALL)

This key contains a list of transit route names that when combined with the link and mode lists define the selection criteria for the report. The default value "ALL" means that every line is eligible for selection. The names may be comma, space or tab delimited and the list may include name ranges. A name range starts the characters that define the beginning of the range followed by two periods and the characters that define the end of the range. For example, ABC..DEF and MW..MW|. The first range includes all lines that start with ABC and are less than or equal to DEF. The second string demonstrates a convenient method of referring to all lines that start with MW. Since the | character is greater than any other number, character or symbol, it captures all possible names in the range.

LINK_REPORT_ONEWAY_# (optional, flag, FALSE)

The link report lists the peak and offpeak ridership on each line that uses at least one of the specified links. By default, the report summarizes ridership in both directions of the link. If this key is set to TRUE, the report only summarizes the ridership in the direction of travel represented by the node pair (i.e., the from-to direction).

On-Off Report Keys

The following keys control the generation of one or more stop boarding and alighting reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

ON_OFF_REPORT_TITLE_# (optional, text, Access Report)

Any text string can be used on this line. This text is printed on the top of the report header.

ON_OFF_REPORT_STOPS_# (optional, list, ALL)

This key contains a list of transit stops were boarding and alighting information is summarized. The default value "ALL" means that every stop is included. The stop numbers may be comma, space, or tab delimited and may include stop ranges. A stop range includes the first stop and last stop numbers separated by two periods (e.g., 300..500).

ON_OFF_REPORT_MODES_# (optional, list, ALL)

This key contains a list of transit modes that when combined with the stop list define the selection criteria for the report. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

ON_OFF_REPORT_DETAILS_# (optional, flag, FALSE)

By default, the on-off report lists the total boardings and alightings by time period for each selected stop. If this key is TRUE, the report shows a boardings and alightings for each transit line that serves the stop. If this key is MODE, the report shows the sum of the boardings and alightings for each transit mode that serves the stop.

NEW_ON_OFF_REPORT_FILE_# (optional, output file)

In addition to creating a report in the printout file, the program can save the summary results to a formatted data file for additional processing and analysis. This key specifies the name and location of the data file. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

NEW_ON_OFF_REPORT_FORMAT_# (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

Access Report Keys

The following keys control the generation of one or more node access reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

ACCESS_REPORT_TITLE_# (optional, text, Access Report)

Any text string can be used on this line. This text is printed on the top of the report header.

ACCESS_REPORT_STOPS_# (optional, list, ALL)

This key contains a list of transit stops were access information is summarized. The default value "ALL" means that every stop is included. The stop numbers may be comma, space, or tab delimited and may include stop ranges. A stop range includes the first stop and last stop numbers separated by two periods (e.g., 300..500).

ACCESS_REPORT_MODES_# (optional, list, ALL)

This key contains a list of access modes that when combined with the stop list define the selection criteria for the report. The default value "ALL" means that any access mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 13..15).

ACCESS_REPORT_DETAILS_# (optional, flag, FALSE)

By default, the access report lists the total arrivals and departures by time period for each selected stop. If this key is TRUE, the report shows the arrivals and departures from each node connected to the stop. If this key is MODE, the report shows the sum of the arrivals and departures for each access mode connected to the stop.

NEW_ACCESS_REPORT_FILE_# (optional, output file)

In addition to creating a report in the printout file, the program can save the summary results to a formatted data file for additional processing and analysis. This key specifies the name and location of the data file. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

NEW_ACCESS_REPORT_FORMAT_# (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

Stop Report Keys

The following keys control the generation of one or more stop activity reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

STOP_REPORT_TITLE_# (optional, text, Stop Report)

Any text string can be used on this line. This text is printed on the top of the report header.

STOP_REPORT_STOPS_# (optional, list)

This key must contain a list of stops for the stop activity report to be generated. The stop numbers may be comma, space, or tab delimited and may include stop ranges. A stop range includes the first stop and last stop numbers separated by two periods (e.g., 300..500).

STOP_REPORT_MODES_# (optional, list, ALL)

This key contains a list of transit and access modes that when combined with the stop and line lists define the selection criteria for the report. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

STOP_REPORT_LINES_# (optional, list, ALL)

This key contains a list of transit route names that when combined with the stop and mode lists define the selection criteria for the report. The default value "ALL" means that every line is eligible for selection. The names may be comma, space or tab delimited and the list may include name ranges. A name range starts the characters that define the beginning of the range followed by two periods and the characters that define the end of the range. For example, ABC..DEF and MW..MW|. The first range includes all lines that start with ABC and are less than or equal to DEF. The second string demonstrates a convenient method of referring to all lines that start with MW. Since the | character is greater than any other number, character or symbol, it captures all possible names in the range.

STOP_REPORT_TRANSFERS_# (optional, list, NONE)

This key defines the mode numbers that will be used when estimating transfers on tunnel links between near-by stations. For modes included in this list, the transfer calculation is based on the arrivals plus departures divided by two. By default all arrivals and departures made using access modes are assumed to enter the station nodes from outside of the node group.

Total Report Keys

The following keys control the generation of one or more line summary reports. All of the keys with the same number extension (_#) are processed as a group. A separate report is generated for each group.

TOTAL_REPORT_TITLE_# (optional, text, Total Report)

Any text string can be used on this line. This text is printed on the top of the report header.

TOTAL_REPORT_LINES_# (optional, list, ALL)

This key contains a list of transit route names that will be aggregated into a ridership summary. The default value "ALL" means that every line is included. The names may be comma, space or tab delimited and the list may include name ranges. A name range starts the characters that define the beginning of the range followed by two periods and the characters that define the end of the range. For example, ABC..DEF and MW..MW|. The first range includes all lines that start with ABC and are less than or equal to DEF. The second string demonstrates a convenient method of referring to all lines that start with MW. Since the |

character is greater than any other number, character or symbol, it captures all possible names in the range.

TOTAL_REPORT_MODES_# (optional, list, ALL)

This key contains a list of transit modes that when combined with the line list define the selection criteria for the report. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

NEW_TOTAL_REPORT_FILE_# (optional, output file)

In addition to creating a report in the printout file, the program can save the summary results to a formatted data file for additional processing and analysis. This key specifies the name and location of the data file. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

NEW_TOTAL_REPORT_FORMAT_# (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, SQLITE3, and VERSION3.

Link Ridership Keys

The following keys control the generation of one or more transit link performance files. All of the keys with the same number extension (_#) are processed as a group. A separate file is created for each group.

NEW_LINK_RIDER_FILE_# (optional, output file)

This key specifies the name and location of a new link performance file that will be created by the program. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file will contain the boarding, alighting, rider and performance data for each link and transit line specified by the selection criteria.

NEW_LINK_RIDER_FORMAT_# (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

LINK_RIDER_MODES_# (optional, list, ALL)

This key contains a list of transit modes that when combined with the line list define the selection criteria for the output file. The default value "ALL" means that any mode is eligible for selection. The mode numbers may be comma, space, or tab delimited and may include mode ranges. A mode range includes the first mode and last mode numbers separated by two periods (e.g., 3..5).

LINK_RIDER_LINES_# (optional, list, ALL)

This key contains a list of transit route names that when combined with the mode list define the selection criteria for the output file. The default value "ALL" means that every line is eligible for selection. The names may be comma, space or tab delimited and the list may include name ranges. A name range starts the characters that define the beginning of the range followed by two periods and the characters that define the end of the range. For example, ABC..DEF and MW..MW|. The first range includes all lines that start with ABC and are less than or equal to DEF. The second string demonstrates a convenient method of referring to all lines that start with MW. Since the | character is greater than any other number, character or symbol, it captures all possible names in the range.

LINK_RIDER_PEAK_HOURS_# (optional, decimal, 6.0, 1.0..10.0)

This key defines the number of hours of services that are used to calculate the vehicle miles or hours of travel statistics. The Peak Ridership file includes the frequency of service expressed as the number of minutes between buses or trains. The number of peak period runs made by a transit route is the value of this key times 60 minutes in an hour divided by the line frequency. The number of runs times the route length and travel time is saved in the PK_VMT and PK_VHT fields.

LINK_RIDER_PEAK_FACTOR_# (optional, decimal, 1.0, 1.0..10.0)

The peak factor is used to estimate the number of riders that use the link during the peak hour. The peak hour riders are calculated as the maximum peak period ridership in the AB or BA directions divided by this factor. The value is typically slightly less than the number of peak hours (e.g., 5 vs. 6).

LINK_RIDER_PEAK_CAPACITY_# (optional, decimal, 1.0, 1.0..1000.0)

The peak capacity value is used in calculating the load factor field. It may represent the number of cars per train, the number of seats per vehicle, or the total passenger capacity of the vehicle. The default value of 1.0 generates the number of riders per vehicle run during the peak hour. The load factor is calculated as the maximum of the number of riders per run in the peak period in the AB or BA directions divided by the peak factor and the peak capacity.

LINK_RIDER_OFFPEAK_HOURS_# (optional, decimal, 10.0, 1.0..20.0)

This key defines the number of hours of services that are used to calculate the vehicle miles or hours of travel statistics. The Offpeak Ridership file includes the frequency of service expressed as the number of minutes between buses or trains. The number of offpeak period runs made by a transit route is the value of this key times 60 minutes in an hour divided by the line frequency. The number of runs times the route length and travel time is saved in the OP_VMT and OP_VHT fields.

LINK_RIDER_XY_FILE_# (optional, input file)

This key specifies the name and location of an input data file that will be read by the program if the output format for the link rider file is specified as ARCVIEW. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file contains the X and Y coordinates of the transit network nodes.

LINK_RIDER_XY_FORMAT_# (optional, text, TAB_DELIMITED)

If the Link Rider XY file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

LINK_RIDER_SIDE_OFFSET_# (optional, decimal, 0 feet, 0..1000 feet)

If this key is greater than zero, the shape vector used to draw the transit links in each direction are offset from the link centerline by the specified distance.

Link Shape Keys

The following keys can be used to add link shape coordinates to the ArcGIS version of each link rider file.

LINK_SHAPE_FILE (optional, input file)

This key specifies the name and location of an input Arcview shape file that will be used to add shapes to the links included in each link rider file that is created in ArcGIS format. The file name needs to include the *.shp extension and include coordinates that are compatible with the node coordinates provided in the LINK_RIDER_XY_FILEs. The shape file also needs to include A-node and B-node numbers that match the transit stop numbers. If an A-node to B-node or B-node to A-node index is not found in the Arcview shape file, the program will draw a straight link between the node coordinates. If the link matches the B-node to A-node index, the shape coordinates are entered in reverse order in the link rider file.

LINK_SHAPE_ANODE (optional, text, ANODE)

This key specifies the field name in the link shape file that stores the A-node number for the link shape.

LINK_SHAPE_BNODE (optional, text, BNODE)

This key specifies the field name in the link shape file that stores the B-node number for the link shape.

Link Shape Keys

The following keys can be used to add link shape coordinates to the ArcGIS version of each link rider file.

SERVICE_FILE (optional, input file)

This key specifies the name and location of an optional service file that can be used to override the default service level data for specific transit routes. It is most frequently used to define the number of cars or seat capacity for trains in the load factor calculations. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

SERVICE_FORMAT (optional, text, TAB_DELIMITED)

If the service file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

SERVICE_LINE_FIELD (optional, text, LINE)

This key specifies the field name in the service file that stores the line name for the service data.

SERVICE_PEAK_FIELD (optional, text, PEAK)

This key specifies the field name in the service file that stores the peak period service data.

SERVICE_OFFPEAK_FIELD (optional, text, OFFPEAK)

This key specifies the field name in the service file that stores the offpeak service data.

Route Difference Keys

The following keys control the generation of one or more route difference reports. All of the keys with the same number extension (_#) are processed as a group. A separate difference report is generated for each group.

BASE_ROUTE_FILE_# (optional, input file)

This key specifies the name and location of an input route file that will be compared to the Alternative Route file to identify differences in the peak and offpeak period headways, travel times, and stop nodes. This file is typically created by running the NetPrep program to merge the peak and offpeak Cube route files into a composite route file and then the ArcNet program is used to summarize the route nodes. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

BASE_ROUTE_FORMAT_# (optional, text, DBASE)

If the Base Route file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

ALTERNATIVE_ROUTE_FILE_# (optional, input file)

This key specifies the name and location of an input route file that will be compared to the Base Route file to identify differences in the peak and offpeak period headways, travel times, and stop nodes. This file is typically created by running the NetPrep program to merge the peak and offpeak Cube route files into a composite route file and then the ArcNet program is used to summarize the route nodes. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

ALTERNATIVE_ROUTE_FORMAT_# (optional, text, DBASE)

If the Alternative Route file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, SQLITE3, and VERSION3.

Report Options:

LINESUM_REPORT_# (optional, text)

This key is typically used to select reports written to the printout file. The "#" at the end of the report keyword represents the report number (e.g., LINESUM_REPORT_1). The key can be provided with additional numbers to specify additional reports. The reports are typically generated in numerical order (i.e., 1, 2, 3…). Since LineSum is a special purpose report generation program, these keys are not required to generate output reports.

The string parameter associated with a report keyword is limited to the following options:

LINE_REPORT
LINK_REPORT
ON_OFF_REPORT
ACCESS_REPORT
STOP_REPORT
TOTAL_REPORT
DIFFERENCE_REPORT

The above reports are printed in the "*.prn" file that is generated in the same directory as the control file used to run the program. Each of above reports is described below:

LINE REPORT

The line report summarizes the boardings, alightings, and ridership for an individual line or the combined ridership of lines that serve the same links. The report is controlled by the control keys that start with LINE_REPORT_*. A separate report is generated for each group of LINE_REPORT_* keys. An example of the line report is shown below. The report is designed for 172 character lines with 65 lines per page. The report can be copied into word processing software for custom printing. It will fit on an 8.5x11 inch landscape page using a 7 point courier font and adjusted margins or an 8.5x14 inch landscape page using a 9 point courier font.

```
Title: ORANGE Line: GAINESVILLE to BOWIE
Lines: WMORNA

                              --------------- 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

Gainesvill    4.23   5.32  6217     0   6217    797     0    797   7014     0   7014      0    70     70      0    22     22      0    92     92   7014    92   7106
Sudley Roa    5.23   6.43  7193    75  13335    545     6   1335   7738    81  14670     24   311    356      2   181    201     26   492    557   7764   573  15227
Centrevill    3.30   4.25  7736   241  20830   2060    12   3386   9796   253  24216     72   861   1143     20   551    732     92  1412   1875   9888  1665  26091
Stringfell    2.60   3.30  7244    40  28034   1322     2   4707   8566    42  32741     40   251   1357      4    43    772     44   294   2129   8610   336  34870
Fair Oaks     4.90   5.66  8942   327  36650   2308    35   6979  11250   362  43629    109  1527   2776     33   752   1492    142  2279   4268  11392  2641  47897
Vienna        2.39   3.63  9616   229  46038   4381    66  11295  13997   295  57333    185  1631   4220     55  1043   2483    240  2674   6703  14237  2969  64036
Dunn Lorin    2.49   3.93  7595  5594  48040   2189   750  12735   9784  6344  60775   1070  1851   5002    581   772   2675   1651  2623   7677  11435  8967  68452
West Falls    2.09   2.89  7020   527  54536   2535   167  15103   9555   694  69639    530   377   4849    274   159   2561    804   536   7410  10359  1230  77049
East Falls    2.51   3.90  3472  2157  55852   1401   561  15942   4873  2718  71794    666   732   4914    200   458   2821    866  1190   7735   5739  3908  79529
Ballston      0.49   1.65  4542  3591  56805   4398   837  19506   8940  4428  76311    692  4824   9046    573  3138   5385   1265  7962  14431  10205 12390  90742
Virginia S    0.49   1.97  1866   813  57857   2134   148  21494   4000   961  79351    242  1139   9944     86   750   6048    328  1889  15992   4328  2850  95343
Clarendon     0.67   2.48  2314   882  59290   1847   252  23088   4161  1134  82378    337  1562  11168    213  1003   6838    550  2565  18006   4711  3699 100384
Court Hous    0.91   1.67  2812  2430  59673   2615   889  24815   5427  3319  84488    563  3832  14440    719  2105   8223   1282  5937  22663   6709  9256 107151
Rosslyn       1.35   3.03  8675 13565  54785   3577  4266  24124  12252 17831  78909   5584  6786  15641   2788  3874   9311   8372 10660  24952  20624 28491 103861
Foggy Bott    0.57   2.03  1045  7832  47996   1180  4326  20978   2225 12158  68974    546  5334  20428    674  3740  12375   1220  9074  32803   3445 21232 101777
Farragut W    0.38   0.94   357 11080  37275   1132  4326  17784   1489 15406  55059   5270  6047  21205   3173  4521  13723   8443 10568  34928   9932 25974  89987
McPherson     0.46   1.06   345  7105  30513    188  2802  15169    533  9907  45682    144  5597  26659    169  3537  17091    313  9134  43750    846 19041  89432
Metro Cent    0.29   0.88  6850 15587  21775   1572  5914  10828   8422 21501  32603   5520  7842  28980   3688  2628  16033   9208 10470  45013  17630 31971  77616
Federal Tr    0.41   2.04    21  5589  16207     18  3136   7710     39  8725  23917    186  2987  31781    103  2714  18644    289  5701  50425    328 14426  74342
Smithsonia    0.59   2.23     9  3565  12652     40  1291   6458     49  4856  19110     69   469  32183    185   233  18691    254   702  50874    303  5558  69984
L'Enfant P    0.33   1.88  2182  7695   7138    983  2364   5075   3165 10059  12213   8321  7359  31220   4076  2906  17522  12397 10265  48742  15562 20324  60955
Federal Ce    0.57   1.86   120  1834   5424    188   638   4628    308  2472  10052    227  4006  35001    328  3382  20574    555  7388  55575    863  9860  65627
Capitol So    0.50   1.89   841  2280   3985    421  2733   2316   1262  5013   6301   2683  7831  40149   2705  2953  20823   5388 10784  60972   6650 15797  67273
Eastern Ma    0.63   1.91   145   939   3191    104   563   1857    249  1502   5048    742  1677  41082    661   900  21064   1403  2577  62146   1652  4079  67194
Potomac Av    0.66   0.95   248   465   2976     80   326   1613    328   791   4589   1511   279  39849   1553    87  19600   3064   366  59449   3392  1157  64038
Stadium Ar    2.19   3.93   356   732   2599    271   765   1119    627  1497   3718   2960   575  37467   3257   376  16720   6217   951  54187   6844  2448  57905
Minnesota     0.91   1.92   424   493   2531    144   329    933    568   822   3464   2402   311  35375   1403    60  15375   3805   371  50750   4373  1193  54214
Deanwood      1.15   1.87   150   432   2249     81   229    785    231   661   3034   1785    84  33673   1370    34  14042   3155   118  47715   3386   779  50749
Cheverly      1.89   2.94    87   474   1861     18   136    668    105   610   2529   2083    73  31662    702    17  13356   2785    90  45018   2890   700  47547
Landover      1.36   2.66   116   352   1625     46   122    593    162   474   2218   2544    68  29186    711    34  12680   3255   102  41866   3417   576  44084
New Carrol    5.50   6.13    25  1533    119      7   576     22     32  2109    141   3055   163  26295   1716    12  10979   4771   175  37274   4803  2284  37415
Woodmoore     1.60   1.88     0     3    118      0     0     22      0     3    140   8971     0  17324   3112     0   7867  12083     0  25191  12083     3  25331
Bowie Town    1.40   1.89     0   107      8      0    22      0      0   129      8   6321     1  11004   5973     0   1893  12294     1  12897  12294   130  12905
Crain High                            8                  0                   8          11004                1893               12897               12897     8

Total        55.04  91.00       98565             38582             137147             76458              43000             119458             256605

Max           5.50   6.43  9616 15587  59673   4398  5914  24815  13997 21501  84488  11004  7842  41082   5973  4521  21064  12897 10784  62146  20624 31971 107151

Passenger Miles            1229802            338893            1568695            734259             350587            1084846            2653541
Passenger Hours              36238             11324              47562             24346              12211              36557              84119
Average Trip Length (miles)   12.5               8.8               21.3               9.6                8.2               17.8               39.0
Average Trip Length (minutes) 22.1              17.6               39.7              19.1               17.0               36.1               75.8
```

The stop sequence is set by the direction of travel by the first line in the group. If other lines or the offpeak period adds stops to the sequence, these stops are shown as branches at the end of the stop list. The data presented in the AB direction columns shows the ridership profile with boardings (on) and alightings (off) at each station and the net ridership on the link between the current stop and the stop that follows. In this example, 6,217 riders board at Gainesville in the peak AB direction, 74 get off and 7,193 get on at Sudley Road, resulting in 13,335 riders leaving Sudley Road towards Centerville. Based on the ons and offs the number of riders between Sudley Road and Centerville should be 13,336. This discrepancy is caused by the rounding error included in the Cube ridership files. The data in the file are stored as integers while the transit assignment using floating point numbers.

The data presented in the BA direction columns is read from bottom to top. Boardings at the last station are shown as riders for the station above. Some of these riders my exit at this station and other may board with the net ridership shown on the next station above. This presentation format permits the total daily ridership columns to show the riders in both directions as a single value in the top to bottom orientation.

Also note that transit assignments such as these typically report riders in production-attraction format. This can be helpful in depicting the peak orientation of the line. To estimate the boardings at a given station in origin-destination format, you would normally add the ons and offs together and divide by two.

LINK REPORT

The link report summarizes the ridership on each line that traverses a sequence of links. The summary can be restricted to a specified set of modes and lines and the specified direction. The report is controlled by the control keys that start with LINK_REPORT_*. A separate report is generated for each group of LINK_REPORT_* keys. An example of the link report is shown below. In this example, 68,207 riders use the segment of Red Line "A" (WMREDA) between stops 8014 and 8017 in the peak period.

```
Title: Red Line in Downtown
Links: 8014-8015-8016-8017

                    -----A->B Direction-----   -----B->A Direction-----   --------- Total --------
Mode   Line            Peak  Off-Peak   Daily     Peak  Off-Peak   Daily     Peak  Off-Peak   Daily

  3    WMREDA         68207    37008   105215   103476    54651   158127   171683    91659   263342
  3    WMREDB         45433    27487    72920    72442    43894   116336   117875    71381   189256

Total  2             113640    64495   178135   175918    98545   274463   289558   163040   452598

Maximum               68207    37008   105215   103476    54651   158127   171683    91659   263342
Passenger Miles       54596    32155    86751    75653    45523   121176   130248    77678   207926
Passenger Hours        2723     1594     4316     3833     2280     6113     6556     3874    10429
```

ON-OFF REPORT

The on-off report summarizes the boardings and alightings for a specified list of transit stops. The summary can be restricted to a specified set of modes and lines. The report is controlled by the control keys that start with ON_OFF_REPORT_*. The report for each group can also be generated in one of three formats. If the Details key is FALSE, the report will show a single boarding and alighting summary for each stop in the list. This format is shown below.

```
Title: Downtown Stations
Modes: All

            ---- Peak ----   -- Offpeak ---   ---- Daily ---
Stop         Board  Alight    Board  Alight    Board  Alight

8015         56308   85064    18473   33460    74781  118524  Metro Center
8016         30419   55174    19707   40066    50126   95240  Gallery Place
8018         24897   33839    22336   23782    47233   57621  Union Station
8037         34713   58485    14394   27195    49107   85680  L'Enfant Plaza

Total       146337  232562    74910  124503   221247  357065
```

If the Details key is MODE, the summary generates a separate row for each mode that uses the stop. In the example below, only mode 3 lines serve the specified stops.

```
Title: Downtown Stations
Modes: All

                   ---- Peak ----   -- Offpeak ---   ---- Daily ---
Stop         Mode   Board  Alight    Board  Alight    Board  Alight

8015            3   56308   85064    18473   33460    74781  118524  Metro Center

8016            3   30419   55174    19707   40066    50126   95240  Gallery Place

8018            3   24897   33839    22336   23782    47233   57621  Union Station

8037            3   34713   58485    14394   27195    49107   85680  L'Enfant Plaza

Total              146337  232562    74910  124503   221247  357065
```

If the Details key is TRUE, the summary generates a row for each line that uses the stop. In the example below, four lines serve stop 8015 with a total peak period boardings of 56,308.

```
Title: Downtown Stations
Modes: All

                                ---- Peak ----   -- Offpeak ---   ---- Daily ---
Stop         Mode   Line         Board  Alight    Board  Alight    Board  Alight

8015            3   WMORNA       12265   23433     5267    8624    17532   32057  Metro Center
                3   WMORND        5847    2860        0       0     5847    2860
                3   WMREDA       25759   47348     7112   19117    32871   66465
                3   WMSILB       12437   11423     6094    5719    18531   17142
        Total       4            56308   85064    18473   33460    74781  118524

8016            3   WMGRNA        5963    9327     5378    9483    11341   18810  Gallery Place
                3   WMGRNB        4306   15752        0       0     4306   15752
                3   WMGRNC        2609    3775        0       0     2609    3775
                3   WMGRND           0       0     5644   10854     5644   10854
                3   WMREDA       17541   26320     8685   19729    26226   46049
        Total       5            30419   55174    19707   40066    50126   95240

8018            3   WMREDA       24897   33839    22336   23782    47233   57621  Union Station

8037            3   WMGRNA        4910   10871     1852    9092     6762   19963  L'Enfant Plaza
                3   WMGRNB        3523   17616        0       0     3523   17616
                3   WMGRNC        2505    5444        0       0     2505    5444
                3   WMGRND           0       0     1917    9590     1917    9590
                3   WMORNA       10182   15205     5303    5358    15485   20563
                3   WMORND        4103    2302        0       0     4103    2302
                3   WMSILB        9490    7047     5322    3155    14812   10202
        Total       7            34713   58485    14394   27195    49107   85680

Total                           146337  232562    74910  124503   221247  357065
```

Note that transit assignments typically report riders in production-attraction format. This can be helpful in estimating the boardings and alightings by time of day. For example, the boardings at 8015 in the AM peak period are estimates as 26,154 (i.e., 56,308/2). To estimate the boardings at a given station in origin-destination format, you would normally add the boarding and alightings together and divide by two. In other words, 8015 is estimated to have 70,686 (i.e., (56,308+85,064)/2) boardings in the AM and PM peak periods.

ACCESS REPORT

Access reports summaries the same type of information as the on-off reports focusing on the riders that arrive or depart each stop using transit access links. The ACCESS_REPORT_* keys generate a report for each key group. If the Details key is FALSE, the report shows the total arrivals and departures for each stop. This format is shown below.

```
Title: Downtown Stations
Modes: All

        ---- Peak ----  -- Offpeak ---  ---- Daily ---
Stop    Arrive  Depart  Arrive  Depart  Arrive  Depart

8015      8497   37261    9670   24661   18167   61922  Metro Center
8016      9595   34348   11713   32079   21308   66427  Gallery Place
8018     30314   39258   24393   25836   54707   65094  Union Station
8037      5174   28946    4901   17701   10075   46647  L'Enfant Plaza

Total    53580  139813   50677  100277  104257  240090
```

If the Details key is MODE, the summary generates a separate row for each access mode that serves the stop. In the example below, mode 12 and 13 links serve the specified stops.

```
Title: Downtown Stations
Modes: All

              ---- Peak ----  -- Offpeak ---  ---- Daily ---
Stop      Mode Arrive  Depart  Arrive  Depart  Arrive  Depart

8015        12   2288   30990    2302   18458    4590   49448  Metro Center
            13   6209    6271    7368    6203   13577   12474
        Total    8497   37261    9670   24661   18167   61922

8016        12   3324   28139    5510   24711    8834   52850  Gallery Place
            13   6271    6209    6203    7368   12474   13577
        Total    9595   34348   11713   32079   21308   66427

8018        12   3724   30662    1642   20783    5366   51445  Union Station
            13  26590    8596   22751    5053   49341   13649
        Total   30314   39258   24393   25836   54707   65094

8037        12   5174   28946    4901   17701   10075   46647  L'Enfant Plaza

Total           53580  139813   50677  100277  104257  240090
```

If the Details key is TRUE, the summary generates a row for each link that uses the stop. In the example below, five access links serve stop 8015. The first row shows that 746 riders arrive and 17,672 riders depart at stop 8015 in the peak period on the link from node 20111.

```
Title: Downtown Stations
Modes: All

                      ---- Peak ----  -- Offpeak ---  ---- Daily ---
Stop     Mode   From  Arrive  Depart  Arrive  Depart  Arrive  Depart

8015       12  20111     746   17672     945   10020    1691   27692  Metro Center
           12  20119    1139    7964     615    5677    1754   13641
           12  20514     182    5354     259    2761     441    8115
           12  21608     221       0     483       0     704       0
           13   8016    6209    6271    7368    6203   13577   12474
        Total      5    8497   37261    9670   24661   18167   61922

8016       12  20123     699    4114    1252    4875    1951    8989  Gallery Place
           12  20133     207    2659     325    2313     532    4972
           12  20156    1692    9661    2711    8545    4403   18206
           12  20557     576   11703    1213    8978    1789   20681
           12  91813     150       2       9       0     159       2
           13   8015    6271    6209    6203    7368   12474   13577
        Total      6    9595   34348   11713   32079   21308   66427

8018       12   9001    1587     467     268    1501    1855    1968  Union Station
           12  20258     474    7796     209    6121     683   13917
           12  20259     200    2786     104    2372     304    5158
           12  20281    1463   19613    1061   10789    2524   30402
           13   8128   17075    5263   15716    4194   32791    9457
           13   8136    9515    3333    7035     859   16550    4192
        Total      6   30314   39258   24393   25836   54707   65094

8037       12   8538     474   11550    1044    7367    1518   18917  L'Enfant Plaza
           12   9057    3580      60    1667     597    5247     657
           12  20045     322   13914     786    7124    1108   21038
           12  20046     798    3422    1404    2613    2202    6035
        Total      4    5174   28946    4901   17701   10075   46647

Total                  53580  139813   50677  100277  104257  240090
```

STOP REPORT

The stop report summarizes the ridership on each line serving a group of stops. The summary can be restricted to a specified set of modes and lines. The report is controlled by the control keys that start with STOP_REPORT_*. A separate report is generated for each group of STOP_REPORT_* keys. An example of the stop report is shown below. This example shows the combined activity at the two stop nodes that make up the Rosslyn station. It shows the number of riders that arrive on each line, the number of riders that alight and board the line, and the net result that depart on the line. This summary includes both transit lines and access links.

If the summary is generated for a single stop, the number of transfers can be approximated by subtracting the number arriving on access links from the total number of boardings. Due to rounding error in the Cube ridership file, it is advisable to report the number of transfers as the average of the boardings minus access arrivals and the alightings minus access departures. The example below, however, presents a significant complication. In this case the mode 13 link represents a pedestrian tunnel between the two station nodes (8065 and 8129). The riders using the tunnel are arrivals at one node and departures from the other node. The software detects this situation and reports the arrival and departures on the mode 13 link as one half of the reported value. It is not possible from the available data, however, to know how many of the travelers that use link 13 board a transit line or depart through an access link attached to the other station. The software assumes all tunnel users board another line and estimates the number of transfers between the Rosslyn lines in the peak period as 15,879 (i.e., ((56,228 – 184 – 10,767 – 29,395) + (80,316 – 35,044 – 29,395)) / 2).

```
Title: Rosslyn Station
Stops: 8065 (Rosslyn), 8129 (Rosslyn (19th & Lee Hwy))

                        ------------Peak-----------   ----------Off-Peak---------   -----------Daily-----------
Mode   Line             Arrive   Off     On  Depart   Arrive   Off     On  Depart   Arrive   Off     On  Depart

  3    WMBLUA            18854   8753   5188  15285        0      0      0      0    18854   8753   5188  15285
  3    WMBLUB            43271  20799   8323  30796    38233  12973   7479  32739    81504  33772  15802  63535
  3    WMBLUC            26508  12756   5391  19144        0      0      0      0    26508  12756   5391  19144
  3    WMORNA            75314  20351  14259  69225    34126   8140   6365  32347   109440  28491  20624 101572
  3    WMORND             9483   4422   6680  11743        0      0      0      0     9483   4422   6680  11743
  3    WMSILB            52326  13235  16387  55477    29039   6612   8144  30574    81365  19847  24531  86051
 11    11                  184      0      0      0      231      0      0      0      415      0      0      0
 12    12                10767      0      0  35044     9186      0      0  15162    19953      0      0  50206
 13    13                29395      0      0  29395    10805      0      0  10805    40200      0      0  40200

Total  9                266102  80316  56228 266109   121620  27725  21988 121627   387722 108041  78216 387736
Transfers                              15879                          1762                          17641
Maximum                  75314  20799  16387  69225    38233  12973   8144  32739   109440  33772  24531 101572
```

TOTAL REPORT

The total report summarizes the total boardings, the maximum link ridership and the passenger miles and hours of travel for selected modes and lines. The report is controlled by the control keys that start with TOTAL_REPORT_*. A separate report is generated for each group of TOTAL_REPORT_* keys. An example of the total report is shown below. The report is designed for 165 character lines with 65 lines per page. The report can be copied into word processing software for custom printing. It will fit on an 8.5x11 inch landscape page using a 7 point courier font and adjusted margins or an 8.5x14 inch landscape page using a 9 point courier font.

```
Title: Metro

                      ------------------- Peak Period ----------------   ------------------- Off Peak ------------------   --------------- Daily -------------------
                        Dist    Time ------Boardings----- ---Max Load--    Dist    Time ------Boardings----- ---Max Load--   -------Boardings------   ----Passenger----
Mode  Line            (miles) (min)   A->B   B->A  Total   A->B   B->A   (miles) (min)   A->B   B->A  Total   A->B   B->A     A->B    B->A   Total      Miles     Hours

  3   WMBELTB            14.7   23.4  17722  11441  29163  10028   9037     14.7   23.4   3915   4636   8551   2120   3953    21637   16077   37714     246916      6566
  3   WMBLUA             28.8   70.0  21404  21857  43261  14286   8457      0.0    0.0      0      0      0      0      0    21404   21857   43261     243634     10375
  3   WMBLUB             40.1   78.0  41546  36235  77781  26658  14173     40.1   78.0  41846  32800  74646  28440  14529    83392   69035  152427    1028932     37266
  3   WMBLUC             30.6   58.0  25246  13379  38625  17225   7868      0.0    0.0      0      0      0      0      0    25246   13379   38625     267987      9350
  3   WMGRNA             22.8   47.0  43527  33644  77171  24600  19196     22.8   47.0  33103  22102  55205  18960  14506    76630   55746  132376     583601     20550
  3   WMGRNB             34.2   47.0  36369  32577  68946  20480  20987      0.0    0.0      0      0      0      0      0    36369   32577   68946     407656     10076
  3   WMGRNC             22.1   47.0   6011  13751  19762   3518  10544      0.0    0.0      0      0      0      0      0     6011   13751   19762      81471      3494
  3   WMORNA             55.0   91.0 104937  60248 165185  66290  25082     55.0   91.0  38726  35697  74423  24723  13749   143663   95945  239608    2167777     70879
  3   WMORND             18.8   57.0  17180  15750  32930   6889   6034      0.0    0.0      0      0      0      0      0    17180   15750   32930      82532      4848
  3   WMREDA             31.6   65.0 169356 144923 314279  80400  58140     31.6   65.0  88377  90599 178976  45965  40396   257733  235522  493255    2955330    105535
  3   WMSILB             35.2   75.0  67085  51493 118578  35476  18229     35.2   75.0  30077  35166  65243  17794  13316    97162   86659  183821    1535824     55872
  3   WMYELA             18.9   60.0  29599  45658  75257  10645  18565     18.9   60.0  11641  16478  28119   5935   8443    41240   62136  103376     362256     18869
  3   WMGRND              0.0    0.0      0      0      0      0      0     26.6   47.0  17899  23856  41755   8238  17621    17899   23856   41755     178357      6167

Total 13                352.8  718.4 579982 4809561060938  80400  58140    244.9  486.4 265584 261334 526918  45965  40396   845566  742290 1587856   10142275    359845
```

DIFFERENCE REPORT

The difference report compares the attributes of an Alternative network to the attributes of a Base network. It lists each line in the Base and/or Alternative network that have some type of difference. The first few lines in the report extract below show an example of lines that are in the Alternative, but not in the Base (i.e., all Base attributes are zero). These are followed by the WMBLUA line that share the same headway, but have different run times and number of stop nodes. In other words, the Metrorail Blue line was extended in the Alternative network.

Note that each line is in list is reported twice. This represents each direction of a two way line. The return direction of a line will normally be identified with the word "Return" in the comment field. The comment fields in this example were truncated to fit on the page, but the "Return" label can still be seen in the comment field for the second and fourth lines.

```
 Difference Report  ------------------- Base ----------------   ------------ Alternative ----------------
                    ----- Peak -----   ----Offpeak-----   Num    ----- Peak -----   ----Offpeak-----   Num
 Mode  Line         Headway  RunTime   Headway  RunTime  Nodes   Headway  RunTime   Headway  RunTime  Nodes

   3   WMBELTA          0        0         0        0      0         6        0        12        0       6  WMATA
   3   WMBELTA          0        0         0        0      0         6        0        12        0       6  WMATA Return|WMATA
   3   WMBELTB          0        0         0        0      0         6        0        12        0       7  WMATA|WMATA
   3   WMBELTB          0        0         0        0      0         6        0        12        0       7  WMATA Return|WMATA Return
   3   WMBLUA          10       64        12       64     28        10       78        12       78      31  WMATA;FRANCONIA/SPRINGFIE
   3   WMBLUA          10       64        12       64     28        10       78        12       78      31  WMATA;FRANCONIA/SPRINGFIE
   3   WMBLUB           0        0         0        0      0        15       70         0        0      27  WMATA;FRANCONIA/SPRINGFIE
   3   WMBLUB           0        0         0        0      0        15       70         0        0      27  WMATA;FRANCONIA/SPRINGFIE
   3   WMGRNA           5       47        12       47     21         7       47         0        0      22  WMATA;GREENBELT STATION;
   3   WMGRNA           5       47        12       47     21         7       47         0        0      22  WMATA;GREENBELT STATION;
   3   WMORNA           6       57        12       57     26         6       91        12       91      34  WMATA;VIENNA STATION;NEW
   3   WMORNA           6       57        12       57     26         6       91        12       91      34  WMATA;VIENNA STATION;NEW
   3   WMORND           0        0         0        0      0        15       57         0        0      21  WMATA;VIENNA STATION;STAD
   3   WMORND           0        0         0        0      0        15       57         0        0      21  WMATA;VIENNA STATION;
   3   WMREDA           2       65         6       65     27         2       65         6       65      28  WMATA;SHADY GROVE STATION
   3   WMREDA           2       65         6       65     27         2       65         6       65      28  WMATA;SHADY GROVE STATION
   3   WMSILB          15       85         0        0     34         5       75        12       75      29  WMATA;VA772 STATION;
   3   WMSILB          15       85         0        0     34         5       75        12       75      29  WMATA;VA772 STATION
   3   WMYELA           6       27         0        0     13         5       29        12       29      15  WMATA;MTVernon;HUNTINGTON
   3   WMYELA           6       27         0        0     13         5       29        12       29      15  WMATA;MTVernon;
   3   WMYELB          15       60         0        0     22        15       35         0        0      19  WMATA;FRANC/SPRINGFLD STA
   3   WMYELB          15       60         0        0     22        15       35         0        0      19  WMATA;FRANC/SPRINGFLD STA
   4   AMTK670         60       65         0        0     13        60       65         0        0      12  AMTRAK;Washington-Union
```

Link Rider File Fields

The link rider files summarize the combined ridership from all lines that utilize each link of the selected route network. The files are controlled by the control keys that contain LINK_RIDER_*. A separate file is generated for each group of LINK_RIDER keys. The output is typically saved in TAB_DELIMITED or ARCVIEW formats. The TAB_DELIMITED format is convenient for processing in Excel. The ARCVIEW format is intended for generating ArcGIS maps. If the ARCVIEW format is selected, a node coordinate file is required to construct the shape records. In either case, the file contains the following data fields.

ANODE	(Integer)	A node number
BNODE	(Integer)	B node number
AB_PK_RIDE	(Integer)	Peak period riders in the AB direction
AB_PK_FREQ	(Double , Minutes)	Peak period frequency in the AB direction
AB_OP_RIDE	(Integer)	Offpeak riders in the AB direction
AB_OP_FREQ	(Double, Minutes)	Offpeak frequency in the AB direction
AB_DAY_RIDE	(Integer)	Daily riders in the AB direction
BA_PK_RIDE	(Integer)	Peak period riders in the BA direction
BA_PK_FREQ	(Double, Minutes)	Peak period frequency in the BA direction
BA_OP_RIDE	(Integer)	Offpeak riders in the BA direction
BA_OP_FREQ	(Double, Minutes)	Offpeak frequency in the BA direction
BA_DAY_RIDE	(Integer)	Daily riders in the BA direction
PK_RIDE	(Integer)	Peak period riders in both directions
PK_FREQ	(Double, Minutes)	Peak period frequency in both directions

PK_DIST	(Double, Miles)	Peak period link length
PK_TIME	(Double, Minutes)	Peak period link travel time
PK_PMT	(Double, Miles)	Peak period passenger miles of travel
PK_PHT	(Double, Hours)	Peak period passenger hours of travel
PK_VMT	(Double, Miles)	Peak period vehicle miles of travel
PK_VHT	(Double, Hours)	Peak period vehicle hours of travel
OP_RIDE	(Integer)	Offpeak period riders in both directions
OP_FREQ	(Double, Minutes)	Offpeak period frequency in both directions
OP_DIST	(Double, Miles)	Offpeak period link length
OP_TIME	(Double, Minutes)	Offpeak period link travel time
OP_PMT	(Double, Miles)	Offpeak period passenger miles of travel
OP_PHT	(Double, Hours)	Offpeak period passenger hours of travel
OP_VMT	(Double, Miles)	Offpeak period vehicle miles of travel
OP_VHT	(Double, Hours)	Offpeak period vehicle hours of travel
DAY_RIDE	(Integer)	Daily riders in both directions
DAY_PMT	(Double, Miles)	Daily passenger miles in both directions
DAY_PHT	(Double, Hours)	Daily passenger hours in both directions
DAY_VMT	(Double, Miles)	Daily vehicle miles in both directions
DAY_VHT	(Double, Hours)	Daily vehicle hours in both directions
PEAK_HOUR	(Double)	Maximum directional riders in one hour
LOAD_FAC	(Double)	Maximum directional riders per vehicle
FROM	(String)	Station name at the A node
TO	(String)	Station name at the B node

ModeChoice (version 5.x.19)

The ModeChoice program is used to:

1. Apply a nested-logit or multi-nomial mode choice model to zone-based trip tables.
    a. Read trip tables and one or more mode-specific skim files in TPPlus or TransCAD format.
    b. Trip tables can be subdivided by income or some other traveler attribute.
    c. Mode choice constants can vary by geographic market segments and traveler attribute.
    d. Short and long walk accessibility to transit by origin and destination zone is used to determine mode shares by combined access markets.
    e. A user script interface is available for manipulating the skim and zonal attributes used for each origin-destination choice.
2. Apply an iterative model calibration process to match mode choice results to calibration targets.
    a. Calibration targets can be defined by mode, market segment, and traveler attribute.
    b. Exit criteria are control by a maximum number of iterations and a RMSE convergence value.
    c. Input constants with minimum and maximum constraints are used as seed values for the calibration process.
    d. The output calibration file can be used for mode choice applications or as input to additional calibration iterations.
3. Generate reports summarizing the mode shares, market segments, and calibration results.
4. Generate text files summarizing the mode shares, market segments, and calibration results.
5. Generate model results in the input format required by FTA's SUMMIT program.
6. Generate production and attraction zone mode split summary files.

ModeChoice is a console-based program that runs in a command window on either Windows or Linux. The command syntax is:

ModeChoice [-flag] [control_file]

The control_file is the directory path and file name of a text file that contains the control strings expected by the program. If a file name is not provided, the program will prompt the user to enter a file name. The program automatically creates a printout file based on the control_file name. If the file name includes an extension (e.g., ".ctl"), the extension is removed and ".prn" is added. The printout file will be created in the current working directory and will overwrite an existing file with the same name.

Additional information about program syntax, command lines flags, control keys and comments can be found in the Execution Service document.

A typical ModeChoice control file is shown below. These keys can be defined in a variety of different ways to perform the tasks listed above.

```
TITLE                    WMATA AM Peak HBW Mode Choice

TRIP_FILE                TRIPS\HBWAM.DAT
SKIM_FILE_2              SKIMS\HWYAM.SKM
SKIM_FILE_3              SKIMS\TSKM_CRAM.DAT
SKIM_FILE_4              SKIMS\TSKM_ABAM.DAT
SKIM_FILE_5              SKIMS\TSKM_MRAM.DAT
```

```
SKIM_FILE_6              SKIMS\TSKM_BMAM.DAT
ZONE_FILE                ZONEV2_V2.txt
NEW_TRIP_FILE            TRIPS\HBW_AM_MS.tpp
NEW_TRIP_FORMAT          TPPLUS

MODE_CONSTANT_FILE       ..\Controls\HBW_AM_Constant.txt
MODE_CHOICE_SCRIPT       ..\Scripts\Mode_Choice_Script.txt

SEGMENT_MAP_FILE         ..\Controls\Segment_Map.txt
ORIGIN_MAP_FIELD         AREA_TYPE
DESTINATION_MAP_FIELD    AREA_TYPE

TRIP_PURPOSE_LABEL       Peak Home-Based Work
TRIP_PURPOSE_NUMBER      1          //---- HBW=1, HBS=2, HBO=3, NHB=4 ----
TRIP_TIME_PERIOD         1          //---- PEAK=1, OFFPEAK=2 ----

PRIMARY_MODE_CHOICE      AUTO, TRANSIT
MODE_CHOICE_NEST_1       AUTO = SOV, HOV
MODE_CHOICE_NEST_2       HOV = SR2, SR3
MODE_CHOICE_NEST_3       TRANSIT = WALK, PNR, KNR
MODE_CHOICE_NEST_4       WALK = WK_CR, WK_BUS, WK_BUS_MR, WK_MR
MODE_CHOICE_NEST_5       PNR = PNR_CR, PNR_BUS, PNR_BUS_MR, PNR_MR
MODE_CHOICE_NEST_6       KNR = KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
NESTING_COEFFICIENT      0.5

VEHICLE_TIME_VALUE       -0.030
WALK_TIME_VALUE          -0.060
DRIVE_ACCESS_VALUE       -0.045
WAIT_TIME_VALUE          -0.075
TRANSFER_COUNT_VALUE     -0.000
PENALTY_TIME_VALUE       -0.075
TERMINAL_TIME_VALUE      -0.075
COST_VALUE_TABLE_1       -0.00185
COST_VALUE_TABLE_2       -0.00092
COST_VALUE_TABLE_3       -0.00059
COST_VALUE_TABLE_4       -0.00044

MODE_ACCESS_MARKET_1     SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, WK_MR, PNR_CR, PNR_BUS,
                         PNR_BUS_MR, PNR_MR, KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_2     SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, PNR_CR, PNR_BUS,
                         PNR_BUS_MR, PNR_MR, KNR_CR, KNR_BUS, KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_3     SOV, SR2, SR3, WK_CR, WK_BUS, WK_BUS_MR, PNR_CR, PNR_BUS,
                         PNR_BUS_MR, KNR_CR, KNR_BUS, KNR_BUS_MR
MODE_ACCESS_MARKET_4     SOV, SR2, SR3, PNR_CR, PNR_BUS, PNR_BUS_MR, PNR_MR, KNR_CR,
                         KNR_BUS, KNR_BUS_MR, KNR_MR
MODE_ACCESS_MARKET_5     SOV, SR2, SR3, PNR_CR, PNR_BUS, PNR_BUS_MR, KNR_CR, KNR_BUS,
                         KNR_BUS_MR
MODE_ACCESS_MARKET_6     SOV, SR2, SR3

NEW_TABLE_MODES_1        SOV = Drive Alone
NEW_TABLE_MODES_2        SR2 = Shared Ride 2
NEW_TABLE_MODES_3        SR3 = Shared Ride 3
NEW_TABLE_MODES_4        WK_CR = Walk to Commuter Rail
NEW_TABLE_MODES_5        WK_BUS = Walk to Bus
NEW_TABLE_MODES_6        WK_BUS_MR = Walk to Bus/Metrorail
NEW_TABLE_MODES_7        WK_MR = Walk to Metrorail
NEW_TABLE_MODES_8        PNR_CR, KNR_CR = PNR/KNK to Commuter Rail
NEW_TABLE_MODES_9        PNR_BUS = PNR to Bus
NEW_TABLE_MODES_10       KNR_BUS = KNR to Bus
NEW_TABLE_MODES_11       PNR_BUS_MR = PNR to Bus/Metrorail
NEW_TABLE_MODES_12       KNR_BUS_MR = KNR to Bus/Metrorail
NEW_TABLE_MODES_13       PNR_MR = PNR to Metrorail
NEW_TABLE_MODES_14       KNR_MR = KNR to Metrorail

NEW_MODE_SUMMARY_FILE    Results/HBW_AM_Summary.txt
NEW_MARKET_SEGMENT_FILE Results/HBW_AM_Segment.txt
NEW_MODE_SEGMENT_FILE    Results/HBW_AM_Mode_Seg.txt
NEW_FTA_SUMMIT_FILE      Results/HBW_AM_Summit.bin
NEW_PRODUCTION_FILE      Results/HBW_AM_Productions.txt
NEW_ATTRACTION_FILE      Results/HBW_AM_Attractions.txt
```

```
MODECHOICE_REPORT_1      MODE_CHOICE_SCRIPT
MODECHOICE_REPORT_2      MARKET_SEGMENT_REPORT
MODECHOICE_REPORT_3      MODE_SUMMARY_REPORT
MODECHOICE_REPORT_4      MODE_VALUE_SUMMARY
```

Execution Service Keys

The ModeChoice program inherits the following control keys from Execution Service.

TITLE (optional, text)

REPORT_DIRECTORY (optional, path)

REPORT_FILE (optional, output file)

REPORT_FLAG (optional, flag, FALSE)

PAGE_LENGTH (optional, integer, 65, >= 0)

PROJECT_DIRECTORY (optional, text)

DEFAULT_FILE_FORMAT (optional, text, TAB_DELIMITED)

Selection Service Keys:

The ModeChoice program inherits the following control keys from Selection Service.

SELECT_ORIGIN_ZONES (optional, list, ALL, > 0)

SELECT_DESTINATION_ZONES (optional, list, ALL, > 0)

Input-Output File Keys:

TRIP_FILE (required, input file)

This key specifies the name and location of the input person trip file that will be read by the program and distributed to various travel modes. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file may contain multiple tables representing trip purposes, income groups, or auto ownership levels.

TRIP_FORMAT (optional, text, TPPLUS)

If the Trip file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TRANSCAD, CUBE, TPPLUS, TRANPLAN, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, and SQLITE3.

NEW_TRIP_FILE (required, output file)

This key specifies the name and location of the output trip file were the mode choice results are stored. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file will contain multiple tables representing trip assigned to each mode.

AECOM

NEW_TRIP_FORMAT (optional, text, TPPLUS)

This key is used to override the default file format for creating data files. The options include TRANSCAD, CUBE, TPPLUS, TRANPLAN, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, and SQLITE3.

SELECT_TRIP_TABLES (optional, text, ALL)

This key is used to select table names from the input trip file for processing. If the key is ALL or not provided, all tables in the input trip file will be processed. The key is interpreted as a comma or space delimited list of text labels that must match the table names found in the trip file.

SKIM_FILE_# (required, input file)

This key specifies the name and location of one or more input data files containing one or more travel attributes (e.g., time, distance and cost) between zones. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The _# is used to identify multiple input files. Each input file typically represents the travel attributes of a given travel mode.

SKIM_FORMAT (optional, text, TPPLUS)

If the Skim file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TRANSCAD, CUBE, TPPLUS, TRANPLAN, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, and SQLITE3. The skim format applies to all input skim files.

ZONE_FILE (required, input file)

This key specifies the name and location of the input zone file that defines attributes of the trip origin and destination. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file may contain area types, terminal time values, parking costs, density measures, and other zone-related attributes required by the mode choice model.

ZONE_FORMAT (optional, text, TAB_DELIMITED)

If the Zone file includes a definition file, this key is ignored. If a definition file is not provided, this key identifies the file format so that the program can create a definition file. The format options include TEXT, BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, ARCVIEW, and SQLITE3.

MODE_CONSTANT_FILE (optional, input file)

This key specifies the name and location of the input file that defines mode specific constants. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. This file is typically the output file of the model calibration process. It is assumed to be a standard TRANSIMS data file with an accompanying definition file. A detailed description of the file structure is found in the Model Structure section of this document.

MODE_CHOICE_SCRIPT (required, input file)

This key specifies the name and location of the input file that defines a user program or script for the mode choice model. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The model script sets the values used in the utility function for each mode using various tables from the skim files and the attributes of the origin or destination zone record. Details about the scripting language available to a model user can be found in the User Program guide. Details about the file and field names expected by the ModeChoice program are found in the Model Structure section of this document.

Market Segment Keys:

SEGMENT_MAP_FILE (optional, input file)

This key specifies the name and location of the input file that defines market segment numbers. The full path and file name for the input file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. This file maps an attribute of the origin zone and an attribute of the destination zone to a market segment code. It is assumed to be a standard TRANSIMS data file with an accompanying definition file. A detailed description of the file structure is found in the Model Structure section of this document.

ORIGIN_MAP_FIELD (optional, text, SEGMENT)

This key identifies the field name in the zone file that will be used to associate a given origin zone with a specific market segment. The key is only read if a segment map file is provided.

DESTINATION_MAP_FIELD (optional, text, SEGMENT)

This key identifies the field name in the zone file that will be used to associate a given destination zone with a specific market segment. The key is only read if a segment map file is provided.

Model Structure Keys:

TRIP_PURPOSE_LABEL (optional, text, Peak HBW)

This is a user-defined text string that is printed at the top of output reports for general information.

TRIP_PURPOSE_NUMBER (optional, integer, 1, 1..100)

This is a user-defined code number that identifies the trip purpose within the model script. The value is passed to the model script through the variable TRIP.PURPOSE.

TRIP_TIME_PERIOD (optional, integer, 1, 1..100)

This is a user-defined code number that identifies the time period within the model script. The value is passed to the model script through the variable TRIP.PERIOD.

PRIMARY_MODE_CHOICE (required, text)

This key defines the mode labels assigned to the primary mode choice. It includes a list of two or more user-defined text strings separated by a comma or space (e.g., AUTO, TRANSIT). The names defined here are used to build attribute fields recognized by the model script (e.g., AUTO.TIME) and define relationships and processing options specified using other control keys. They are also used by output reports to identify the mode. All mode names much be unique and follow the standard variable name syntax define in the User Program guide.

MODE_CHOICE_NEST_# (optional, text)

A nested logit model structure is specified using one or more of these keys. Each key associates a mode label defined in a higher level key with a comma or space delimited list of user-defined text strings representing the nested mode names. For example, if the primary mode choice includes AUTO as one of its mode labels, the key

MODE_CHOICE_NEST_1 AUTO = SOV, HOV

subdivides the AUTO mode into two submodes labeled SOV and HOV. The new mode names defined by this key are used to build attribute fields recognized by the model script (e.g., SOV.TIME) and define relationships and processing options specified using other control keys. They are also used by output reports to identify the mode. Any number of nested records can be defined. The only restriction is that the mode labels specified to the left of the equal sign must have been defined in the primary mode list or the mode list of a nest key with a lower key number (i.e., _#).

NESTING_COEFFICIENT_# (optional, decimal, 0.5, 0.0..1.0)

In a nested model, the model script should define the attributes of each mode that is not nested. The attributes and utility of the nested mode are defined by the logsum of its nested members. This logsum is combined with other modes in a higher level mode split using a nesting coefficient. This key defines the nesting coefficient for the mode nest with the same key number (i.e., _#). The key may also be specified without a key number (i.e., NESTING_COEFFICIENT). In this case, the key value is applied to all nests. The default coefficient is 0.5.

Utility Parameter Keys:

The utility of a given mode is defined as the sum of a series of mode attributes multiplied by attribute weights (e.g., utility = a * x + b * y + c). The following keys define the weights for attributes set by the model script. Additional information about how these weights are applied can be found in the Model Structure section of this document.

VEHICLE_TIME_VALUE (optional, decimal, -0.02, 0, -0.03..-0.02)

The vehicle time value is the weight generally associated with in-vehicle travel time. The value of this key is multiplied by the mode attribute field called TIME. TIME is typically defined in minutes.

WALK_TIME_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The walk time value is the weight generally associated with walk access or out-of-vehicle time. The value of this key is multiplied by the mode attribute field called WALK. WALK is typically defined in minutes.

DRIVE_ACCESS_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The drive access value is the weight generally associated with drive access time to transit for park-n-ride or kiss-n-ride modes. The value of this key is multiplied by the mode attribute field called AUTO. AUTO may be defined as time in minutes or distance in miles.

WAIT_TIME_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The wait time value is the weight generally associated with waiting for the first transit vehicle. It could also represent the total waiting time for all transit boardings. The value of this key is multiplied by the mode attribute field called WAIT. WAIT is typically defined in minutes.

LONG_WAIT_VALUE (optional, decimal. 0, 0, -1.0..-0.02)

Most models calculate waiting time as a function of the headway of the available transit routes between two stops. One half of the headway is typically used for frequent transit service. If the transit option has infrequent service, the mode choice model may include an average wait time that travelers arrive prior to each scheduled departure (using the WAIT field) plus a penalty factor that attempts to capture the disutility associated with infrequent service. The long wait value key can be used to include an additional penalty for infrequent service. The value of this key is multiplied by the mode attribute field called LWAIT. LWAIT is typically defined in minutes, but may represent other values as well.

TRANSFER_TIME_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

If a transit trip requires a transfer between transit routes, the second wait may be valued by the traveler differently from the waiting time for the first transit route. The transfer time value is the weight generally associated with waiting to board the second or third transit route. The value of this key is multiplied by the mode attribute field called XWAIT. XWAIT is typically defined in minutes.

TRANSFER_COUNT_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The transfer count value is the disutility associated with the number of transfer included in the transit path. The key value is multiplied by the mode attribute field called XFER. XFER is the number of transfers.

PENALTY_TIME_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The penalty time value can be used to include additional time-related attributes in the utility function. This value of this key is multiplied by the mode attribute field TPEN. TPEN may be defined in minutes or simply be a 0/1 flag value that defines when to include the penalty for a given mode.

TERMINAL_TIME_VALUE (optional, decimal, 0, 0, -1.0..-0.02)

The terminal time value is the weight generally associated with the extra time required to complete the trip at a given origin or destination. It may represent the time required to park the car and walk to the destination for an auto trip or additional access disutility for transit trips. The value of this key is multiplied by the mode attribute field called TERM. TERM is typically defined in minutes.

COST_VALUE_TABLE_# (optional, decimal, 0, 0, -5.0..0.0)

The cost value is the weight generally associated with travel cost. The value of this key is multiplied by the mode attribute field called COST. The COST field will typically include transit fares and/or parking and auto operating costs in cents.

Note there is only one COST field, but there can be multiple cost values. Each cost value is mapped directly to one of the tables in the input trip file. The first cost value (i.e., COST_VALUE_TABLE_1) is applied to the trips included in the first trip table in the trip file, or if the SELECT_TRIP_TABLES key is provided, the first table included in the selection list. The cost value key is the only value key that changes based on the trip table. All other path attributes and the cost of the trip do not change as a function of the trip table being processed. In other words, the path is assumed to be the same for each trip table being processed.

Multiple cost values are typically used to include different "values of time" or in this case values of cost in the mode choice model. If, for example, the input trip file contains multiple tables representing income groups, the modeler might wish to make the relative impact of transit fares or parking costs higher for low income traveler than for high income travelers. Assuming income increases with table number, the cost value for table #1 would be a larger negative number than the cost value for table #2. This makes the disutility of low income trips higher as costs increase and as a consequence reduces the importance of travel time differences in the mode choice. This can be interpreted as a lower value of time for low income travelers and a higher value of time of high income travelers.

Output Specification Keys:

MODE_ACCESS_MARKET_# (optional, text, , SOV, SR2, SR3...)

In addition to travel markets based on origin-destination combinations and traveler attributes such as income or auto ownership, separate mode choice calculations can be made for each subset of the origin and destination zone that has access to various transit services. In its simplest form this is based on the percentage of a zone that can walk to a transit stop within a specified distance (e.g., 0.5 mile). Separate access markets could also be defined for longer walks to premium transit services or for drive access markets.

Each access market is defined using a mode access market key. The key number (_#) corresponds to a field in the model script (e.g., TRIP.ACCESS1) that defines the share of the total trips that are included in each access market. The sum of access field values should total 1.0 in order to distribute all of the trips to a single access market.

The mode access market key includes a comma or space delimited list of the mode names that are available to travelers within the corresponding access market. The mode names are defined by the PRIMARY_MODE_CHOICE and MODE_CHOICE_NEST_# keys. Mode names may be used in all access markets or only a few access markets. If the model is nested, only the non-nested modes should be included in the access market keys.

The following is a very simple example of how access markets are defined and used. Let's assume a very simple model with two modes (AUTO and TRANSIT). In order to choose the TRANSIT mode, the traveler's origin and destination must be within walking distance of a transit stop. This results in two access markets – the share of the trips between the origin and destination that have walk access to transit at both ends of the trip and the share of trips that don't have a transit option. If we assume the input zone file includes a transit walk coverage field called COVERAGE, the model script will define the access travel markets using a calculation like:

```
TRIP.ACCESS1 = ORG.COVERAGE * DES.COVERAGE        //---- walk to transit market ----
TRIP.ACCESS2 = 1.0 – TRIP.ACCESS1                 //---- no transit option ----
```

In this case, two mode access market keys would be included in the control file:

```
MODE_ACCESS_MARKET_1        AUTO, TRANSIT
MODE_ACCESS_MARKET_2        AUTO
```

The program will split the input trips into two access markets and apply the mode choice calculations to each market based on the modes that are available within that market. In this case, all trips in market #2 will choose AUTO and a percentage of the trips in market #1 will choose AUTO. Note that the disutility of the AUTO choice is the same for both access markets, but the number of mode options available to the traveler vary between access markets.

NEW_TABLE_MODES_# (optional, text)

Mode choice calculations are made for each access market and mode name defined using the PRIMARY_MODE_CHOICE and MODE_CHOICE_NEST_# keys. Each mode name and access market can be saved to the output trip file in a separate table or combined into one or more aggregate tables. The mode names and access markets assigned to each output table are defined using this key. The key specifies a comma or spaced delimited list of mode names followed by an equal sign and the name given to the output table. For example, the key

NEW_TABLE_MODES_1 SOV = Drive Alone

would create a table called "Drive Alone" in the new trip file and save the mode shares from the SOV mode to this table. If the key does not include an equal sign, the name of the output table is equal to the name of the mode (e.g., SOV).

In the example shown above, the SOV trips from each access market are summed and saved as a single value. If the user wishes to save the SOV trips for a given access market into a separate output table, the access market code number can be appended to the mode name to select the subset of SOV trips generated by the access market. This is done using the following syntax:

NEW_TABLE_MODES_1 SOV.1 = SOV Access Market 1

The syntax that combines multiple modes into an aggregate output table would look like this:

NEW_TABLE_MODES_1 SOV, SR2, SR3 = Auto Trips

This command sums the trips assigned to SOV, SR2, and SR3 modes for all access markets and outputs the total value in a table called "Auto Trips".

OUTPUT_TRIP_FACTOR (optional, decimal, 1.0, 1.0..1000.0)

This factor is applied to the trips assigned to each mode prior to writing the value to the output matrix file. It is primarily used to compensate or correct for the number of significant digits written to the matrix. All internal calculations use double floating point numbers.

Summary Data File Keys:

NEW_MODE_SUMMARY_FILE (optional, output file)

In addition to writing the mode choice summary report to the printout file, the user may choose to create a tab delimited file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. An example of the type of file created by this key is shown below.

Mode	M1	M2	M3	M4	Total	Percent
AUTO	258444.3	529517.5	439434	551002.3	1778398.2	76.13
TRANSIT	135016.8	157786.5	123960.4	140720.7	557484.5	23.87
SOV	191003.8	403410.9	333950.7	423731.2	1352096.6	57.88
HOV	67440.5	126106.6	105483.3	127271.2	426301.6	18.25
SR2	50558.9	93265.9	77212.8	93278.4	314316	13.46
SR3	16881.6	32840.7	28270.6	33992.8	111985.7	4.79
WALK	114416.5	109483.1	71750.1	62246.3	357896	15.32
PNR	8230.8	32785.3	38241.8	64343	143600.8	6.15
KNR	12369.5	15518.1	13968.5	14131.5	55987.6	2.4
WK_CR	130.9	751.6	423.9	301.6	1608	0.07
WK_BUS	61009.1	40828.5	18401.3	14596.7	134835.6	5.77
WK_BUS_MR	27673	29394.8	25147.2	19474.7	101689.7	4.35
WK_MR	25603.5	38508.2	27777.7	27873.2	119762.7	5.13
PNR_CR	803.8	3730.2	4156.6	6022	14712.5	0.63
PNR_BUS	2492.6	3280.1	3248.3	3691.1	12712.1	0.54
PNR_BUS_MR	1065	4131.2	4719.1	7067.4	16982.8	0.73
PNR_MR	3869.4	21643.8	26117.8	47562.4	99193.4	4.25
KNR_CR	191.3	787.3	1072.7	1232.9	3284.3	0.14
KNR_BUS	2102.7	2358.5	2438.6	1935.9	8835.7	0.38
KNR_BUS_MR	1453.1	2731.9	2914.6	3322.8	10422.4	0.45
KNR_MR	8622.4	9640.3	7542.6	7640	33445.3	1.43
Total	393461.2	687304	563394.5	691723.1	2335882.7	100

NEW_MARKET_SEGMENT_FILE (optional, output file)

If a segment map file is provided, the mode choice results can also be summarized by geographic market segments. The user may choose to create a tab delimited file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. Selected records from a market segment file are shown below.

Segment	Mode	M1	M2	M3	M4	Total	Percent
1	AUTO	1665	4149.4	3149.3	4346	13309.7	14.92
1	TRANSIT	22678.5	25289.1	14442.1	13477.5	75887.2	85.08
1	SOV	334.2	1258.3	1066.4	1768.5	4427.4	4.96
1	HOV	1330.8	2891.1	2082.9	2577.5	8882.3	9.96
1	SR2	972.3	2299.8	1700.5	2179.4	7152	8.02
1	SR3	358.5	591.3	382.4	398.1	1730.3	1.94
1	WALK	19962.8	22667.8	12812.8	11849.9	67293.3	75.44
1	PNR	255.6	834.6	661.4	690.3	2441.9	2.74
1	KNR	2460.1	1786.7	968	937.3	6152.1	6.9
1	Total	24343.5	29438.5	17591.4	17823.5	89196.9	100
2	AUTO	480.7	1003	716.1	901.8	3101.7	21.99
2	TRANSIT	4033.2	3696.1	1883	1390.1	11002.4	78.01
2	SOV	335.3	745.1	539.7	709.5	2329.6	16.52
2	HOV	145.5	257.9	176.4	192.4	772.1	5.47
2	SR2	113.2	209.6	145.8	163.8	632.3	4.48
2	SR3	32.3	48.3	30.6	28.6	139.8	0.99
2	WALK	3934.3	3583.9	1815.3	1340.5	10674	75.68
2	PNR	10.3	51.3	33.6	24.5	119.7	0.85
2	KNR	88.6	60.9	34.1	25	208.6	1.48
2	Total	4513.9	4699.1	2599.2	2291.9	14104	100

NEW_MODE_SEGMENT_FILE (optional, output file)

If a segment map file is provided, the mode choice results can also be summarized by mode and market segment in a matrix format. The user may choose to create a tab delimited file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. An example of the type of file (21 modes by 12 market segments) created by this key is shown below.

Mode	1	2	3	4	5	6	7	8	9	10	11	12	Total
AUTO	13309.7	3101.7	19400.4	19161.5	44786.9	19101.6	138996.7	736479.3	33094.1	32432.7	90662	627871.8	1778398
TRANSIT	75887.2	11002.4	27745	14810.4	138662.3	13537.3	48291.6	27722.6	115600.8	25051.7	35870.3	23303	557484.5
SOV	4427.4	2329.6	15782.6	15276.9	14760.7	13117.2	111079.6	595715.2	6428.8	16827.6	61747.4	494603.5	1352097
HOV	8882.3	772.1	3617.7	3884.6	30026.2	5984.4	27917.1	140764	26665.3	15605.1	28914.7	133268.3	426301.6
SR2	7152	632.3	3002.8	3175.1	24028.4	4790.4	22746	115934	13657.8	7060.7	14581.8	97554.7	314316
SR3	1730.3	139.8	615	709.5	5997.7	1194	5171.1	24830	13007.5	8544.4	14332.8	35713.6	111985.7
WALK	67293.3	10674	26528.1	14771.1	48481.3	6506.7	36205.5	25607.2	59769.3	15747.6	25540.3	20771.6	357896
PNR	2441.9	119.7	486.9	15.8	71194.5	5330.4	8698.5	1393	39354.8	6039.6	6657.5	1868.1	143600.8
KNR	6152.1	208.6	730	23.4	18986.5	1700.2	3387.5	722.4	16476.7	3264.4	3672.5	663.3	55987.6
WK_CR	0	0	0	7.4	337	34.1	473.3	142.3	335	74.9	128.9	75.1	1608
WK_BUS	29966.6	3808	13541.7	4979.7	13749	1490.2	14959.6	12498.4	14415	5344.9	8158.7	11923.8	134835.6
WK_BUS_MR	1255.9	369.8	1838.7	7698.5	17851	2686.1	11485.2	9623.6	24659.8	5488.6	10998.1	7734.2	101689.7
WK_MR	36070.8	6496.1	11147.6	2085.6	16544.2	2296.4	9287.4	3343	20359.5	4839.2	6254.5	1038.4	119762.7
PNR_CR	0.5	0	0.2	0	6464.5	399.6	1011.8	145.7	4206.3	884.9	1438.7	160.3	14712.5
PNR_BUS	451.1	0.7	173.7	2.9	4708.8	19.4	637.6	635.6	3804.4	438.4	364.2	1475.2	12712.1
PNR_BUS_MR	200.6	33.4	117.9	4.9	4841.1	846.9	1396.5	211.7	6414.3	1052.8	1684.3	178.3	16982.8
PNR_MR	1789.7	85.6	195.1	8.1	55180	4064.5	5652.6	400	24929.7	3663.6	3170.3	54.3	99193.4
KNR_CR	1.4	0.1	0.3	0	1265.9	96.3	239.4	33.5	1097.5	247	274.5	28.5	3284.3
KNR_BUS	243.1	0.7	58.3	2	4024.3	28.9	561.1	266.7	2662.8	368.3	262.9	356.7	8835.7
KNR_BUS_MR	0	0	0.6	4.3	3504.9	608.2	871.2	177	3481.1	650.1	904.6	220.4	10422.4
KNR_MR	5907.6	207.9	670.9	17.1	10191.4	966.9	1715.9	245.2	9235.4	1999	2230.5	57.6	33445.3
Total	89196.9	14104	47145.4	33971.9	183449.2	32638.8	187288.2	764201.9	148694.9	57484.4	126532.3	651174.8	2335883

NEW_FTA_SUMMIT_FILE (optional, output file)

The ModeChoice program can summarize and write the data needed as input for the FTA SUMMIT program used to generate User Benefits data for the FTA New Starts process. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. The file is in binary format, ready for input to SUMMIT.

NEW_PRODUCTION_FILE (optional, output file)

The program can summarize and write mode choice results for each origin zone into an output file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. Selected records from a production file are shown below.

ZONE	AUTO	TRANSIT	SOV	HOV	SR2	SR3	WALK	PNR	KNR	TOTAL
472	383.62	204.45	302.52	81.1	66.09	15.01	190.34	10.05	4.06	588.07
473	904.98	208.47	721.08	183.91	152.44	31.47	21.99	138.03	48.44	1113.45
474	343.3	84.85	271.91	71.39	59.04	12.35	1.78	67.11	15.97	428.15
475	1655.22	440.95	1307.14	348.08	285.97	62.12	157.14	205.26	78.55	2096.17
476	651.74	157.82	514.36	137.37	112.9	24.47	40.12	93.09	24.61	809.56
477	681.38	187.03	538.46	142.92	117.37	25.55	80.53	83.09	23.41	868.41
478	808.68	244.02	634.93	173.75	142.33	31.43	77.61	135.59	30.82	1052.7
479	1100.15	402.31	868.95	231.2	189.26	41.94	280.13	97.49	24.7	1502.46

NEW_PRODUCTION_FORMAT (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, and SQLITE3.

NEW_ATTRACTION_FILE (optional, output file)

The program can summarize and write mode choice results for each destination zone into an output file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. Selected records from an attraction file are shown below.

ZONE	AUTO	TRANSIT	SOV	HOV	SR2	SR3	WALK	PNR	KNR	TOTAL
9	39.41	73.65	11.02	28.39	19.48	8.91	52.6	15.63	5.41	113.06
10	372.72	1233.57	101.57	271.16	178.93	92.23	769.35	340.52	123.69	1606.29
11	2658.32	5733.63	732.34	1925.98	1271.47	654.51	4271.26	1102.56	359.81	8391.95
12	1412.7	7487.63	390.27	1022.43	688.88	333.55	3785.5	2744.38	957.75	8900.33
13	491.85	520.07	133.8	358.05	237.29	120.76	472.82	34.45	12.8	1011.92
14	794.73	3916.72	215.43	579.3	386.77	192.53	2097.99	1369.51	449.22	4711.45
15	990.74	1053.47	273	717.74	478.48	239.26	948.45	74.63	30.39	2044.21
16	271.53	381.26	74.39	197.14	131.31	65.82	348.43	24.23	8.6	652.79
17	1372.38	6779.72	380.88	991.5	674.29	317.21	3205.85	2608.55	965.32	8152.1
18	1107.57	5191.1	305.9	801.66	540.41	261.25	2646.39	1894.61	650.1	6298.67

NEW_ATTRACTION_FORMAT (optional, text, TAB_DELIMITED)

This key is used to override the default file format for creating data files. The options include BINARY, FIXED_COLUMN, COMMA_DELIMITED, SPACE_DELIMITED, TAB_DELIMITED, CSV_DELIMITED, DBASE, and SQLITE3.

Model Calibration Keys:

The ModeChoice program can be operated in production mode or calibration mode. In production mode the choice model is applied once and the outputs are generated. In calibration mode, the choice model is applied and the differences between the calibration targets and the mode specific results are evaluated. The estimation errors are used to adjust the mode specific constants and the mode choice model is applied again. The process continues until the maximum number of iterations are reached or the model converges.

CALIBRATION_TARGET_FILE (optional, input file)

This key specifies the name and location of the calibration target file. The full path and file name for the calibration file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. This key must be defined for mode constant calibration to take place. The target file is a tab delimited text file with fields and rows that correspond to the model structure, market segments, and trip table names. A sample target file for models without market segments is shown below. This is followed by an example of a target file with market segments. The Min_Const and Max_Const fields define the maximum and minimum constant values the calibration process can generate. Once this value is reached, no additional adjustments to the mode constant area made.

Mode	M1	M2	M3	M4	Target	Min_Const	Max_Const
AUTO	229903	483088	401304	500736	1615030	-8	8
TRANSIT	126842	153369	123126	143790	547128	-8	8
SOV	168206	367728	305782	386757	1228473	-8	8
HOV	61697	115360	95522	113980	386559	-8	8
SR2	45524	85097	70194	84263	285078	-8	8
SR3	16173	30262	25329	29717	101481	-8	8
WALK	106378	103466	71806	57586	339236	-8	8
PNR	8984	33809	37915	70214	150922	-8	8
KNR	11480	16094	13405	15991	56970	-8	8
WK_CR	604	700	574	438	2316	-8	8
WK_BUS	57844	38988	17901	15735	130468	-8	8
WK_BUS_MR	23642	27408	22471	17165	90686	-8	8
WK_MR	24288	36370	30860	24248	115768	-8	8
PNR_CR	1693	5166	4992	8048	19899	-8	8
PNR_BUS	2419	3367	2898	4197	12882	-8	8
PNR_BUS_MR	1330	3802	3326	5696	14154	-8	8
PNR_MR	3543	21474	26698	52272	103987	-8	8
KNR_CR	400	1180	1186	1693	4459	-8	8
KNR_BUS	1956	2404	2269	2321	8950	-8	8
KNR_BUS_MR	1526	2456	2140	2641	8762	-8	8
KNR_MR	7599	10055	7810	9336	34800	-8	8

Segment	Mode	Target	Min_Const	Max_Const
1	AUTO	4035	-8	8
1	TRANSIT	76656	-8	8
1	SOV	3081	-8	8
1	HOV	954	-8	8
1	SR2	766	-8	8
1	SR3	188	-8	8
1	WALK	71598	-8	8
1	PNR	3901	-8	8
1	KNR	1157	-8	8
1	WK_CR	3	-8	8
1	WK_BUS	35139	-8	8
1	WK_BUS_MR	6281	-8	8
1	WK_MR	30176	-8	8
1	PNR_CR	5	-8	8
1	PNR_BUS	622	-8	8
1	PNR_BUS_MR	239	-8	8
1	PNR_MR	3035	-8	8
1	KNR_CR	10	-8	8
1	KNR_BUS	93	-8	8
1	KNR_BUS_MR	54	-8	8
1	KNR_MR	999	-8	8
2	AUTO	3260	-8	8
2	TRANSIT	7690	-8	8

CALIBRATION_SCALING_FACTOR (optional, decimal, 1, 1.0..5.0)

This factor is applied to the calibration targets prior to the calibration process. It is used to adjust the calibration targets to a different analysis year or to adjust the trip units.

MAX_CALIBRATION_ITERATIONS (optional, integer, 20, 1..1000)

The calibration process adjusts the mode specific constants over multiple iterations. This key defines the maximum number of iterations that may be performed before the process exits.

CALIBRATION_EXIT_RMSE (optional, decimal, 5, .0..50.0)

The calibration process will exit after a maximum number of iterations or the root mean squared errors between the mode targets and the estimated trips are less than the value of this key.

NEW_MODE_CONSTANT_FILE (optional, output file)

After the last calibration iteration, the program writes the final set of mode specific constants in the same format as the input calibration target file. The full path and file name for the new mode constant file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key.

NEW_CALIBRATION_DATA_FILE (optional, output file)

During the calibration process, the program writes the estimated mode shares to the output printout file. The user may also choose to write the results of each iteration to a tab delimited text file that can be processed or formatted by Excel for documentation and analysis purposes. The full path and file name for the output file is constructed by appending the value of this key to the value of the PROJECT_DIRECTORY key. A few samples lines from a calibration data file are shown below.

Table	Mode	Target	Trips	Diff	Percent	Constant
Total	AUTO	1784560	1778398	-6162	-0.30%	0.051099
Total	TRANSIT	551324	557484	6160	1.10%	-0.1639
Total	SOV	1357098	1352097	-5001	-0.40%	0.04324
Total	HOV	427462	426302	-1160	-0.30%	-0.13717
Total	SR2	315245	314316	-929	-0.30%	-0.04398
Total	SR3	112217	111986	-231	-0.20%	0.123453
Total	WALK	351892	357896	6004	1.70%	2.684534
Total	PNR	143519	143601	82	0.10%	-4.41545
Total	KNR	55913	55988	75	0.10%	-5.67752
Total	WK_CR	1714	1608	-106	-6.20%	-1.89604
Total	WK_BUS	130468	134836	4368	3.30%	-3.7537
Total	WK_BUS_I	109731	101690	-8041	-7.30%	1.230198
Total	WK_MR	109979	119763	9784	8.90%	3.251272
Total	PNR_CR	14726	14713	-13	-0.10%	-1.44801
Total	PNR_BUS	12881	12712	-169	-1.30%	-1.72846
Total	PNR_BUS_	17125	16983	-142	-0.80%	0.161845
Total	PNR_MR	98787	99193	406	0.40%	0.410116
Total	KNR_CR	3300	3284	-16	-0.50%	-0.1991
Total	KNR_BUS	8950	8836	-114	-1.30%	0.788159
Total	KNR_BUS_	10603	10422	-181	-1.70%	1.221137
Total	KNR_MR	33060	33445	385	1.20%	-0.57578
M1	SOV	192805	191004	-1801	-0.90%	-0.96654
M1	SR2	52182	50559	-1623	-3.10%	-1.68618
M1	SR3	18538	16882	-1656	-8.90%	-2.16409

Report Options:

MODECHOICE_REPORT_# (optional, text)

This key is typically used to select reports written to the printout file. The "#" at the end of the report keyword represents the report number (e.g., MODECHOICE_REPORT_1). The key can be provided with additional numbers to specify additional reports. The reports are typically generated in numerical order (i.e., 1, 2, 3…).

The string parameter associated with a report keyword is limited to the following options:

MODE_CHOICE_SCRIPT
MODE_CHOICE_STACK
MODE_SUMMARY_REPORT
MARKET_SEGMENT_REPORT
CALIBRATION_REPORT
TARGET_DATA_REPORT
MODE_VALUE_SUMMARY
SEGMENT_VALUE_SUMMARY

The above reports are printed in the "*.prn" file that is generated in the same directory as the control file used to run the program. Each of above reports is described below:

MODE CHOICE SCRIPT

The mode choice script report is a simple listing of the processing logic included in the file identified by the Mode Choice Script key. An extract from a mode choice script report is shown below.

```
Mode Choice Script

      //---- WMATA Mode Choice Script -- 1/25/12 ----
      INTEGER HBW, HBS, HBO, NHB, PEAK, ENDDEF
      REAL PEF_P, PEF_A, BREAK1, BREAK2, BREAK3, BREAK4, FACTOR,
        WK_P1, WK_P2, WK_P3, WK_P4, WK_A1, WK_A2, WK_A3, WK_A4, WK_PEF,
        DR_P1, DR_P2, DR_P3, DR_P4, DR_A1, DR_A2, DR_A3, DR_A4, DR_PEF,
        CBD, TERM_TIME, PARK_COST, AUTO_OP, OCC3, TIME1, TIME2, TIME3,
        SHORT_WK_P, SHORT_WK_A, LONG_WK_P, LONG_WK_A, WK_P, DR_P, WK_A, DR_A,
        WK_MR_P, WK_MR_A, WK_ALL_P, WK_ALL_A, WK_ACCESS, WK_METRO,
      ENDDEF
      //---- purpose/period codes ----
      HBW = 1
      HBS = 2
      HBO = 3
      NHB = 4
      PEAK = 1
      //---- auto operating cost ---
      AUTO_OP = 10
      //---- auto occupancy for 3+ ----
      OCC3 = 3.5
      //---- terminal time ----
      IF (TRIP.PURPOSE != NHB) THEN
        TERM_TIME = ORG.TERM_TIME_HOME + DES.TERM_TIME_HOME
      ELSE
        TERM_TIME = ORG.TERM_TIME_OTHER + DES.TERM_TIME_OTHER
      ENDIF
```

MODE CHOICE STACK

The mode choice stack report shows how the processing logic included in the file identified by the Mode Choice Script key is converted to compiled commands. An extract from a mode choice stack report is shown below.

```
Mode Choice Stack

 18) Real          AUTO_OP
 19) Real          3.500000
 20) Assign        =
 21) Real          OCC3
 22) Integer       Trip.PURPOSE
 23) Integer       NHB
 24) Relation      NE
 25) Logical       If False, Jump to 32
 26) Integer       Org.TERM_TIME_HOME
 27) Integer       Des.TERM_TIME_HOME
 28) Math          +
 29) Assign        =
 30) Real          TERM_TIME
 31) Logical       Jump to 37
 32) Integer       Org.TERM_TIME_OTHER
 33) Integer       Des.TERM_TIME_OTHER
 34) Math          +
 35) Assign        =
 36) Real          TERM_TIME
 37) Integer       Trip.PURPOSE
 38) Integer       HBW
 39) Relation      EQ
 40) Logical       If False, Jump to 47
 41) Integer       Des.HBW_PARK_COST
 42) Real          2.000000
 43) Math          /
 44) Assign        =
 45) Real          PARK_COST
 46) Logical       Jump to 70
 47) Integer       Trip.PURPOSE
 48) Integer       HBS
 49) Relation      EQ
 50) Logical       If False, Jump to 57
 51) Integer       Des.HBS_PARK_COST
 52) Real          2.000000
 53) Math          /
 54) Assign        =
 55) Real          PARK_COST
 56) Logical       Jump to 70
 57) Integer       Trip.PURPOSE
 58) Integer       HBO
 59) Relation      EQ
```

MODE SUMMARY REPORT

The mode summary report shows the trips allocated to each mode and from each trip table and the overall mode shares and table shares. An example of the mode summary report is shown below.

Mode Summary Report -- Peak Home-Based Work

Trips by Mode	Table M1	Table M2	Table M3	Table M4	Total	Percent
AUTO	235766.1	493333.5	412580.7	517518.8	1659199.1	76.72%
TRANSIT	120970.0	143115.0	111844.1	127007.6	502936.7	23.26%
SOV	171940.2	370539.3	307945.2	387997.6	1238422.3	57.27%
HOV	63825.8	122794.3	104635.5	129521.2	420776.8	19.46%
SR2	46540.9	89120.0	76030.5	96147.1	307838.5	14.23%
SR3	17284.9	33674.2	28605.0	33374.1	112938.2	5.22%
WALK	102405.2	100095.1	64984.1	55596.8	323081.3	14.94%
PNR	7642.0	29519.7	34858.5	59376.5	131396.7	6.08%
KNR	10922.8	13500.1	12001.5	12034.2	48458.7	2.24%
WK_CR	196.7	796.2	486.1	388.8	1867.9	0.09%
WK_BUS	51639.9	36387.0	16819.2	13846.4	118692.6	5.49%
WK_BUS_MR	27863.5	28236.9	22940.5	16448.9	95489.7	4.42%
WK_MR	22705.2	34674.9	24738.4	24912.7	107031.2	4.95%
PNR_CR	998.8	3912.9	4229.2	5728.7	14869.7	0.69%
PNR_BUS	2182.2	3067.4	3261.6	3957.1	12468.3	0.58%
PNR_BUS_MR	1203.7	4176.5	4466.9	5757.8	15605.0	0.72%
PNR_MR	3257.2	18362.9	22900.7	43932.9	88453.7	4.09%
KNR_CR	180.5	626.3	871.6	1021.6	2700.0	0.12%
KNR_BUS	1619.1	1817.1	1936.0	1579.9	6952.1	0.32%
KNR_BUS_MR	1624.1	2679.2	2657.9	2587.7	9549.0	0.44%
KNR_MR	7499.1	8377.5	6535.9	6845.1	29257.5	1.35%
Total	356815.1	636581.1	524529.5	644623.7	2162549.4	100.00%
Percent	16.50%	29.44%	24.26%	29.81%	100.00%	

MARKET SEGMENT REPORT

The market segment report has the same layout as the mode summary report, but lists the trips and mode shares for each market segment. In this case market segments can be defined as part of the mode choice model or simply included to summarize travel markets of special interest. An example of the market segment report is shown below.

Market Segment Report #2 -- Peak Home-Based Work

Trips by Mode	Table M1	Table M2	Table M3	Table M4	Total	Percent
AUTO	482.7	1010.6	721.0	906.7	3121.1	22.13%
TRANSIT	4031.2	3688.4	1878.1	1385.2	10983.0	77.87%
SOV	336.2	750.6	543.7	713.7	2344.3	16.62%
HOV	146.4	260.0	177.4	193.0	776.8	5.51%
SR2	113.9	211.3	146.6	164.5	636.2	4.51%
SR3	32.6	48.7	30.8	28.5	140.6	1.00%
WALK	3932.7	3576.6	1810.9	1336.1	10656.3	75.55%
PNR	10.5	51.5	33.5	24.4	120.0	0.85%
KNR	88.0	60.4	33.7	24.7	206.7	1.47%
WK_CR	0.0	0.0	0.0	0.0	0.0	0.00%
WK_BUS	1654.4	1246.1	542.0	348.3	3790.8	26.88%
WK_BUS_MR	124.5	123.2	72.2	53.3	373.2	2.65%
WK_MR	2153.7	2207.3	1196.8	934.5	6492.2	46.03%
PNR_CR	0.0	0.0	0.0	0.0	0.0	0.00%
PNR_BUS	0.3	0.2	0.1	0.1	0.7	0.01%
PNR_BUS_MR	3.1	13.0	9.4	8.5	33.9	0.24%
PNR_MR	7.1	38.3	24.0	15.8	85.3	0.60%
KNR_CR	0.0	0.0	0.0	0.0	0.1	0.00%
KNR_BUS	0.4	0.2	0.1	0.1	0.8	0.01%
KNR_BUS_MR	0.0	0.0	0.0	0.0	0.0	0.00%
KNR_MR	87.7	60.2	33.5	24.6	205.9	1.46%
Total	4513.9	4699.1	2599.2	2291.9	14104.0	100.00%
Percent	32.00%	33.32%	18.43%	16.25%	100.00%	

CALIBRATION REPORT

The calibration report shows the difference between the target value and the estimated number of trips after each iteration of the convergence process. It shows the difference, percent error, and the current set of mode-specific constants. The Minutes column to the rights translates the mode constant to an equivalent number of minutes in a vehicle. Large values imply that a significant portion of the explanatory power of the model is defined by the mode constant. The bottom of the report shows the absolute error, average error, percent root mean squared error (RMSE), and r-squared values for the overall model. The RMSE value is one of the stopping criteria for the iterative process.

```
Calibration Report for Iteration #2 -- Peak Home-Based Work

Trips by Mode           Target     Trips  Difference  Percent    Constant  Minutes

AUTO                   1784560   1779442       -5118    -0.3%    0.054487
TRANSIT                 551324    556441        5117     0.9%   -0.175085
SOV                    1357098   1353008       -4090    -0.3%    0.043464      2.5
HOV                     427462    426433       -1029    -0.2%   -0.137922
SR2                     315245    314443        -802    -0.3%   -0.043735     -0.8
SR3                     112217    111990        -227    -0.2%    0.122807      0.5
WALK                    351892    357402        5510     1.6%    2.676592
PNR                     143519    143201        -318    -0.2%   -4.407045
KNR                      55913     55837         -76    -0.1%   -5.669874
WK_CR                     1714      1610        -104    -6.1%   -1.822108     23.6
WK_BUS                  130468    133868        3400     2.6%   -3.776571      7.3
WK_BUS_MR               109731    102249       -7482    -6.8%    1.316363     49.7
WK_MR                   109979    119675        9696     8.8%    3.176108     65.2
PNR_CR                   14726     14644         -82    -0.6%   -1.446001    -91.3
PNR_BUS                  12881     12730        -151    -1.2%   -1.714174    -93.6
PNR_BUS_MR               17125     16961        -164    -1.0%    0.171273    -77.9
PNR_MR                   98787     98866          79     0.1%    0.407098    -75.9
KNR_CR                    3300      3270         -30    -0.9%   -0.193783   -101.9
KNR_BUS                   8950      8842        -108    -1.2%    0.801548    -93.7
KNR_BUS_MR               10603     10430        -173    -1.6%    1.238855    -90.0
KNR_MR                   33060     33295         235     0.7%   -0.586834   -105.2

Abs.Error = 1.1%    Avg.Error = 0.4%     RMSE = 6.2%     R-Squared = 1.000
```

TARGET DATA REPORT

The target data report lists the calibration target trips by mode and sub-mode included in the file referenced by the mode value summary report shows the weighted average value of each attribute provided by the Calibration Target File key. An example of the target data report is shown below.

Target Data Report -- Peak Home-Based Work

Trips by Mode	M1	M2	M3	M4	Target	Min_Const	Max_Const
AUTO	263525	532091	439427	549517	1784560	-8.00	8.00
TRANSIT	129936	155213	123968	142206	551323	-8.00	8.00
SOV	192805	405029	334830	424434	1357098	-8.00	8.00
HOV	70720	127061	104597	125084	427462	-8.00	8.00
SR2	52182	93729	76862	92472	315245	-8.00	8.00
SR3	18538	33332	27735	32612	112217	-8.00	8.00
WALK	109971	107221	74833	59864	351889	-8.00	8.00
PNR	8647	32191	35980	66703	143521	-8.00	8.00
KNR	11318	15801	13156	15639	55914	-8.00	8.00
WK_CR	447	518	425	324	1714	-8.00	8.00
WK_BUS	57844	38988	17901	15735	130468	-8.00	8.00
WK_BUS_MR	28607	33164	27190	20770	109731	-8.00	8.00
WK_MR	23074	34552	29317	23036	109979	-8.00	8.00
PNR_CR	1253	3823	3694	5956	14726	-8.00	8.00
PNR_BUS	2419	3367	2898	4197	12881	-8.00	8.00
PNR_BUS_MR	1609	4600	4024	6892	17125	-8.00	8.00
PNR_MR	3366	20400	25363	49658	98787	-8.00	8.00
KNR_CR	296	873	878	1253	3300	-8.00	8.00
KNR_BUS	1956	2404	2269	2321	8950	-8.00	8.00
KNR_BUS_MR	1846	2972	2589	3196	10603	-8.00	8.00
KNR_MR	7219	9552	7420	8869	33060	-8.00	8.00

Number of Calibration Targets = 21

MODE VALUE SUMMARY

The mode value summary report shows the weighted average value of each attribute provided by the mode script for each travel mode. This report is helpful in reviewing the attributes to ensure they are in the expected range. An example of the mode value summary report is shown below.

Mode Value Summary Report -- Peak Home-Based Work

Average Mode Values	TIME	WALK	AUTO	WAIT	LWAIT	XWAIT	TPEN	TERM	COST	XFER	BIAS	PEF	CBD	Trips
SOV	33.05	0.00	0.00	0.00	0.00	0.00	0.00	4.95	266.12	0.00	0.00	0.00	-0.02	1238422
SR2	40.23	0.00	0.00	0.00	0.00	0.00	0.00	5.37	190.22	0.00	0.00	0.00	0.00	307839
SR3	50.48	0.00	0.00	0.00	0.00	0.00	0.00	5.42	143.84	0.00	0.00	0.00	0.00	112938
WK_CR	39.85	14.97	0.00	14.66	0.00	0.00	3.06	0.00	341.66	0.55	0.00	-0.62	0.45	1868
WK_BUS	28.03	13.47	0.00	8.04	0.00	0.00	6.30	0.00	153.01	0.26	0.00	0.84	0.43	118693
WK_BUS_MR	37.90	12.03	0.00	9.75	0.00	0.00	7.17	0.00	348.68	1.41	0.00	-0.57	0.43	95490
WK_MR	16.29	8.55	0.00	3.15	0.00	0.00	2.00	0.00	198.34	0.37	0.00	1.98	0.61	107031
PNR_CR	47.37	10.02	14.58	13.70	0.00	0.00	2.75	2.06	414.99	0.30	0.00	2.60	0.77	14870
PNR_BUS	51.08	4.32	7.29	7.49	0.00	0.00	5.16	1.06	317.13	0.03	0.00	2.12	0.75	12468
PNR_BUS_MR	46.28	6.57	6.94	8.79	0.00	0.00	7.08	1.23	444.00	1.26	0.00	2.51	0.71	15605
PNR_MR	25.73	3.16	14.93	3.41	0.00	0.00	2.00	7.53	576.93	0.29	0.00	2.58	0.86	88454
KNR_CR	42.12	8.59	4.10	14.60	0.00	0.00	2.68	0.00	330.76	0.27	0.00	2.49	0.75	2700
KNR_BUS	56.27	5.33	6.42	8.18	0.00	0.00	5.31	0.00	322.54	0.06	0.00	2.33	0.80	6952
KNR_BUS_MR	49.73	6.24	5.94	10.23	0.00	0.00	7.08	0.00	468.70	1.28	0.00	2.30	0.69	9549
KNR_MR	17.86	2.95	5.89	3.09	0.00	0.00	2.00	0.00	249.10	0.26	0.00	2.24	0.79	29258

SEGMENT VALUE SUMMARY

The segment value summary report has the same layout as the mode value summary report, but lists the model attributes and mode shares for each market segment. In this case market segments can be defined as part of the mode choice model or simply included to summarize travel markets of special interest. An example of the segment value report is shown below.

```
Segment Value Summary Report #9 -- Peak Home-Based Work

Average Mode Values      TIME    WALK    AUTO    WAIT   LWAIT   XWAIT    TPEN    TERM    COST    XFER    BIAS     PEF     CBD   Trips

SOV                     57.03    0.00    0.00    0.00    0.00    0.00    0.00    7.44  648.07    0.00    0.00    0.00   -1.00    6491
SR2                     59.93    0.00    0.00    0.00    0.00    0.00    0.00    7.28  385.16    0.00    0.00    0.00    0.00   13754
SR3                     52.89    0.00    0.00    0.00    0.00    0.00    0.00    6.92  223.25    0.00    0.00    0.00    0.00   13071
WK_CR                   40.54   15.52    0.00   11.75    0.00    0.00    2.65    0.00  372.68    0.30    0.00    0.29    1.00     339
WK_BUS                  28.36   13.29    0.00    8.49    0.00    0.00    6.17    0.00  153.04    0.23    0.00    1.39    1.00   14258
WK_BUS_MR               35.07    9.73    0.00    9.14    0.00    0.00    7.01    0.00  400.77    1.28    0.00   -0.49    1.00   24992
WK_MR                   16.17    9.70    0.00    2.91    0.00    0.00    2.00    0.00  279.55    0.23    0.00    2.49    1.00   20345
PNR_CR                  58.91    6.33   14.90   14.11    0.00    0.00    2.89    2.66  570.98    0.41    0.00    2.79    1.00    4162
PNR_BUS                 53.52    3.80    7.42   10.49    0.00    0.00    5.18    1.06  440.06    0.04    0.00    2.84    1.00    3801
PNR_BUS_MR              45.96    5.74    6.73    9.99    0.00    0.00    7.00    1.06  533.02    1.17    0.00    2.79    1.00    6364
PNR_MR                  27.60    3.18   14.51    3.06    0.00    0.00    2.00   10.55  696.37    0.21    0.00    2.73    1.00   24738
KNR_CR                  45.69    6.27    4.11   15.01    0.00    0.00    2.76    0.00  456.51    0.32    0.00    2.70    1.00    1087
KNR_BUS                 54.73    5.03    7.35   10.31    0.00    0.00    5.29    0.00  458.59    0.06    0.00    2.82    1.00    2659
KNR_BUS_MR              46.57    5.19    6.86   10.63    0.00    0.00    7.00    0.00  556.65    1.17    0.00    2.81    1.00    3480
KNR_MR                  21.27    2.85    7.10    3.14    0.00    0.00    2.00    0.00  370.29    0.23    0.00    2.66    1.00    9154
```

Model Structure

This section describes how the control keys, key values, and supporting data files work together to define various model structures.

Mode Constant File

At a minimum the mode constant file needs to include a mode and constant field. It interprets fields named "mode", "m", "mod", and "modes" as the mode field and "constant" or "const" as the constant field. The text strings found in each row of the mode field must equal the mode labels defined by the primary and nested mode choice keys. For example, if the model structure includes the following keys:

```
PRIMARY_MODE_CHOICE      AUTO, TRANSIT
MODE_CHOICE_NEST_1       AUTO = SOV, HOV
```

The mode field values should be AUTO, TRANSIT, SOV, or HOV.

If the model includes market segment constants, a market segment field should be included in the mode constant file using the field name "segment", "seg", "market", or "s". A new set of mode specific constants should be provided for each market segment.

If the input trip file includes multiple tables or the select trip tables key includes multiple table names, the mode constant file may include a constant value corresponding to each table name. These are typically mode specific constants related to income groups or auto ownership levels. They only apply to non-nested modes. A sample mode constant file for a nested logic model with four income groups (m1..m4) and one market segment is shown below:

MODE	CONSTANT	M1	M2	M3	M4
AUTO	-0.04163	0	0	0	0
TRANSIT	0.123351	0	0	0	0
SOV	0.0564	-1.00472	0.316212	-0.03065	0.525459
HOV	-0.17925	0	0	0	0
SR2	-0.03414	-1.7059	-0.4329	-0.78516	-0.35202
SR3	0.095915	-2.19206	-0.91932	-1.26013	-0.84587
WALK	2.786009	0	0	0	0
PNR	-4.16415	0	0	0	0
KNR	-5.38748	0	0	0	0

WK_CR	-2.09436	2.504276	1.771577	0.982812	0.463658
WK_BUS	-3.30456	4.466809	1.897428	0.220162	-0.41669
WK_BUS_MR	0.336553	3.480035	1.759397	1.231243	0.613881
WK_MR	3.692878	3.179454	2.368986	3.619977	1.794977
PNR_CR	-0.9984	1.578119	1.161049	0.48174	0.326164
PNR_BUS	-2.02325	2.075262	1.112592	0.444009	0.138738
PNR_BUS_MR	-0.36019	1.791723	1.135733	0.446216	0.182777
PNR_MR	0.49127	1.728489	1.13469	0.435312	0.28503
KNR_CR	0.519867	1.689123	1.143168	0.591982	0.339448
KNR_BUS	0.859016	2.211065	0.821654	0.127109	-0.38996
KNR_BUS_MR	0.512033	2.149108	1.083957	0.332159	-0.12394
KNR_MR	-0.41711	2.424966	1.005164	0.161685	-0.27005

Segment Map File

The segment map file expects to find three fields in the data file. It interprets field names "origin", "org", "o", or "i" as the origin field; "destination", "des", "d", or "j" as the destination field; and "segment", "seg", "market", or "s" as the market segment field. If the file does not include a file header, the program assumes the fields are in origin, destination, market segment order. Each field is assumed to be an integer number. The numbers found in the origin and destination fields are values assigned to the origin map field and the destination map field for a given zone in the zone file.

A typical segment map file is shown below. This file uses seven origin and destination area type codes to define 12 market segments.

ORG	DES	SEGMENT
1	1	1
1	2	2
1	3	2
1	4	3
1	5	3
1	6	4
1	7	4
2	1	9
2	2	10
2	3	11
2	4	11
2	5	11
2	6	12
2	7	12
3	1	1
3	2	2
3	3	3
3	4	3
3	5	3

3	6	4
3	7	4
4	3	7
4	4	7
4	5	7
4	6	8
4	7	8
5	1	9
5	2	10
5	3	11
5	4	11
5	5	11
5	6	12
5	7	12
6	1	5
6	2	6
6	3	7
6	4	7

Mode Choice Script Interface

The program creates a number of custom file records to set values for use or update by the user-defined mode choice script. The zone data file, for example, is split into an "ORG" file and a "DES" file to give the user access to the zone-based attributes at the origin and destination ends of the trip. Each of the data fields included in the input zone file is copied to the ORG and DES files provided to the mode choice script. For example, if the zone file includes an AREA_TYPE field, the command ORG.AREA_TYPE will return the value of the AREA_TYPE field corresponding to the origin zone number.

The program also adds fields to the input trip file to expand functionality. Normally the trip file will include the origin and destination zone numbers and the number of trips. The program adds a "PURPOSE", "PERIOD", "SEGMENT", and several "ACCESS#" fields to the trip file. The value of the Trip Purpose Number control key is written to the "PURPOSE" field and the value of Trip Time Period control key is written to the "PERIOD" field. If a segment map is provided, the segment value generated by the segment map is written to the "SEGMENT" field. An access field is created for each of the Mode Access Market keys defined in the control file. The user script should calculate the market share for each access market and store that share in the corresponding TRIP.ACCESS# field.

The program also creates an attribute or value file for each of the user-defined modes. The file names correspond to the mode names defined by the Primary Mode Choice and Mode Choice Nest keys. For example, the keys

```
PRIMARY_MODE_CHOICE      AUTO, TRANSIT
MODE_CHOICE_NEST_1       AUTO = SOV, HOV
```

Create four files for use by the modeling script. The file names will be "AUTO", "TRANSIT", "SOV" and "HOV". Each of these files will include a program-defined series of field names. These field names are listed below along with the control key that is applied to the value of each field when calculating the mode utility.

Description	**Field Name**	**Control Key**
In-vehicle travel time	TIME	VEHICLE_TIME_VALUE
Walk access time	WALK	WALK_TIME_VALUE
Drive access time	AUTO	DRIVE_ACCESS_VALUE
Initial waiting time	WAIT	WAIT_TIME_VALUE
Long initial wait time	LWAIT	LONG_WAIT_VALUE
Transfer wait time	XWAIT	TRANSFER_TIME_VALUE
Additional travel time	TPEN	PENALTY_TIME_VALUE
Terminal Time	TERM	TERMINAL_TIME_VALUE
Travel cost	COST	COST_VALUE_TABLE_#
Transfer penalty	XFER	TRANSFER_COUNT_VALUE
Mode-specific bias	BIAS	1
Pedestrian Environment Factor	PEF	1
CBD constant	CBD	1
Mode constant value	CONSTANT	1
Table-specific constants	[table names]	1

Given the mode example listed above, the user can include in-vehicle travel time, terminal time, and cost in the utility equation by setting the following variables in the model script.

```
AUTO.TIME = SKIM2.TIME
AUTO.COST = 10 * SKIM2.DISTANCE + SKIM2.TOLL + DES.PARK_COST
AUTO.TERM = ORG.TERM_TIME + DES.TERM_TIME
```

The program will than calculate the mode utility by exponentiating the following equation:

Auto [x] = AUTO.TIME * VEHICLE_TIME_VALUE + AUTO.TERM * TERMINAL_TIME_VALUE +
AUTO.COST * COST_VALUE_TABLE [x] + AUTO.CONSTANT + AUTO.[x]

Where [x] corresponds to each subdivision of trips defined in the input trip table. This is typically a distribution of trips by income group or auto ownership.

Note that the CONSTANT and [x] fields do not need to be set by the model script. These values are pre-set with the constant values included in the Mode Constant File. The program does, however, permit the user to modify these constant values for a particular interchange to implement special market segment adjustments. The constant values are re-initialized each time the script is executed.

The Auto [x] utility is used in the mode share calculation for each access market. The access market share (TRIP.ACCESS#) is multiplied by the number of trips in the corresponding trip table [x] to define the number of trips in the access market. These trips are then distributed to the modes included in the corresponding Mode Access Market key based on the model structure defined by the Primary Mode Choice and Mode Choice Nest keys. The sum of the mode utilities in the lowest level nest are calculated first. The log of this sum is factored by the nesting constant and added to the sum of the mode utilities included in the next higher nest. This continues until the primary mode choice is reached. The primary mode choice model splits the access market trips into the primary modes based on the ratio of the mode utility to the sum of the mode utilities. The shares assigned to each nested mode are calculated in the same way.

Simple Example

This section describes how to configure the software to implement a simple mode choice model involving a primary choice between auto and transit with a nested transit choice between walk and drive access. The trip table only includes total home-based work person trips. Three skim files are provided. The first is an auto skim, the second is the walk to transit skim and the third is the drive to transit skim.

The control file includes the following keys.

```
TRIP_FILE                TRIPS\HBW.MTX
SKIM_FILE_1              SKIMS\AUTO.SKM
SKIM_FILE_2              SKIMS\WALK.SKM
SKIM_FILE_3              SKIMS\DRIVE.SKM
ZONE_FILE                ZONE.TXT
NEW_TRIP_FILE            TRIPS\HBW_MS.MTX

MODE_CONSTANT_FILE       HBW_Constant.txt
MODE_CHOICE_SCRIPT       Mode_Choice_Script.txt

PRIMARY_MODE_CHOICE      AUTO, TRANSIT
MODE_CHOICE_NEST_1       TRANSIT = WALK, DRIVE
NESTING_COEFFICIENT      0.5

VEHICLE_TIME_VALUE       -0.02
WALK_TIME_VALUE          -0.04
DRIVE_ACCESS_VALUE       -0.03
WAIT_TIME_VALUE          -0.05

MODE_ACCESS_MARKET_1     AUTO, WALK, DRIVE
MODE_ACCESS_MARKET_2     AUTO, DRIVE
MODE_ACCESS_MARKET_3     AUTO

NEW_TABLE_MODES_1        AUTO
NEW_TABLE_MODES_2        WALK = Walk Access
NEW_TABLE_MODES_3        DRIVE = Drive Access
```

The mode constant file would look something like the following:

```
MODE        CONSTANT
AUTO         0.051099
TRANSIT     -0.163902
WALK         2.684534
DRIVE       -4.415451
```

At a minimum, the mode choice script should include the following:

```
//---- transit access markets ----

TRIP.ACCESS1 = ORG.WALK_COVERAGE * DES.WALK_COVERAGE
TRIP.ACCESS2 = (1 - ORG.WALK_COVERAGE) * DES.WALK_COVERAGE
TRIP.ACCESS3 = 1 - TRIP.ACCESS1 - TRIP.ACCESS2
```

```
//---- auto ----

AUTO.TIME = SKIM1.TIME

//---- walk access to transit ----

IF (SKIM2.TIME > 0) THEN
  WALK.TIME = SKIM2.TIME
  WALK.WALK = SKIM2.WALK
  WALK.WAIT = SKIM2.WAIT
ENDIF

//---- drive access to transit ----

IF (SKIM3.TIME > 0) THEN
  DRIVE.TIME = SKIM3.TIME
  DRIVE.WALK = SKIM3.WALK
  DRIVE.DRIVE = SKIM3.DRIVE
  DRIVE.WAIT = SKIM3.WAIT
ENDIF
```

Note that the mode names defined by the Primary Mode Choice and Mode Choice Nest keys in the control file are the same as the mode names in the mode constant file and the mode choice script. The mode utilities will be calculated as:

Auto = exp (-0.02 * SKIM1.TIME + 0.051099)

Walk = exp (-0.02 * SKIM2.TIME + -0.04 * SKIM2.WALK + -0.05 * SKIM2.WAIT + 2.684534)

Drive = exp (-0.02 * SKIM3.TIME + -0.04 * SKIM2.WALK + -0.03 * SKIM3.DRIVE +
-0.05 * SKIM3.WAIT + -4.415451)

Transit = exp (0.5 * log (Walk + Drive) + -0.163902)

The model includes three mode access markets based on the percentages of the origin and destination zone that are within walking distance of transit service. The first market has walk access at the origin and destination ends of the trip. As a result, all three modes are an option in this market segment. The second market is not within walking distance at the origin end, but is within walking distance at the destination end of the trip. In this case, a walk access transit trip is not an option, but a drive access transit trip is a possibility. The third market is not within walking distance at both ends of the trip, so the only option is auto.

The mode shares for access market #1 are calculated as:

Auto Share = Auto / (Auto + Transit)

Transit Share = Transit / (Auto + Transit)

Walk Share = Transit Share * Walk / (Walk + Drive)

Drive Share = Transit Share * Drive / (Walk + Drive)

The mode shares for access market #2 are calculated as:

Auto Share = Auto / (Auto + Drive)

Drive Share = Drive / (Auto + Drive)

All of the trips in access market #3 are assigned to the auto mode.